SeqDPT - Adapter-Entfernung und Qualitätskontrolle

Benjamin Strauch

Freie Universität Berlin
E-mail: *b.strauch@fu-berlin.de*

ZUSAMMENFASSUNG

SeqDPT ist eine Sammlung von Programmen zur Verarbeitung von NGS-Reads. Wir werden hier auf das Adapter- und Qualitäts-Trimming eingehen. Dies ermöglicht es, Adapter-Verunreinigungen und Basen niederer Qualität aus Sequenz-Reads zu entfernen.

1 EINFÜHRUNG

1.1 Adapter

Heutzutage werden viele DNA-Sequenzen über Next-Generation-Sequenzierung (NGS) gewonnen. Hierbei werden viele einzelne Stücke einer größeren Sequenz generiert und diese dann einzeln sequenziert. Eine bedeutende Technologie ist hierbei die der der Firma *Illumina*. Bei dieser Methode werden im Laufe der Sequenzierung die zu sequenzierenden Stücke (auch ,,Inserts" genannt) am 5'- und am 3'-Ende mit Adaptern ligiert, die gewisse technische Funktionen erfüllen. Angesichts dieser Anordnung sieht man leicht, dass abgelesene Reads, die am Sequenzanfang beginnen, über gegenüberliegende das Sequenzende hinaus in Adapter-Anteile laufen können.

1.2 Basenqualitäten

Ein weiteres Problem der modernen Sequenziertechnologien ist, dass die Sicherheit, mit der die Basen der DNA bestimmt werden kann, mit der Länge des Reads abnimmt. Dies wird als *Qualität* der Basen bezeichnet. Man ist für spätere Anwendungen daran interessiert, größere Regionen von Basen schlechter Qualität vom Ende der Reads abzutrennen. Auch dies leistet die Software.

2 METHODEN

Die Implementierung geschah mit der C++-Bibliothek ,,Seqan" (Döring *et al.*, 2008). Die Vorzüge dieser Sprache und Bibliothek sind, dass sie uns erlauben ein sehr schnelles Programm zu entwickeln, was für die Verarbeitung großer Datenmengen essentiell ist.

2.1 Adaptersuche durch Sequenz-Alignment

2.1.1 Single-end Reads. Die Beschreibung der Adapter-Verunreinigung suggeriert schnell eine naheliegende Methode zur Erkennung dieser Adapter. Wir machen uns nämlich zu nutze, dass ein Adapterpräfix, das am Ende eines Reads vorkommt, perfekt mit dem Anfang der Adaptersequenz übereinstimmen sollte. Man kann diese Übereinstimmung nun mit einem Overlap-Alignment, also einer Variante des klassischen Smith-Waterman-Algorithmus, bestimmen.

2.1.2 Paired-end Reads. Noch besser lassen sich Adapter in sogenannten *paired-end* Reads finden. Hierbei wurde der Insert von beiden Enden sequenziert, so dass die 5'-3' Sequenz eine Kontaminierung mit dem 3'-Adapter und die komplementäre Sequenz eine Kontaminierung mit dem reversen Komplement des 5'-Adapters enthalten könnte. Da die beiden Reads aber von einem Insert kommen gilt: Der Read ist *genau dann* verunreinigt, wenn die gepaarten Sequenzen sich überlappen. Man kann nun im gepaarten Fall in einem ersten Schritt eine Überlappung suchen. Auch dies geschieht wieder über eine *free-shift*-Variante des Smith-Waterman-Algorithmus, indem man die 1. Sequenz mit dem reversen Komplement der 2. Sequenz aligniert. Falls es ein signifikantes Alignment gibt, hat man die Überlappung gefunden, aus der man nun die originale Insert-Größe rekonstruieren kann. Wenn ein Read der Länge n größer als der Insert der Größe k ist, so müssen $n - k$ Basen Adapter-Verunreinigung sein.

Eine ähnliche Methode wurde erfolgreich in einer Veröffentlichung von Lindgreen (2012) beschrieben und wir konnten auch ebenso gute Ergebnisse für gepaarte Reads erzielen.

2.2 Qualitäts-Trimming

Das Entfernen der schlechten Basen vom Ende der Reads gehen wir standardmäßig mit einem fenster-basierten Ansatz an. Das bedeutet, dass wir ein Fenster gewisser Länge, voreingestellt 5, vom 5'- zum 3'-Ende des Reads laufen lassen. In dem Fenster wird jeweils die durchschnittliche Basenqualität ausgerechnet. Sobald das Fenster sich an einer Position befindet, bei dem diese Qualität das erste Mal eine angegebene Grenze unterschreitet, wird ab Beginn des Fensters abgeschnitten. Diese Methode ermöglicht eine etwas geglättete Erkennung der schlechten Basen und schneidet so beim Übergang von guten in schlechte Basen ab.

Die implementierte Methode ist auch im Programm *Trimmomatic* [1] implementiert. Die Güte bestätigt sich in Analysen mit FastQC. Wir haben auch noch eine Trimming-Methode, die simpel das längste Suffix schlechter Basen abschneidet und eine Methode, die vom bekannten Aligner *BWA* genutzt wird implementiert.

[1] http://www.usadellab.org/cms/index.php?page=trimmomatic

Tabelle 1: Vergleich der Programmfunktionen (S - Single-end, P - Paired-end)

(a) Güte und Geschwindigkeit des Adapter-Trimmings.

Programm	Sens.		Spez.		Laufzeit	
	S	P	S	P	S	P
Trimmomatic	54,3 %	57,4 %	99,7 %	91,2 %	116,4s	205,6s
AdapterRemoval	95,9 %	48,6 %	81,9 %	100 %	66,4s	285s
Flexbar	99,1 %	77,1 %	47,2 %	47 %	144,5s	265,9s
SeqDPT	89,5 %	98,8 %	68,65 %	99,9 %	6.25/37.7	11.9/47.8s

(b) Qualitäts-Trimming, für Phred-Grenze = 20

Programm	Laufzeit.		Qual		
	S	P	25 %	Mittel	75 %
Trimmomatic	4.93s	7.5s	29	33	36
AdapterRemoval	64s	293s	4	31,5	36
Flexbar	5s	14s	4	31,3	36
SeqDPT	4.86/0.15s	10.8/1.4s	29	33	36

3 ERGEBNIS

3.1 Analyse des Adapter-Trimmings

3.1.1 Gütemaße des Adapter-Trimmings Um die Güte des Adapter-Trimmings feststellen zu können, benötigten wir Wissen über die tatsächliche Adapter-Verunreinigung. Dafür haben wir Illumina-Reads mit dem Read-Simulator *SimSeq* [2] simuliert und hatten daher Informationen über kontaminierte Reads.

Für den Test wurden jeweils 100000 Sequenzen mit je einer von sechs üblichen Barcode-Sequenzen in den 3'-Adapter eingebettet simuliert. Die Barcodes wurden integriert um realistische Testbedingungen zu schaffen. Im vorgegebenen Adapter-Muster für die Programme wurden die Barcode-Basen als NNNNNN dargestellt.

Wir können nun die Güte eines Adapter-Trimmings bestimmen, indem wir die Originaldaten mit den getrimmten Daten vergleichen: Wie weit wurden die Reads jeweils getrimmt? Zu weit (falsch positiv), genau richtig (richtig positiv), zu wenig (falsch negativ) oder wurde korrekterweise gar nichts getrimmt (richtig negativ).

Mit dieser Methode konnten wir verschiedene Programme vergleichen und gewisse Kennwerte ausrechnen. Interessant für uns waren hier *Sensitivität* (wurde möglichst viel Adapter entfernt?) und *Spezifität* (wurde möglichst wenig *echte* Sequenz entfernt?).

3.1.2 Verglichene Software und Einstellungen Zum Vergleich haben wir die Programme *Trimmomatic*, *AdapterRemoval* und *Flexbar* (Dodt *et al.*, 2012) genutzt und versucht, bei diesen die Einstellungen so gut wie möglich unseren anzupassen, so dass ähnliche Testbedingungen herrschten. So haben wir natürlich immer die gleichen Adapterreferenzen genutzt und bei allen Programmen eingestellt, dass Reads möglichst nicht bei zu kurzer Länge entfernt werden sollen, damit die reine Adapter-Entfernung gut gemessen wird. Die Kennwerte sind in Tabelle 1a eingetragen, die Laufzeit von SeqDPT ist zur Anschaulichkeit in IO/Prozessierung unterteilt.

3.2 Qualitätsveränderung durch Basentrimming

Neben dem Adapter-Trimming können wir auch die Verbesserung der Qualität bestimmen. In Tabelle 1b sind die Laufzeiten und für die Qualitäten der letzten 5 Basen nach dem Schneiden die Werte für 25/75 %-Quantile und Mittelwert der Basenqualitäten notiert. Vorher betrugen sie in der Referenzdatei ca. 4, 30,4 und 36.

[2] https://github.com/jstjohn/SimSeq

4 DISKUSSION

Im Falle von single-end Reads ist die Spezifität eher mittelmäßig, was an unserer aggressiven Standardeinstellung liegt, auch bei kleinen Treffern potenzielle Adapter abzutrennen. Dies führt zu einer uns wichtigeren höheren Sensitivität. Wir haben diese Abwägung getroffen, da eventuell fälschlich entfernte Basen ohnehin am Ende des Reads liegen und damit eher von minderer Qualität sind.

Im Falle von paired-end Reads ist es uns nicht nur möglich, ohne vorheriges Wissen über Adaptersequenzen die Reads zu säubern, Spezifität und Sensitivität sind außerdem ausgezeichnet.

Beim Entfernen der Enden niedriger Qualität schneidet SeqDPT mit Trimmomatic am besten ab, da diese beide auf den Fenster-basierten Ansatz setzen. Flexbar und AdapterRemoval hingegen schneiden nur ein möglichst großes zusammenhängendes schlechtes Stück, wodurch sie Basen weiter im Read verpassen.

Darüber hinaus ist unser Programm, unter anderem aufgrund seiner Implementierung in C++, Nutzung der schnellen Seqan-Bibliothek und Multithreading beim Entfernen der Adapter wesentlich schneller als die Referenzprogramme. Es ist also möglich große Datenmengen zu verarbeiten, wie sie beispielsweise bei einer Analyse eines ganzen Genoms oder mehrerer Proben gleichzeitig anfallen.

5 SCHLUSSFOLGERUNG

Wir haben gesehen, dass sich das Programm verlässlich zur Entfernung von Adaptern eignet und auch die Entfernung schlechter Basen wie erwartet funktioniert. Gerade die, von neben SeqDPT nur von AdapterTrimming, implementierte Überlappung von gepaarten Reads kann die Adapter sehr gut erkennen.

LITERATUR

Dodt, M., Roehr, J. T., Ahmed, R., and Dieterich, C. (2012). Flexbar—flexible barcode and adapter processing for next-generation sequencing platforms. *Biology*, **1**(3), 895–905.

Döring, A., Weese, D., Rausch, T., and Reinert, K. (2008). SeqAn an efficient, generic C++ library for sequence analysis. *BMC Bioinformatics*, **9**, 11.

Lindgreen, S. (2012). AdapterRemoval: easy cleaning of next-generation sequencing reads. *BMC Res Notes*, **5**, 337.