Univerzita Karlova

Přírodovědecká fakulta

Studijní program: Bioinformatika

Studijní obor: Bioinformatika

Lucie Korená

Tvorba a hodnocení kvality genomových assembly

Construction and quality assessment of the genome assemblies

Bakalářská práce

Vedoucí práce: Mgr. Roman Leontovyč

Praha, 2020

Prohlášení:

Prohlašuji, že jsem závěrečnou práci zpracovala samostatně a že jsem uvedla všechny použité zdroje a literaturu. Tato práce ani její podstatná část nebyla předložena k získání jiného nebo stejného akademického titulu.

V Praze, 5.6.2020

Lucie Korená

Poděkování

Chtěla bych poděkovat svému školiteli panu Mgr. Romanu Leontovyčovi za trpělivost a cenné rady při psaní mé bakalářské práce a infrastruktuře Metacentrum, na které probíhaly výpočty k praktické části práce. Dále bych chtěla poděkovat svým přátelům a rodině, kteří mě podporovali po celou dobu studia.

Abstrakt

Detailní znalost genetické informace studovaného organismu je stěžejní pro mnohá odvětví moderního výzkumu. Současné sekvenační technologie neumožňují přečíst celou molekulu DNA vcelku, proto jsou získávány pouze úseky genomové sekvence, které samotné nejsou dostatečně informativní. Cílem genomicko-bioinformatického přístupu je složit tyto úseky do původní genomové sekvence – genomové assembly. Jedná se o náročný proces, ke kterému je potřeba výkonná počítačová infrastruktura, specializované softwary a expertní personál. Existuje celá řada softwarů (assemblerů), jejichž cílem je zrekonstruovat původní genetickou informaci daného organismu, které se liší ve velikosti skládaného genomu a druhu organismu. Výsledná kvalita genomové assembly je závislá na typu assembleru a nastavení jeho parametrů. Je tedy vhodné vytvořit několik assembly a jejich kvalitu následně vyhodnotit na základě technických a biologických metrik. Tato práce popisuje základní metody masivně paralelního sekvenování, dále se zabývá algoritmy skládání genomových assembly a popisuje metriky, pomocí kterých se vyhodnocuje kvalita výsledných genomových assembly. Praktická část je zaměřena na tvorbu assembly ptačí motolice *Trichobilharzia szidati* pomocí dvou programů a následné zhodnocení kvality obou assembly.

Klíčová slova: genomika, genomové assembly, bioinformatika, masivně paralelní sekvenování, kvalita assembly, algoritmus

Abstract

Detailed information of the genome of the studied organism is crucial for many fields of modern research. Actual sequencing technologies are not able to read the whole DNA molecule at once therefore only fragments of the genetic information are obtained, which are not sufficiently informative on their own. The goal of the genomic-bioinformatic approach is to assemble these fragments into complete original information – genome assembly. The process of the genome assembly is demanding in terms of computational power, software equipment and expert staff. Many assemblers – programs for genome assembly are available differing in performance, size of the analyzed genome or target organism. The quality of final assembly is fully dependent on assembler and setting of inner parameters. In practice, multiple assemblies are constructed and their quality evaluated according to the technical and biological parameters. The presented thesis describes current high throughput sequencing technologies, different approaches and algorithms for genome assembly and methodology for their quality assessment. The practical part is focused on assembly and its quality assessment using Illumina data of the bird fluke *Trichobilharzia szidati*.

Keywords: genomics, genome assembly, bioinformatics, High throughput sequencing, quality assembly, algorithm

Obsah

Slovník použitých pojmů	
Adaptory	Krátké sekvenčně specifické oligonukleotidové sekvence, které jsou ligované k 5’ a 3’ konci každého DNA fragmentu v sekvenační knihovně v rámci přípravy k NGS sekvenování
Alignment	Výpočetní metoda, která srovnává 2 a více sekvencí pod sebe tak, aby stejné nukleotidové báze či aminokyseliny ležely pod sebou. Dělí se na **pairwise** (srovnává dvě sekvence) a **multiple** (srovnává více jak dvě sekvence)
Assembler	Software, který sestavuje z kratších sekvenačních fragmentů delší celky
Assembly	Složení fragmentovaných sekvencí do větších struktur, na základě jejich překryvu, případně pomocí referenční sekvence
DNA fragment	Úsek sekvence DNA vzniklý fyzikálním nebo chemickým štěpením větší molekuly DNA
Kontig	První stupeň tvorby assembly, kdy jsou ready na základě jejich překryvů spojeny do delších sekvencí neobsahujících mezery
Misassembly	Oblast v assembly, která obsahuje rozlehlé inzerce, delece, inverze nebo přeskupení, které jsou výsledkem špatného sestavení
Primer	Krátký oligonukleotid na který se váže DNA polymeráza a inkorporuje komplementární nukleotidové báze do nově vznikajícího reverzního řetězce
Referenční genom	Kompletní sekvence genomu, která může být použita pro mapování krátkých DNA sekvencí při porovnání genomů z různých jedinců
Read	Záznam sekvence nukleotidů ze sekvenačních platforem • **Krátké ready** – ready velikosti 50-300 bp získané ze sekvenačních platforem jako jsou: SOLiD, Illumina, IonTorrent • **Dlouhé ready** – ready velikosti v řádech desetitisíců až statisíců bází (výjimečně milionů) získané ze sekvenačních platforem jako jsou: Oxford Nanopore Technologies nebo PacBio
Scaffold	Druhý stupeň v procesu tvorby assembly, kdy jsou 2 a více kontigů spojených do většího celku
Sekvenační knihovny	Sada modifikovaných DNA fragmentů, které jsou připraveny k sekvenování
Templát	Rekombinantní sekvence, která je tvořena adaptorovou sekvencí (ke které se váže univerzální primer) a cílovou sekvencí, jejichž pořadí nukleotidů není známo

1 Úvod

Sekvenování neboli proces, při kterém se zjišťuje pořadí nukleotidových bází v sekvencích DNA nebo RNA, prochází v poslední době velkým rozvojem, který s sebou přináší snadnou a rychlou dostupnost biologických dat. Jeden z prvních projektů na sekvenování genomu (The Human Genome Project) pomocí Sangerovy metody sekvenování trval přibližně 13 let a stál kolem 2,7 miliardy dolarů. Současné sekvenační platformy dokážou sekvenovat genomy v kratší časové době (několik týdnů) za cenu v řádech tisíců či desetitisíců. Avšak jednou z hlavních nevýhod těchto metod je omezená délka fragmentů nukleových kyselin, které jsou schopny najednou osekvenovat. Při celogenomovém sekvenování jsou čteny krátké úseky DNA, které je nutno skládat do původní genetické informace pomocí počítače.

Softwary, které slouží ke skládání těchto genomových úseků dohromady se nazývají assemblery. Existuje jich celá řada a jsou specifické pro konkrétní sekvenační data a velikost genomu daného organismu. Vzhledem k tomu, že vývoj sekvenačních platforem je velice rychlý, tak vznikají stále nové přístupy pro skládání genomové assembly, a proto nelze naprogramovat univerzální assembler, který by vyhovoval všem sekvenačním platformám. Při sestavování je vhodné použít více assemblerů s různým nastavením vstupních parametrů. K vyhodnocení kvality výsledných assembly se využívají technické a biologické parametry, které popisují spojitost (technické parametry) a úplnost (biologické parametry) genomu.

Tato práce se v teoretické části zaměřuje na základní popis sekvenačních technologií, následnou tvorbu assembly (popis jednotlivých algoritmů) a popis metrik a programů hodnotících kvalitu vytvořené assembly. V praktické části práce byla provedena tvorba assembly organismu *Trichobilharzia szidati* na základě sekvenačních dat získaných z platformy Illumina použitím assemblerů ABySS a MaSuRCA. Následně bylo provedeno zhodnocení sestavené assembly pomocí programu hodnotí technické parametry (QUAST) a biologické parametry (BUSCO).

2 Sekvenování

2.1 Techniky sekvenování

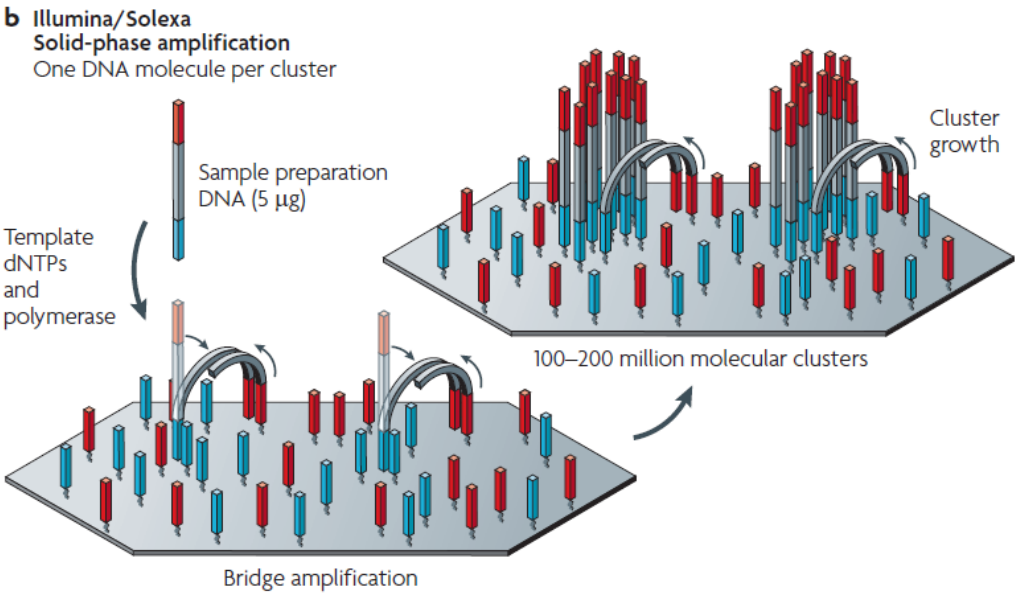
První sekvenační metody vznikly v roce 1977, jednalo se o Sangerovu dideoxynukleotidovou sekvenovací metodu (Sanger *et al.* 1977) a o Maxam and Gilbertovu metodu chemického štěpení (Maxam & Gilbert 1977). Následovaly sekvenační techniky tzv. *nové generace (NGS)*, které představovaly další krok v sekvenování DNA. Výhodou sekvenování nové generace je, že celá knihovna templátů DNA je imobilizována na dvourozměrný povrch a paralelně probíhá sekvenování na všech templátech (Shendure *et al.* 2017). Tyto nejnovější metody kombinují různé přístupy, co se týče sestavování templátových sekvencí, samotného sekvenování či detekce struktury (příklady sekvenačních platforem nové generace jsou uvedeny v Tab. 1).

Tab. 1: Příklady sekvenačních platforem nové generace (Levy & Myers 2016)

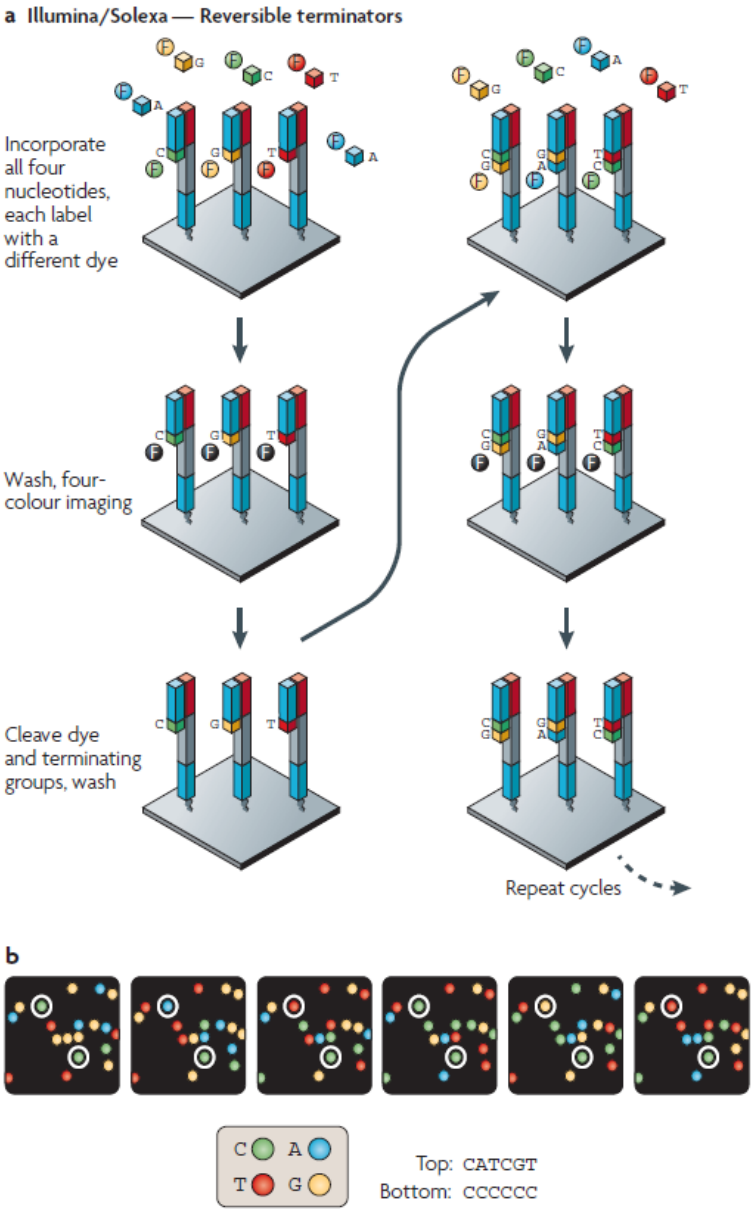
Platforma	**Amplifikace**	**Sekvenační přístup**	**Detekce**	**URL**
Illumina	Klonální	Sekvenace syntézou	Optická	**http://www.illumina.com**
Oxford Nanopore	Single molecule	Real-time sequencing	Nanooptická	**http://www.nanoporetech.com**
Pacific Biosciences	Single molecule	Sekvenace syntézou	Optická	**http://www.pacb.com**
SOLiD	Klonální	Sekvenace ligací	Optická	**http://www.thermofisher.com/us/en/home/brands/applied-biosystems.html**

2.1.1 Illumina

Technologie vyvíjená původně firmou Solexa, kterou v roce 2007 koupila společnost Illumina (Liu *et al.* 2012). Výsledkem sekvenace jsou krátké ready (100-300 bp) s chybovostí ≤ 1% (Schirmer *et al.* 2015). Princip fungování je založen na sekvenování syntézou. Na fragmenty DNA jsou ligovány k oběma koncům specifické adaptory. Takto upravené fragmenty jsou následně amplifikovány (zmnoženy) pomocí tzv. *amplifikace na mostech* (Obr. 1), kdy se na skleněné destičce vyskytují náhodně rozmístěné adaptory (obsahující oligonukleotidové sekvence) kompatibilní jak s adaptory na jednom konci fragmentů, tak i adaptory kompatibilní s adaptory na druhém konci fragmentů. Po navázání dochází k amplifikaci. Sekvenování probíhá na základě detekce jednotlivých bází, které jsou začleňovány do řetězce DNA při syntéze reverzního řetězce, jedná se o tzv. metodu *cyclic reversible termination (CRT)*, při které se používají *reverzní terminátory* neboli nukleotidy (A, G, C, T), které jsou fluorescenčně značené a po inkorporaci zastavují syntézu DNA. Po inkorporaci daného nukleotidu dojde k odmytí nezačleněných nukleotidů a provede se detekce, která slouží k identifikaci inkorporovaného nukleotidu (Obr. 2). Následně proběhne štěpení, které odstraní inhibující skupinu a fluorescenční barvivo z *reverzního terminátoru*. Před zahájením dalšího cyklu proběhne ještě promytí (Metzker 2010).



Obr. 1: **Amplifikace na mostech – Illumina**
Fragment DNA s navázanými adaptory na obou koncích (na každém konci je odlišný typ adaptoru) se inkorporuje na skleněnou destičku posetou adaptory kompatibilní s adaptory na fragmentech DNA. Následně pomocí PCR se namnoží původní fragmenty a ohýbají se k sousedním adaptorům, které jsou kompatibilní k danému konci fragmentu (Metzker 2010) .



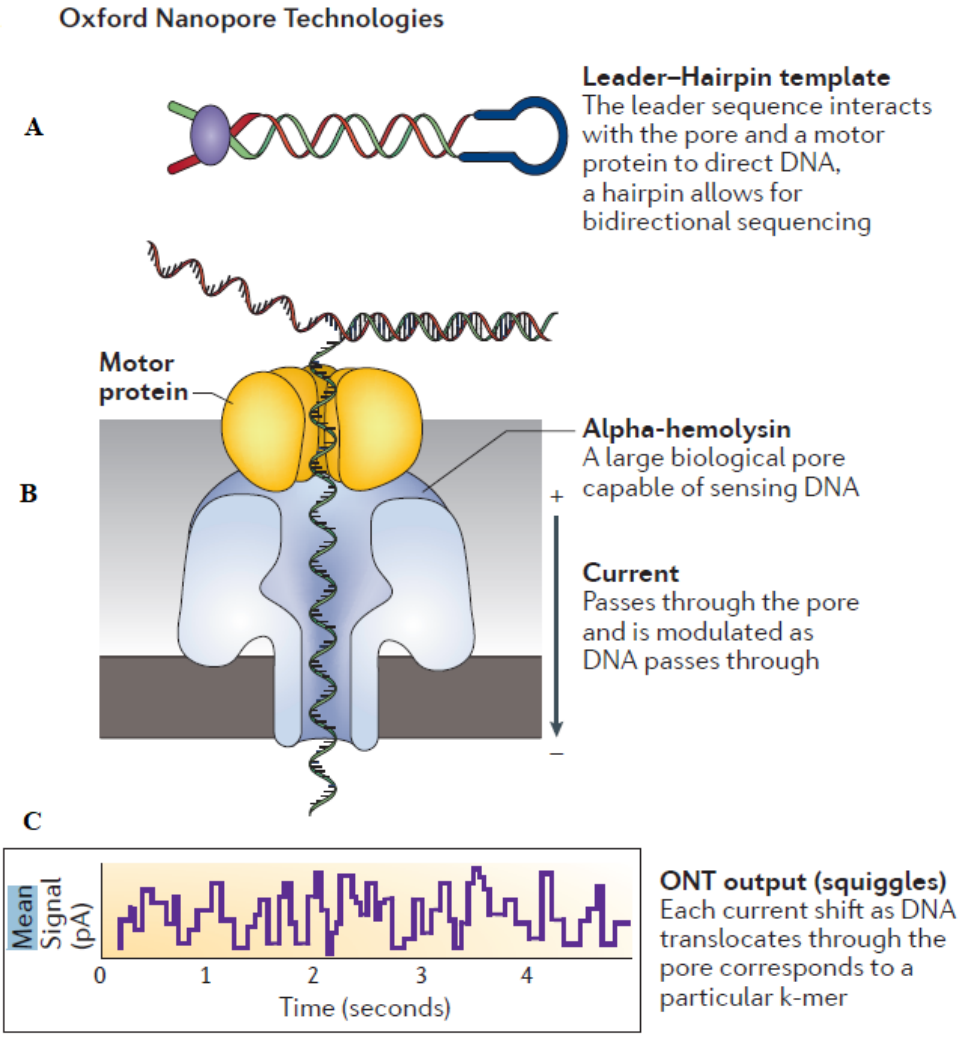
Obr. 2: **Sekvenování pomocí CRT**
Názorně zobrazený popis metody CRT, kdy se v každém kroku inkorporuje jeden ze 4 reverzibilních nukleotidů (A, G, C, T), který po inkorporaci vyzařuje fluorescenční signál, který je zaznamenán detektorem. Následně dojde k odmytí ostatních reverzibilních nukleotidů a probíhá detekce. Po detekci proběhne štěpení, při kterém se odstraní inhibující skupina a fluorescenční barvivo z nově inkorporovaného nukleotidu a celý cyklus se opakuje (Metzker 2010).

2.1.2 Nanopore – Oxford Nanopore Technologies

V roce 2014 byla uvedena na trh první sekvenační platforma na bázi nanopore sekvenování (MinION od společnosti Oxford Nanopore Technologies). Technologie nanopore poskytuje dlouhé sekvenační ready o velikosti v řádu desítek až stovek tisíc bází (nejdelší známý osekvenovaný úsek byl - 2 272 580 bází (online zdroj č. 1)). Dlouhé ready řeší problém s repetitivními úseky v genomu. Doba sekvenování zde trvá minuty a náklady

na zařízení jsou zde menší než u sekvenačních platforem generující krátké ready. Avšak jednou z hlavních nevýhod je vyšší chybovost než u platformy Illumina.

K určení pořadí jednotlivých bází dochází přímou detekcí složení nativní jednovláknové DNA. DNA je zpočátku fragmentována na úseky velikosti 8-10 kb, na které jsou ligovány dva různé adaptory tzv. *leader* (vedoucí adaptor) a *hairpin* (vlásenka). Kvůli neexistenci přesného nasměrování adaptoru na konkrétní konec fragmentu existují 3 možné sekvenační knihovny: leader-hairpin, leader-leader a hairpin-hairpin. Nejlepší variantou je leader-hairpin, neboť leader adaptor obsahuje sekvenci potřebnou pro nasměrování DNA do pórů a hairpin umožňuje sekvenaci dopředného i zpětného řetězce. U varianty hairpin-hairpin k sekvenaci nedochází, neboť chybí adaptor, který DNA fragment nasměruje do póru (Goodwin *et al.* 2016). Samotné sekvenování probíhá pomocí průchodu upraveného DNA fragmentu s iontovým proudem proteinovým pórem (Obr. 3-B), kde každá báze je specifická poklesem proudové amplitudy (Bayley 2006). Změnu proudové amplitudy v čase zaznamenává detektor, který je umístěný vedle póru. Vysoká rychlost translokace DNA přes pór způsobuje problém ve správném rozlišení dané báze, proto sekvenační data obsahují specifickou chybovost. Míra chybovosti se vývojem novějších metod, které zpomalují průchod bází póry, zlepšuje (McCombie *et al.* 2019).



Obr. 3: **Nanopore sekvenování**
A – DNA fragment s navázaným leader a hairpin adaptorem, B – průběh nanopore sekvenování, kdy jedno vlákno DNA prochází pórem a jednotlivé báze DNA mění amplitudu iontového proudu, procházejícím pórem, C – výstupní záznam změn iontového proudu na základě kterého je přečtena sekvence DNA (Metzker 2010)

2.2 Typy sekvenování

Výstupem sekvenačních platforem jsou ready různých délek závisející na zvolené sekvenační platformě. Existují 3 základní typy readů:

2.2.1 Single-end reads

Nejjednodušší typ readů. Sekvenování probíhá od jednoho konce fragmentu ke druhému. Délka readů závisí na předem zvolené sekvenační platformě. Jedná se o vysoce kvalitní data získané za krátkou dobu. Nesou informaci o primární struktuře sekvenované sekvence a o kvalitě nasnímaného signálu pro každý nukleotid. Jednotlivé ready jsou spojeny přes překrývající se oblasti do větších oblastí zvaných kontigy. Při skládání do větších úseků se potýkají s problémy jako jsou repetitivní úseky, polymorfismy, chybějící

data (při malém sekvenačním pokrytí genomu), proto se při skládání readů kombinují ještě s pair-end nebo mate-pair ready.

2.2.2 Pair-end reads

Typ párových readů s inzerty délky 200–800 bp mezi oběma konci. Sevence DNA je sekvenována z obou konců fragmentu (online zdroj č. 2). Poskytují nejen informaci o sekvenci, ale také o poloze fragmentu, proto se často využívají v kombinaci se single-reads, neboť umožňují detekovat repetitivní oblasti či strukturní varianty genomu (Li *et al.* 2015).

2.2.3 Mate-pair reads

Typ párových readů s inzerty délky 2-5 kb mezi oběma konci (asi 10x větší než u pair-end reads). Kromě informace o sekvenci poskytují i informaci o fyzické vzdálenosti mezi dvěma ready. Usnadňují sestavení genomové assembly z krátkých readů (řešení problému s repetitivními úseky v genomu). Pomocí mapování kontigů na mate-pair reads lze určit pořadí kontigů a jejich orientace (Van Nieuwerburgh *et al.* 2012).

3 Genomové assembly

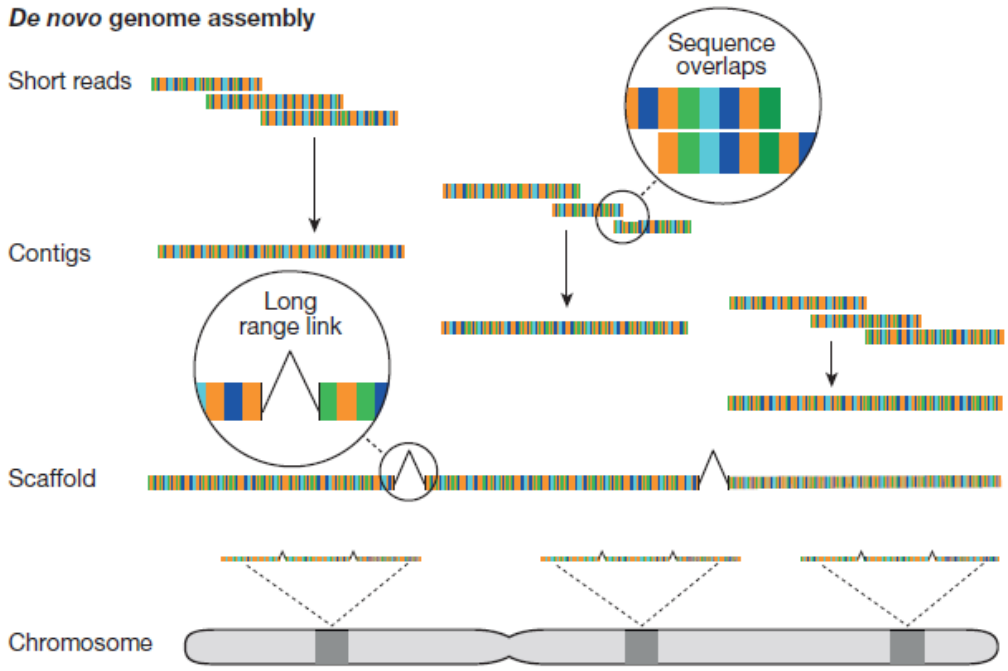
S rozvojem masivně paralelních sekvenačních technologií se zvyšuje dostupnost takzvaných „-omics“ sekvenačních dat. I přes významný posun ve vývoji sekvenačních technologií generujících dlouhé ready (Oxford Nanopore Technologies (online zdroj č. 3), PacBio (online zdroj č. 4)), jsou intenzivně využívané sekvenační platformy, produkující krátké ready (50 – 300bp) např. Illumina (online zdroj č. 5). Sekvenační ready je nutné softwarově sestavit do delších sekvenačních celků pomocí tzv. assemblerů, kde je v ideálním případě výsledkem sestavený genom na úrovni jednotlivých chromozómů.

3.1 Typy assemblerů

Rozdílné assemblery jsou vhodné pro různé sekvenační technologie a pro různě velké sestavované genomy. Assembly dělíme na tzv. *de novo* neboli sestavování genomu bez referenčního genomu a na tzv. *reference based*, kdy je k dispozici referenční genom. Tato bakalářská práce se zabývá softwary pro *de novo* sestavení genomové assembly, neboť praktická část práce se zabývá *de novo* assembly.

3.1.1 Assemblery pro krátká čtení

Při sestavování assembly jsou v první fázi na základě překryvů jednotlivých readů sestavovány delší úseky (kontigy) neobsahující mezery, ty jsou dále spojovány pomocí párových readů (mate-pair nebo paired-end) do větších celků tzv. scaffoldů (Obr. 4). Pokud mezi sousedními kontigy ve scaffoldu nedochází k překryvu, tak se mezi nimi vyskytují mezery, které jsou vyplněny pomocí dalších nezávislých readů a tím se dokončí tvorba assembly (Sohn & Nam 2016).



Obr. 4: **Sestavení de novo genome assembly**
V prvním kroku se krátké ready na základě jejich překryvů skládají v kontigy. Následně jsou spojovány kontigy do větších úseků zvaných scaffoldy. V ideálním případě bychom měli dostat genomovou assembly, která je na úrovni chromozómů (Shendure *et al.* 2017).

Při sestavování assembly se assemblery potýkají s několika problémy, které musí co nejefektivněji vyřešit:

- **Oprava sekvenčních chyb**, která musí být provedena před začátkem sestavování assembly nebo během sestavování, neboť by se tyto chyby dostaly do výsledných kontigů a následně do scaffoldů a vytvořily by tam artefakty (Sohn & Nam 2016). Každá sekvenační platforma dělá tyto chyby, např. u platformy Illumina je známo, že generuje krátké ready s ≤ 1% náhodných sekvenčních chyb (Schirmer *et al.* 2015).
- **Nerovnoměrná hloubka čtení**, která je způsobená tím, že některé úseky genomu mohou být osekvenovány víckrát než jiné. Nerovnoměrné pokrytí způsobuje vznik mezer ve výsledné assembly.
- **Topologická složitost repetitivních úseků**, pokud velikost readů je dostatečně velká, že pokrývá tyto repetitivní úseky, pak tento problém nenastává, ale platformy generující krátké ready (např. Illumina) nepokrývají svou délkou tuto oblast, proto je vhodné využít kombinace s dlouhými ready (Sohn a Nam 2016).

- **Algoritmická složitost** – tvorby grafů a průchody grafy jsou relativně náročné na paměť a na výpočet. De Bruijnovy grafy vyžadují značnou část RAM[1] paměti a počítají v dlouhých časových intervalech (několik dnů až týdnů) (Sohn & Nam 2016).

Assemblery pro krátké ready se liší dle toho, jaký typ algoritmu pro sestavení assembly využívají, dělíme je na algoritmy využívající de Bruijnovy grafy, Overlap layout consensus či Greedy algoritmus. Některé assemblery využívají i kombinaci těchto algoritmů (př. MaSuRCA – pracuje na principu Overlap layout consensus a de Bruijnova grafu). Tyto algoritmy jsou popsány v následující podkapitole (3.2).

3.1.2 Assemblery pro dlouhá čtení

Dlouhé ready jsou získávány z platforem jako jsou například PacBio a Oxford Nanopore Technologies. Dlouhé ready mohou vyřešit některé problémy, se kterými se setkávají assemblery pro krátké ready, jako je například již zmíněný problém s rozpoznáváním repetitivních úseků. Tento problém však není úplně vyřešen, neboť existují tak dlouhé repetitivní úseky, které ani dlouhé ready nepokryjí, jako například: *long interspersed nuclear elements* (LINEs) nebo *long terminal repeats* (LTRs) (Ashton *et al.* 2015). Platformy generující dlouhé ready mají nedostatek spočívající ve vyšší chybovosti vzniklých readů, v této době se jejich chybovost pohybuje mezi 10-15 % (Vasudevan *et al.* 2020), proto se často při sestavování assembly využívá kombinace s krátkými ready s chybovostí ≤ 1% (Schirmer *et al.* 2015). Na této bázi jsou založené hybridní assemblery (Deshpande *et al.* 2013).

3.2 Algoritmy k sestavení assembly

Existuje několik druhů algoritmů pro tvorbu assembly.

3.2.1 De Bruijnovy grafy (DBG)

Jeden z dosud nejlepších algoritmů pro sestavování genomové assembly pro krátké ready získané z platforem Illumina a SOLiD (Miller *et al.* 2010). Funguje na principu sestavení orientovaného grafu reprezentujícího překryvy mezi ready, kde z každého readu jsou vygenerovány podřetězce (k-mery) délky $L-k+1$, kde L je délka readu a k je daná délka k-meru. Kontigy a scaffoldy jsou získávány inverzní transformací optimální cesty

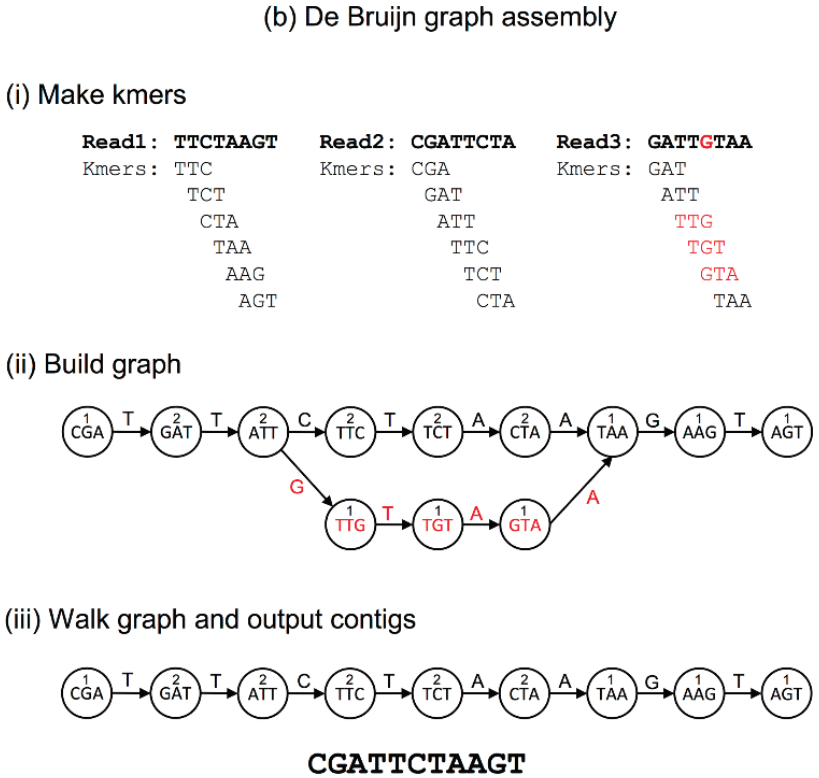
[1] RAM paměť je operační paměť počítačů neboli paměť, ve které jsou uloženy běžící programy a jejich data, čím větší kapacita je k dispozici, tím je plynulejší chod programů

v sestaveném de Bruijnově grafu mezi sekvencemi (Sohn & Nam 2016). De Bruijnovy grafy rozdělujeme na 2 typy Hamiltonovský a Eulerovský (Compeau *et al.* 2011).

Hamiltonovský de Bruijnův graf spočívá v tom, že zde každý k-mer reprezentuje uzel v grafu a pokud se jeho (*k-1*) dlouhá přípona shoduje s (*k-1*) dlouhou předponou jiného k-meru, tak mezi těmito dvěma uzly bude vést hrana (Obr. 5). Assemblery fungující na tomto principu jsou například: SOAPdenovo (Luo *et al.* 2012), SGA (Simpson & Durbin 2012), ABySS (Simpson *et al.* 2009), Meraculous (Chapman *et al.* 2011) a Velvet (Zerbino & Birney 2008). Algoritmus spočívá v hledání Hamiltonovské cesty v takto vytvořeném grafu, což znamená, že hledá cestu, která každý vrchol navštíví právě jednou (Sohn & Nam 2016). Tento problém však patří mezi NP těžké problémy[2] pro větší počet uzlů v grafu (Compeau *et al.* 2011). Proto se na vyhledávání takové cesty využívá různých heuristik.

Eulerovský de Bruijnův graf reprezentuje uzly a hrany opačným způsobem než Hamiltonovský. K-mery jsou zde hrany a překrývající se (*k-1*) dlouhé úseky zde reprezentují uzly. Zde je řešení komplikovaného grafového problému takové, že se hledají Eulerovské cesty, které procházejí všemi hranami, z nichž každá z hran je navštívena právě jednou (Pevzner *et al.* 2001). Assemblery, které využívají tento typ grafů jsou například: EULER (Pevzner *et al.* 2001), SPAdes (Bankevich *et al.* 2012), ALLPATHS-LG (Gnerre *et al.* 2011) a MaSuRCA (Zimin *et al.* 2013). Tyto assemblery vytvářejí lepší assembly pro velké genomy než assemblery fungující na základě Hamiltonovských de Bruijnových grafů (Earl *et al.* 2011).

[2] NP těžké problémy je třída problémů, které nejsou řešitelné v polynomiálním čase pomocí deterministického algoritmu, takže vykazují velkou časovou složitost. Tento problém je převoditelný na jakýkoliv jiný NP těžký problém



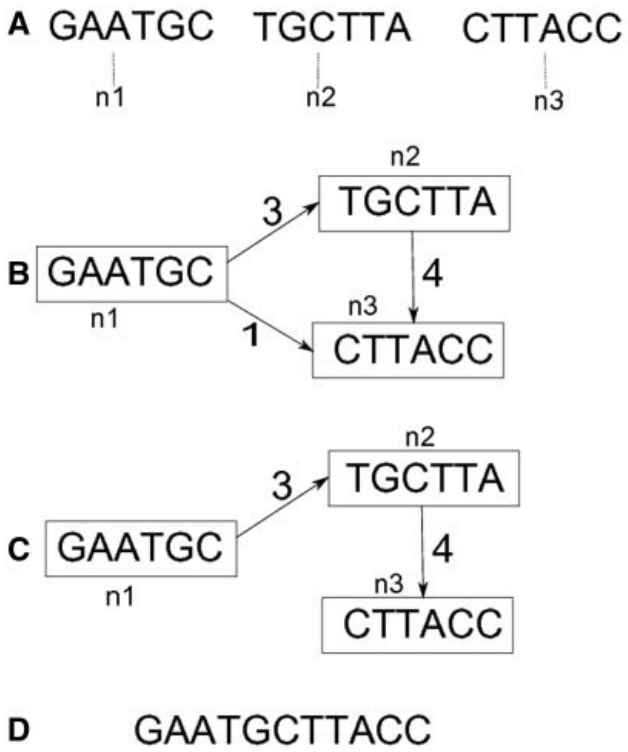
Obr. 5: **Příklad sestavení a průchodu de Bruijnovým grafem**
(i) Tvorba k-merů – na vstupu jsou 3 ready, ze kterých jsou vygenerovány k-mery velikosti 3, **(ii) Tvorba Hamiltonovského de Bruijnova grafu** – každý ze vzniklých k-merů zde reprezentuje vrchol v grafu a hrany mezi vrcholy reprezentují překryvy jednotlivých dvou k-merů (červené části značí polymorfismy, a proto vzniká větev v grafu), čísla v uzlech značí, kolikrát je daný k-mer ve vstupních readech zastoupený, **(iii) Průchod grafem** – hledání maximálního průchodu grafem (cesty s nízkým pokrytím mohou být ignorovány), průchodem jsou získány kontigy – zde kontig: CGATTCTAAGT (Ayling et al. 2019)

Klíčovým parametrem je zde určení správné velikosti k-meru. Je třeba zvolit takovou hodnotu, která bude dostatečně malá na to, aby nebyly opomenuty všechny možné překryvy readů, ale také dostatečně velká na to, aby nedocházelo k tomu, že budou vznikat falešné překryvy (He *et al.* 2013). Optimální velikost k-meru nelze předem spolehlivě určit, proto se v praxi často čerpá ze zkušeností s podobnými datovými soubory (Chikhi & Medvedev 2014).

3.2.2 Overlap layout consensus (OLC)

Tento typ algoritmů se využívá především při zpracovávání dlouhých readů, neboť nepřevádí ready na k-mery. Jedná se o tříkrokový přístup, při kterém se nejdříve na základě překryvů jednotlivých readů vytvoří *overlap graph* (graf překryvů), jednotlivé délky překryvů jsou získány výpočtem pairwise sequence alignmentu. Ze vzniklého grafu se odstraní dle transitivního pravidla (vysvětleno níže) přebytečné hrany, tzv. *layout*. V posledním kroku probíhá hledání cesty s maximálním ohodnocením hran, která bude sloužit jako *consensus* sekvence (Sohn & Nam 2016). OLC algoritmus se skládá ze tří částí:

- **Overlap graph (graf překryvů)** – každý read reprezentuje jeden uzel v grafu a hrany mezi jednotlivými uzly jsou na základě překryvů daných dvou sekvencí čili pokud je sufix jednoho uzlu shodný s prefixem druhého uzlu, tak mezi těmito uzly povede v grafu orientovaná hrana a bude mít váhu, která je rovna délce prefixu (sufixu) (Obr. 6 B). Pro zjištění délky překryvu se využívá tvorby pairwise sequence alignmentu pro každý pár readů (nejčastěji pomocí algoritmu Smith-Waterman).
- **Layout** – odstranění přebytečných hran z overlap grafu na základě transitivního pravidla čili pokud vede hrana z vrcholu *i* do vrcholu *j*, z vrcholu *j* do vrcholu *k* a z vrcholu *i* do vrcholu *k*, potom je méně ohodnocena hrana vedoucí z vrcholu *i* vymazána (Obr. 6 C).
- **Consensus** – hledání nejpravděpodobnější cesty v grafu neboli cesty, která bude mít nejvyšší ohodnocení. Výsledkem je sestavená consensus sekvence (Obr. 6 D)



Obr. 6: **Overlap layout consensus (OLC)**
A – na vstupu jsou 3 fragmenty DNA, **B** – **Kontrukce overlap grafu:** každý ze zadaných fragmentů představuje vrchol ve vznikajícím grafu a hodnota hran spojující dané dva vrcholy se odvíjí od délky jejich překryvu (délka sufixu jedné sekvence shodná se stejnou délkou prefixu druhé sekvence), délka tohoto překryvu se dá vypočítat pomocí pairwise sequence alignmnetu, **C – Layout:** odstranění přebytečných hran z grafu na základě transitivního pravidla, **D – Hledání Hamiltonovské cesty:** v grafu hledáme takovou cestu, která projde všemi vrcholy grafu právě jednou a bude mít nejvyšší váhu, pomocí této cesty se vygeneruje kontig (zde kontig: GAATGCTTACC) (Chen et al. 2017)

Softwary využívající tento typ algoritmu jsou například: Arachne (Batzoglou 2002), Celera Assembler (Myers 2000), CAP3 (Huang 1999), PCAP (Huang & Yang 2005), Phrap (Bastide & McCombie 2007), Phusion (Mullikin 2003) a Newbler (Margulies *et al.* 2005).

3.2.3 Greedy algoritmus

V první iteraci algoritmu probíhá výpočet překryvů mezi jednotlivými ready. Dle intezity překryvu se přiřadí příslušné skóre, na základě kterého se iterativně slučují nejvíc skórující ready. Tento postup se opakuje do doby, kdy už nelze další ready sloučit. Algoritmus se vyznačuje lehkou implementací, avšak ta je vykompenzována výraznou paměťovou náročností, takže použitelnost tohoto algoritmu je omezena pouze na sestavení malých genomů (bakterie, jednobuněčná eukaryota) (Pop *et al.* 2002).

Assembler fungující na tomto principu je SSAKE (Warren *et al.* 2007), dobře funguje pro krátké ready. Ukládá si krátké ready a jejich frekvence v hašovací tabulce a hledá nejlepší shody prostřednictvím *prefix tree* neboli stromu předpon. Z SSAKE se odvodily další assemblery jako například JR-Assembler (Chu *et al.* 2013).

3.2.4 Hybridní algoritmy

Algoritmy kombinující krátké kvalitní ready a dlouhé chybové ready získané z různých sekvenačních platforem jako jsou Illumina a Oxford Nanopore Technologies. Dělí se na dva typy. První typ assemblerů při opravě chyb vzniklých při sekvenování opravuje dlouhé ready pomocí krátkých readů. Takto opravené dlouhé ready jsou následně použité k tvorbě výsledné assembly. Tento proces je však výpočetně i časově náročný (Deshpande *et al.* 2013). Assemblery využívající tento přístup jsou například PBcR (Koren *et al.* 2012) a Nanocorr (Goodwin *et al.* 2015). Druhý typ assemblerů vytvoří kontigy z krátkých readů a scaffoldy sestavují pomocí dlouhých readů (Deshpande *et al.* 2013). Assemblery tohoto typu jsou například Cerulean (Deshpande *et al.* 2013), hybridSPades (Antipov *et al.* 2016), AHA (Rasko *et al.* 2011) a DBG2OLC (Ye *et al.* 2016).

3.3 De novo assemblery

V této podkapitole budou popsány principy dvou *de novo* assemblerů, kde každý z nich funguje na základě jiného druhu algoritmu.

3.3.1 ABySS (de Bruijnův graf)

Celým jménem Assembly By Short Sequencing. Program vhodný pro sestavování assembly z velkého množství krátkých readů například ze sekvenační platformy Illumina. Je založen na bázi de Bruijnova grafu (Hamiltonova), který umožňuje paralelní výpočet

algoritmu rozdělit mezi několik počítačů, proto je assembler rychlý a i přesný. Tím, že se výpočty dají paralelizovat na více počítačů dojde k výraznému ušetření množství paměti (Simpson *et al.* 2009). Princip sestavení assembly probíhá ve dvou fázích:

V **první fázi** probíhá tvorba de Bruijnova grafu, během které jsou z readů délky *l* vygenerovány ready velikosti (*l-k+1)*, kde *k* je velikost k-meru, a sekvence s neznámými bázemi jsou vyřazeny. Před sloučením tohoto grafu do kontigů jsou z grafu odstraněny vrcholy a hrany vzniklé chybami v sekvenování, které obvykle vytváří v grafu tzv. *dead-end uzly*, ze kterých se nedá dostat do jiných uzlů (slepé uličky). Tyto uzly jsou způsobené spojením správných a chybných k-merů (Simpson *et al.* 2009). Tato fáze je výrazně paměťově náročná, proto novější verze ABySS (ABySS 2.0) využívá tzv. *Bloom filtr*[3], který řádově snižuje paměťovou složitost (Jackman *et al.* 2017).

Ve **druhé fázi** jsou využity pair-end ready (pokud jsou k dispozici) pomocí nichž dochází ke spojování kontigů do větších úseků. Na základě alignmentu pair-end readů a kontigů z první fáze dojde k vytvoření sady propojených kontigů. Tato sada propojených kontigů je následně filtrována, aby se předešlo zanesení chyb do výsledné assembly. Každý kontig C_i má sadu propojených kontigů P_i pomocí které se provede průchod de Bruijnovým grafem za účelem nalezení jedinečné cesty vedoucí od C_i přes všechny kontigy v P_i. Tento proces se opakuje pro každý kontig C_i a v posledním kroku jsou spojeny konzistentní cesty ke generování kontigů výsledné assembly (Simpson *et al.* 2009).

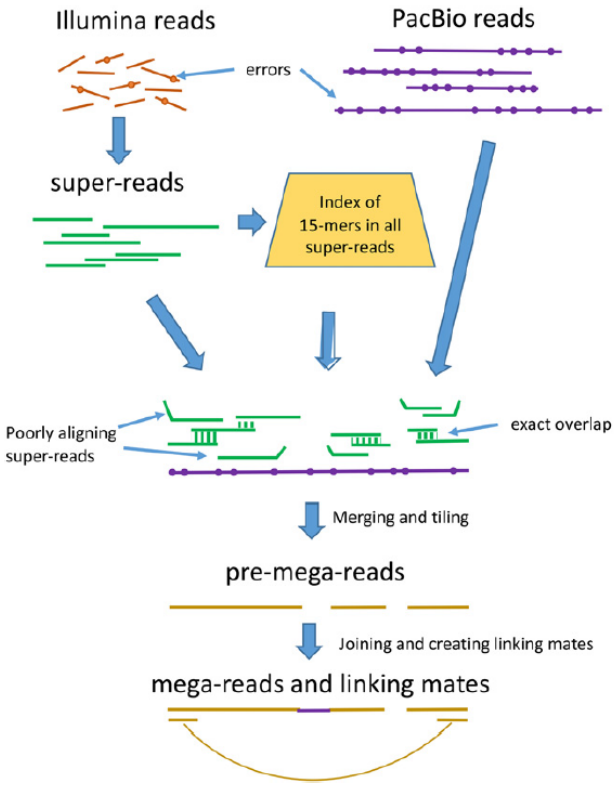
3.3.2 MaSuRCA (de Bruijnův graf / Overlap layout consensus)

Assembler celým jménem Maryland Super-Read Celera Assembler. Typ hybridního assembleru, který využívá výpočetní efektivitu de Bruijnových grafů a flexibilitu Overlap layout consensus metody. Assembly je sestavováno buď pouze pomocí krátkých readů, nebo v kombinaci s dlouhými ready, kdy jsou výsledné assembly výrazně kvalitnější. Kombinují se například data z platformy Illumina spolu s dlouhými ready z platformy PacBio (Zimin *et al.* 2013).

V prvním kroku se z krátkých readů vytvoří tzv. *super-ready*, což jsou původní ready rozšířené (na 3‘ i 5‘ konci). V dalším kroku jsou super-reads rozděleny na ready délky *k* (většinou k = 15). Pro každý read z této databáze se spočítá sequence alignment ke každému dlouhému readu a aplikuje se algoritmus Overlap layout consensus, jehož výsledkem jsou

[3] Bloom filtr – datová struktura, která umožňuje přidávat a ověřovat přítomnost prvků v množině v konstantním čase bez nutnosti uchovávání daných prvků, čímž výrazně snižuje paměťové nároky

tzv. *pre-megareads*, což jsou dlouhé ready obsahující mezery. Tyto mezery mohou být vyplněny kompatibilními úseky v dlouhých readech, ale vzhledem k chybovosti dlouhých readů k tomuto vyplnění dochází pouze, když se v daném úseku překrývá více dlouhých readů navzájem. Pokud tyto mezery nelze vyplnit (úsekem z dlouhého readu), vytvoří se tzv. *linked pair of reads* neboli spojovací vazba mezi ready překlenující mezeru. Po případném vyplnění mezer a vytvoření spojovací vazby přes mezery vzniknou tzv. *mega-reads* (Zimin *et al.* 2017). V posledním kroku jsou mega-reads poskládány do kontigů a následně do scaffoldů pomocí assembleru CABOG (Miller *et al.* 2008) (Obr. 7).



Obr. 7: **Tvorba mega-reads**
V prvním kroku se z krátkých readů vytvoří super-reads, z kterých následně je vytvořená databáze super-readů délky k. Pro každý read z databáze je proveden sequence alignment ke každému dlouhému readu. Na základě překryvů krátkých readů jsou sestaveny pre-mega reads. Sloučením těchto readů a případném vyplnění mezer pomocí kompatibilních úseků v dlouhých readech, či tvorbou spojovacích vazeb přes mezeru, vznikají mega-ready (Zimin *et al.* 2017).

4 Hodnocení kvality genomové assembly

K sestavení assembly je k dispozici množství programů, u kterých má uživatel navíc možnost nastavení různých parametrů výrazně ovlivňujících výsledek. Nastavení jednotlivých parametrů je třeba na konkrétních datech optimalizovat a výsledkem je tedy množství různých assembly, které je třeba z hlediska kvality vyhodnotit.

Pro porovnávání assemblerů byla v roce 2011 spuštěna soutěž Assemblathon 1, kde na simulovaných datech byla vyhodnocena kvalita jednotlivých assembly od různých assemblerů (Earl *et al.* 2011). Tvůrci assemblerů nejdříve sestaví assembly s defaultním nastavením a následně se pokouší vylepšovat parametry tak, aby byla vytvořena co nejlepší assembly. Simulovaná data se však od reálných dat mohou lišit, proto assemblery sestavující nejlepší assembly dle Assemblathonu 1 nemusí vykazovat stejně dobré výsledky na reálných datech (Baker 2012). Proto v roce 2013 byla spuštěna soutěž Assemblathon 2, kde assemblery pracují s reálnými daty (Bradnam *et al.* 2013). Výsledky Asseblathonu 2 ukazují, že většina assemblerů vykazuje dobré výsledky v jedné určité metrice, zatímco v jiných metrikách má výsledky horší. Dále bylo ukázáno, že assembler, který sestavil nejlepší assembly jednoho organismu, tak u jiného organismu sestavil mnohem horší assembly. Sestavená assembly také závisela na velikosti genomu daného organismu. Je dobré si uvědomit k jakému účelu genomové assembly bude sloužit a dle toho se zaměřit na konkrétní hodnotící metriku (Bradnam *et al.* 2013).

4.1 Metriky hodnocení kvality genomové assembly

Kvalitu genomové assembly je možné hodnotit z hlediska technických a biologických parametrů. Technické parametry reprezentují spojitost neboli *continuity* výsledné assembly a zahrnují metriky délky kontigů a scaffoldů, počty misassemblies v kontizích (reprezentující strukturní chyby) a funkční elementy v assembly (pokrytí genomu, procento GC nukleotidů a další). Technické parametry však dostatečně nereprezentují úplnost neboli *completness* výsledné assembly, která je vyhodnocena pomocí biologických parametrů na základě ortologních genů[4], které se v daných organismech vyskytují. Tyto ortologní geny jsou uloženy v databázích CEGMA (Parra *et al.* 2007) a BUSCO (Simão *et al.* 2015).

[4] Ortologní gen – typ genu, který se vyvinul od společného předka, který si zachovává stejnou funkci v průběhu evoluce

4.1.1 Technické parametry

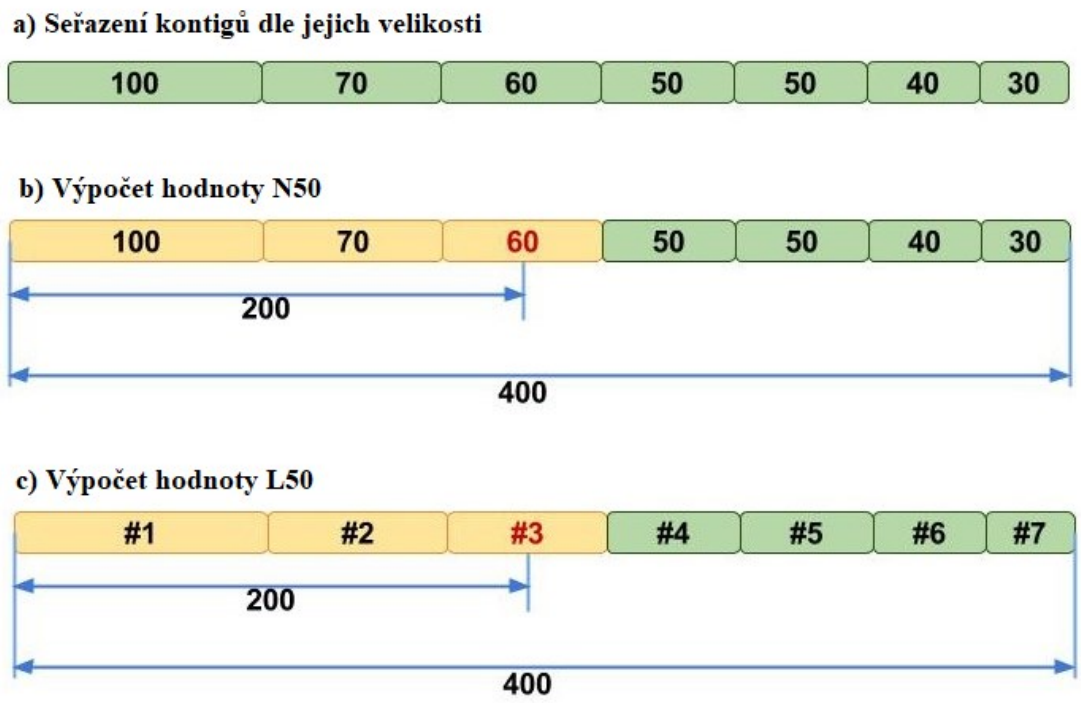
Nejpoužívanější metriky, které představují standartní měřítko pro hodnocení kvality. Dle těchto parametrů jsou jednotlivé assemblery navzájem porovnávány.

Délka a počet kontigů a scaffoldů

Typ metriky, který může být spočítán pro hodnocení assembly s i bez referenčního genomu (výjimkou je NGx – zde je referenční genom potřebný). Vyhodnocuje se zde celkový počet kontigů, čím má assembly méně kontigů, tím je kvalitnější, neboť osekvenovaný genom je méně rozfragmentovaný, ideální počet kontigů by se rovnal počtu chromozómů daného organismu. Kvalita nezávisí pouze na počtu kontigů, ale také na délce nejdelšího kontigu, kde s délkou roste kontinuální informace o genomu. Podobný princip platí i pro vyhodnocování scaffoldů.

- **Počet kontigů** – celkový počet kontigů ve výsledné assembly
- **Nejdelší kontig** – délka nejdelšího kontigu v assembly
- **Nejdelšího scaffold** – délka nejdelšího scaffoldu v assembly
- **Počet scaffoldů** – celkový počet scaffoldů ve výsledné assembly
- **Metrika Nx** – metrika vyjadřující délku kontigu takovou, že součet kontigů této délky a kontigů větších je roven nebo větší než x% ($0 \leq x \leq 100$) délky sestavené assembly (součtu délek všech kontigů), pokud x = 50, tak se jedná o median délek kontigů (Earl *et al.* 2011)
    - nejpoužívanější hodnotou je N50, postup získání této hodnoty je následující:
        1) Seřazení kontigů dle jejich velikosti (Obr. 8 a) od největšího po nejmenší
        2) Sčítání délek kontigů (od začátku), dokud se součet nerovná polovině délky celkové assembly, délka kontigu, u kterého se součet zastavil, je výsledná hodnota N50 (Obr. 8 b)
    - podobným způsobem lze vypočítat tuto hodnotu pro různé hodnoty x (N25, N80, N90) – seřazení kontigů dle velikosti probíhá všude stejně, ale sčítání se zastaví dle zadané hodnoty x (až hodnota součtu obsahuje alespoň x% délky assembly) (Videvall 2017)

- **Metrika NGx** – normalizovaná metrika Nx, výpočet hodnoty probíhá stejným způsobem jako u Nx metriky, akorát velikost assembly je nahrazená velikostí referenčního genomu
- **Metrika Lx** – metrika vyjadřující počet kontigů jejichž součet délek je větší či roven hodnotě x ($0 \leq x \leq 100$). Nejprve se provede seřazení kontigů dle jejich délky (od nejdelšího po nejkratší). Následně se provádí stejný součet jak u Nx metriky, akorát se nezaznamenává délka kontigu, u kterého se součet zastaví, ale počet kontigů, které tento součet tvoří (Obr. 8 c) (Videvall 2017).



Obr. 8: **Výpočet hodnot N50 a L50**
a) seřazení kontigů dle velikosti od největšího po nejmenší, **b) výpočet hodnoty N50** pomocí sčítání délek kontigů, dokud se součet délek nerovná polovině délky výsledného assembly (zde se uvažuje polovina, neboť se provádí výpočet hodnoty N50, pokud by se prováděl výpočet například N70, tak se sčítání délek provádí do doby, dokud se součet délek nerovná 70% délky výsledného assembly) a hodnota N50 odpovídá hodnotě 60 kbp, **c) výpočet hodnoty L50** stejně jako u výpočtu hodnoty N50 se sčítá délka kontigů, dokud se součet délek nerovná polovině délky výsledné assembly, ale nebere se v úvahu délka assembly, nýbrž počet kontigů, které tento součet tvoří (Videvall 2017)

Misassemblies a strukturní variace

Misassemblies jsou oblasti v sestaveném assembly, které obsahují rozsáhlé inzerce, delece, inverze či přeskupení, které jsou důsledkem špatného sestavení assembly. Jedná se o metriky popisující strukturní chyby vzniklé při sestavování kontigů a jsou závislé na referenčním genomu (Muggli *et al.* 2015). Nalezení těchto chybných oblastí je problematické, neboť skutečné biologické sekvence mohou vykazovat podobné znaky a

nemusí se zrovna jednat o misassembly (Salzberg & Yorke 2005). Zde se uvádí informace jako je počet kontigů obsahující misasembly, celkový počet bází v kontigu obsahující misassembly a informace o lokálních misassemblies.

Reprezentace funkčních elementů genomu

Metriky, které jsou většinou závislé na existenci referenčního genomu (kromě procenta GC nukleotidů). Zde se již hodnotí funkční aspekty složení genomového assembly jako jsou procento pokrytí genomu, počet duplikací, procento GC nukleotidů, počet mismatches na 100 kb a počet delecí a inzercí na 100 kb (Gurevich *et al.* 2013).

4.1.2 Biologické parametry

Technické metriky nemohou poskytnout žádné informace týkající se genového složení assembly. Tyto metriky popisují tzv. *completeness* neboli úplnost genomové assembly, což je velmi důležitý hodnotící parametr. S rostoucím počtem osekvenovaných genomů roste počet známých genů, které jsou přítomny u konkrétních druhů organismů (Seppey *et al.* 2019). První publikovanou databází sloužící k mapování ortologních genů v assembly byla databáze CEGMA (Parra *et al.* 2007), které byla v roce 2015 ukončena podpora z důvodů ukončení financování (online zdroj č. 6) a byla nahrazena databází BUSCO (Simão *et al.* 2015). Databáze poskytují ortologní geny, které by se v dané assembly měly vyskytovat. Pokud v assembly nelze identifikovat větší množství daných BUSCO genů, tak je to známka toho, že výsledná assembly je špatně složená nebo daný organismus je nedostatečně osekvenovaný.

CEGMA (Core Eukaryotic Genes Mapping Approach)

Jedná se o výpočetní metodu sloužící k anotaci genů v nově sestavené assembly. Obsahuje databázi konzervovaných proteinových rodin vyskytujících se v eukaryotických organismech jako jsou: *Homo sapiens, Drosophila melanogaster, Arabidopsis thaliana, Caenorhabditis elegans, Saccharomyces cerevisiae* a *Schizosaccharomyces pombe*. Gen patří do databáze, pokud je přítomen pouze v jedné kopii minimálně u 4 z těchto organismů. Pro zajištění spolehlivosti anotace genů zahrnuje CEGMA použití skrytých Markovových modelů (HMM)[5] proteinových rodin (Parra *et al.* 2007)

[5] HMM – pravděpodobnostní modely, které zapouzdřují evoluční změny, ke kterým došlo v sadě souvisejících sekvencí pomocí multiple sequence alignmentu, zachycují informace specifické pro danou polohu v každém sloupci alignmentu

Anotace genů zahrnuje využití několika bioinformatických nástrojů. Pomocí TBLASTN se provede porovnání vstupní sekvence s databázovými sekvencemi a identifikují se odpovídající kandidátní oblasti v sekvenci, které mohou obsahovat ortologní geny. Následně se pro každý gen v databázi vytvoří skrytý Markovovův model pomocí softwaru HMMER (Eddy 1998). Každý tento profil je následně zpracován pomocí anotačního systému GenWise (Birney 2004) a dochází k vytvoření predikce genu pro každou kandidátní oblast v sekvenci. Tyto predikce jsou dále zpracovány programem Geneid (Parra 2003), pomocí něhož se provede porovnání s každým skrytým Markovovým modelem a dochází k filtrování, během něhož se zachovají jen ty genomové predikce, které jsou lepší než zadaný treshold. Treshold hodnota je hodnota maximálního skóre zarovnání všech genů nevyskytujících se v jádře se skrytými Markovovými profily (Parra *et al.* 2009).

BUSCO (Benchmarking Universal Single-Copy Orthologs)

Databáze obsahující tzv. *single-copy genes* neboli geny vyskytující se v jedné kopii na konkrétním místě v genomu, které mohou mít ortology u různých druhů organismů. Geny jsou zařazeny do kategorií. Gen patří do dané kategorie, pokud se vyskytuje v alespoň 90 % organismů dané kategorie. Verze 3 obsahuje 28 eukaryotických datových souborů a 16 prokaryotických datových souborů. Při hodnocení kvality assembly z dané kategorie organismů by tyto geny měly být nalezeny. Pokud ve výsledné assembly existuje větší množství BUSCO genů, které zde nelze identifikovat, tak je to známka toho, že sekvenační přístupy nedokázaly plně zachytit očekávaný obsah genu (Seppey *et al.* 2019).

Vyhledávání v této databázi pracuje na principu hledání motivů pomocí skrytých Markovových modelů, které byly vytvořeny z multiple sequnce alignmentu ortologních genů z této databáze a zachycují konzervovanou sekvenci genu. Jako vstup přijímá buď genom, genomovou sadu nebo transkriptom. V prvním kroku probíhá analýza vstupní sekvence s konsenzuálními sekvencemi databáze pomocí lokálního alignmentu. Dle nalezené shodné oblasti *(candidate region)* se extrahují genové modely na základě blokových profilů[6] a nakonec se určí skóre zadané sekvence proti skrytým Markovovým modelům konkrétních BUSCO genů (Seppey *et al.* 2019).

[6] Blokové profily – matice, která obsahuje polohově specifické frekvence blokových sloupců v multiple sequence alignmentu

4.2 Programy hodnotící kvalitu genomové assembly

Jedním z prvních z programů zaměřených na kontrolu kvality genomových assembly je Plantagora (Barthelson *et al.* 2011). Tento program je však určen jen pro hodnocení rostlinných genomů. Dalším vhodným nástrojem je CAGE (Salzberg *et al.* 2012), který vyhodnocuje technické parametry (popsané v předchozí podkapitole) a pro vyhodnocení je nutná existence referenčního genomu. Nevýhodou CAGE je, že není paralelizovaný (Gurevich *et al.* 2013). Níže jsou popsané dva často používané nástroje k hodnocení QUAST (Gurevich *et al.* 2013) a SQUAT (Yang *et al.* 2019), který zahrnuje i hodnocení kvality vstupních readů.

4.2.1 QUAST

Nástroj pro hodnocení kvality genomových assembly sloužící k vyhodnocení *de novo* i *reference based* assembly. Vyhodnocuje všechny výše zmíněné technické metriky za pomoci jiných softwarů jako jsou například Plantagora (Barthelson *et al.* 2011), GAGE (Salzberg *et al.* 2012), GeneMark.hmm (Lukashin 1998), GlimmerHMM (Majoros *et al.* 2004) a rozšiřuje je o další metriky jako je například počet genů a operonů v assembly vyhodnocené na základě uživatelem poskytnutého anotovaného seznamu pozic genů a operonů v referenčním genomu.

Na výstupu poskytuje souhrnné tabulky a grafy, které jsou velmi užitečné pro vzájemné porovnávání více zkonstruovaných assembly. Grafy lze rozdělit do několika skupin: *Nx grafy* (reprezentují hodnoty Nx, NGx, NAx a NGAx metrik), *grafy kumulativních četností* (kontigy jsou zde uspořádány od největšího po nejmenší – dle počtu bází), *graf vyjadřující obsah GC* (zobrazena distribuce GC v kontizích), *grafy zarovnaných kontigů* (reprezentující zarovnání neboli alignment kontigů s referenčním genomem a pozice misassemblies v těchto kontizích) a *srovnávací diagramy několika metrik* (jako například počet genů, operonů či pokrytí genomu) (Gurevich *et al.* 2013).

4.2.2 SQUAT

Jedná se o nástroj pro vyhodnocení kvality sestaveného assembly, kdy v procesu *pre-assembly* hodnotí kvalitu vstupních readů a v procesu *post-assembly* se hodnotí kvalita sestaveného assembly (Yang *et al.* 2019).

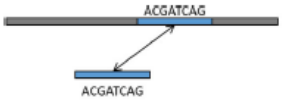
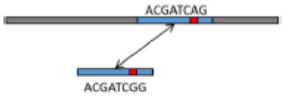
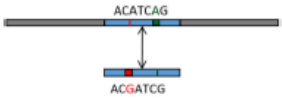
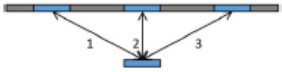
V **pre-assembly** se hodnotí kvalita vstupních readů na základě kvality jednotlivých bází, ze kterých se read skládá. Nejprve je vypočtena základní statistika vstupního FASTAQ souboru (počet bází a readů, minimální/maximální/průměrná délka readu, frekvence

jednotlivých bází a distribuce CG bází) a pro každý read se vypočítá tzv. *base quality score* (skóre jednotlivých bází v readu, jak dobře byly jednotlivé báze osekvenovány). Vstupní ready jsou dle tohoto skóre kategorizovány do 3 skupin. Rozdělení readů je vizualizováno v HTML výstupním souboru (Yang *et al.* 2019).

V **post-assembly** procesu slouží vytvořená assembly jako reference, na kterou se budou sekvenční ready mapovat. Mapování probíhá na základě dvou odlišných algoritmů BWA-backtrack (Li & Durbin 2009) a BWA-MEM (Li 2013), které patří do softwaru BWA[7]. Během mapování jsou nejednoznačně osekvenované ready (mají nejednoznačný base-calling) označeny typem N. Ostatní ready jsou rozděleny do šesti skupin, dle jejich mapování na výslednou assembly (Obr. 9):

- Typ F – ready, které nelze namapovat na assembly (failed to map)
- Typ M – ready, které se vyskytují na více místech v assembly (multi-mapped)
- Typ P – ready, které se vyskytují právě na jednom místě v assembly a shodují se s assembly v celé délce (perfectly-matched)
- Typ S – ready, které se vyskytují právě na jednom místě v assembly, ale neshodují se v celé délce (obsahují nějakou substituci)
- Typ C – ready, které obsahují na obou stranách alignmentu tzv. *clips* – read není zarovnán k assembly od začátku do konce, ale koncové části readu jsou odříznuty
- Typ O – ostatní ready, které obsahují inzerce či delece

[7] BWA – software určený pro mapování sekvencí proti referenčnímu genomu. Skládá se z BWA-backtrack, BWA-SW a BWA-MEM.

P		Perfectly-matched reads
S		Reads with substitution errors
C		Reads containing clips
O		Reads with other errors
M		Multi-mapped reads
F		Failed-to-map/Unmapped reads
N		Reads containing N

Obr. 9: **Rozdělení vstupních readů v SQUAT** (Yang *et al.* 2019)

Algoritmus BWA-MEM provádí lokální alignment, zatímco BWA-backtracking provádí globální alignment. Po rozdělení vstupní readů do těchto 7 kategorií, probíhá jejich analýza a rozdělení na špatně mapované a dobře mapované na základě sloupcového grafu, kde jsou tyto ready zaneseny (v pořadí: P, S, C, O, M, F, N). Výstupem post-assembly je procentuální zastoupení jednotlivých sedmi kategorií readů a základní statistika získaná z program QUAST (popsán výše) uložené v HMTL formátu (Yang *et al.* 2019).

5 Závěr

Předkládaná práce pojednává o problematice genomové assembly. Popisuje základní principy sekvenování, algoritmy na jejichž principu pracují assemblery a metriky, pomocí kterých probíhá vyhodnocení sestavené assembly. V praktické části bylo provedeno sestavení genomu ptačí motolice *Trichobilharzia szidati* pomocí dvou odlišných assemblerů a zhodnocení sestavené assembly pomocí softwaru hodnotící technické parametry a softwaru hodnotící biologické parametry.

Existuje celá řada sekvenačních platforem, které sekvenují různě dlouhé úseky genomu s určitou chybovostí. Neexistence sekvenační platformy schopné osekvenovat celou genomovou sekvenci bez chyby vede k tomu, že se stejná data osekvenují na více platformách a využije se výhod daného stroje.

K sestavení genomové assembly byla navržena celá řada assemblerů, které skládají genomovou assembly jak pouze z krátkých readů či pouze z dlouhých readů, tak pomocí kombinace obou druhů readů dohromady (tzv. hybridní assemblery). Z experimentálních výsledků pro assembler MaSuRCA vyplývá, že kombinace krátkých a dlouhých readů dává kvalitnější assembly. Většina assemblerů umožňují nastavení vstupních parametrů, které také výrazně ovlivňují kvalitu výsledné assembly. Výběr assembleru se odvíjí od sekvenačních dat, které jsou k dispozici, a velikosti genomu sekvenovaného organismu. Nicméně vybrat správný assembler a nastavit správně vstupní parametry je složité, neboť neexistuje univerzální nastavení, vše se odvíjí od povahy zkoumaného genomu a vstupních sekvenačních dat. Doporučuje se vyzkoušet více složení genomových assembly pomocí odlišných assemblerů s různými vstupními parametry a tyto assembly následně vyhodnotit dle příslušných metrik (technické, biologické).

K vyhodnocení kvality genomové assembly byly navrženy technické a biologické parametry, které obsahují hodnotící metriky. Ukázalo se, že žádná z metrik plně nepopisuje kvalitu výsledné assembly, neboť technické parametry hodnotí assembly pouze na základě délky sestavených kontigů, scaffoldů a počtu strukturních variací a neberou v úvahu biologicky funkční elementy genomu. Biologické parametry zase nevystihují spojitost sestavené assembly. Proto při vyhodnocení kvality je vhodné dívat se na výsledky jak z technických, tak z biologických metrik.

K vyřešení problémů, které se vztahují k tvorbě genomových assembly by bylo vhodné vymyslet sekvenační platformu, která bude sekvenovat co největší fragmenty sekvencí s nízkou chybovostí. Tento proces by vyřešil problém s repetitivními úseky v genomu a tvorba genomové assembly by nebyla tak výpočetně náročná.

6 Praktická část

6.1 Úvod

Cílem praktické části bylo zjistit na stejné datové sadě, který z assemblerů (ABySS, MaSuRCA) sestaví kvalitnější assembly. K hodnocení kvality sestavených assembly byly využity softwary QUAST a BUSCO.

Analýza probíhala na datech – ptačí motolice *Trichobilharzia szidati* získaných v rámci projektu 50 Helminth Genomes Intiative (online zdroj č. 7). Z důvodu nedostatečné kvality výsledného assembly však nebyl sestavený genom publikován.

Trichobilharzia szidati je ptačí motolice z čeledi *Schistosomatidae*. V rámci životního cyklu střídá 2 hostitele. Mezihostitelem je vodní plž *Lymnaea stagnalis* a konečným hostitelem je vodní ptactvo řádu *Anseriformes*. Tato motolice může infikovat i savce včetně člověka, kde nákaza způsobuje alergickou kožní reakci tzv. cerariovou dermatitidu (Horák *et al.* 2002).

Pro *T. szidati* není k dispozici referenční genom, proto probíhala tvorba *de novo* genomové assembly. Pro představu velikosti genomu, zastoupení GC párů a dalších informací byly využity blízce příbuzné organismy *Trichobilharzia regenti* a *Schistosoma mansoni* (Tab. 2), ke kterým je k dispozici referenční genom.

K sestavení assembly byly vybrány assemblery ABySS a MaSuRCA. **ABySS** je vhodný pro sestavování assembly velkých genomů a je vhodný pro zpracování dat ze sekvenační platformy Illumina. V soutěži Assemblathon 2 sestavil kvalitní assembly, konkrétně pro sestavení genomu hada (*Boa* constrictor) dosáhl nejvyššího celkového skóre kvality genomové assembly (Bradnam *et al.* 2013). **MaSuRCA** je hybridní assembler, který je vhodný jak pro krátké ready, tak pro dlouhé ready. Funguje na bázi de Bruijnových grafů a Overlap layout consensus algoritmu.

Tab. 2: Základní charakteristiky genomů *Trichobilharzia regenti* a *Schistosoma mansoni*

	Trichobilharzia regenti	***Schistosoma mansoni***
Velikost genomu:	701 762 036 bp	409 579 008 bp
Délka scaffoldů:	702 Mb	410 Mb
Nejdelší scaffold:	141,1 kB	88,9 Mb
Délka N50:	7,7 kb	50,5 Mb
Procento GC párů:	37,4	35,5

6.2 Metodika

Výpočetní kapacita byla využívána v rámci infrastruktury Metacentrum.

Tvorba assembly

Pro sestavení assembly byla použita data z Whole Genome Sequencing of *Trichobilharzia szidati* (https://www.ncbi.nlm.nih.gov/sra/ERR119617). Sekvenační data byla získána z platformy Illumina HiSeq 2000 a jedná se o pair-end ready o velikosti 100 bp. Celkem je k dispozici 91 000 000 pair-end readů. Pro zkonstruování výsledné assembly byly vybrány assemblery ABySS a MaSuRCA.

Assembler ABySS byl spuštěn ve verzi abys-2.2.3 s následujícími parametry: -pe; -k=64; zbytek parametrů ponechán na defaultních hodnotách.

Assembler MaSuRCA byl spuštěn ve verzi masurca-3.2.6 s následujícími parametry: JF_SIZE = 8191580160; GRAPH_KMER_SIZE = auto; USE_LINKING_MATES = 1; EXTENDED_JUMP_READS = 0

Hodnocení kvality assembly

Pro vyhodnocení kvality sestavených assembly byl použity softwary QUAST a BUSCO.

Software QUAST byl spuštěn ve verzi quast-4.6.3

- ABySS: `quast.py t_szidati-scaffolds.fa`
- MaSuRCA: `quast.py final.genome.scf.fasta`

Software BUSCO byl spuštěn ve verzi busco-3.0.2 a pouze na výslednou assembly od assembleru MaSuRCA (dle QUAST kvalitnější assembly) a byla zvolena databáze eukaryota_odb9

```
run_BUSCO.py -m genome -i final.genome.scf.fasta -o BUSCO_OUTPUT
-l /software/busco/3.0.2/src/db/eukaryota_odb9
```

6.3 Výsledky a diskuze

Hodnocení kvality sestavených genomových assemblery bylo provedeno pomocí softwaru QUAST (Tab. 3). Z výsledků je patrné, že assembler MaSuRCA sestavil podstatně kvalitnější assembly, a proto bylo na těchto datech provedeno i hodnocení biologických metrik pomocí softwaru BUSCO (Tab. 4).

V porovnání s příbuznými organismy (*Trichobilharzia regenti* a *Schistosoma mansoni*) je genomové assembly získané z assembleru MaSuRCA podobné assembly *Trichobilharzia regenti,* v metrice N50, nejdelší scaffold a v procentu GC párů. Ale i tak výsledné genomové assembly nedosahuje požadované kvality, což může být způsobeno tím, že před prací s těmi ready nebyla provedena žádná úprava vstupních readů.

Tab. 3: Výsledky hodnocení kvality assembly pomocí softwaru QUAST

	ABySS	**MaSuRCA**
Délka scaffoldů	9 691 788 bp	1 196 066 613 bp
Nejdelší scaffold	13,7 kb	106,9 kb
Délka N50	0,67 kb	6,4 kb
Procento GC párů	37.24	37.46
Celkový počet scaffoldů	13232	303407

Tab. 4: Výsledky hodnocení kvality assembly pomocí softwaru BUSCO

	MaSuRCA
Complete BUSCOs[1]	51
Complete and single-copy BUSCOs	43
Complete and duplication BUSCOs	8
Fragmented BUSCOs[2]	68
Missing BUSCOs[3]	184
Total BUSCO groups searched	303

[1] **Complete BUSCOs** jsou geny, které se ve výsledné genomové assembly vyskytují jak single-copy, tak na více místech (duplication), jedná se o geny, které skórovaly v celém očekávaném rozsahu skóre (zahrnují Complete and single-copy BUSCOs a Complete and suplication BUSCOs)

[2] **Fragmented BUSCOs** jsou geny, jejichž BUSCO skóre nebylo příliš malé ani dostatečně velké na to, aby patřily mezi Complete BUSCOs

[3] **Missing BUSCOs** jsou geny, které nevykazují žádné shody, nebo jejich BUSCO skóre je příliš malé

7 Zdroje

Antipov, D., Korobeynikov, A., McLean, J. S., & Pevzner, P. A. (2016). hybridSPAdes: an algorithm for hybrid assembly of short and long reads. *Bioinformatics*, **32**(7), 1009–1015.

Ashton, P. M., Nair, S., Dallman, T., … O'Grady, J. (2015). MinION nanopore sequencing identifies the position and structure of a bacterial antibiotic resistance island. *Nature Biotechnology*, **33**(3), 296–300.

Ayling, M., Clark, M. D., & Leggett, R. M. (2019). New approaches for metagenome assembly with short reads. *Briefings in Bioinformatics*.

Baker, M. (2012). De novo genome assembly: what every biologist should know. *Nature Methods*, **9**(4), 333–337.

Bankevich, A., Nurk, S., Antipov, D., … Pevzner, P. A. (2012). SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing. *Journal of Computational Biology*, **19**(5), 455–477.

Barthelson, R., McFarlin, A. J., Rounsley, S. D., & Young, S. (2011). Plantagora: Modeling Whole Genome Sequencing and Assembly of Plant Genomes. *PLoS ONE*, **6**(12), e28436.

Bastide, M., & McCombie, W. R. (2007). Assembling Genomic DNA Sequences with PHRAP. *Current Protocols in Bioinformatics*, **17**(1).

Batzoglou, S. (2002). ARACHNE: A Whole-Genome Shotgun Assembler. *Genome Research*, **12**(1), 177–189.

Bayley, H. (2006). Sequencing single molecules of DNA. *Current Opinion in Chemical Biology*, **10**(6), 628–637.

Birney, E. (2004). GeneWise and Genomewise. *Genome Research*, **14**(5), 988–995.

Bradnam, K. R., Fass, J. N., Alexandrov, A., … Korf, I. F. (2013). Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. *GigaScience*, **2**(1), 10.

Chapman, J. A., Ho, I., Sunkara, S., Luo, S., Schroth, G. P., & Rokhsar, D. S. (2011). Meraculous: De Novo Genome Assembly with Short Paired-End Reads. *PLoS ONE*, **6**(8), e23501.

Chen, Q., Lan, C., Zhao, L., Wang, J., Chen, B., & Chen, Y.-P. P. (2017). Recent advances in sequence assembly: principles and applications. *Briefings in Functional Genomics*, **16**(6), 361–378.

Chikhi, R., & Medvedev, P. (2014). Informed and automated k-mer size selection for genome assembly. *Bioinformatics*, **30**(1), 31–37.

Chu, T.-C., Lu, C.-H., Liu, T., Lee, G. C., Li, W.-H., & Shih, A. C.-C. (2013). Assembler for de novo assembly of large genomes. *Proceedings of the National Academy of Sciences*, **110**(36), E3417–E3424.

Compeau, P. E. C., Pevzner, P. A., & Tesler, G. (2011). How to apply de Bruijn graphs to genome assembly. *Nature Biotechnology*, **29**(11), 987–991.

Deshpande, V., Fung, E. D. K., Pham, S., & Bafna, V. (2013). Cerulean: A Hybrid Assembly Using High Throughput Short and Long Reads. In A. Darling & J. Stoye, eds., *Algorithms in Bioinformatics*, Vol. 8126, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 349–363.

Earl, D., Bradnam, K., St. John, J., … Paten, B. (2011). Assemblathon 1: A competitive assessment of de novo short read assembly methods. *Genome Research*, **21**(12), 2224–2241.

Eddy, S. R. (1998). Profile hidden Markov models. *Bioinformatics*, **14**(9), 755–763.

Gnerre, S., MacCallum, I., Przybylski, D., … Jaffe, D. B. (2011). High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proceedings of the National Academy of Sciences*, **108**(4), 1513–1518.

Goodwin, S., Gurtowski, J., Ethe-Sayers, S., Deshpande, P., Schatz, M. C., & McCombie, W. R. (2015). Oxford Nanopore sequencing, hybrid error correction, and de novo assembly of a eukaryotic genome. *Genome Research*, **25**(11), 1750–1756.

Goodwin, S., McPherson, J. D., & McCombie, W. R. (2016). Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, **17**(6), 333–351.

Gurevich, A., Saveliev, V., Vyahhi, N., & Tesler, G. (2013). QUAST: quality assessment tool for genome assemblies. *Bioinformatics*, **29**(8), 1072–1075.

He, Y., Zhang, Z., Peng, X., Wu, F., & Wang, J. (2013). De novo assembly methods for next generation sequencing data. *Tsinghua Science and Technology*, **18**(5), 500–514.

Horák, P., Kolárová, L., & Adema, C. (2002). Biology of the schistosome genus Trichobilharzia. *Advances in Parasitology*, **52**, 155–233.

Huang, X. (1999). CAP3: A DNA Sequence Assembly Program. *Genome Research*, **9**(9), 868–877.

Huang, X., & Yang, S. (2005). Generating a Genome Assembly with PCAP. *Current Protocols in Bioinformatics*, **11**(1).

Jackman, S. D., Vandervalk, B. P., Mohamadi, H., … Birol, I. (2017). ABySS 2.0: resource-efficient assembly of large genomes using a Bloom filter. *Genome Research*, **27**(5), 768–777.

Koren, S., Schatz, M. C., Walenz, B. P., … Phillippy, A. M. (2012). Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nature Biotechnology*, **30**(7), 693–700.

Levy, S. E., & Myers, R. M. (2016). Advancements in Next-Generation Sequencing. *Annual Review of Genomics and Human Genetics*, **17**(1), 95–115.

Li, H. (2013). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *ArXiv:1303.3997 [q-Bio]*. Retrieved from http://arxiv.org/abs/1303.3997

Li, H., & Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, **25**(14), 1754–1760.

Li, Y.-L., Weng, J.-C., Hsiao, C.-C., Chou, M.-T., Tseng, C.-W., & Hung, J.-H. (2015). PEAT: an intelligent and efficient paired-end sequencing adapter trimming algorithm. *BMC Bioinformatics*, **16**(Suppl 1), S2.

Liu, L., Li, Y., Li, S., … Law, M. (2012). Comparison of Next-Generation Sequencing Systems. *Journal of Biomedicine and Biotechnology*, **2012**, 1–11.

Lukashin, A. (1998). GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Research*, **26**(4), 1107–1115.

Luo, R., Liu, B., Xie, Y., … Wang, J. (2012). SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *GigaScience*, **1**(1), 18.

Magoc, T., Pabinger, S., Canzar, S., … Salzberg, S. L. (2013). GAGE-B: an evaluation of genome assemblers for bacterial organisms. *Bioinformatics*, **29**(14), 1718–1725.

Majoros, W. H., Pertea, M., & Salzberg, S. L. (2004). TigrScan and GlimmerHMM: two open source ab initio eukaryotic gene-finders. *Bioinformatics*, **20**(16), 2878–2879.

Margulies, M., Egholm, M., Altman, W. E., … Rothberg, J. M. (2005). Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, **437**(7057), 376–380.

Maxam, A. M., & Gilbert, W. (1977). A new method for sequencing DNA. *Proceedings of the National Academy of Sciences*, **74**(2), 560–564.

McCombie, W. R., McPherson, J. D., & Mardis, E. R. (2019). Next-Generation Sequencing Technologies. *Cold Spring Harbor Perspectives in Medicine*, **9**(11), a036798.

Metzker, M. L. (2010). Sequencing technologies — the next generation. *Nature Reviews Genetics*, **11**(1), 31–46.

Miller, J. R., Delcher, A. L., Koren, S., … Sutton, G. (2008). Aggressive assembly of pyrosequencing reads with mates. *Bioinformatics*, **24**(24), 2818–2824.

Miller, J. R., Koren, S., & Sutton, G. (2010). Assembly algorithms for next-generation sequencing data. *Genomics*, **95**(6), 315–327.

Muggli, M. D., Puglisi, S. J., Ronen, R., & Boucher, C. (2015). Misassembly detection using paired-end sequence reads and optical mapping data. *Bioinformatics*, **31**(12), i80–i88.

Mullikin, J. C. (2003). The Phusion Assembler. *Genome Research*, **13**(1), 81–90.

Myers, E. W. (2000). A Whole-Genome Assembly of Drosophila. *Science*, **287**(5461), 2196–2204.

Parra, G. (2003). Comparative Gene Prediction in Human and Mouse. *Genome Research*, **13**(1), 108–117.

Parra, G., Bradnam, K., & Korf, I. (2007). CEGMA: a pipeline to accurately annotate core genes in eukaryotic genomes. *Bioinformatics*, **23**(9), 1061–1067.

Parra, G., Bradnam, K., Ning, Z., Keane, T., & Korf, I. (2009). Assessing the gene space in draft genomes. *Nucleic Acids Research*, **37**(1), 289–297.

Pevzner, P. A., Tang, H., & Waterman, M. S. (2001). An Eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences*, **98**(17), 9748–9753.

Pop, M., Salzberg, S. L., & Shumway, M. (2002). Genome sequence assembly: algorithms and issues. *Computer*, **35**(7), 47–54.

Rasko, D. A., Webster, D. R., Sahl, J. W., … Waldor, M. K. (2011). Origins of the *E. coli* Strain Causing an Outbreak of Hemolytic–Uremic Syndrome in Germany. *New England Journal of Medicine*, **365**(8), 709–717.

Salzberg, S. L., Phillippy, A. M., Zimin, A., … Yorke, J. A. (2012). GAGE: A critical evaluation of genome assemblies and assembly algorithms. *Genome Research*, **22**(3), 557–567.

Salzberg, S. L., & Yorke, J. A. (2005). Beware of mis-assembled genomes. *Bioinformatics*, **21**(24), 4320–4321.

Sanger, F., Nicklen, S., & Coulson, A. R. (1977). DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, **74**(12), 5463–5467.

Schirmer, M., Ijaz, U. Z., D'Amore, R., Hall, N., Sloan, W. T., & Quince, C. (2015). Insight into biases and sequencing errors for amplicon sequencing with the Illumina MiSeq platform. *Nucleic Acids Research*, **43**(6), e37.

Seppey, M., Manni, M., & Zdobnov, E. M. (2019). BUSCO: Assessing Genome Assembly and Annotation Completeness. In M. Kollmar, ed., *Gene Prediction*, Vol. 1962, New York, NY: Springer New York, pp. 227–245.

Shendure, J., Balasubramanian, S., Church, G. M., … Waterston, R. H. (2017). DNA sequencing at 40: past, present and future. *Nature*, **550**(7676), 345–353.

Simão, F. A., Waterhouse, R. M., Ioannidis, P., Kriventseva, E. V., & Zdobnov, E. M. (2015). BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics*, **31**(19), 3210–3212.

Simpson, J. T., & Durbin, R. (2012). Efficient de novo assembly of large genomes using compressed data structures. *Genome Research*, **22**(3), 549–556.

Simpson, J. T., Wong, K., Jackman, S. D., Schein, J. E., Jones, S. J. M., & Birol, I. (2009). ABySS: A parallel assembler for short read sequence data. *Genome Research*, **19**(6), 1117–1123.

Sohn, J., & Nam, J.-W. (2016). The present and future of *de novo* whole-genome assembly. *Briefings in Bioinformatics*, bbw096.

Van Nieuwerburgh, F., Thompson, R. C., Ledesma, J., … Head, S. R. (2012). Illumina mate-paired DNA sequencing-library preparation using Cre-Lox recombination. *Nucleic Acids Research*, **40**(3), e24–e24.

Vasudevan, K., Devanga Ragupathi, N. K., Jacob, J. J., & Veeraraghavan, B. (2020). Highly accurate-single chromosomal complete genomes using IonTorrent and MinION sequencing of clinical pathogens. *Genomics*, **112**(1), 545–551.

Videvall, E. (2017, March 29). What's N50? Retrieved from https://www.molecularecologist.com/2017/03/whats-n50/

Warren, R. L., Sutton, G. G., Jones, S. J. M., & Holt, R. A. (2007). Assembling millions of short DNA sequences using SSAKE. *Bioinformatics*, **23**(4), 500–501.

Yang, L.-A., Chang, Y.-J., Chen, S.-H., Lin, C.-Y., & Ho, J.-M. (2019). SQUAT: a Sequencing Quality Assessment Tool for data quality assessments of genome assemblies. *BMC Genomics*, **19**(9), 238.

Ye, C., Hill, C. M., Wu, S., Ruan, J., & Ma, Z. (Sam). (2016). DBG2OLC: Efficient Assembly of Large Genomes Using Long Erroneous Reads of the Third Generation Sequencing Technologies. *Scientific Reports*, **6**(1), 31900.

Zerbino, D. R., & Birney, E. (2008). Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research*, **18**(5), 821–829.

Zimin, A. V., Marçais, G., Puiu, D., Roberts, M., Salzberg, S. L., & Yorke, J. A. (2013). The MaSuRCA genome assembler. *Bioinformatics*, **29**(21), 2669–2677.

Zimin, A. V., Puiu, D., Luo, M.-C., … Salzberg, S. L. (2017). Hybrid assembly of the large and highly repetitive genome of Aegilops tauschii, a progenitor of bread wheat, with the MaSuRCA mega-reads algorithm. *Genome Research*, **27**(5), 787–792.

8 Online zdroje

1. Longer and longer: DNA sequence of more than two million bases now achieved with nanopore sequencing. Oxford Nanopore Technologies [online] [cit. 2020-05-25]. Dostupné z: https://nanoporetech.com/about-us/news/longer-and-longer-dna-sequence-more-two-million-bases-now-achieved-nanopore
2. What is mate pair sequencing for? [online] [cit. 2020-04-03]. Dostupné z: https://www.ecseq.com/support/ngs/what-is-mate-pair-sequencing-useful-for
3. Oxford Nanopore Technologies: https://nanoporetech.com/
4. PacBio: https://www.pacb.com/
5. Illumina: https://www.illumina.com/
6. Goodbye CEGMA, hello BUSCO! ACGT [online] [cit. 2020-05-06]. Dostupné z: http://www.acgt.me/blog/2015/5/18/goodbye-cegma-hello-busco
7. *Comparative genomics of the major parasitic worms | Nature Genetics* [online] [cit. 2020-06-02]. Dostupné z: https://www.nature.com/articles/s41588-018-0262-1