Bioinformatics, 31(24), 2015, 4000–4002
doi: 10.1093/bioinformatics/btv501
Advance Access Publication Date: 26 August 2015
Applications Note

OXFORD

Sequence analysis

Bayexer: an accurate and fast Bayesian demultiplexer for Illumina sequences

Haisi Yi[1,2], Zhe Li[3,*], Tao Li[1,*] and Jindong Zhao[1,4]

[1]Key Laboratory of Algal Biology, Institute of Hydrobiology, Chinese Academy of Sciences, Wuhan, Hubei 430072, China, [2]University of Chinese Academy of Sciences, Beijing 100049, China, [3]State Key Laboratory of Systematic and Evolutionary Botany, Institute of Botany, Chinese Academy of Sciences, Beijing 100093, China and [4]College of Life Science, Peking University, Beijing 100871, China

*To whom correspondence should be addressed.
Associate Editor: John Hancock

Received on June 29, 2015; revised on August 14, 2015; accepted on August 19, 2015

Abstract

Summary: Demultiplexing is used after high-throughput sequencing to *in silico* assign reads to the samples of origin based on the sequenced reads of the indices. Existing demultiplexing tools based on the similarity between the read index and the reference index sequences may fail to provide satisfactory results on low-quality datasets. We developed Bayexer, a Bayesian demultiplexing algorithm for Illumina sequencers. Bayexer uses the information extracted directly from the contaminant sequences of the targeting reads as the training dataset for a naïve Bayes classifier to assign reads. According to our evaluation, Bayexer provides higher capability, accuracy and speed on various real datasets than other tools.

Availability and implementation: Bayexer is implemented in Perl and freely available at https://github.com/HaisiYi/Bayexer.

Contact: litao@ihb.ac.cn or lizhe@ibcas.ac.cn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Although the rapid improvement of next-generation sequencing is dramatically increasing data output and reducing sequencing costs, multiplexing, which allows large numbers of samples to be sequenced simultaneously during a single sequencing run, is also raising sample throughput. Commonly, multiplexing is achieved by adding different specifically designed short sequences, referred to as indices, to the sequencing adapters of each sample. The *in silico* process after sequencing to determine the origin sample of each read by comparing the sequenced indices (SIs) with the reference indices (RIs) is called demultiplexing.

Because sequencing errors occur, SIs do not always perfectly match the RIs. The demultiplexers (CASAVA, bcl2fastq) provided by Illumina allow for 0, 1 or 2 Hamming distances between the SIs and RIs. Splitaake (https://github.com/faircloth-lab/splitaake) and deindexer (https://github.com/ws6/deindexer) use Levenshtein distances to tolerate the IN-DEL (insertion and deletion) errors. Recently developed, deML (Renaud *et al.*, 2015) uses the quality scores of SI bases to assign each read to the origin sample with the maximum likelihood.

These approaches work well for high-quality reads but for low-quality reads the results may be unsatisfactory due to misassignment or assignment failure. It is widely reported that there are various types of inherent error profiles in Illumina sequencing data (Dohm *et al.*, 2008; Harismendy *et al.*, 2009; Hoffmann *et al.*, 2009; Kircher *et al.*, 2009; Nakamura *et al.*, 2011), which suggests that certain miscalls may be more likely than others, and an error-containing SI probably originated from one certain RI. Additionally, it is a ubiquitous phenomenon during Illumina sequencing that insert sequences shorter than the read length result in adaptor contamination on the ends of common reads and the index bases are often contained in the contaminant sequences. Therefore, demultiplexing could be seen as a classification problem with RIs identified from contaminant sequences as classes, and the corresponding SIs

in the separate index reads as observations. Based on this idea, we developed a novel application, Bayexer, which applies a naïve Bayes classifier (NBC) to the demultiplexing of Illumina sequences. Bayexer extracts training datasets directly from the contaminant sequences of targeting reads and then uses them to train the NBC and assigns the reads. The evaluation on real sequencing datasets showed the great advantage of Bayexer compared with other tools.

Bayexer is compatible with both single-index and dual-index sequencing and supports various Illumina sequencers.

2 Methods

Given a multiplexed dual-index paired-end sequence dataset, let S be all the observed SI pairs and R be all the RI pairs in it. Every s ($s \in S$), both reads having l bases, is iteratively split into overlapping k-mers, where $k = 2l, l, l-1, l-2, \ldots, 1$. For each s, the set of k-mers with size k is denoted by W_k, and the set of all k-mers is W. Let $c(r)$ be the observed number of r ($r \in R$), $c(w)$ be the observed count of k-mer w ($w \in W$) and $c(w|r)$ be the count of w originating from r. For each read cluster, Bayexer searches every pre-supplied RI-containing adapter-1 sequence in the first read and the RI-containing sequences of adapter-2 in the second read. If the adapter sequences containing r are detected in both reads and the SI pair is s, Bayexer takes this into the training set and adds one to $c(r)$, $c(w)$ and $c(w|r)$ for $\forall w \in W$. An example of this procedure is shown in Supplementary Figure S1 and Table S1. When the search is completed for all read clusters, any given s can be assigned by the NBC according to the maximum a posteriori decision rule as:

$$\hat{r} = argmax_r \left(P(r) \times \prod_{w \in W'} \frac{c(w|r) + \alpha_r}{c(r) + \alpha_r N} \right) \quad (1)$$

where N is the number of observed different k-mers in the current feature. To eliminate zeros and reduce the effect of uneven training sets, a modified smooth factor α_r (Supplementary Methods) is applied to the NBC. The priori probabilities $P(r)$ can be either supplied by users based on their prior knowledge (e.g. the DNA amount of each sample) or inferred from the input data itself by Bayexer (Supplementary Methods).

For every given s, we included a feature selection procedure to form $W' \subset W$, based on the following observations: (i) longer k-mers contain more complete information of the error profile and (ii) some k-mers in the dataset may have frequencies too low to provide reliable estimation. For a given s, Bayexer investigates all $w \in W_k$ from $k = 2l$ down to $k = 1$. When all the $w \in W_k$ have $c(w)$ greater than a pre-defined threshold value T, all the $w \in W_i$ with $i < k$ are abandoned and all the $w \in W_j$ with $j > k$ having $c(w)$ smaller than T are also abandoned, then all the remaining w form the W'.

We tested Bayexer and deML, which was reported to be the most robust demultiplexer available (Renaud *et al.*, 2015), on three real sequencing datasets. To evaluate the accuracy of our algorithm, we used the dual-index paired-end MiSeq dataset described in Renaud *et al.* (2015) that contains 99 libraries of polymerase chain reaction (PCR) products of a human genome region and a library of PhiX DNA fragments as a control. By mapping these reads to the reference genomes (Supplementary Methods), their samples of origin were identified and used to measure the misassignment rates of demultiplexing. To evaluate the robustness of Bayexer and deML, we used them to demultiplex a single-index GAIIx dataset that we produced, which had an average index quality score of 2.01. It consisted of a mate-pair library of *Microcystis aeruginosa* TAIHU98 whole-genome (NZ_ANKQ00000000.1) shotgun sequences and two libraries of metagenomic sequences (Supplementary Methods). We assembled the demultiplexed *M.aeruginosa* reads and investigated the statistics of the constructed unitigs and their similarity to the reference genome. To evaluate the applicability and performance of the two tools, we tested them on a 384-plex NextSeq500 run that included 153 078 446 read clusters (https://basespace.illumina.com/run/11358363).

3 Results

Among the total of 15 245 844 read clusters in the MiSeq dataset, we detected 9 635 948 clusters unambiguously aligned to the human genome and 4 933 839 to the PhiX (Supplementary Table S2). In the mapped reads, we found 435 177 different SIs. We counted the frequencies of every SI and computed the proportion of PhiX clusters in every one of them. Supplementary Figure S2 indicates that most of the SIs, especially those with greater frequencies, were much more likely to have originated from one certain RI than others, and thus could be safely assigned. We evaluated the accuracy of our estimation by comparing these proportions with the probability values estimated by Bayexer and found that 95.41% of the SIs containing 99.53% of the read clusters have estimation deviation smaller than 0.05. Figure 1 shows that the SIs with higher frequencies have more accurate estimation, whereas the estimation for low-frequency SIs may not be accurate enough, which suggested that the vast majority of the reads were correctly assigned by Bayexer and the misassignment rate could be controlled by excluding the low-frequency SIs. When assigning both 100% reads, Bayexer showed a total misassigned read number 22.67% smaller than deML. The advantage of Bayexer was more obvious for reads with lower quality scores, while on higher quality reads the two tools performed comparably (Supplementary Fig. S3 and Table S4). The misassignment rates on different total assignment rates are shown in Supplementary Figure S4 and Table S3.

The assembly of the *M.aeruginosa* reads from the GAIIx run demultiplexed by Bayexer had much less fragments, higher N50, more similar size and GC content to the reference genome, and higher sequence identity to the reference genome compared with deML (Supplementary Table S5). This suggests that even on an extremely low-quality dataset, Bayexer can still achieve a very high assignment rate while providing highly reliable demultiplexed reads.

The performance evaluation on all three datasets shows that Bayexer also required a much shorter runtime, especially on the huge GAIIx dataset and highly multiplexed NextSeq500 dataset, where it was five times faster than deML (Supplementary Table S6).

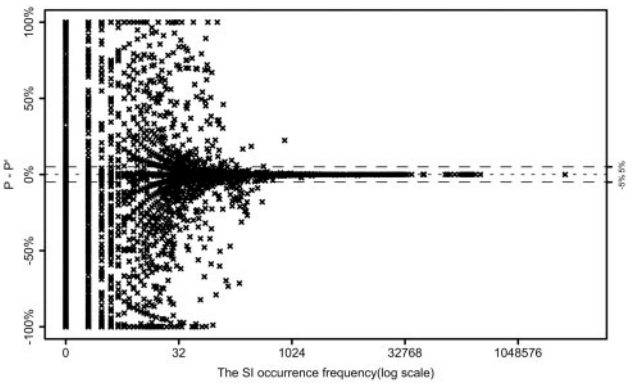

Fig. 1. *P-P'* with respect to the SI frequency. *P* is the true proportion of PhiX clusters in all the clusters of a certain SI, and *P'* is the estimated probability of this SI originating from PhiX

In summary, we present Bayexer, an Illumina sequence demultiplexer which outperforms other available tools on both accuracy and speed. In particular, the great advantage of accuracy on low-quality reads indicated that Bayexer can significantly improve the utilization ratio of low-quality sequencing data and thus benefit a wide range of high-throughput sequencing projects.

Acknowledgements

The authors thank Gabriel Renaud for his help with the use of deML.

Funding

Autonomous Projects of the State Key Laboratory of Freshwater Ecology and Biotechnology [2011FBZ31 and 2011FBZ32].

Conflict of Interest: none declared.

References

Dohm,J.C. *et al.* (2008) Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Res.*, **36**, e105.

Harismendy,O. *et al.* (2009) Evaluation of next generation sequencing platforms for population targeted sequencing studies. *Genome Biol.*, **10**, R32.

Hoffmann,S. *et al.* (2009) Fast mapping of short sequences with mismatches, insertions and deletions using index structures. *PLoS Comput. Biol.*, **5**, e1000502.

Kircher,M. *et al.* (2009) Improved base calling for the Illumina genome analyzer using machine learning strategies. *Genome Biol.*, **10**, R83.

Nakamura,K. *et al.* (2011) Sequence-specific error profile of Illumina sequencers. *Nucleic Acids Res.*, **39**, e90.

Renaud,G. *et al.* (2015) deML: robust demultiplexing of Illumina sequences using a likelihood-based approach. *Bioinformatics*, **31**, 770–772.