Mini-FastQC

Daniel Kersting

Department of Computer Science, Free University of Berlin, Takustr. 9, 14195 Berlin, Germany
E-mail: *dkersting@zedat.fu-berlin.de*

ABSTRACT

Mini-FastQC ist eine kleinere Version des bekannten bioinformatischen Programms FastQC welches eine Programm zur Qualitätskontrolle von FastQ-Dateien repräsentiert. Mini-FastQC berechnet alle nötigen Werte für eine Qualitätskontrolle. Unter anderem GC-Gehalt, Kmer-Gehalt, Überexpremierte Seqenzen und eine pro Base Sequenzqualität. Dieses Paper beschreibt die Vorgehensweise des Teilprogramms zur ermittlung des Kmer-Gehalt und eine übersichtliche Ausgabe.

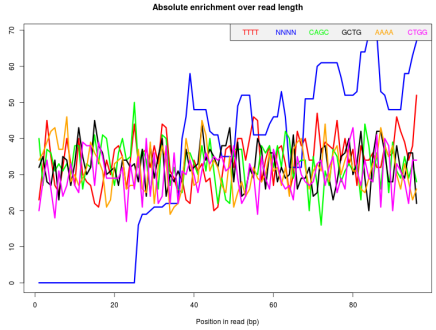

Figure 1. Kmer-Gehalt

1 EINLEITUNG

Einer der ersten Schritte zur Analyse von NGS-Daten (next generation sequencing) ist es, eine hochwertige und angemessene Maßnahme zur ermittlung der Daten zu nutzen.Dieses Paper gibt einen Einblick in "SeqC" ein NGS-Qualitätsprogramm unter Nutzung der Seqan-Bibliothek. Es wurde nach den Vorbildern bestehender Werkzeuge, wie FastQC (Andrews, 2010) und die FastX Toolkit (Gordon und Hannon, 2010) entwickelt. Der Begriff k-mer bezieht sich in der Regel auf ein bestimmtes n-Tupel oder n-Gramm von Nukleinsäure-oder Aminosäuresequenzen, welche verwendet werden, um bestimmte Regionen innerhalb von Biomolekülen zu identifizieren durch welche Proteinsequencen oder Genvorhersagen in der DNA lokalisiert werden können. Es werden entweder k-mer-Strings als solche für die Suche nach Regionen von Interesse verwendet werden oder k-mer Statistiken, welche diskrete Wahrscheinlichkeitsverteilungen in Bezug auf die Anzahl von möglichen k-mer-Kombinationen (oder vielmehr Permutationen mit Wiederholungen) geben verwendet. Spezifische kurze k-mere werden als Oligomere respektive "Oligos" bezeichnet.

2 METHODEN

Die Kmere werden durch ein C++ Programm ermittelt und gezählt. Ein R-Script hilft bei der Ploterstellung, welcher die ersten 6 meist vorkommenden Kmere grafisch über die Basenpaare darstellt. Ein HTML-File bietet im Anschluss an die Plot erstellung eine Übersicht über alle Plots. Von der "Pro Base Sequenzqualität" über den "GC-Gehalt" bis hin zu dem "Kmer-Gehalt".

2.1 Kmer-Gehalt

Der Kmer-Content beschreibt das Vorkommen von Kmeren über den Read. Hierbei werden die 6 höchst expremierten Kmere aufgezeigt. Auf der x-Achse sind die Basenpaare gezeigt und auf der y-Achse das absolute Vorkommen der Kmere.
Jeder Read wird einzeln durchlaufen und seine Kmere ermittelt. Zum speichern der Daten dient eine Map, welche als Schlüssel die Position in dem Read hat und als Wert eine weitere Map enthält in welcher der Schlüssel das Kmer und der Wert das Vorkommen dieses Kmers an besagter Position ist.

std::map¡unsigned,std::map¡std::string,unsigned¿ ¿

Hierzu gibt es noch eine zweite map, welche als Schlüssel den Kmer hat und als Wert das gesamte Vorkommen des Kmers in allen Reads.

std::map¡std::string,unsigned¿

Nach der Ermittlung der Kmere und ihrer Vorkommen, werden die 6 höchst expremierten Kmere in einer tsv-Datei gespeichert. Die tsv-Datei hat die Form einer Matrix. Die erste Zeile beinhaltet die Kmere und die erste Spalte beinhaltet die Position. In der dazugehörigen Zelle steht das Vorkommen des Kmers(Spalte) an seiner Position(Zeile).

2.2 Plotgenerierung

Jede berechnete Statistik wird in einer tsv-Datei gespeichert, welche dann mit Hilfe eines R-scripts eingelesen werden und für jede Statistik einen Plot erzeugt, welcher dann wiederum als png-Datei abgespeichert wird. Die Plots (png-Dateien) werden bei einem erneuten Aufruf des R-scripts einfach überschrieben.

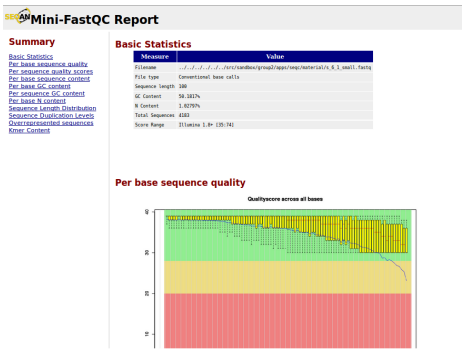

Figure 2. Ausgabe über eine HTML-Seite

2.3 Ausgabe

Die Ausgabe respektive das Zusammenführen der einzelnen Plots erfolgt über eine HTML-Datei. Diese bindet die Plots ein und listet sie auf. Zu Anfang steht eine kleine Zusammenfassung der Berechnungen, welche unteranderem den Dateinamen, den GC-Gehalt und die absolute Anzahl an Reads aufweist. An dem linken Seitenrand gibt es eine kleine Navigationsleiste, welche es erlaubt direkt zu der gewünschten Statistik zu wechseln.

3 ERGEBNISSE UND DISKUSSION

3.1 Ein- und Ausgabe

Die Anwendung wurde erfolgreich auf Sanger, Illumina HighSeq 2000 Solexa, Ion Torrent PGM und 454 GS FLX Datensätzen getestet. Auf allen Datensätzen unterscheidet sich die Ausgabe des FastQC von der unseren. Jedoch erwiesen sich diese Unterschiede nur als leichte Abweichungen von den Werten die FastQC errechnete. Zum Vergleichen der beiden Ausgaben, wurde ein Perl-script geschrieben, welche die beiden Ausgabedatein 1 zu 1 Vergleicht und die Unterschiede in eine Ausgabedatei schreibt. SeqC bietet nicht die Möglichkeit von Adapterentfernung oder Trimming. Dies wären noch Ergänzungsmöglichkeiten, welche das Programm noch komfortabler um nicht zu sagen interessanter machen würde.

3.2 Leistung

Die Anforderungen an die CPU-Zeit der Anwendung sind enorm über denen der FastQC. Eine Auflage von einer Datei 7GB (ENA: SRR066417 1) verwendet 13.2 min CPU-Zeit, welche 70min abgebrochen wurde. Der Speicherbedarf schien etwa bei 420MB pro Lauf stabil zu liegen während FastQC lediglich 190MB benötigt. Weitere Untersuchungen werden wahrscheinlich zeigen, dass die Effizienz der Bibliothek nicht vollständig ausgenutzt worden ist.

4 CONCLUSION

Die Anwendung SeqC repräsentiert die grundlegende Implementierung die ein *NGS-Quality-Reporting-Tool* mit sich bringen sollte. Die Software wurde in C++ mit Hilfe der Seqan-Bibliothek implementiert und kann daher von effizienten Algorithmen und Datenstrukturen zur Sequenzanalyse optimal profitieren. Jedoch müssen einige Verbesserungen in Bezug auf Effizienz bezüglich des Speicherverbrauchs und der Laufzeit vorgenommen werden ehe die Applikation "Marktreif" wäre.