Genomika Ewolucyjna i Populacyjna

Ogólnogenomowe resekwencjonowanie dla genomiki populacji

Ćwiczenia 1 Składanie genomów

Mateusz Konczal

Instrukcja skonstruowana jest w taki sposób, że zadania i materiały do samodzielnej pracy (lub wspólnej dyskusji), na podstawie których wystawiane będą oceny z ćwiczeń, znajdują się w niebieskich ramkach. Reszta zadań wykonywanych jest na podstawie wprost podanych poleceń lub wspólnie z prowadzącym ćwiczenia.

Ćwiczenia opracowanie zostały na podstawie skryptu Aruna Sethuramana z San Diego State University:

https://github.com/arunsethuraman/biomi609spring2022

oraz tutoriali EBI:

https://www.ebi.ac.uk/training/online/sites/ebi.ac.uk.training.online/files/user/18/private/velvet-practical_part-1.pdf

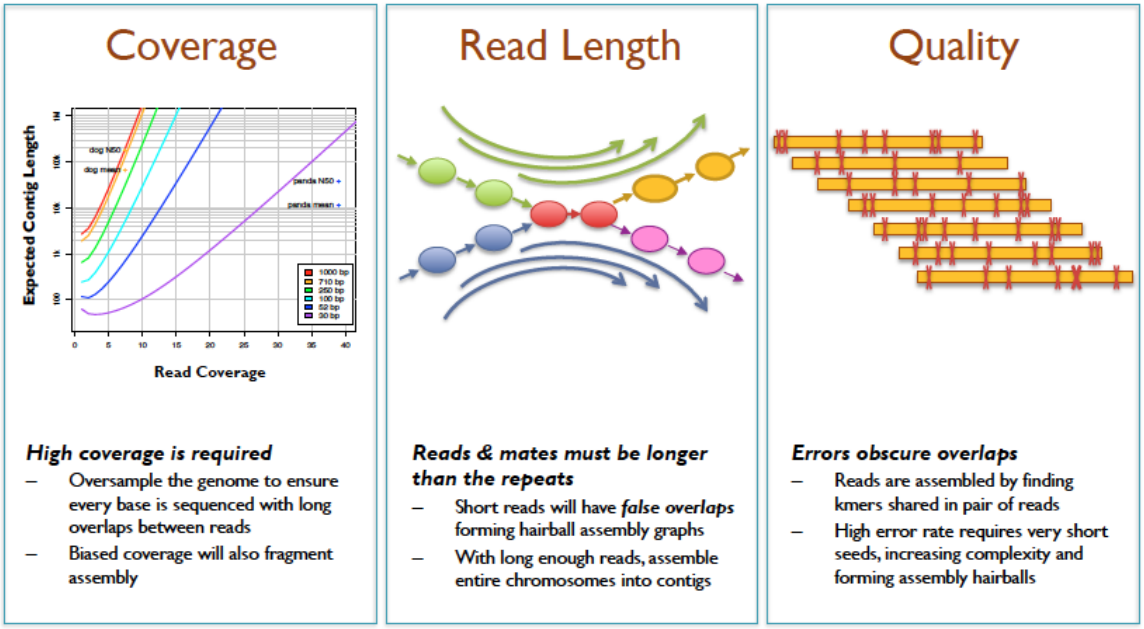
https://www.ebi.ac.uk/training/online/sites/ebi.ac.uk.training.online/files/user/18/private/velvet-practical_part-2.pdf

W celu przeprowadzenia zdecydowanej większości analiz z zakresu genomiki populacyjnej i ewolucyjnej potrzebny jest genom referencyjny. Obecnie dla wielu gatunków takie genomy są już dostępne. Nie zmienia to faktu, że składanie genomów ma cały czas spore znaczenie w bioinformatyce. Dzieje się tak dlatego, że osobniki różnią się zmiennością w wariantach strukturalnych, które trudno wykryć za pomocą mapowania odczytów do jednego genomu referencyjnego. Dodatkowo, mikroorganizmy i wirusy szybko ewoluują i zmieniają się z roku na rok. W takim przypadku, genom referencyjny staje się szybko przestarzały. Złożenie nowego genomu pozwala nie tylko uaktualnić referencję, ale także pozwala porównać go z poprzednią wersją. Dlatego dla mikroorganizmów często składa się genomy z poszczególnych próbek i potem je przyrównuje, zamiast mapować odczyty do genomu referencyjnego.

Składanie genomu (*ang. genome assembly*) jest trudnym problemem z wieloma aspektami z tym związanymi: (1) aspekt biologiczny – gatunki mogą mieć różną ploidalność, heterozygotyczne fragmenty mogą być trudne do rozwiązania, genomy mogą zawierać różną ilość sekwencji powtarzalnych (2) aspekt techniczny dot. sekwencjonowania – genomy są ogromne i błędy sekwencjonowania mogą prowadzić do błędów w dalszych analizach. (3) aspekt obliczeniowy – duże i skomplikowane genomy wymagają sporych mocy obliczeniowych i ogromnych zasobów pamięci podręcznej; (4) aspekt dot. dokładności – ponieważ nie wiemy jaki jest prawdziwy genom, który składamy, trudno jest oszacować dokładność i jakość przeprowadzonych analiz.

Na poniższej rycinie przedstawione są „składniki" potrzebne do dobrego złożenia genomu.

Ingredients for a good assembly



Current challenges in *de novo* plant genome sequencing and assembly
Schatz MC, Witkowski, McCombie, WR (2012) *Genome Biology*. 12:243

Podczas tych ćwiczeń skupimy się na złożeniu genomu wirusa SARS-CoV2. Zsekwencjonowane próbki wirusa dostępne są w publicznych bazach danych, np. www.gisaid.org i https://www.ncbi.nlm.nih.gov/sra . Genomy większości wirusów są bardzo małe w stosunku do organizmów eukariotycznych, co umożliwi nam sprawne przeprowadzenie obliczeń w trakcie trwania ćwiczeń. Przy odpowiednio dużych zasobach obliczeniowych, podobne analizy mogą być wykonane dla bardziej skomplikowanych genomów, chociaż w tym wypadku obecnie stosuje się inne technologie sekwencjonowania (długie odczyty).

DYSKUSJA:

Przedyskutuj z prowadzącym i innymi studentami różnice pomiędzy najpopularniejszymi technologiami sekwencjonowania, obecnymi na rynku. Jak różnice te mogą wpływać na proces składania genomów? Jakie technologie są najlepsze do tego celu?

Cel 1 – Instalacja programów Velvet, SRA Toolkit oraz FastQC

Uruchom komputer oraz otwórz wirtualną maszynę z zainstalowanym Linuxem. Otwórz terminal oraz stwórz katalog roboczy w Twoim katalogu domowym. Przejdź do katalogu roboczego i stwórz w nim podkatalog Tools, a następnie ściągnij i skompiluj program Velvet, który dostępny jest w podanym niżej repozytorium:

https://github.com/dzerbino/velvet

Dodaj katalog, w którym zainstalowany został velvet do ścieżki tak, tak żeby program dostępny był z każdego miejsca w terminalu:

```
export PATH=$PATH:/sciezka/do/katalogu/velvet_1.2.10
```

Następnie zainstaluj SRA Toolkit w dedykowanym dla niego katalogu, oraz dodaj go do ścieżki, korzystając z podpowiedzi podanych poniżej:

```
wget https://ftp-trace.ncbi.nlm.nih.gov/sra/sdk/2.11.3/sratoolkit.2.11.3-centos_linux64.tar.gz

tar -zxvf sratoolkit.2.11.3-centos_linux64.tar.gz

vdb-config --interactive

export PATH=$PATH:/ sciezka/do/katalogu/sratoolkit.2.11.2-centos_linux64/bin
```

Wykorzystując analogiczne podejście ściągnij i dodaj do ścieżki program FastQC.

Cel 2 – Pobranie danych dla SARS-CoV2, oraz ich procesowanie za pomocą SRA Toolkit. Analiza jakości i oczyszczanie za pomocą FastQC oraz trimmomatic

Na początek musimy pobrać dane dla wybranego genomu/próbki z bazy danych SRA. Otwórz stronę Short Read Archive NCBI a następnie wyszukaj próbkę SRX14015288.

ZADANIE:

Odczytaj podstawowe informacje o próbce SRX14015288. Odpowiedz na pytania:

Jaka technologia została użyta do sekwencjonowania?

Ile odczytów udało się uzyskać?

Ile nukleotydów zostało zsekwencjonowanie?

W zakładce „Data access” znajdź link do plików (typ „SRA Normalized”) a następnie pobierz je za pomocą programu wget do katalogu roboczego.

Teraz należy przeformatować ten plik do stadardowego formatu FASTQ. Użyjemy do tego jednego z programów dostępnych w ramach SRA Toolkit:

```
fastq-dump -I --split-files SRR17854410
```

Polecenie to powinno stworzyć parę plików. Otwórz pliki za pomocą programu less i przypominij sobie co znajduje się w plikach FASTQ i jak informacja ta jest zapisana. Poniżej porótce opisz ten format.

PYTANIE:

Przypominaj sobie co znajduje się w plikach FASTQ i jak informacja ta jest zapisana. Poniżej pokrótce opisz ten format.

ZADANIE:

Przeanalizuj jakość danych w plikach FASTQ za pomocą programu FastQC. Zwróć uwagę na średnią jakość nukleotydów oraz zanieczyszczenie adapterami. Poniżej przedstaw wyniki i interpretację.

ZADANIE:

Użyj programu trimmomatic, żeby usunąć sekwencje o niskiej jakości. Możesz wykorzystać poniższe polecenie:

```
trimmomatic PE -threads 3 SRR17854410_1.fastq SRR17854410_2.fastq
SRR17854410_1_trimmed.fastq SRR17854410un_1.fastq
SRR17854410_2_trimmed.fastq SRR17854410un_2.fastq TRAILING:30
```

Przeanalizuj wyniki programem FastQC i sprawdź w jaki sposób zmieniły się wyniki odpowiedź opisz poniżej. W jaki sposób zmodyfikowałbyś/zmodyfikowałabyś powyższe polecenie, żeby poprawić efektywność tego kroku. Odpowiedz na to pytanie posiłkując się instrukcją obsługi programu:
http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/TrimmomaticManual_V0.32.pdf

Cel 3 – Składanie genomu za pomocą programu VELVET

Odczyty, które ściągnęliśmy z bazy danych SRA, to stosukowo krótkie sekwencje połączone ze sobą w pary. Każda parach pochodzi z jednej sekwencji DNA o nie zaznanej dokładanie długość, aczkolwiek zazwyczaj mających kilkaset par zasad.

Podczas wykonywania obliczeń liczyć będziemy czas potrzebny na każde zadanie. W wierszu poleceń wystarczy przed konkretnym poleceniem uruchomić program `time`, które na koniec podsumuje nam czas potrzebny do wykonania konkretnego zadania. Składanie genomów bywa procesem wymagającym dużo czasu i zasobów, dlatego warto też w trakcie obliczeń monitorować zużycie zasobów, np. poprzez program `top`. Możesz otworzyć go w drugim oknie terminala i obserwować podczas realizacji opisanych poniżej zadań.

Składanie genomu wykonywać będziemy za pomocą programu `velvet`. Manual do programu znajduje się pod podanym poniżej adresem:

https://github.com/dzerbino/velvet/wiki/Manual

Zacznijmy nasze analizy używając k-merów długości 25, a wyniki zapiszmy do katalogu o nazwie run_25:

```
time velveth run_25 25 -shortPaired -separate \
-fastq SRR17854410_1_trimmed.fastq SRR17854410_2_trimmed.fastq
```

W przeciągu kilku minut obliczenia powinny się zakończyć.

PYTANIA:

Co to jest kmer?

Sprawdź jakie pliki zostały wygenerowane w katalogu run_25. Opisz ich zawartość. Co znajduje się w pliku log?

PYTANIE:

W tym samym katalogu co wcześniej uruchom velvetg, za pomocą polecenia:

```
time velvetg run_25
```

Sprawdź jakie nowy pliki powstały w katalogu run_25. Co zawierają? Jak możesz zinterpretować wartość N50, która znajduje się w pliku log?

W powyższym przykładzie użyliśmy domyślnych ustawień programu velvetg. Sprawdź jakie parametry możliwe są do zmodyfikowania. Uruchom program velvetg bez żadnych dodatkowych parametrów. W takim przypadku powinna pokazać nam się lista dostępnych opcji. Zwróć szczególną uwagę na dwa paramtery -cov_cutoff oraz -ex_cov . Co one oznaczają?

Opcja -cov_cutcoff pozwala na usunięcie kontigów, dla których pokrycie (*ang. coverage*) kmerów jest zbyt niskie, sugerując niską jakość, lub pochodzenie z innych źródeł niż analizowany genom (zanieczyszczenia). Druga opcja – „-exp_cov" jest wykorzystywana, do ustalenia oczekiwanego pokrycia w zsekwencjonowanej próbki.

W jaki sposób możemy otrzymać informacje o oczekiwanym pokryciu w naszej próbce? Jedną, prostą, metodą jest wykorzystanie wzoru C = LN/G, gdzie C – to pokrycie; L – długość odczytów, L – liczba odczytów; G – długość genomu. Znając oczekiwaną długość genomu (dla wirusa SARS-Cov2 jest to 29800 p.z.), liczbę zsekwencjonowanych odczytów (np. 12478) oraz długość tych odczytów (np. 150 p.z.), łatwo możemy obliczyć oczekiwane pokrycie, które w tym przypadku wyniesie 62,8. Innymi słowy oczekiwać będziemy, że każdy nukleotyd w genomie reprezentowany jest przez średnio przez 63 odczyty. Gdy nie znamy oczekiwanej wielkości genomu, możemy posiłkować się rozkładem częstości kmerów. Na końcu tego ćwiczenia podane są odnośniki do materiałów dodatkowych, gdzie możesz dowiedzieć się więcej na ten temat.

Przeprowadźmy teraz podobne obliczenia co wcześniej, tylko z zmodyfikowanymi opcjami. Zmieńmy nazwę pliku contig.fa, tak żeby nie został on nadpisany przez nowe analizy:

```
mv run_25/contig.fa run_25/contig.fa.0
```

```
time velvetg run_25 -cov_cutoff 16
```

Zapisz wynik składania genomu a następnie wykonaj obliczenia z dwoma parametrami `-cov_cutoff 16` and `-exp_cov 63`. Do tego momentu velvetg ignorował informację o parach sekwencji. Od momentu użycia tych dwóch parametrów velvetg stara się oszacować wielkość insertu, czy odległość pomiędzy końcami sekwencji z pary (po usunięciu adapterów). Polecenia będą wyglądały w następujący sposób:

```
mv run_25/contig.fa run_25/contig.fa.1
time velvetg run_25 -cov_cutoff 16 -exp_cov 63
```

Ile czasu potrzeba do wykonania poszczególnych obliczeń? Jaka wielkość insertu została oszacowana dla naszych danych?

ZADANIE:

Zapisz wynik poprzednich analiz. Następnie wykonaj uruchom program velvetg jeszcze raz, tym razem pozwól mu ustalić parametry -cov_cutoff oraz -exp_cov w sposób automatyczny. Sprawdź jak to zrobić w opisie poszczególnych funkcji. Jakie wartości zostały wybrane? Jak wyglądało Twoje polecenie i w jaki sposób zmieniły się wartości N50?

Cel 4. Ocena jakości genomów – program QUAST

Przejdź do stworzonego wcześniej katalogu Tools oraz zainstaluj tam program QUAST, który służy do oceny jakości genomów. Repozytorium zawierające ten program znajduje się tutaj:

https://github.com/ablab/quast

Żeby zainstalować program wykonaj następujące polecenia:

```
git clone https://github.com/ablab/quast.git
cd quast
./install.sh
```

Po zainstalowaniu programu dodaj go do ścieżki, tak żeby dostępny był z każdego miejsca. Pamiętaj, że dodanie programu do ścieżki jest aktywne tylko dla działającej sesji. Po wygaśnięciu sesji, trzeba wykonać te polecenia jeszcze raz. Alternatywnie można dodać tego typu polecenia do pliku konfigurującego basha, np. ~/.bashrc.

Porównamy teraz uzyskane przez nas genomy z genomem referencyjnym SARS_covV2. Genom referencyjny możesz znaleźć na stronie NCBI. Ściągnij go w postaci pliku fasta kopiując sekwencję, lub używając narzędzie NCBI o nazwie datasets download (https://www.ncbi.nlm.nih.gov/datasets/docs/v1/). Ściągnij również annotację w postaci pliku GFF. Następnie porównaj złożone przez siebie genomy, oraz genom referencyjny, za pomocą poniższego polecenia:

```
quast.py run_25/contigs.fa run_25/contigs.fa.0 -r reference.fasta -g GCf*.gz
```

ZADANIE:
Odczytaj i zinterpretuj wyniki wygenerowane przez program QUAST.

Materiały dodatkowe:

Inne programy używane do składania genomów

Velvet jest stosunkowo prostym narzędziem. Obecnie istnieją na rynku bardziej wyrafinowane programy, które produkują lepszej jakości genomy. Poniżej znajduje się lista kilku takich programów, które możesz eksplorować:

1) ABySS - https://github.com/bcgsc/abyss
2) SOAPdenovo2 - https://github.com/aquaskyline/SOAPdenovo2
3) SPAdes - http://cab.spbu.ru/software/spades/

Szacowanie wielkości genomu oraz oczekiwanego pokrycia za pomocą analizy rozkładu częstości kmerów

Poniżej znajdują się dwa tutoriale, pokazujące w jaki sposób można szacować rozkład częstości kmerów. Na maszynie wirutalnej zainstalowany jest program jellyfish, więc z sukcesem takie analizy można przeprowadzić (choć tylko dla stosukowo krótkich kmerów (max. 15), ze względu na ograniczenia RAM)

https://ucdavis-bioinformatics-training.github.io/2020-Genome_Assembly_Workshop/kmers/kmers

https://bioinformatics.uconn.edu/genome-size-estimation-tutorial/