Bayesian decoding and confidence estimation for deep combinatorial sequence indexing with Pheniqs

Lior Galanti[1], Dennis Shasha[2] and Kristin C Gunsalus[1]*

Abstract

Background: Systems biology increasingly relies on deep sequencing with combinatorial index tags called barcodes to associate individual biological sequences with their respective sample, individual cell, and/or molecule. Accurate data interpretation critically depends on discerning the correct origin of each sequence of interest, and hence on the confidence of the final classification confidence. This, in turn, is cumulatively impacted by the decoding confidence of the various barcodes. Preserving a classification quality score is especially desirable when the decision to assign or discard a read depends on factors only available downstream. These may include the observed distribution of expected barcodes within a mixture of bulk sequencing samples, or the number of individual cells that can be reliably detected in single-cell sequencing experiments. Despite these advantages, computation and reporting of decoding confidence scores have not been widely adopted. To improve overall performance at this critical stage of sequence analysis, we have developed a flexible and robustly engineered software package, Pheniqs, that supports arbitrarily complex barcoding schemes and enables standardized reporting of decoded index sequences with classification confidence scores.

Results: Pheniqs implements both a standard minimum distance decoder and a probabilistic decoder that classifies sequence reads based on the full posterior probability of observed barcode indexes. Pheniqs consults basecalling quality scores and prior distributions to compute the posterior decoding error probability of individual barcodes and reports the resulting barcode sequences and confidence scores in standardized SAM auxiliary tags. Pheniqs can be extended with new error models and relies on intuitive confidence thresholds for fine-tuning decoding accuracy. Performance evaluation using synthetic data indicates that Pheniqs achieves greater accuracy than simple MDD or maximum likelihood estimation by correctly computing the full posterior classification probability. An optimized multi-threaded implementation assures that Pheniqs is both faster and scales better with large and complex barcode sets than existing tools.

Conclusions: Using full posterior probabilities to decode barcodes with Pheniqs is faster and more accurate than existing tools, especially those who depend on minimum distance. With the availability of an efficient implementation and the increased reliance on combinatorial barcode for classification we encourage the community to switch to more accurate probabilistic methods.

Keywords: demultiplex; barcode; confidence

Background

High-throughput sequencing with multiplexed sample barcodes is now standard practice, and single-cell applications are rapidly gaining in popularity. To enhance resolution and to control for different quantification biases, supplemental barcodes have been introduced to tag the individual cell and molecule from which sequence reads originate. Increasingly complex barcoding schemes are being devised to accommodate new experimental applications [1], and cellular indexing protocols may include several successive rounds of barcoding, increasing the potential combinatorial space exponentially [2]. Highly uneven barcode distributions and noise reads that do not belong to any of the classes are common, adding to the challenge of sequence classification.

*Correspondence: kcg1@nyu.edu
[1] Center for Genomics & System Biology, Department of Biology, New York University, 12 Waverly Place, 10003 New York, United States
Full list of author information is available at the end of the article

Existing barcode decoding tools were designed to handle reads that are multiplexed with a single, relatively small, set of sample indexes and thus are not well adapted to the rising use of novel experimental designs with custom multi-component barcoding schemes and associated complexities of the sequence data. Pheniqs overcomes many of the limitations of these first-generation decoders by rethinking key aspects of current decoding requirements: scalability, assumptions about the number of tags and their location, and the ability to explicitly account for base calling quality scores, prevalent foreign or noise sequences, and uneven barcode distributions.

Pheniqs (**PH**ilology **EN**coder w**I**th **Q**uality **S**tatistics, pronounced *phoenix*) combines a generic and extensible approach to barcode decoding with flexible configuration options that easily accommodate custom applications. A robust and efficient code base and extensive documentation support individual installations and large-scale production facilities alike. Pheniqs implements both a standard minimum distance decoder (MDD) and a probabilistic decoder that consults basecalling quality scores as well as priors to compute the full posterior decoding probability for classification rather than a simple maximum likelihood estimate.

Several existing tools for demultiplexing barcode sets are currently available. Illumina's bcl2fastq tool is geared specifically toward sample demultiplexing of Illumina data and relies on quality assessment procedures that are not obviously accessible to end-users. Standalone, peer reviewed, classifiers include deML [3], Bayexer [4], and Axe [5]. DeML is capable of handling a single barcode set with either one or two segments. While it consults base calling qualities and reports classification scores in Sequence Alignment/Map (SAM) [6] auxiliary tags, it does not accurately reflect the probability of correct classification. Instead, it estimates maximum likelihood only for the top candidates that are within a short Hamming distance, under restrictive assumptions that do not hold in most complex cellular indexing designs – specifically, that barcodes are uniformly distributed, and that foreign sequences (i.e. those that should not match any barcode) are extremely rare. Bayexer attempts to train a naive Bayes classifier by studying the error pattern when the insert sequence is shorter than the read length and results in a second observation of the barcode. This approach, although potentially more accurate in those cases, will fail when such redundancy is not present. Axe uses a set of pre-computed prefix trees to find a match within a given Hamming distance and can handle barcodes that differ in length. Although this method can be very fast, it ignores quality scores, and it is highly sensitive to upstream errors in the prefix and so sacrifices accuracy. None of the above tools account for the prior sample distribution or foreign sequence frequency, compute or report the posterior classification probability for individual reads, or handle multiple indexes from both sample and cellular tags. In addition, they all lack the ability to address read fragments in arbitrary offsets or to decode barcodes with more than two segments, and so often require custom preprocessing to reposition the barcode.

Implementation
Probabilistic decoding
Classification based on barcodes involves extracting a subsequence r from an observed read, along with the basecall quality scores associated with the individual nucleobases in r, and decoding the original sequence s. Let $r \in \{A, C, G, T, N\}^n$ be an observed sequence of length n extracted from the read and $\mathcal{B} \subseteq \{A, C, G, T\}^n$ a given set of distinct barcodes where each $b \in \mathcal{B}$ identifies an individual class. A decoder is denoted as a decision function $\phi : \{A, C, G, T, N\}^n \mapsto \mathcal{B} \cup \varepsilon$ where ε denotes a decoding failure for a foreign sequence, i.e., $s \notin \mathcal{B}$.

A maximum likelihood decoder will identify the barcode $\hat{b} \in \mathcal{B}$ which maximizes the posterior probability that $\hat{b}$ was sequenced given that r was observed, assuming it is not a foreign sequence.

$$\hat{b} = \arg\max_{b \in \mathcal{B}} P(b|r) \qquad (1)$$

Applying Bayes' rule we can compute $P(b|r)$ using

$$P(b|r) = \frac{P(r|b)P(b)}{P(r|b \notin \mathcal{B})P_\varepsilon + \sum_{b' \in \mathcal{B}} P(r|b')P(b')} \quad (2)$$

where P_ε is the prior probability of encountering foreign sequences and $P(r|b \notin \mathcal{B}) = 1/4^n$, as foreign sequences are assumed to produce random observations.

The *Phred-adjusted maximum likelihood decoder (PAMLD)* implemented by Pheniqs solves **Equation 2** by computing $P(r|b)$ for each $b \in \mathcal{B}$ from the base calling quality scores [7]. $P(b)$, the expected fraction of reads identified by b, can be either estimated from the data or provided *a priori* by the user.

When $P(r|\hat{b}) < 1/4^n$ the initial evidence supporting the classification is inferior to that provided by a random sequence, indicating the $\hat{b}$ recovered in **Equation 1** is more likely to be noise. The *noise filter* considers those a decoding failure without further consideration. In addition, the *confidence filter* declares a failure if $P(\hat{b}|r) \leq C$ where C is a user provided confidence threshold for the minimum acceptable probability of a correct decoding. The probability of a decoding error is

$$P_{\text{decoding\_error}}(\hat{b}, r) = 1 - P(\hat{b}|r) \qquad (3)$$

Therefore, directly estimating $P(\hat{b}|r)$ allows Pheniqs to report an intuitive classification confidence for every read, while the governing threshold C allows researchers to choose between assignment confidence and yield of classified reads.

By contrast, deML [3] assumes that P_ε is infinitesimally small and that samples are uniformly pooled, thereby suggesting that for every $b \in \mathcal{B}$

$$P(b) = \frac{1 - P_\varepsilon}{|\mathcal{B}|} \approx \frac{1}{|\mathcal{B}|} \tag{4}$$

Under such conditions $P(b|r) \propto P(r|b)$ and **Equation 1** can be simplified to

$$\hat{b} = \underset{b \in \mathcal{B}}{\arg \max}\, P(r|b) \tag{5}$$

While such assumptions simplify maximum likelihood estimation of $\hat{b}$, they are often grossly imprecise. A relatively low yield in (e.g. single-cell) experiments that rely on several layers of combinatorial indexing often results in an extremely uneven barcode distribution, with P_ε representing a significant portion of the sequenced DNA. Furthermore, implementations that refrain from computing $P(\hat{b}|r)$ do not report the posterior classification probability.

Software architecture

Pheniqs is distributed as a compiled binary with a command line interface that accepts a JSON encoded configuration file. Using a familiar syntax that mimics python array slicing Pheniqs can decode multiple barcodes located anywhere in the sequencing read. It uses tokens extracted from read segments by addressing either the 5' end, 3' end or both and optionally reverse complemented to construct the output template segments, as well as sample, cellular and molecular barcodes (Figure 1). This generic approach allows for arbitrary manipulation that accommodates any potential barcoding scheme and experimental design and often obviates the need for pre and post processing.

By directly interfacing with the low level HTSlib C API Pheniqs can read, write, and manipulate either uncompressed or gzip compressed FASTQ files as well as the SAM file format or any of its binary compressed variants BAM and CRAM. Unlike FASTQ, the SAM format can encode sequencing data in a single, smaller file that supports richer metadata annotations. Applications that consume and produce a single file can use POSIX standard streams to avoid the speed and storage bottlenecks associated with reading and writing temporary files from slow media.

Pheniqs reports the decoded sample, cellular and molecular barcodes as well as their corresponding quality scores and the posterior decoding error probability in SAM auxiliary tags. It supports declaring standard SAM read groups to be associated with the sample barcodes and can optionally perform an exhaustive quality assessment during processing that is included in the final report.

Results

Accuracy analysis with synthetic data

Since indel errors occur at a much lower rate than substitutions [8], we used a simple model that introduces an error on a nucleobase according to the quality score and substitution frequencies made available with LRSim[9]. We first paired each read from the run published with deML with a perfect barcode sequence sampled from the estimated prior distribution. To simulate foreign sequences we sampled from a random offset on the *PhiX174* genome. We then generated multiple files with position-independent substitution errors simulated on the barcode bases with quality scores calibrated to increasing error rates.

Each library was evaluated as a binary classifier, so a correct assignment was counted as a true positive (TP) while an incorrect assignment was counted as a false negative (FN) for the correct library and as a false positive (FP) for the incorrectly assigned library. We then summed up the values from all libraries and computed the *precision* $\left(\frac{TP}{FP+TP}\right)$, *recall* $\left(\frac{TP}{FN+TP}\right)$ and *F-score* (harmonic average of the precision and recall). Reads that were marked by either Pheniqs or deML as failing quality control were considered to be unclassified for this analysis.
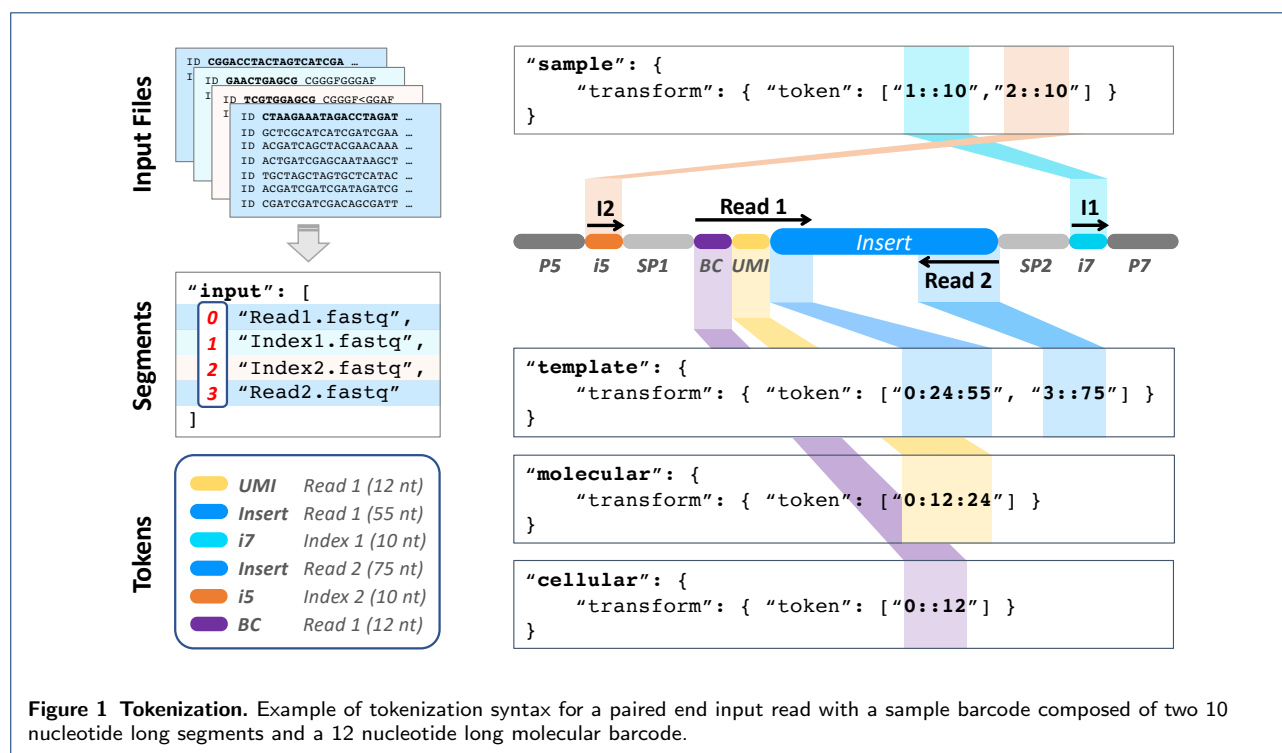
Prior estimation

To study the effect of accounting for priors on precision and recall, we estimated the noise prior P_ε and the individual library priors $P(b)$ for each $b \in \mathcal{B}$ from the high confidence reads. Let S_ε be the number of reads filtered by the *noise filter*, S_b the number of reads classified to b with confidence higher than C and $S_\mathcal{B} = \sum_{b \in \mathcal{B}} S_b$. An estimator for the noise prior is

$$\hat{P}_\varepsilon = \frac{S_\varepsilon}{S_\varepsilon + S_\mathcal{B}} \tag{6}$$

and for an individual barcode is

$$\hat{P(b)} = \frac{S_b}{S_\varepsilon + S_\mathcal{B}} \tag{7}$$

$\hat{P}_\varepsilon$ and $\hat{P(b)}$ assume that low confidence reads, i.e. reads that passed the *noise filter* but failed the *confidence filter*, come from the same distribution. When compared to the true known prior (Figure 1) for the

Figure 1 Tokenization. Example of tokenization syntax for a paired end input read with a sample barcode composed of two 10 nucleotide long segments and a 12 nucleotide long molecular barcode.

simulated data, the error in estimation was negligible for rates lower than 1 per 1000, the expected error rate on most Illumina platforms. Although the error in estimating $\hat{P}(b)$ remained low, the error in estimating $\hat{P}_\varepsilon$ grew higher with the error rate due to the difficulty in distinguishing noise from low confidence. This suggests that platforms with high error rates would benefit from prior estimation that accounts for this.

As expected PAMLD performs best with the true prior but only marginally worse with the estimated prior; it performs least well with a uniform prior, as is used by deML. In all 3 cases Pheniqs outperforms deML in both precision and recall (Figure 2).

Conclusion

Author details
[1] Center for Genomics & System Biology, Department of Biology, New York University, 12 Waverly Place, 10003 New York, United States. [2] Courant Institute, Department of Computer Science, New York University, 251 Mercer St, 10012 New York, United States.

References
1. Stuart, T., Satija, R.: Integrative single-cell analysis. Nature Reviews Genetics **20**(5), 257–272 (2019). doi:10.1093/bioinformatics/btp324

2. Cao, J., Packer, J.S., Ramani, V., Cusanovich, D.A., Huynh, C., Daza, R., Qiu, X., Lee, C., Furlan, S.N., Steemers, F.J., Adey, A., Waterston, R.H., Trapnell, C., Shendure, J.: Comprehensive single-cell transcriptional profiling of a multicellular organism. Science **357**(6352), 661–667 (2017). doi:10.1126/science.aam8940
3. Renaud, G., Stenzel, U., Maricic, T., Wiebe, V., Kelso, J.: deml: robust demultiplexing of illumina sequences using a likelihood-based approach. Bioinformatics **31**(5), 770–772 (2015). doi:10.1093/bioinformatics/btu719
4. Yi, H., Li, Z., Li, T., Zhao, J.: Bayexer: an accurate and fast bayesian demultiplexer for illumina sequences. Bioinformatics **31**(24), 4000–4002 (2015). doi:10.1093/bioinformatics/btv501
5. Murray, K.D., Borevitz, J.O.: Axe: rapid, competitive sequence read demultiplexing using a trie. Bioinformatics, 432 (2018). doi:10.1093/bioinformatics/bty432
6. Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., Subgroup, .G.P.D.P.: The sequence alignment/map format and samtools. Bioinformatics **25**(16), 2078–2079 (2009). doi:10.1093/bioinformatics/btp352
7. Edgar, R.C., Flyvbjerg, H.: Error filtering, pair assembly and error correction for next-generation sequencing reads. Bioinformatics **31**(21), 3476–3482 (2015). doi:10.1093/bioinformatics/btv401
8. Franziska Pfeiffer, C.G.: Systematic evaluation of error rates and causes in short samples in next-generation sequencing. Scientific Reports **8**(1), 10950 (2018)
9. Luo, R., Sedlazeck, F.J., Darby, C.A., Kelly, S.M., Schatz, M.C.: Lrsim: A linked-reads simulator generating insights for better genome partitioning. Computational and Structural Biotechnology Journal **15**, 478–484 (2017). doi:10.1016/j.csbj.2017.10.002
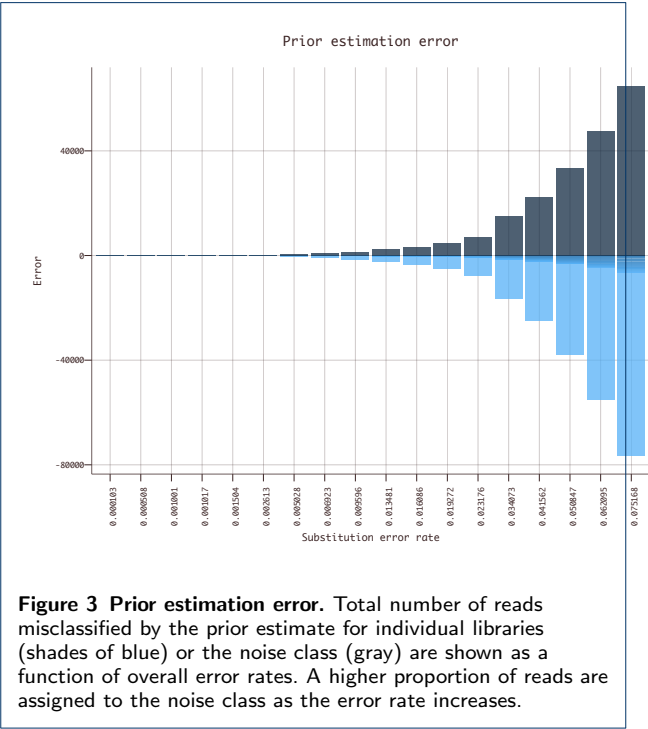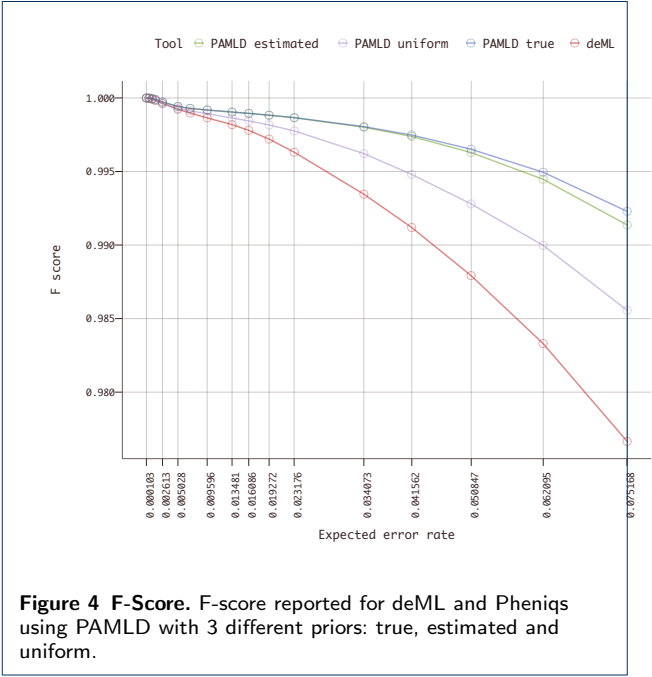
Figures
Tables
Additional Files
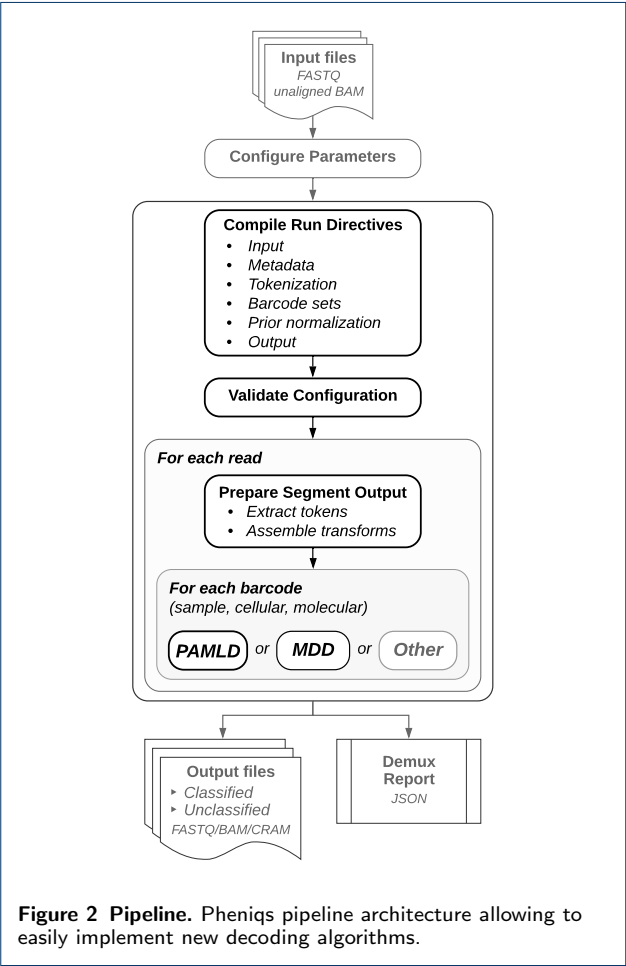
Figure 2 Pipeline. Pheniqs pipeline architecture allowing to easily implement new decoding algorithms.

Figure 4 F-Score. F-score reported for deML and Pheniqs using PAMLD with 3 different priors: true, estimated and uniform.

Figure 3 Prior estimation error. Total number of reads misclassified by the prior estimate for individual libraries (shades of blue) or the noise class (gray) are shown as a function of overall error rates. A higher proportion of reads are assigned to the noise class as the error rate increases.