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

IDENTIFIKACE GENŮ VE SQUIGGLECH ZE SEKVENACE NANOPÓREM
GENE IDENTIFICATION IN NANOPORE SQUIGGLES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Nikita Talanin**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Vojtěch Bartoň**
ADVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Biomedicínská technika a bioinformatika**

Ústav biomedicínského inženýrství

Student: Nikita Talanin ***ID:*** 218498

Ročník: 3 ***Akademický rok:*** 2022/23

NÁZEV TÉMATU:

Identifikace genů ve squigglech ze sekvenace nanopórem

POKYNY PRO VYPRACOVÁNÍ:

1) Vytvořte literární rešerši na téma sekvenace nanopórem. Zaměřte se na popis výstupního formátu dat. 2) Sestavte z dostupných dat vhodný dataset bakteriálních genomů a proveďte výběr vhodných genů pro identifikaci. 3) V datasetu identifikujte úseky signálů reprezentující sekvence vybraných genů. 4) Navrhněte metodu identifikace genu ve squigglech. 5) Realizujte navržený postup ve vhodném programovém prostředí. 6) Statisticky vyhodnoťte úspěšnost vašeho řešení a výsledky diskutujte.

DOPORUČENÁ LITERATURA:

BRANTON, Daniel, David W DEAMER, Andre MARZIALI, et al. Nature Biotechnology. 2008, 26(10). ISSN 1087-0156. Dostupné z: doi:10.1038/nbt.1495

RANG, Franka J., Wigard P. KLOOSTERMAN a Jeroen DE RIDDER. From squiggle to basepair: computational approaches for improving nanopore sequencing read accuracy. Genome Biology. 2018, 19(1). ISSN 1474-760X. Dostupné z: doi:10.1186/s13059-018-1462-9

Termín zadání: 6.2.2023 ***Termín odevzdání:*** 14.8.2023

Vedoucí práce: Ing. Vojtěch Bartoň

doc. Ing. Jana Kolářová, Ph.D.
předseda rady studijního programu

ABSTRAKT

Sekvenace nanopórem je nová a rychle se rozvíjející technologie, která umožňuje přímé sekvenování jednovláknové DNA a RNA v reálném čase. Výsledkem sekvenace je takzvaný squiggle, což je časová řada intenzit proudů při průchodu nukleotidů nanopórem. Identifikace genů v těchto squigglech je klíčovým krokem pro využití těchto dat v genomických studiích.

Tato bakalářská práce se zabývá vývojem a testováním metody pro automatickou identifikaci genů ve squigglech ze sekvenace nanopórem. Cílem bylo vytvořit systém, který by mohl rychle a přesně identifikovat geny v squigglech, a tím podpořit další analýzy a interpretaci dat z nanopórové sekvenace.

Metoda využívá konvoluční neuronové sítě (CNN), které byly úspěšně použity v mnoha jiných oblastech bioinformatiky. Pro trénování modelu byl použit velký dataset squigglů, které byly označeny podle genu, který reprezentují. Výsledky ukazují, že systém je schopen s určitou přesností identifikovat geny ve squigglech a že může být účinným nástrojem pro analýzu dat z nanopórové sekvenace.

KLÍČOVÁ SLOVA

Nanopore sekvenování, klasifikace genů, bioinformatika

ABSTRACT

Nanopore sequencing is a new and rapidly developing technology that allows for the direct sequencing of single-stranded DNA and RNA in real-time. The result of the sequencing is the so-called squiggle, which is a time series of current intensities as nucleotides pass through the nanopore. Identifying genes in these squiggles is a crucial step for the utilization of this data in genomic studies.

This bachelor's thesis focuses on the development and testing of a method for automatic gene identification in squiggles from nanopore sequencing. The aim was to create a system capable of quickly and accurately identifying genes in squiggles, thereby supporting further analysis and interpretation of nanopore sequencing data.

The method utilizes Convolutional Neural Networks (CNN), which have been successfully used in many other areas of bioinformatics. A large dataset of squiggles, labeled according to the gene they represent, was used to train the model. The results show that the system can identify genes in squiggles with a certain level of accuracy and can be an effective tool for nanopore sequencing data analysis

KEYWORDS

Nanopore sequencing, gene classification, bioinformatics

TALANIN, Nikita. *Identifikace genů ve squigglech ze sekvenace nanopórem*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2023, 45 s. Bakalářská práce. Vedoucí práce: Ing. Vojtěch Bartoň

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Nikita Talanin
VUT ID autora:	218498
Typ práce:	Bakalářská práce
Akademický rok:	2022/23
Téma závěrečné práce:	Identifikace genů ve squigglech ze sekvenace nanopórem

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné 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 autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych využil této příležitosti a poděkoval svému vedoucímu práce, Ing. Vojtěchu Bartoňovi, za jeho neocenitelnou pomoc a podporu během tvorby této práce. Jeho hluboké pochopení tématu a nekonečná trpělivost měly nezměrný vliv na můj výzkum. Jsem velmi vděčný za jeho nasměrování, za trpělivost při odpovědích na mé dotazy a za to, že mi dal prostor pro vývoj mé vlastní kreativity a myšlenek. Ing. Vojtěchu Bartoni, vaše mentorství bylo pro mě nesmírně cenné a budu si toho vážit i do budoucna. Děkuji vám za vaši inspiraci a vedení.

Obsah

Seznam obrázků

Úvod

Sekvenování je proces stanovení pořadí nukleotidů v celém genomu nebo jen v určitém úseku DNA. První metody sekvenování představili Sanger a Coulson v sedmdesátých letech 20. století. V současné době se jím říká „sekvenování první generace“. Jako první metody měly celý rozsah nevýhod a nedokázaly splnit všechny požadavky vědecké společnosti. Proto rozvoj a zkoumání nových technik neustále pokračoval. Druhá generace sekvenátorů neboli „next generation“ už měla výrazně lepší vlastnosti a byla používána v obou projektech The Human Genome Project. Metody druhé generace ale měly omezení v délkách jednotlivých čtení, proto stále zůstával požadavek na zlepšení a vývoj dalších technik. Nejmodernějším metodám se říká „third generation“ a patří k nim techniky, kterým pro sekvenování stačí jen jedna molekula a dokážou z této molekuly stanovit bázi jednu po druhé. Za necelých 50 let se sekvenování významně rozvíjelo, a to jak ze strany přesnosti, tak ze strany rychlosti a ceny sekvenování. Za poslední dobu sekvenování znamenalo revoluci v oblasti přírodních a biologických věd. Moderní rozsah využití sekvenačních technik je vskutku široký, včetně genetiky, molekulární biologie, archeologie, kriminalistiky, medicíny, zemědělství, šlechtění a mnoha dalších.

Jedním z příkladů sekvenátorů třetí generace je Oxford Nanopore Sequencing. Tato metoda je založena na měření proudu iontů při průchodu molekuly DNA přes pór v umělé membráně. Výstupem tohoto typu sekvenování je tzv. „squiggle“ proudový záznam, který obsahuje naměřené hodnoty proudu. Ze squiggle záznamu nelze ihned stanovit nukleotidovou sekvenci zkoumané molekuly DNA. Proto squiggle potřebuje následné zpracování zvané „basecalling“. Pod pojmem „basecalling“ se rozumí řada matematických a statistických operací pro stanovení pořadí nukleotidů ze squiggle záznamu. Moderní systémy vždy používají algoritmy strojového učení a neuronoví sítě pro provedení basecallingu.

Tato bakalářská práce bude věnována zejména programovacím algoritmům pro analýzu squiggle záznamů ze sekvenování bakterie Klebsiella pneumoniae. Bude navržen algoritmus, který pomocí neuronových sítí dokáže identifikovat určité geny v neznámých squigglech na základě učení z už zpracovaných a určených squigglů.

1 Sekvenování DNA

1.1 DNA

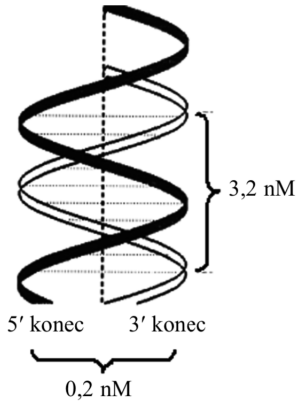
DNA neboli deoxyribonukleová kyselina je nukleová kyselina, která je hlavním nositelem genetické informace ve většině organismů, až na některé výjimky např. RNA-víry. DNA je biologický makromolekulární polymer a skládá se z nukleotidů. Všechny nukleotidy mají podobné složení a skládají se z:

- zbytku kyseliny ortofosforečné
- pětiuhlíkatého cukru (pentózy)
- dusíkaté báze

Podle druhu zastoupené pentózy rozlišujeme kyselinu deoxyribonukleovou (DNA) a kyselinu ribonukleovou (RNA).

V DNA jsou zastoupeny zejména čtyři druhy nukleotidů, které se od sebe liší dusíkatými bázemi. Právě tyto báze nebo spíše jejich pořadí je podstatné pro přenos genetické informace. Rozlišují se purinové nukleové báze, kam patří adenin a guanin, a pyrimidinové báze, cytosin a thymin.

Reální trojrozměrný dvoušroubovicový model molekuly DNA představili v roce 1953 James D. Watson a Francis Crick s použitím rentgenové difrakční analýzy. [1]. Rentgenová difrakční analýza prokázala, že sekundární struktura molekul DNA je dvojitá šroubovice a určila její rozměry. Podobný pokus o rok dřív provedli Rosalind Franklinová a Raymond Gosling. Bylo zjištěno, že poměr purinových a pyrimidinových bází je 1. Naopak obsah GC se u organismů může značně odlišovat. [1]

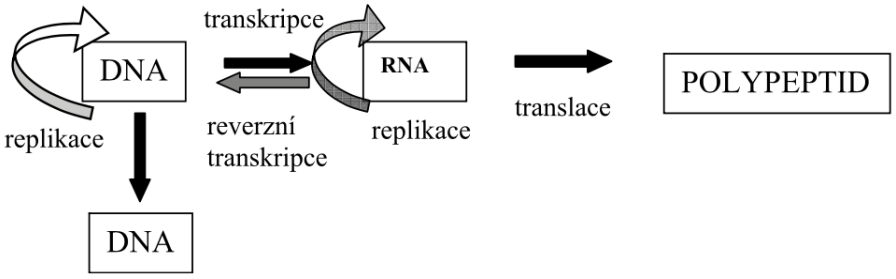


Obr. 1.1: Model a rozměry molekuly DNA. (upravené z [1])

První úspěšnou izolaci DNA provedl švýcarský lékař Friedrich Miescher v roce 1869, když v nemocnici zkoumal složení hnisu z obvazu. Nukleové kyseliny izoloval z bílých krvinek a pojmenoval je nukleiny. Obecné chemické složení nukleových kyselin

popsal Phoebus Levine jen na začátku 20. století. Ale o skutečné funkci nukleových kyselin, ale jejich význam v přenosu informací mezi generacemi se objevil až o 40 let později v experimentech Oswalda Averyho, Colina MacLeoda a Maclyna McCarta.

O čtyři roky později Francis Crick představil centrální dogmu molekulární biologie, které obsahovalo vzájemné vztahy mezi DNA, RNA a proteiny. [1] Po doplnění o současné znalosti centrální dogma umožňuje podat základní přehled o funkcích a způsobech přenosu genetické informace, jak je znazorněno na následujícím schématu.



Obr. 1.2: Schéma centrálního dogmatu genetiky (upravené z [1])

1.2 Sekvenování DNA

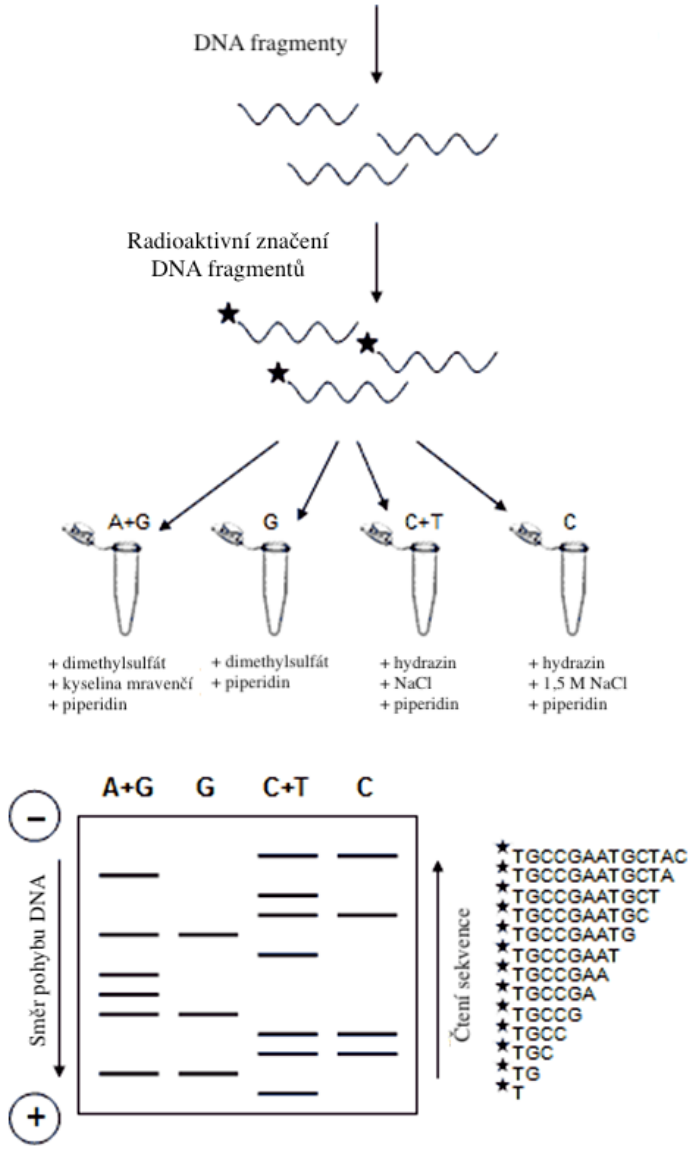
Sekvenování DNA je zobecněný termín pro biochemické postupy, jejichž cílem je zjišťování přesného pořadí nukleotidů v molekule deoxyribonukleové kyseliny. První sekvenování krátkého úseku DNA provedl Ray Wu z Cornellovy univerzity v roce 1970, celý výzkum mu trval 3 roky a stanovil pořadí jen 12 nukleotidů z okrajových regionu genomu víru.[6] Pro sekvenování DNA, které patří v současnosti mezi nejrozšířenější způsoby analyzování biologického materiálu, byl vyvinut poměrně velký počet různých technologií a metod. Používané techniky pro sekvenování DNA můžeme rozdělit na tři generace. Metody sekvenování druhé generace (neboli next generation sequencing) způsobily do jisté míry revoluci v genetice, a to jak z pohledu vědeckých otázek, na něž lze hledat řešení, tak z pohledu zcela nových technik výzkumné práce.

1.2.1 První generace metod sekvenování

Maxamova–Gilbertova metoda

Allan Maxam a Walter Gilbert v roce 1977 vyvinuli svou metodu chemického sekvenování s použitím radionuklidu. Ke konci krátkého úseku DNA přidává radionuklid,

většinou radioaktivní izotop fosforu. Vzorky jsou stejně jako v Sangerově metodě rozděleny do čtyř skupin. Každá skupina bude vystavena působení různých chemikálií, které budou štěpit molekulu DNA v určitých místech a základě nukleové báze, která se v tomto místě nachází. Poté se elektroforézou v polyakrylamidovém gelu rozštěpené úseky seřadí podle délky. Po použití filmu citlivého na rentgenové záření jsou všechny sekvence s radioaktivně označeným koncem patrné jako proužky na autoradiogramu. Podle pozice těchto proužků v porovnání se sousedními se dá odvodit původní umístění bází v DNA. [7]



Obr. 1.3: Schéma Maxamovy–Gilbertovy metody sekvenování (převzato z [2])

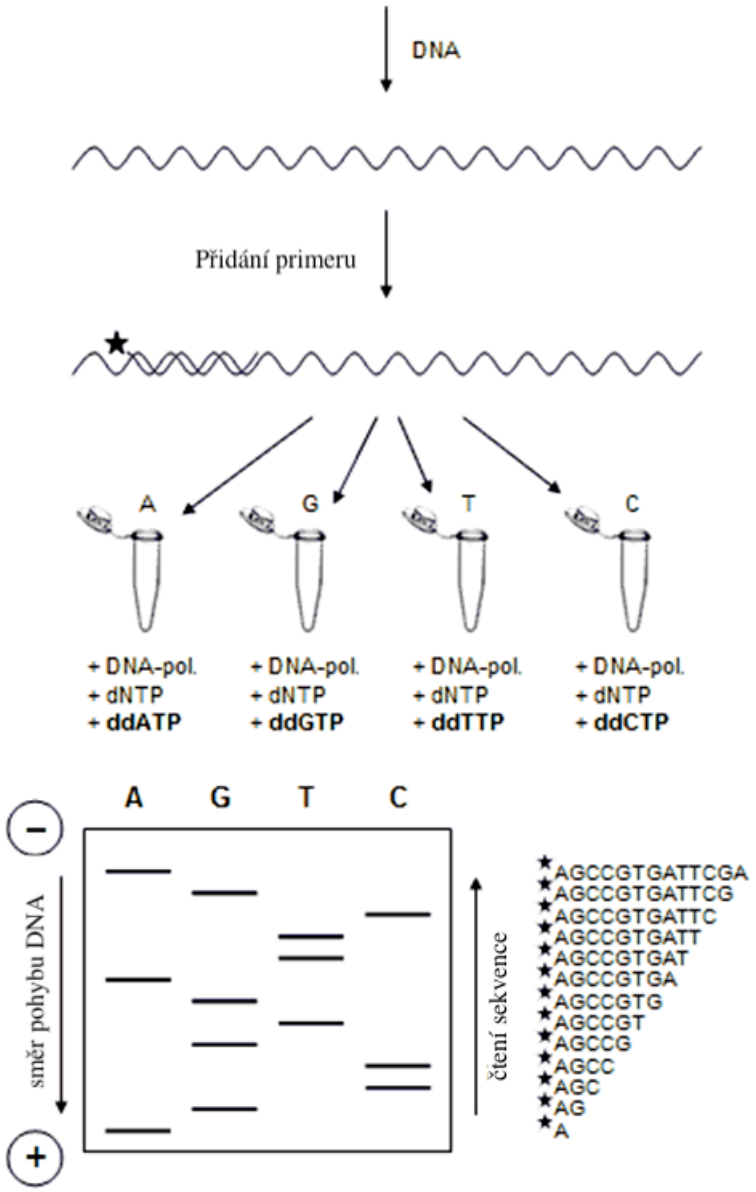
Sangerovo sekvenování

V roce 1977 Sanger s týmem spolupracovníků popsal novou metodu sekvenování oligonukleotidů pomocí enzymatické polymerace. Tato metoda z počátku byla známa jako metoda ukončení řetězce (chain termination). Metoda spočívala v katalyzované enzymatické reakci, která polymerizuje fragmenty DNA, komplementární k původní neznámé DNA.

Na začátku P-primer je nasazen na určitý úsek DNA (je komplementární tomu úseku) a slouží jako začátek polymerizace. V přítomnosti DNA-polymerázy od primeru pokračuje normální katalytická polymerizace a prodloužení nového komple-

mentárního řetězce. Tato reakce probíhá ve čtyřech různých zkumavkách, které se liší přítomností modifikovaného nukleosidu určitého typu. Těmto nukleosidům se říká terminátory neboli dideoxynukleosidtrifosfáty (ddNTP). Tyto terminátory různých typů (ddATP, ddCTP, ddGTP, ddTTP podle adeninu, cytosinu, guaninu a thyminu) zaprvé neumožnují následné navázaní nukleosidů a tím zastavují polymerizaci a zadruhé jsou různě fluorescenčně zabarvené.

Ve výsledku v každé zkumavce budou DNA řetězce o různých délkách, které mají stejné 5-konce (P-primery) a opačný 3-konec, ten byl určen a zastaven použitým terminátorem v dané zkumavce. Pomocí elektroforézy dále lze seřadit řetězce dle velikostí a z fluorescenčního zabarvení stanovit typ nukleosidu, který je umístěn na této pozici. [7]



Obr. 1.4: Schéma Sangerovy metody sekvenování (převzato z [2])

1.2.2 Next generation metody sekvenování

Většina sekvenačních metod tzv. druhé generace (Next Generation Sequencing, NGS) je zase založena na principu syntézy DNA podle templátu, ale na rozdíl od Sangerovy metody nebo Maxamovy–Gilbertovy metody jsou schopny detekovat přidávání bází jednu po druhé a zároveň sekvenovat tisíce až miliony rozdílných molekul DNA najednou. [8]

Jejich hlavní nevýhodou oproti Sangerovu sekvenování je krátká maximální délka výsledných sekvencí, která se dnes obvykle pohybuje zhruba od 100 až po 500 bází

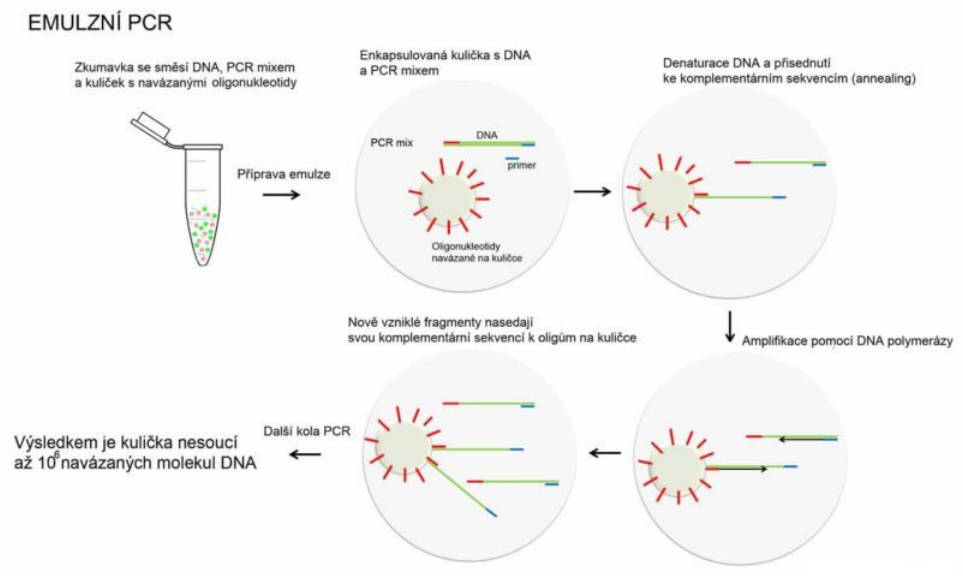
(Sangerova metoda nabízí až 1 000 bází). Nevýhodu představuje i menší přesnost a častější chyby při čtení DNA. [8]

U všech těchto metod je DNA nejdříve „nastříhána“ na relativně krátké části a na jejich konci je přilepen adaptér – velmi krátká molekula DNA o přesně dané sekvenci. Slouží k následnému navázání sekvenovaného úseku DNA na pevný povrch. Takto upravené DNA se říká sekvenační knihovna. Po uchycení DNA pomocí adaptéru na povrchu, na kterém bude docházet k sekvenaci, je každý řetězec DNA namnožen (většinou se používá PCR reakce), čímž vznikne skupina neboli klastr identických molekul DNA koncentrovaných v jednom místě (sekvence adaptéru je komplementární ke krátkým řetězcům DNA uchyceným na sekvenačním povrchu). Toto namnožení slouží k zesílení výsledného signálu, což umožní jeho zachycení kamerou, neboť signál z pouhé jedné molekuly DNA by nebyl dostatečně silný.

454 Roche

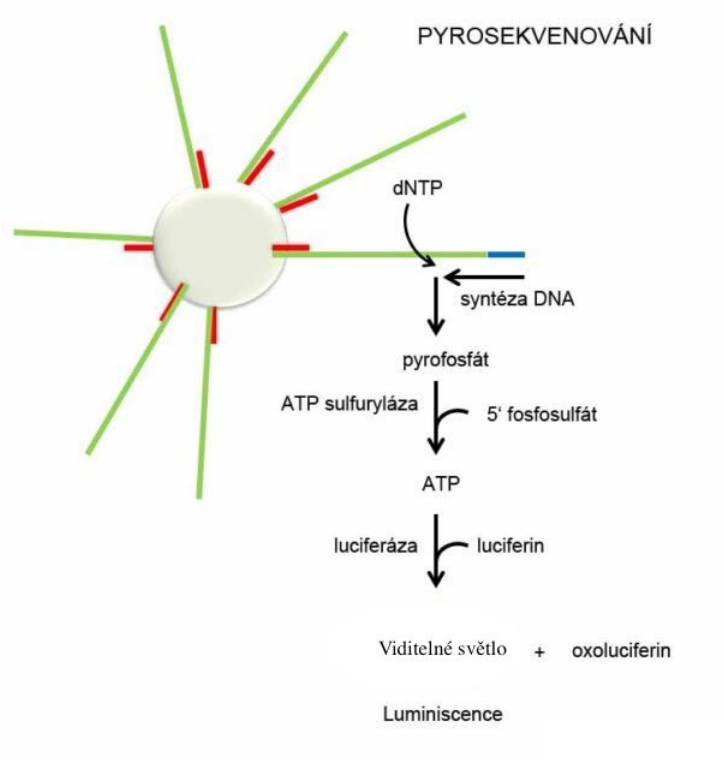
První komerční realizací metody druhé generace byla tzv. 454 sekvenování. Tuto techniku objevil a publikoval v roce 2005 Jonathan Rothberg. Praktická realizace měla do praxe velký komerční a vědecký úspěch. Nejmodernější verze dokazovaly analyzovat více než 1 milion molekul zároveň s délkou čtení kolem 700 až 1000 bází. [9]

Po fragmentaci DNA, přidání adapterů a denaturaci dvoušroubovice jsou úseky amplifikovány emulzní PCR reakcí. Pomocí adapterů jsou pak fragmenty nasazeny na speciální kuličky o velikosti 28 mikrometrů. Každá kulička je pak zapouzdřena do jamek na pikotitrační destičce, které obsahují všechny potřebné enzymy pro pyrosekvenování.



Obr. 1.5: Schéma 454 Roche sekvenování (převzato z [2])

Pyrosekvenování se zakládá na principu, při kterém během přidání každé nové báze do rostoucího řetězce DNA se uvolní molekula zvaná pyrofosfát (PPI). Uvolněný PPI se potom stane součástí několika na sebe navazujících enzymatických reakcí, na jejichž konci čeká enzym luciferáza. Ten vydá světelný záblesk, který lze zachytit vysoce citlivou kamerou. Při 454 sekvenování je v určitém momentě přidán do reakční směsi vždy pouze jeden typ báze a v okamžiku, kdy je tato báze vložena do rostoucího řetězce DNA, dojde přes uvolněný pyrofosfát a luciferázu ke světelnému záblesku. Kamera snímá celou destičku a podle toho, která komůrka se rozsvítí, pozná, kde proběhlo přidání báze, a podle intenzity světla kolik bází bylo přidáno najednou. Čtyři typy nukleotidů jsou do směsi přidávány jeden po druhém a mezi jednotlivými kroky dochází k odstranění přebytečných nukleotidů. Tím se zajistí, že v reakční směsi je vždy přítomen jediný typ nukleotidu. Počítač následně analyzuje záznam a podle světelných signálů z každé komůrky vytvoří výsledné sekvence odpovídající templátu DNA v jednotlivých komůrkách. [7]



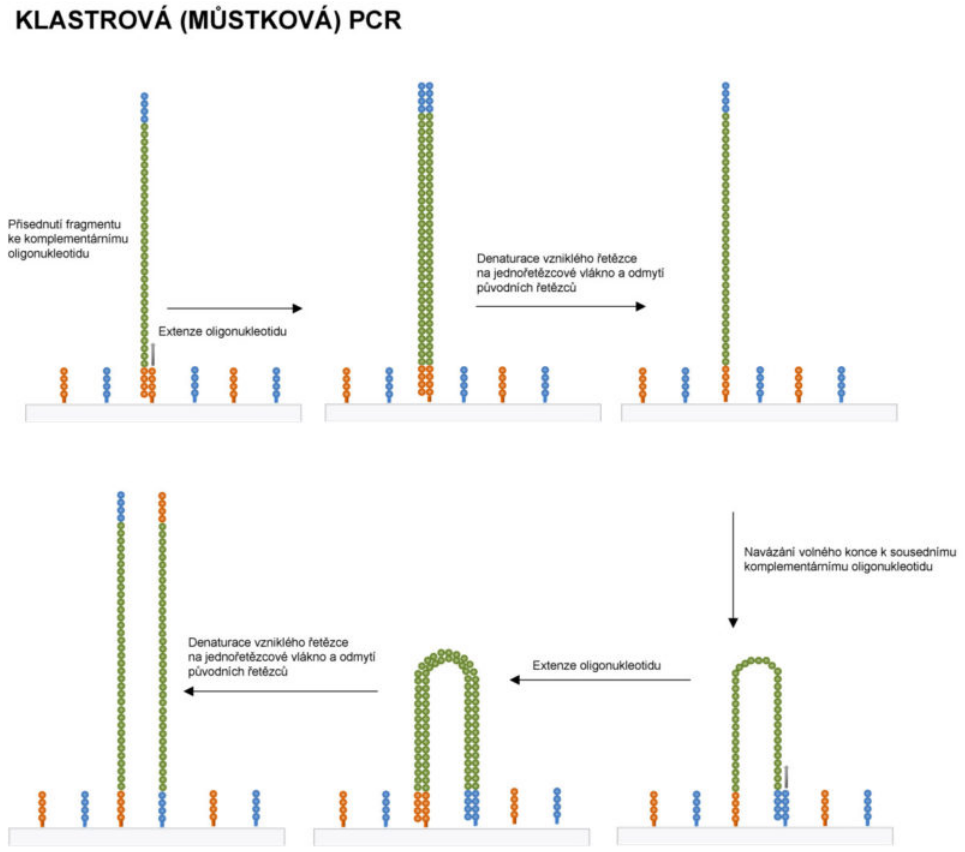
Obr. 1.6: Princip pyrosekvenování (převzato z [2])

Illumina

V současné době je nejpoužívanější metodou druhé generace technologie od společnosti Illumina. Hlavní nevýhodou techniky je používání kratších sekvencí, délka čtení se pohybuje kolem sta bází.

Při sekvenování technikou Illumina jsou pomocí adaptérů jednotlivé nastříhané molekuly DNA přichyceny na malou destičku. Každá molekula se pak namnoží (PCR reakce) tak, že na destičce vznikne velký počet klastrů, kde každý klastr obsahuje jen vzájemně identické molekuly DNA.

Vlastní sekvenační proces pak využívá podobného mechanismu jako Sangerovo sekvenování, kdy jsou do rostoucího řetězce zařazeny báze s navázanou fluorescenční barvou (každý nukleotid má specifickou barvu), které syntézu zastaví. Oproti Sangerovu sekvenování je tato blokace syntézy vratná. Po přečtení nově přidané báze vysoce citlivou kamerou dojde k enzymatickému odstranění jak fluorescenčního značení, tak blokující části molekuly a může proběhnout další kolo reakce, tedy přidání další báze. Kamera v každém jednotlivém kole syntézy řetězce DNA snímá signál z celé destičky a podle rozdílné fluorescence pozná, jaká báze byla přidána u každé z milionů skupin. Počítač pak opět analyzuje záznam krok po kroku a podle toho, jak se mění fluorescenční signál v rámci každé skupiny (skládající se z identických



Obr. 1.7: Klastrová (můstková) PCR reakce (převzato z [2])

molekul DNA), zrekonstruuje přesnou sekvenci molekul DNA v příslušné skupině. Illumina má velmi vysokou přesnost čtení DNA (uvádí se mezi 99 až 99,9 %). [8] Nejčastější chybou je špatně přečtená báze, tedy záměna jednoho nukleotidu. ¨

1.2.3 Metody třetí generace

Používání jen jedné molekuly pro celé sekvenování je charakteristickým znakem třetí generace sekvenátorů bez nutnosti aplikace PCR amplifikace nebo paralelního sekvenování. Proto je v praxi jejich hlavní výhodou cena za jednu bázi a rychlost provedení pokusů. Třetí generace taky má mnohem lepší charakteristiky týkající se délky čtení (20 až 200 tisíc bází), oproti druhé generaci, která dokazovala zvládnout délku sekvence jen o 1000 bází. Ale nevýhodou je relativně nižší přesnost čtení, společnosti PacBio a Oxford Nanopore uvádí přesnost čtení kolem 87-88 %. [**?**]

Nanopórové sekvenování

Využití nanopóru pro sekvenování bylo poprvé popsáno na konci 80. let několika americkými vědci. David Deamer, George Church a Hagan Bayley nezávisle na sobě

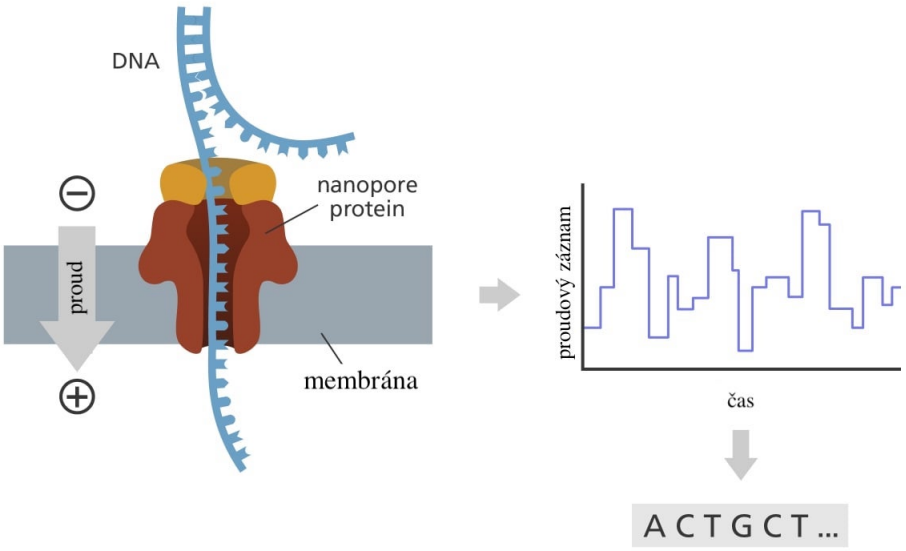
popsali sekvenování jedné molekuly DNA pomocí elektroforézy a membránových nanoporů. Hlavním principem nanoporového sekvenování je zabudování biologického nanopóru do umělé membrány se stálým elektrickým odporem a přiložení elektrického napětí na celý systém.[10]. Vzorkovány proudový záznam bude konečným pozorovatelným výstupem systému, který bude možný potom dekódovat na pořadí nukleotidů. Tomuto procesu se říká basecalling.

Před samotným sekvenováním je potřeba určitá příprava molekuly DNA podle jednoho ze standardních protokolů. Zaprvé molekula bude ostříhaná na fragmenty délky 8-20 kbp. Na oba konce budou připojeny adaptory a jeden konec navíc bude obsahovat motorový protein, který má za úkol rozepnout dvoušroubovici DNA a protlačit jednořetězcové vlákno přes nanopór. Také existuje možnost použití „hairpin" adapteru na jeden z konců původního fragmentu, který umožní sériový průchod obou DNA řetězců. To umožňuje sekvenovat dva komplementární DNA vlákna, čím lze zlepšit výslednou kvalitu a přesnost sekvenování. [10] [11] Využití nanopóru pro sekvenování bylo poprvé popsáno na konci 80. letech několika americkými vědci. David Deamer, George Church a Hagan Bayley nezávisle na sobě popsali používání nanopóru pro sekvenování jedné molekuly DNA. Hlavním principem je zabudování biologického nanopóru do umělé membrány se stálým elektrickým odporem a na celý systém je přiloženo elektrické napětí. Průchod a přítomnost molekuly DNA v nanopóru ovlivní elektrické vlastnosti celého systému a vyvolává výchylky v proudu procházejícím přes nanopór.[10] Vzorkovány proudový záznam bude konečným pozorovatelným výstupem systému, který bude možný potom dekódovat na pořadí nukleotidů. Tomuto procesu se říká basecalling.

Biologická membrána s nanoporem je obklopena roztokem elektrolytu. Membrána rozdělí roztok na dvě komory. Napětí je aplikováno na membránu a indukuje elektrické pole, které žene nabité částice, v tomto případě ionty, do pohybu. Uvnitř pórů molekula DNA zaujímá objem, který částečně omezuje tok iontů, což je pozorováno jako pokles iontového proudu. Na základě různých faktorů, jako je geometrie, velikost a chemické složení, se bude měnit velikost iontového proudu a doba trvání translokace. Na základě této modulace v iontovém proudu lze poté snímat a potenciálně identifikovat různé molekuly.[12]

MinION zařízení

Zařízení MinION měří 105 × 23 × 33 mm a váží 87 gramů a je nejmenším sekvenačním zařízením na aktuálním trhu [13]. MinION obsahuje 512 senzorů, každý senzor je propojen s čtyřmi nanopóry, ale lze sekvenovat jenom přes jeden nanopór současně. [10] Vzorkovací frekvence každého senzorů je kolem 3kHz. [14] Pro propojení s počítačem využívá standardní USB3 port a zároveň nemá vysoké hard-



Obr. 1.8: Průchod DNA molekuly nanopórem (upravené z [3])

warové požadavky na počítač. Specializovaný software, který se jmenuje MinKNOW, je multiplatformní (Windows, MacOS, Linux) a splňuje několik základních úkolů: sběr data z MinION, jejich analýza v reálném čase, streamování dat, sledování a identifikace vzorků, pro zajištění správného průběhu pokusů. Pro úspěšný průběh se doporučuje pouze DNA vysoké kvality (délka fragmentů větší než 30 kb). [13]

Výstup systému a datový format fast5

Soubory FAST5 jsou soubory formátu Hierarchical Data Format 5 (HDF5) se specifickým schématem definovaným společností Oxford Nanopore Technologies (ONT) pro ukládání nezpracovaných dat aktuálního signálu generovaných ze zařízení ONT. Na rozdíl od fasta a fastq, fast5 je binární formát, nelze ho otevřít v klasickém textovém editoru a není „human-readable“ formátem. Existují dva typy FAST5: single-FAST5 a multi-FAST5 (poprvé se objevil v září 2018). Soubor multi-FAST5 obsahuje několik čtení v jednom souboru, zatímco jeden soubor single-FAST5 obsahuje pouze jedno čtení. Všechny moderní ONT sekvenátory již používají jenom multi-FAST5 formát.

Formát HDF5

HDF je zkrátka pro hierarchický datový formát (Hierarchical Data Format) a svou strukturou připomíná standardní souborový systém. Stejně jako existují úrovně adresářů neboli složek a souborů v souborovém systému, soubor FAST5 (HDF5) obsahuje skupiny (groups) a datasety. HDF5 objekt je společný pojem pro skupiny a

Obr. 1.9: Zařízení MinION (upravené z [4])

datasety. Skupiny obsahují další skupiny nebo datasety, kde každý dataset už obsahuje homogenní, případně vícerozměrný vektor dat. Objekty HDF5 mohou volitelně obsahovat atributy, což jsou páry klíč–hodnota. Atributy jsou metadata a obsahují další informace o skupině, jsou stejně jak datasety uloženy ve skupinách. HDF5 umožňuje mít přístup k určitému objektu a pracovat s informací po částech bez potřeby extrahovat ostatní data. Díky tomu je HDF5 tak vhodný pro vědecké programování, má flexibilní schéma a dobře zvládá vícerozměrné bloky dat.

Skupiny HDF5 organizují objekty HDF5. Každý soubor HDF5 obsahuje kořenovou skupinu, která může obsahovat další skupiny nebo odkazy na jiné objekty HDF5, tyto objekty mohou být teoreticky ve stejném nebo v jiném souboru HDF5. Odkazy jsou jako cesty k adresářům/souborům v souborovém systému. Odkazy mohou být absolutní, relativní nebo dokonce symbolické. Práce se skupinami a členy skupin (objekty HDF5) je v mnoha ohledech podobná práci s adresáři a soubory v UNIXu.

Datasety HDF5 organizují a obsahují datové hodnoty. Dataset se skládá z metadat (datový typ, velikost dat, technika komprese atd.), která data popisují, a z dat samotných. V každém čtení v rámci souboru FAST5 jsou nalezeny dva datasety: skupina Raw obsahuje nezpracovaný proudový signál, skupina Analyses obsahuje data FASTQ (pouze v případě, že byl proveden base-calling).

Atributy lze volitelně přidat k objektům HDF5. Atributy mají dvě části: název a hodnotu. Atributy jsou přístupné otevřením objektu, ke kterému jsou přidány;

proto nejsou nezávislými objekty. Atribut má obvykle malou velikost a obsahuje podrobnosti o objektu, ke kterému je připojen.

Datový prostor (dataspace) HDF5 musí být definován před definováním datasetu HDF5 nebo atributů. Datový prostor definuje některá metadata, jako je velikost a tvar datasetu nebo atributy dat Datový prostor (dataspace) HDF5 musí být definován před definováním datasetu HDF5 nebo atributů. Datový prostor definuje některá metadata, jako je velikost a tvar datasetu nebo atributy dat (tj. počet dimenzí a velikost každé dimenze vícerozměrného pole, ve kterém jsou data reprezentována).

Stejně jako u UNIXu adresáře a soubory, objekty v souboru HDF5 jsou často popsány uvedením jejich úplných (nebo absolutních) názvů cesty.

- / označuje kořenovou skupinu.
- /read_1 znamená člena kořenové skupiny s názvem read_1.
- /read_1/Raw znamená objekt Raw skupiny read_1, který je zase členem kořenové skupiny.

Pod každou read skupinou jsou následující skupiny:

- /Raw
- /channel_id
- /context_tags
- /tracking_id
- /Analyses

Raw skupina obsahuje jeden dataset a sedm atributů. Dataset je nezpracovaný proudový záznam, což je řada 16bitových celých čísel (HDF5 Datatype = H5T_STD_I16LE). Jedná se o celočíselné hodnoty přímo pocházející z procesu sběru dat (analogově digitální převodník). Tento nezpracovaný signál lze převést na hodnoty pikoampér (pA) pomocí atributů dostupných ve skupině channel_id.

Níže je uvedeno sedm atributů ze skupiny Raw s popisem každého atributu, vzorovými hodnotami a datovým typem.

1. read_number Unikátní pořadové číslo v každém kanálu počítané od nuly. Všimněte si, že ne všechna vygenerovaná čtení jsou čtení „řetězce“, ale pouze čtení řetězce se zapisují do konečného souboru fast5, takže některá čísla přečtení mohou chybět. Příklad hodnoty: 663. Typ dat: 32bitové celé číslo.
2. read_id Unikátní identifikátor pro čtení. Toto je univerzální jedinečný identifikátor (UUID) a měl by být jedinečný pro jakékoli čtení z jakéhokoli zařízení. Příklad hodnoty: 00013387-ed1c-4407-91c4-10d55578e969. Typ dat: Řetězec
3. start_time Čas začátku čtení. Jednotkou pro start_time je „počet vzorků signálu“, takže start_time musí být vydělen vzorkovací frekvencí (atr. Read_xxxx /channel_id /sampling_rate), aby se získal počáteční čas v sekundách (tj. čas od zahájení běhu). Příklad hodnoty: 1054346. Typ dat: 64bitové celé číslo

4. duration Doba trvání čtení. Jednotkou délky čtení je také ‚počet vzorků signálu'. Příklad hodnoty: 8457. Typ dat: 32bitové celé číslo.
5. start_mux Nastavení MUX pro kanál při zahájení čtení, které určuje jaký z 4 nanopórů byl použít pro dané čtení. Může mít hodnoty od 1 do 4. Příklad hodnoty: 3. Typ dat: 8bitové celé číslo.
6. median_before Odhadovaná střední aktuální úroveň signálu těsně před zahájením čtení. Ve většině případů to lze použít jako odhad úrovně otevřených pórů (open pore). Stav otevřených pórů je stav, kdy uvnitř póru není žádný řetězec. Příklad hodnoty: 205.359. Typ dat: 64bitová s pohyblivou řádovou čárkou.
7. end_reason Ukazuje, jestli čtení bylo zastaveno nebo úspěšně dokončeno

Channel_id skupina obsahuje1 atributy, které jsou relevantní pro kanál, který sekvenoval dané čtení. Skupina channel_id má následující atributy:

1. channel_number Číslo kanálu, ve kterém probíhalo čtení. Příklad hodnoty: 411. Datový typ: String
2. digitisation Digitalizace je počet úrovní kvantizace v analogově digitálním převodníku (ADC). To znamená, že pokud je ADC 13bitový, digitalizace je 8192. Příklad hodnoty: 8192,0. Datový typ: 64bitová s pohyblivou řádovou čárkou
3. offset Chyba offsetu ADC. Tato hodnota se přičte při převodu signálu na pikoampéry. Příklad hodnoty: 8,0. Datový typ: 64bitová s pohyblivou řádovou čárkou
4. range Celý rozsah měření v pikoampérech. Příklad hodnoty: 1454.91. Datový typ: 64bitová s pohyblivou řádovou čárkou
5. sampling_rate Vzorkovací frekvence ADC, tj. počet datových bodů shromážděných za sekundu (v Hertzech). Příklad hodnoty: 4000. Datový typ: 64bitová s pohyblivou řádovou čárkou

Z těchto atributů lze digitalizaci, offset a rozsah použít k převodu původního signálu ve skupině Raw na hodnoty pikoampérového proudu:

$$signal\_v\_pico\_amperech = (raw\_signal + offset) * rozsah/digitalizace$$

Skupina context_tags má globální atributy, které popisují běh sekvenování. Atributy ve skupině context_tags jsou uvedeny níže s krátkým popisem. Datový typ hodnoty všech atributů uvedených ve skupině context_tags je String.

1. barcoding_enabled Označuje, zda je během basecallingu povoleno demultiplexování. Příklad hodnoty: 0.
2. experiment_duration_set Označuje dobu trvání experimentu v minutách. Příklad hodnoty: 4320
3. experiment_type Označuje typ experimentu. Příklad hodnoty: genomic_dna
4. local_basecalling Označuje, zda je basecalling povolen nebo není (nastaveno na 1 nebo 0). Příklad hodnoty: 0.

5. package Tento atribut se vztahuje k balíčku Bream. Příklad hodnoty: bream4.
6. package_version Verze balíčku. Příklad hodnoty: 6.0.7.
7. sample_frequency Vzorkovací frekvence. Obvykle je stejná jako vzorkovací frekvence ve skupině channel_id. Příklad hodnoty: 4000.
8. sequencing_kit Sekvenční sada vybraná uživatelem v GUI. Příklad hodnoty: sqk-rbk004. [https://store.nanoporetech.com/sample-prep.html]

Skupina tracking_id má globální atributy pro běh sekvenování a sekvenační zařízení. Tyto jsou většinou pro interní použití ONT. Datový typ všech atributů uvedených ve skupině tracking_id je String.

1. asic_id Application Specific Integrated Circuit identifier (ASIC) průtokové cely (jedinečné číslo čipu). Umožňuje sledování šarží čipů. Příklad hodnoty: 483830637.
2. asic_id_eeprom Identifikátor EEPROM paměti průtokové cely ASIC. Příklad hodnoty: 5741023
3. asic_temp Teplota ve stupních Celsia čipu ASIC na začátku běhu sekvenování. Příklad hodnoty: 29.512911
4. asic_version Použitá verze ASIC. Příklad hodnoty: IA02D
5. auto_update Zda je automatická aktualizace v MinKNOW povolena nebo ne. Příklad hodnoty: 1.
6. auto_update_source Odkaz na zdroj aktualizací MinKNOW. Příklad hodnoty: https://mirror.oxfordnanoportal.com/software/MinKNOW/
7. bream_is_standard Bream je jedním ze softwarů pro řízení sekvenování. Atribut označuje, zda Bream je používán jako standardní balíček pro řízení průběhu sekvenování. Příklad hodnoty: 0
8. configuration_version Verze konfigurace systému v MinKNOW včetně skriptů experimentu. Příklad hodnoty: 4.0.13
9. device_id Sériové ID MinION. Příklad hodnoty: MN31868
10. device_type Typ zařízení, buď MinION, PromethION nebo GridION. Příklad hodnoty: minion
11. distribution_status Stabilní stav, developer, alfa nebo beta. Příklad hodnoty: stable
12. distribution_version Verze MinKNOW. Příklad hodnoty: 20.06.4
13. exp_script_name Název skriptu experimentu, který se spouští, spolu s volitelnými parametry, které jsou mu předány, na základě toho, jaké sady jsou v MinKNOW vybrány pro sekvenování. Příklad hodnoty: sequencing /sequencing_MIN106_DNA : FLO-MIN106 : SQK-RBK004
14. exp_script_purpose Účel skriptu experimentu. Například zda byl experiment skutečným sekvenováním nebo jen simulací. Příklad hodnoty: sequencing_run
15. exp_start_time Čas zahájení běhu sekvenování podle normy ISO 8601. Pří-

klad hodnoty: 2020-12-09T07:43:20Z

16. flow_cell_id Jedinečné ID průtokové cely, které používá ONT ke sledování metrik průtokové cely a záruky. Příklad hodnoty: FAL63964
17. flow_cell_product_code Typ průtokové cely (kód produktu průtokové cely a typu pórů). Příklad hodnoty: FLO-MIN106
18. guppy_version Verze Guppy používaná MinKNOW. Guppy je sada nástrojů pro zpracování dat, která obsahuje algoritmy basecalling Oxford Nanopore Technologies a několik funkcí bioinformatického následného zpracování. Příklad hodnoty: 4.0.9+92ae0933
19. heatsink_temp Teplota (ve stupních Celsia) chladiče na ASIC na začátku sekvenování. Příklad hodnoty: 33.992188
20. hostname Název hostname počítače/stroje, který provádí sekvenování. Příklad hodnoty: NB-UBMI-720
21. installation_type Typ instalace MinKNOW. Příklad hodnoty: nc
22. operating_system Operační systém a verze počítače provádějícího sekvenování. Příklad hodnoty: Windows 10.0
23. protocol_run_id Toto je identifikátor pro experiment (pouze v případě, že název zadaný uživatelem není jedinečný). Má stejnou hodnotu pro každý běh stejné experimentální skupiny. Příklad hodnoty: 8e890e0a-6f9f-4927-8512-116183e66b71
24. run_id Jedinečné ID běhu, které se bude pro každý běh lišit, a to i ve stejné experimentální skupině. Kdykoli MinKNOW spustí skript experimentu pro získávání dat, vygeneruje se nové run_id. Příklad hodnoty: fc10227bc1eb2daff39f4a40bc259a25054f14a6
25. sample_id Sample ID je jméno dané uživatelem pro tento vzorek. Příklad hodnoty: KP1179_KP1193_KP1231_KP1236_KP1272_KP128
26. usb_config Informace o spojení mezi průtokovou kyvetou a počítačem. Příklad hodnoty: MinION_fx3_1.1.1_ONT#MinION_fpga_1.1.0#bulk#Auto
27. version Verze MinKNOW. Příklad hodnoty: 4.0.4

Basecalling

V současné době se basecalling většinou provádí v cloudu Amazon pomocí služby Metrichor, nikoli na místním počítači propojeném se sekvenačním zařízením. [13] Tato služba analyzuje soubory FAST5 vygenerované sekvenačním softwarem a vrátí je s výsledky analýzy. Pracovní postup se skládá z několika kroků. Nejprve se analyzují hodnoty proudu, aby se určilo, které hladiny proudu odpovídají templátovému vláknu a které odpovídají komplementárnímu vláknu. Tato fáze je také potřebná k odstranění proudových hodnot souvisejících s adaptéry na molekule. Dále jsou pou-

žity statistické modely trénované ONT ke stanovení vztahu mezi 5-mery (případně 6-mery) a proudovými hladinami. Poté jsou vypočítány rozdíly mezi modelem a pozorovanými proudovými hodnotami. Nakonec lze pro každou molekulu získat dvě 1D sekvence, z nichž je, pokud možno zkonstruována jedna 2D sekvence. Po provedení basecallingu Metrichor klasifikuje čtení do dvou „tříd čtení“: „pass” a „fail“. [13] Proudový záznam ze souboru FAST5, metadata a data z basecallingu vrácená Metrichorem jsou spojena do souboru FAST5, který lze stáhnout do adresáře zvoleného uživatelem.

Postup basecallingu lze stručně popsat následovně.

1. Eventy na templátovému a komplementárním vláknu jsou zpracovávány zvlášť a poté použity pro 1D basecalling.
2. Pokud je poměr délky sekvence templátu k délce sekvence komplementu po basecallingu mezi 0,5 a 2,0, pokusí se o 2D-basecalling.
3. Pokud je 2D basecalling úspěšné, vytvoření 2D čtení.
4. Výpočet skóre kvality (Q-score) za účelem kvantifikace kvality 2D čtení.

FAST5 se umístí do adresáře „pass“, pokud má 2D čtení střední Q-skóre větší, než 9; zatímco všechny ostatní soubory FAST5 jsou umístěny do adresáře „fail“. Proto adresář „fail“ obsahuje soubory FAST5, které mají:

- Čtení se středním Q-skóre < 9 (kromě čtení templátů a komplementů);
- Čtení 1D templátů a komplementů, u kterých nebylo dosaženo 2D volání báze (buď je poměr příliš velký (>2,0)/příliš malý (<0,5) nebo 2D volání báze selhalo);
- Pouze čtení templátu.

Čtení klasifikované jako „pass“ lze normálně považovat za vysoce kvalitní čtení z experimentu.[13]

2 Dataset genomů z databáze

2.1 Klebsiella pneumoniae

Klebsiella pneumoniae je gramnegativní nepohyblivá tyčinkovitá bakterie. Tento druh bakterie je součastí bikrobiální flory gastrointestinálního traktu, respiračního ústrojí a na kůži, ve vnějším prostředí je přítomen v půdě a ve vodě. Tyto bakterie nejčastěji způsobují uroinfekce a pneumonie, mohou být příčinou sepse. Klebsiellové pneumonie jsou většinou lobární, často se vyskytují u pacientů s oslabeným imunitním systémem. U novorozenců můžou způsobovat bakteriální hnisavou meningitidu a sepse. Přenos je fekálně-orální, kontaktem i vzduchem.

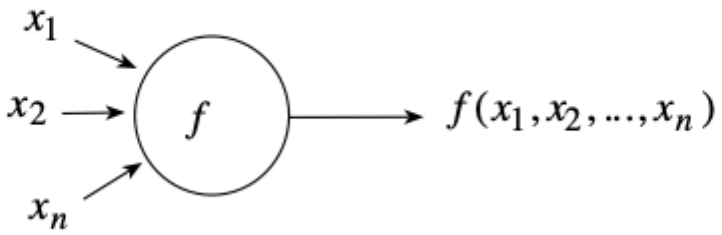
2.2 Popis databáze

Databáze obsahuje 1994 multi-FAST5 souborů. Byly analyzovány izoláty *K. pneumoniae* odebraných v období 09.2014 až 07.2019 především na Interní, hematologické a onkologické klinice FN Brno. Vysokomolekulární DNA byla extrahována pomocí MagAttract HMW DNAKit a k měření čistoty extrahované DNA bylo použito NanoDrop. Koncentrace DNA byla kontrolována fluorometrem Qubit 3.0 a pomocí Agilent 4200 TapeStation byla zkontrolována správná délka izolované DNA. K přípravě sekvenační knihovny pro 27 izolátů byla použita souprava Rapid Barcoding Kit, u zbývajících dvou izolátů byla použita souprava Ligation Sequencing 1D Kit. Sekvenování bylo provedeno pomocí sekvenační platformy MinION s průtokovými celami R9.4.1

3 Konvoluční neuronové sítě pro klasifikaci genů

3.1 Úvod do neuronových sítí

Neuronové sítě jsou v informatice a strojovém učení velmi populárním nástrojem, který je schopen řešit různé typy problémů. Neuronové sítě jsou inspirovány biologickými neurony. Matematickým analogiím zjednodušených neuronů se říká perceptrony. Perceptrony jsou schopny řešit pouze jednoduché matematické úlohy [5]. Každý perceptron přijímá vstupní signály, které jsou váženy a zkombinovány do výsledného výstupu, který se pak předává do další vrstvy. [5]. Trénování neuronové sítě se spočívá v úpravě těchto vah.



Obr. 3.1: Základní schéma perceptronu [5]

Mezi nejznámější a nejpoužívanější typy neuronových sítí patří dopředné neuronové sítě, rekurentní neuronové sítě a konvoluční neuronové sítě.

3.2 Základní principy neuronových sítí

Perceptrony jsou základními stavebními kameny neuronových sítí. Každý uzel, má vstupní signály a výstupní signál. Vstupní signály jsou zpravidla vážené, což znamená, že mají různou váhu v závislosti na důležitosti daného signálu pro výstup. Vážené vstupní signály jsou následně převedeny na výstupní signál pomocí aktivační funkce. Tento proces se opakuje pro každý neuron v síti a umožňuje sítí přizpůsobit se různým vstupům a úlohám, které má řešit. [15]

Pokud si každý uzel v umělé neuronové síti představíme jako primitivní funkci schopnou transformovat svůj vstup na přesně definovaný výstup, pak umělé neuronové sítě nejsou nic jiného než sítě primitivních funkcí. [5]

3.3 Konvoluční neuronové sítě

Každý druh neuronových sítí je určen, respektive lépe zvládá úlohy s různými cíli. Pro zpracování sekvenačních dat byla zvolena jednorozměrná konvoluční neuronová síť.

Jednorozměrná konvoluční neuronová síť (1D-CNN) je varianta konvolučních neuronových sítí (CNN) speciálně navržená pro zpracování jednorozměrných sekvencí dat, jako jsou časové řady, audiosignály nebo textové sekvence. Hlavním rozdílem s klasickou CNN sítí je, že konvoluce v 1D-CNN je aplikována pouze v jedné dimenzi - obvykle po časové ose. [16]

Jednorozměrná konvoluční neuronová síť (1D-CNN) je speciální třída konvolučních neuronových sítí (CNN), která je navržena k zpracování dat s jedním rozměrem. Tato síť se skládá ze tří klíčových částí: konvoluční vrstvy, pooling vrstvy (pooling layers) a plně propojené vrstvy (fully connected layers). Konvoluční vrstva detekuje místní vlastnosti vstupu pomocí malých jader. Pooling vrstva slouží k snížení rozměrů vstupu a pomáhá předejít přeučení. Plně propojená vrstva nakonec slouží k produkci výstupu sítě. Každá z těchto vrstev má své vlastní parametry, které se učí během tréninku sítě [16].

Základní principy těchto vrstev jsou popsány v následujících odstavcích:

- Konvoluční vrstva je základní stavební prvek konvoluční neuronové sítě, a to i v 1D-CNN. Vstupem do konvoluční vrstvy je jednorozměrná datová sekvence, jako je například časová řada, audiosignál nebo proudový signál. Konvoluční vrstva je vybavena filtry, které se skládají z konvolučních jader [16]. Konvoluční jádro (také známé jako filtr) je klíčovou součástí konvoluční vrstvy v konvoluční neuronové síti. Jedná se o malou matici vah, která je aplikována na vstupní data prostřednictvím konvoluční operace.

  V praxi konvoluční jádro "klouže"přes vstupní data a v každém kroku počítá skalární součin svých vah s odpovídajícími vstupními daty. Výsledkem je nová matice (v případě 1D-CNN, jednorozměrná sekvence), která reprezentuje vlastnosti vstupních dat, které byly detekovány konvolučním jádrem.

  Konvoluční jádra jsou typicky učena během procesu trénování sítě, kdy se jejich váhy aktualizují tak, aby co nejlépe předpovídaly cílové hodnoty. [15]

  Konvoluce je matematická operace, která mění dvě funkce do třetí. V kontextu 1D-CNN, konvoluce je aplikována na vstupní data a konvoluční jádro, což vede k nové sekvenci dat, která reprezentuje vlastnosti detekované jádrem [15]. Matematicky lze konvoluci popsat jako:

  $$(y * w)(t) = \int y(a)w(t-a)da, \tag{3.1}$$

  kde y je vstupní data, w je konvoluční jádro a * označuje konvoluční operaci.

- Po konvoluční vrstvě následuje pooling vrstva. Tato vrstva má za úkol snížit rozměry výstupu z konvoluční vrstvy tím, že provede operaci downsamplingu, například max-pooling nebo average-pooling. [15]]
- Na závěr je plně propojená vrstva (fully connected layer), která vytváří konečný výstup sítě. Tato vrstva je propojena se všemi neurony v předchozí vrstvě. Každý neuron v plně propojené vrstvě vypočítá vážený součet svých vstupů a přidá bias. Matematicky lze tento proces popsat jako:

$$y = Wx + b \tag{3.2}$$

kde y je výstup neuronu, W je matice vah, x je vektor vstupů a b je bias [15]. V plně propojené vrstvě je často použita nelineární aktivační funkce, jako je ReLU (Rectified Linear Unit) nebo softmax pro vícevrstvovou klasifikaci, aby se zajistilo, že výstup sítě není pouze lineární kombinací vstupů.

4 Výsledky studentské práce

4.1 Zpracování datasetu

4.1.1 Výběr vhodných genů z datasetu

Multilokusní sekvenční typizace (MLST) je metoda pro charakterizaci izolátů bakteriálních druhů pomocí sekvencí fragmentů obvykle šesti nebo sedmi „house-keeping“ genů. Pro každý „house-keeping“ gen mohou být sekvence o něco odlišné v rámci jednoho bakteriálního druhu. Jsou to odlišné alely a tyto alely definují pro každý izolát na každém z lokusů alelický profil nebo typ sekvence (ST).

Každý izolát druhu je proto jednoznačně charakterizován sérií sedmi celých čísel, která odpovídají alelám v sedmi house-keeping genech. V MLST je počet nukleotidových rozdílů mezi alelami ignorován a sekvencím jsou přidělena různá čísla alel, ať už se liší na jediném nukleotidovém místě nebo na mnoha místech. Důvodem je, že jediná genetická událost vedoucí k nové alele může nastat bodovou mutací (změnou pouze jednoho nukleotidového místa) nebo rekombinantní náhradou (která často změní více míst). Velkou výhodou MLST je, že sekvenční data jsou jednoznačná a alelické profily izolátů lze snadno porovnat s profily ve velké centrální databázi. Většina bakteriálních druhů má dostatečnou variaci v rámci house-keeping genů, aby poskytla mnoho alel na lokus, což umožňuje rozlišit miliardy odlišných alelických profilů pomocí sedmi „house-keeping“ genů.

Pro *Klebsiella pneumoniae* jsou stanoveny 7 house-keeping genů, jsou to: *rpoB*, *gapA*, *mdh*, *pgi*, *phoE*, *infB*, *tonB*. Vzorky sekvencí těchto genů byly staženy z knihovny NCBI. [17]

4.1.2 Zpracování datasetu

Zpracování FAST5 souborů se provádělo v jazyce Python pomocí knihovny h5py. Nejdříve ze souborů byly extrahovány base-calling sekvence a squiggle záznamy.

Hodnoty v extrahovaných squigglech neodpovídaly skutečným proudovým hodnotám. Nejprve bylo zapotřebí přidat offset ke každému vzorku signálů. Hodnoty offsetů byly uloženy v atributu „offset“ ve skupině „channel_id“ pro každé čtení zvlašť. Squiggle záznamy se potom byly převedeny na pikoampérové hodnoty, protože v souborech jsou uloženy jenom jim odpovídající kvantizační hladiny. Pro převod do pikoampérů se potřeboval rozsah měření a počet kvanizačních hladin. Atributy „range“ a „digitisation“ se taky ukládaly ve skupině „channel_id“. Vynásobením rozsahem a vydělením počtem kvantizačních hladin signal byl převeden na pikoampéry.

Poté bylo potřeba ze squiggle signálů vybrat ty časti záznamů, které reprezentovaly nalezené v databázi MLST geny.

Průchod každé báze přes pór trvá nějaký čas, zatím co vzorkovací frekvence snímače je pořad stejná. Proto každému nukleotidu vždy odpovídá několik vzorků squiggle signálu. Aby nebyla potřeba ukládat příslušnost každého vzorku k nějaké bázi, stačilo stanovit jenom vstup (neboli začátek průchodu pórem) pro každý nukleotid. Pro tento úkol byl využit atribut „Move“ ze skupiny „BaseCalled_template“. Tento atribut obsahuje seznam binárních hodnot: „1“ a „0“, kde „1“ znamená, že právě na této pozici se uskutečnil přechod do další báze, zatímco nulová hodnota znamená, že se v póru pořád nacházel stejný nukleotid jako u minulého kroku. Počet jedniček v seznamu je zřejmě rovný délce base-callingem stanovené sekvence. Atribut „Move“ je dvakrát kratší než délka sekvence, protože kontrola přechodu do nové báze se provádí na jednom ze dvou vzorků, délka tohoto kontrolního kroku byla uložena v atributu „block_stride“ ve skupině „basecall_1d_template“.

Každé čtení má na začátku nějaké zašumění, kde se žádný nukleotid ještě neměří, většinou o délce 250-300 vzorků. Tento úsek bylo třeba ostřihnout, informaci o přesné poloze vstupu prvního nukleotidu do póru obsahuje atribut „first_sample_template“.

Ve výsledku pro stanovení poloh nukleotidů ve squigglu bylo potřeba najít indexy jedniček v seznamu „Move“, vynásobit je dvěma, která reprezentuje jeden krok systému a zvlášť do seznamu uložit hodnoty ostřihnutého squigglu s těmito indexy. Tyto hodnoty budou reprezentovat proud při vstupu nové báze do póru. Délky tohoto uloženého seznamu s polohami bází bude rovná délce odpovídající jim sekvence.

Pro vyhledávání MLST genů v souborech bylo zapotřebí vytvořit databáze ve formátu multi-fasta, které by obsahovaly odvedené sekvence každého čtení. Jako popis každé sekvence databáze měly ID čtení, kterému tato sekvence patřila. Pro každý soubor byla vytvořena zvláštní databáze pojmenována podle nazvu FAST5 souboru.

Vyhledávání poloh MLST genů se provádělo BLASTem. Biopython knihovna má algoritmus pro realizaci BLAST vyhledávání vůči lokální databázi. Provedením tohoto algoritmu v 1994 FAST5 souborech bylo nalezeno 53480 úseků house-keeping genů. Výsledky BLAST analýzy se ukládají ve formátu „.xml“ a obsahují taky informace o polohách začátků a konců nalezených úseků a také popisy fasta sekvencí, kde ty úseky byly nalezeny. V tomto případě popisy sekvencí obsahovaly ID čtení. Po provádění algoritmu se vygenerovaly 13951.xml souborů s výsledky BLASTU.

Knihovna Biopython zahrnuje modul NCBIXML, který dokáže zpracovat xml výsledky analýzy a vyhledávat potřebné tagy. Tagy „sbjct_start“ a „sbjct_end“ reprezentují začátky a konce (indexy) úseků v databází, když „title“ poskytuje informaci o nazvu čtení, kde ten úsek byl nalezen.

V předchozím kroku polohy každého nukleotidu byly uloženy do zvláštního seznamu, a tento seznam měl stejnou délku jako sekvence. Proto tagy „sbjct_start“ a „sbjct_end“ se mohly zároveň využit jako indexy v seznamech poloh každého nukleotidu v squigglech. A s těmito indexy bylo pouze zapotřebí přejít na původní squiggly a uložit zvlášť ty časti, které reprezentují nalezeny úseky MLST genů.

Potom výsledky analýzy ze všech souborů bylo zapotřebí uložit dohromady v jednom souboru. HDF5 byl zvolen jako nejvhodnější formát pro ukládání velkých objemů dat a kvůli možnosti používání atributů jako užitečného nástroje pro ukládání doplňujících a vedlejších informací. Soubor má následující strukturu:

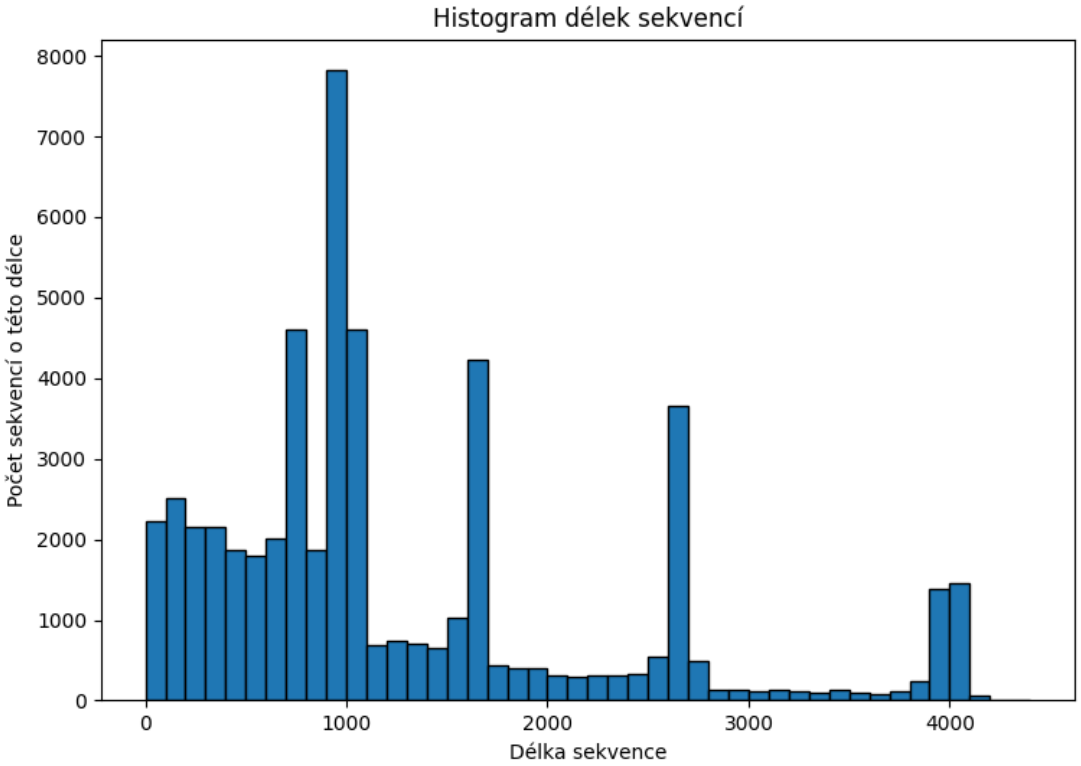
- Hlavní skupina má název typu „alignment_id“ a obsahuje 4 hlavní atributy:
  1. cestu k souboru, kterému patřila daná sekvence;
  2. název souboru FAST5;
  3. ID čtení, kde shoda byla nalezena;
  4. název MLST genu, který byl nalezen v daném úseku.

  Dál do skupiny se ukládají 4 datasety, ty jsou:
  - sequence_of_alignment“ obsahuje sekvenci shody a taky dva atributy „seq_start_pos“ a „seq_end_pos“, kam se ukládají indexy počátku a konce úseku v celé sekvenci tohoto čtení,
  - full_sequence_of_read“ obsahuje celou sekvenci čtení,
  - „squiggle_of_alignment“ obsahuje squiggle záznam nalezené shody a taky dva atributy „squiggle_start_pos“ a „squiggle_end_pos“, kde jsou uloženy indexy počátku a konce v celém squiggle záznamu celého čtení,
  - „full_squiggle“ obsahuje squiggle záznam celého čtení.

  Tento soubor se bude následně využívat jako trénovací dataset.

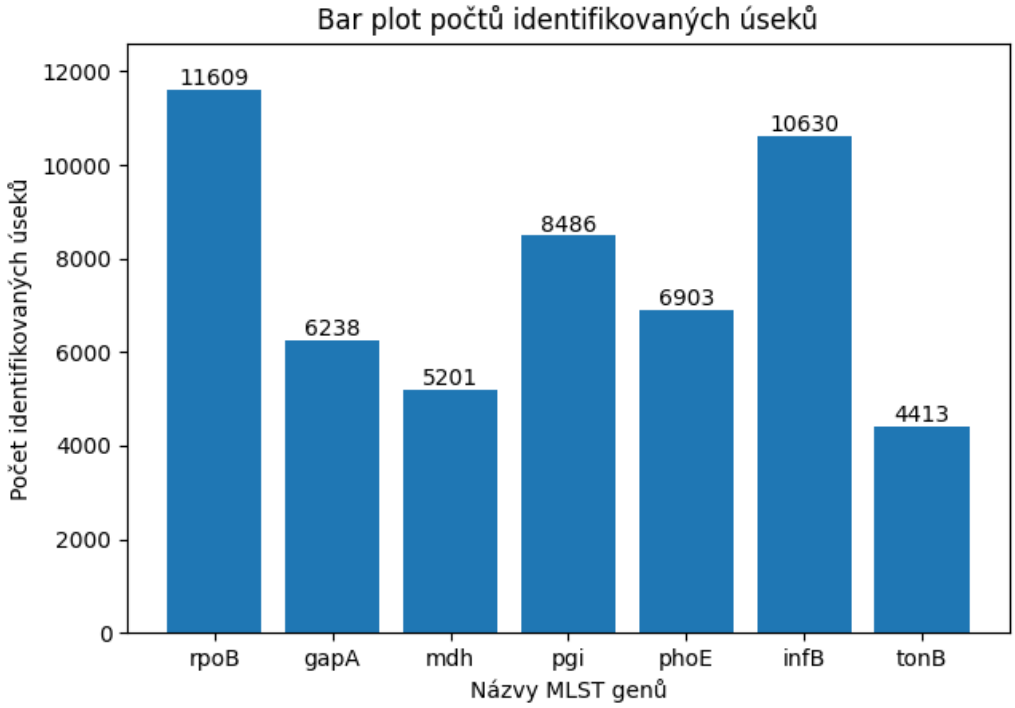
4.1.3 Výsledná data

V pěti souborech bylo celkově identifikováno 53480 úseků MLST genů. Nejdelší úsek má přes 4000 nukleotidů, nejkratší obsahuje pouze 26, ale 48746 úseků jsou delší než 200. Zastoupení délek úseků nebylo rovnoměrné. Nejčastěji se délky úseků nacházely v rozmezí 36 až 1700 bp. Průměrná délka squiggle záznamů v datasetu je 14944 vzorků.



Obr. 4.1: Histogram délek sekvencí úseků MLST genů

Zastoupení MLST genů různých druhů taky nebylo rovnoměrné. Nejvíc identifikovaných úseků patřilo genu rpoB, nejmíň úseků bylo identifikováno pro gen tonB.



Obr. 4.2: Bar plot počtů identifikovných úseků MLST genů s jejich názvy

4.2 Neuronová síť pro klasifikaci genů

4.2.1 Architektura neuronové sítě a její parametry

Pro zpracování datasetu a klasifikaci MLST genů byla navržena konvoluční neuronová síť.

Konvoluční siť obsahovala tři konvoluční vrstvy. Každá vrstva se skládala ze dvou konvolučních podvrstev a jedné pooling vrstvy. Počet filtrů v první vrstvě činil 64 jader v každé podvrstvě, 128 jader v každé podvrstvě druhé vrstvy a 256 jader v každé podvrstvě třetí vrstvy. Tím pádem se první vrstva celkově skládala z 128 filtrů, druhá z 256 filtrů a třetí z 512 filtrů. Všechny konvoluční vrstvy se používaly ReLU jako aktivační funkci.

ReLU (Rectified Linear Unit) je aktivační funkce, která se často používá v různých typech neuronových sítích. Je to jedna z nejpopulárnějších aktivačních funkcí díky svým vlastnostem, které přispívají k efektivnímu učení modelů a řešení některých problémů spojených s gradientním zmizením. Matematicky je ReLU definována

jako:

$$f(x) \quad = \quad max(0, x) \tag{4.1}$$

kde x je vstup do ReLU. [18] To znamená, že pro vstupy x menší než 0 je výstupem funkce ReLU 0, zatímco pro vstupy x, které jsou nezáporné, je výstupem přímo hodnota x.

Také byl nastaven dropout, s podílem 0.5. Dropout je technika regularizace používaná v neuronových sítích s cílem zabránit přeučení (overfitting). Dropout vrstva funguje tak, že během procesu trénování náhodně vypíná (nastavuje na nulu) výstup určitého podílu neuronů v předchozí vrstvě. [19]

Při použití dropoutu v průběhu trénování se v každém kroku (epoše) pro každý neuron náhodně rozhodne, zda bude vypnut s pravděpodobností p, která je hyperparametrem sítě. To znamená, že výstup tohoto neuronu je pro tuto epochu nastaven na nulu. [19]

Dropout pomáhá zvýšit robustnost modelu tím, že přinutí síť učit se více robustní a generalizované vlastnosti, které nejsou závislé na aktivaci konkrétního podmnožiny neuronů. [19]

Po každé konvoluční vrstvě následovala pooling vrstva. V této neuronové sítě byly využity max-pooling vrstvy. Max-pooling bere největší hodnotu z definovaného okna dat, matematicky lze max-pooling operaci popsat jako:

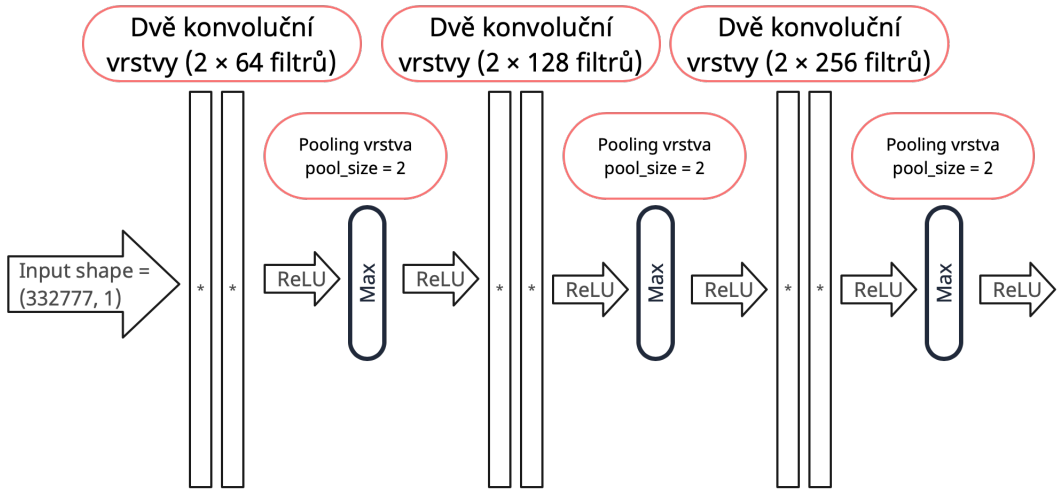
$$y'(t) \quad = \quad max(y(t-k), ..., y(t), ..., y(t+k)), \tag{4.2}$$

kde y' je výstup pooling vrstvy, y je vstup do pooling vrstvy, t je časový index a k je poloviční velikost okna. [20]

Táto neuronová síť měla pool_size (délku okna) nastaveny na 2. Což znamená, že se operace provádí na každých dvou po sobě jdoucích hodnotách vstupních dat. V každé takové dvojici hodnot je vybrána a zachována pouze ta, která má vyšší hodnotu. Tímto způsobem se rozměr vstupních dat snižuje na polovinu, což dávalo možnost dvojnásobně snižovat velikost vstupních dat při přechodu k následující vrstvě. Takže dvojnásobně klesala velikost vstupních dat a zároveň narůstal počet konvolučních filtrů.

V praxi operace max pooling pomáhá modelům získat robustnější reprezentace vlastností a zároveň snižuje počet parametrů a výpočetní náročnost modelu.

Poslední čtyři vrstvy byly plně spojené vrstvy. První vrstva obsahovala 256 neuronů, druhá vrstva měla 128 neuronů, třetí vrstva měla 64 neuronů, všechny tři plně propojené vrstvy používaly ReLU jako aktivační funkci. Poslední vrstva se skládala ze 7 neuronů (počet hledaných MLST-genů) a používala softmax aktivační funkci.



Obr. 4.3: Struktura konvolučních vrstev modelu

Softmax funkce bere na vstupu vektor a převádí ho na pravděpodobnostní distribuci. Jinými slovy, softmax normalizuje výstup na soubor hodnot mezi 0 a 1, které dohromady dají 1.

Pokud je tedy na výstupní vrstvě N neuronů s aktivační funkcí softmax, každý neuron udává pravděpodobnost, že daný vstup patří do třídy, kterou tento neuron reprezentuje. Neuron s nejvyšší pravděpodobností poté určuje třídu, do které vstup patří.

Matematicky je softmax definován pro j-tý prvek výstupního vektoru z jako:

$$softmax(z)_j \quad = \quad exp(z_j)/sum(exp(z_k), \tag{4.3}$$

pro k v (1, ..., N)

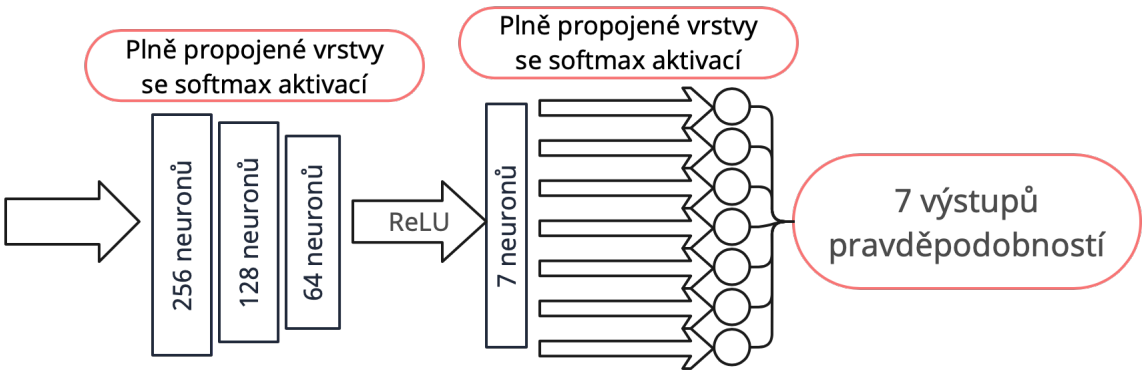
kde exp je exponenciální funkce, a N je počet tříd (a tedy i počet prvků výstupního vektoru). [20]

Neuronová síť se definovala, kompilovala a učila v prostředí Python 3.10 pomocí knihovny Keras 2.12.0.

4.2.2 Trénování sítě

Rozdělení datasetu

Před trénováním byl dataset rozdělen na trénovací, validní a testovací datasety, pro ověření správnosti predikce na neznámých pro model vzorcích. Dataset byl náhodně rozdělen v poměru 80:10:10 s rovnoměrným zastoupením všech MLST-genů v každém datasetu. Proudový záznamy před samotným trénováním bylo zapotřebí

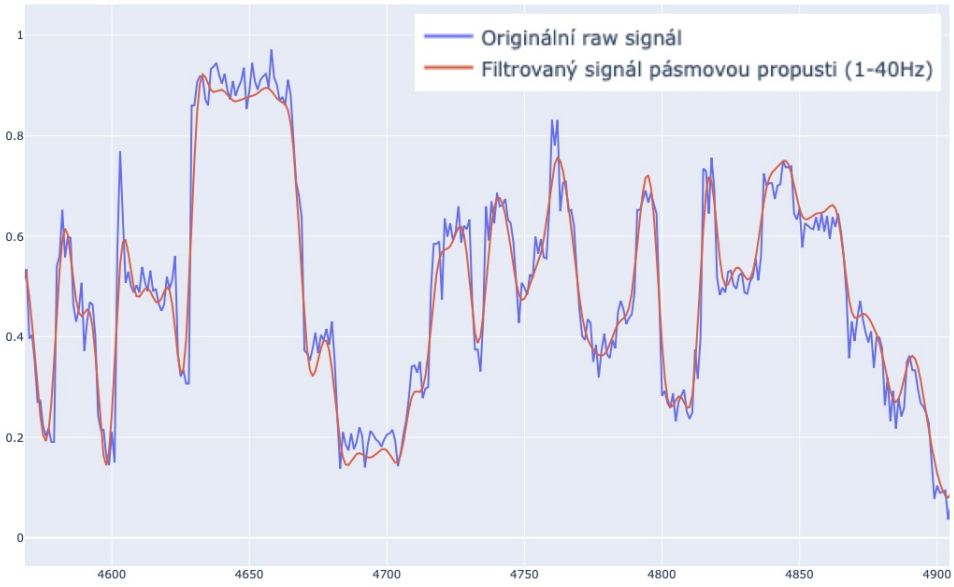


Obr. 4.4: Struktura plně spojených vrstev modelu

předzpracovat. Všechny epizody učení musely mít stejnou délku záznamů, která byla zadána při definování modelu jako rozměr vstupních dat. Nejdelší záznam v celém datasetu se skládal z 665553 vzorků, ale převzorkování pomohlo dvojnásobně zkrátit délky signálů. Proto nejdelší záznam už měl 332777 vzorků a všechny ostatní záznamy se musely prodloužit na tuto délku, tím, že všechny chybějící vzorky signálů byly doplněny nulami.

Předzpracování vzorků

Samotné squiggle signály bylo zapotřebí zpracovat před začátkem trénování. Jelikož předzpracování má zásadní vliv na výsledky učení a úspěšnost predikce modelu. Nejprve se data musela normalizovat. Normalizace znamená přizpůsobení dat tak, aby měla určité statistické vlastnosti nebo se nacházela v určitém rozsahu. Normalizace dat před učením neuronové sítě má několik důležitých výhod a je považována za důležitý krok v přípravné fázi modelování. Normalizace zlepšuje generalizační schopnosti modelu a stabilitu učení, omezuje vliv velkých hodnot a zrychluje průběh trénování. V tomto případě se data normalizovala do rozsahu [0:1]. Poté se data musela filtrovat, kvůli velkému zašumění squiggle záznamů. Filtrace se prováděla pásmovou propustí s propustným pásmem 1 až 40 Hz. Na konci byl signál převzorkován na menší vzorkovací frekvenci, to pomohlo zrychlit učení a snížit výpočetní náročnost.



Obr. 4.5: Příklad vyfiltrovaného squiggle signálu

Předzpracování signálu bylo zprostředkováváno pomocí knihoven scikit-learn 1.3.0 a scipy 1.9.3.

Trénování sítě

Trénování sítě probíhalo s batch sizem (dávkami) po 32 sekvencích a po 3 epochách. Taková velikost batchu a počet epoch byly zvoleny jako nejoptimálnější z hlediska časové a výpočetní náročnosti.

Trénovací dataset obsahoval 80% původního celého datasetu, což se rovnalo skoro 43 tisícům záznamům. Trénování na celém připraveném datasetu by mělo proběhnout za 1337 cyklů.

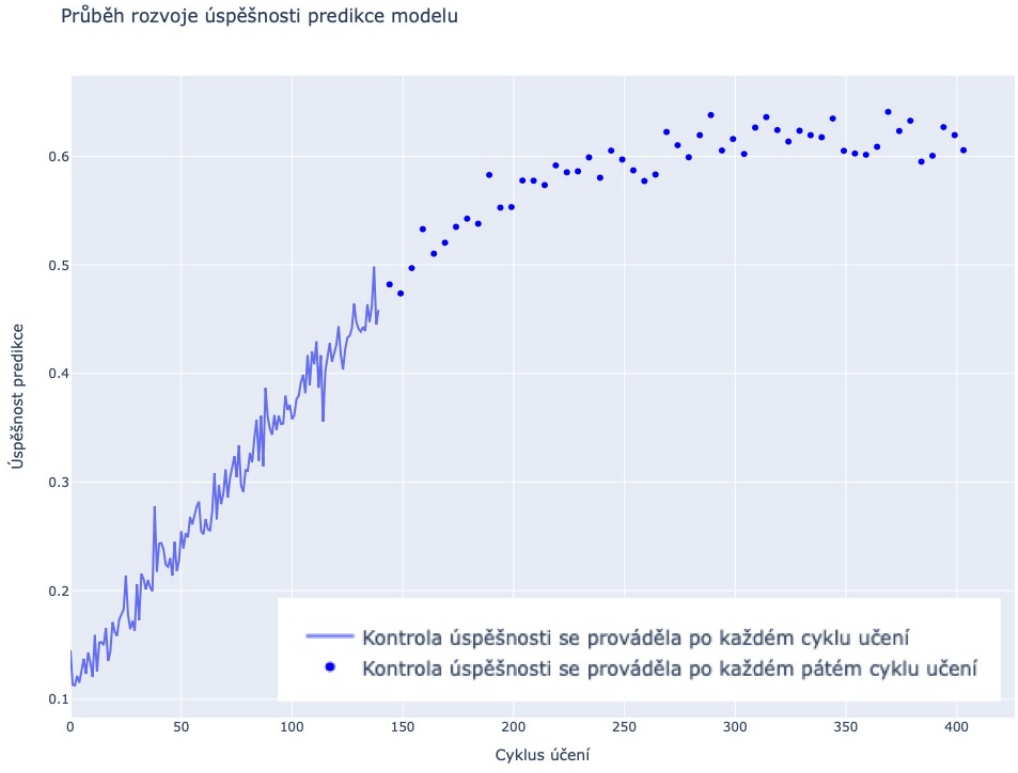
Pro definování a trénování modelu byly vytvořeny dva algoritmy. První algoritmus rozděloval dataset na trénovací, testovací a validní a zvlášť ukládal do souboru informace, do jakého setu patří každý záznam. První program taky definoval prvotní vrstvy modelu, kompiloval a ukládal netrénovaný model a prováděl první cyklus učení. Druhý algoritmus pokračoval trénování z druhého cyklu, na kterém se zastavil první. Rozdělení algoritmů pomohlo ladit proces učení bez nutnosti definovat a kompilovat model znovu. Po každé epoše se prováděl výpočet úspěšnosti modelu na validních datech pro možnost sledování průběhu učení. Frekvence měření úspěšnosti se poté měla snížit pro zrychlení procesu trénování. Následně se kontrola úspěšnosti prováděla pouze jednou za 5 cyklů.

Algoritmy taky vytvářely textový soubor, kam průběžně doplňovaly logy o průběhu učení. Logy obsahovaly informace o ID použitých sekvencí, číslo cyklu trénování, tvar (shape) vstupních dat a úspěšnost predikce na validních datech. U každého kroku se také uvádělo datum a čas provedení. Tyto informace se především používaly během ladění algoritmu a pro ukládání hodnot úspěšnosti modelu v různých cyklech.

Pro učení se používaly procesory Intel(R) Xeon(R) Gold 6128 CPU @ 3.40GHz. Tyto procesory byly poskytnuté GECKO UBMI FEKT VUT. Trénování probíhalo pomocí 18 takovýchto procesorů a 100 Gb operační paměti.

Úspěšnost predikce neuronové sítě

Úspěšnost predikce se měřila jednou za pět cyklů trénování. Na začátku učení úspěšnost činila kolem 10-15 procent, což přibližně odpovídá náhodnému výběru ze 7 genů. Prudký nárůst pokračoval až do 230. cyklu učení, kde úspěšnost dosáhla 60%. Následné zlepšení už nebylo tak rychlé a za dalších 150 cyklů se zlepšilo jen o necelých 5% a na konci trénování model vykazoval úspěšnost 64.9%.



Obr. 4.6: Rozvoj úspěšnosti predikce modelu během trénování

Diskuze dosažených výsledků

Navržený model dokázal klasifikovat neznámé squiggle záznamy do 7 skupin s úspešostí 64.9%.

Možnými faktory, které měly negativní vliv na výslednou generalizační schopnost, jsou:

- Při vyváření datasetu minimální délka jednoho záznamu, který se zapisoval do datasetu, byla omezena na 200 vzorků. V praxi se potom ukázalo, že taková délka signálu může být příliš krátká pro kvalitní reprezentaci úseků genů. Tyto krátké úseky patřily jak do trénovacího datasetu, čímž mohly kazit generalizační schopnosti modelu, tak i do testovácího datasetu, čímž mohly snížit výslednou úspěšnost.
- K ztrátě určité časti informace mohlo dojít během filtrování signálů pásmovou propustí. Propustné pásmo bylo nastaveno na 1 až 40 Hz. Takže byly odfiltrovány vyšší frekvenční složky, které reprezentovaly zašumění, ale do vyšších frekvencí mohly spadat i některé užitečné složky.
- Pro snížení časové výpočetní náročnosti signály byly převzorkovány na dvakrát nižší frekvenci. To nepochybně neslo ztrátu informací, ale velmi zrychlilo průběh trénování kvůli dvakrát kratším signálům.
- Úseky, které reprezentují MLST-geny, se vyhledávaly z databáze izolátů bakterie pomoci BLASTU. Algoritmus BLAST je dostatečně spolehlivý, ale nepochybně se ve výsledném datasetu mohly objevit i špatné vzorky, které mohly ovlivnit učení.

Každá z výše uvedených příčin zvlášť nebo jejich hromadné působení mohlo určitým způsobem ovlivnit generalizační schopnost modelu a výsledky finálního testování. Úspěšnost predikce na úrovni 65% by se dala považovat za relativně vysokou také s ohledem na to, že se klasifikace prováděla do většího počtu skupin.

Závěr

Třetí generace sekvenátorů je špičkou v moderní molekulární biologii a genetice, pro sekvenování molekuly DNA jim stačí pouze jedna molekula, ze které dokážou jeden po jednom stanovit pořadí nukleotidů. Oxford Nanopore je jedna z metodik této generace, která je založena na sekvenování nanopórem, přes který prochází zkoumána DNA molekula. Tato technika je založena na odečítání změn proudu při průchodu molekuly pórem a z těchto vychýlení pomoci matematických a statistických technik lze odvodit pořadí nukleotidu. Této technice sekvenování byla věnována tato bakalářská práce.

V práci byla stručně popsána důležitá teorie, která je nezbytná pro pochopení základních principů nanopórového sekvenování. Do nezbytných teoretických znalosti taky byla zahrnuta struktura formátu FAST5, který se používá sekvenátorem pro ukládání dat. Praktická část bakalářské práce byla zaměřena na navržení vlastní neuronové sítě pro klasifikaci genů z neznámých squiggle záznamů a její učení.

Před samotným trénováním sítě bylo zapotřebí vytvořit dataset, který by obsahoval MLST-geny a odpovídající jim předzpracovány squiggle záznamy pro bakterii Klebsiella pneumoniae, tyto geny pomocí algoritmu BLAST byly vyhledávány v databázi Fakultní nemocnice Brno.

V bakalářské práci byla navržena konvoluční neuronová síť, která bylo naučená předzpracovanými vzorky z již sestaveného datasetu MLST genů. V práci byla podrobně popsána struktura navržené neuronové sítě a principy algoritmů, které se používaly v této neuronové sítě. Výsledná úspěšnost predikce tohoto modelu činí 64.9%.

Poslední kapitola byla věnována faktorům, které mohly negativně ovlivňovat výsledky modelu. Tyto faktory se tykaly především kvality sebraných dat pro učení a jejich předzracování.

Literatura

[1] Berta Otová and Romana Mihalová. *Základy biologie a genetiky člověka*. Karolinum, V Praze, 1. vyd edition, 2012.

[2] Labguide, 2022. URL: https://labguide.cz/.

[3] Your genome, 2023. URL: https://www.yourgenome.org.

[4] Oxford nanopore technologies, 2023. URL: https://nanoporetech.com/products/minion.

[5] Raúl Rojas. *Neural Networks*. Springer, Berlin, 1 edition, 1996.

[6] Charles R. Cantor and Cassandra L. Smith. *Genomics : the science and technology behind the Human Genome Project*. Wiley-Interscience, Hoboken, New Jersey, 1. vyd. edition, 1999.

[7] Lilian T. C. França, Emanuel Carrilho, and Tarso B. L. Kist. A review of dna sequencing techniques. *Quarterly Reviews of Biophysics*, 35(2):169–200, 2002. URL: https://www.cambridge.org/core/product/identifier/S0033583502003797/type/journal_article, doi:10.1017/S0033583502003797.

[8] *Next Generation Genome Sequencing: Towards Personalized Medicine*. Wiley-VCH Verlag GmbH & Co. KGaA, Federal Republic of Germany, dr. michal janitz edition, 2008.

[9] Elaine R. Mardis. Next-generation sequencing platforms. *Annual Review of Analytical Chemistry*, 6(1):287–303, 2013. PMID: 23560931. arXiv:https://doi.org/10.1146/annurev-anchem-062012-092628, doi:10.1146/annurev-anchem-062012-092628.

[10] David Deamer, Mark Akeson, and Daniel Branton. Three decades of nanopore sequencing. *Nature Biotechnology*, 34(5):518–524, 2016. URL: http://www.nature.com/articles/nbt.3423, doi:10.1038/nbt.3423.

[11] Matei David, L. J. Dursi, Delia Yao, Paul C. Boutros, and Jared T. Simpson. Nanocall. *Bioinformatics*, 33(1):49–55, 2016-12-29. URL: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btw569, doi:10.1093/bioinformatics/btw569.

[12] Peng Chen, Jiajun Gu, Eric Brandin, Young-Rok Kim, Qiao Wang, and Daniel Branton. Probing single dna molecule transport using fabricated nanopores. *Nano Letters*, 4(11):2293–2298, 2004-11-01. URL: https://pubs.acs.org/doi/10.1021/nl048654j, doi:10.1021/nl048654j.

[13] Hengyun Lu, Francesca Giordano, and Zemin Ning. Oxford nanopore minion sequencing and genome assembly. *Genomics, Proteomics & Bioinformatics*, 14(5):265–279, 2016. URL: https://linkinghub.elsevier.com/retrieve/pii/S1672022916301309, doi:10.1016/j.gpb.2016.05.004.

[14] Madiha Sultan and Anastassia Kanavarioti. Nanopore device-based fingerprinting of rna oligos and micrornas enhanced with an osmium tag. *Scientific Reports*, 9(1), 2019. URL: https://www.nature.com/articles/s41598-019-50459-8, doi:10.1038/s41598-019-50459-8.

[15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, Cambridge, [2016].

[16] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline, 2016. arXiv:1611.06455.

[17] Laure Diancourt, Virginie Passet, Jan Verhoef, Patrick A. D. Grimont, and Sylvain Brisse. Multilocus sequence typing of klebsiella pneumoniae nosocomial isolates. *Journal of Clinical Microbiology*, 43(8):4178–4182, 2005. URL: https://journals.asm.org/doi/10.1128/JCM.43.8.4178-4182.2005, doi:10.1128/JCM.43.8.4178-4182.2005.

[18] S. Weidman. *Deep Learning from Scratch: Building with Python from First Principles*. O'Reilly Media, Incorporated, 2019. URL: https://books.google.cz/books?id=PRSCwwEACAAJ.

[19] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 06 2014.

[20] Christopher M. Bishop. *Pattern recognition and machine learning*. Springer Science + Business Media, New York, [2006].