Minerva: an alignment- and reference-free approach to deconvolve Linked-Reads for metagenomics

David C. Danko,[1,2] Dmitry Meleshko,[1,2] Daniela Bezdan,[2] Christopher Mason,[2,3] and Iman Hajirasouliha[2,4]

[1] *Tri-Institutional Computational Biology and Medicine Program, Weill Cornell Medicine of Cornell University, New York, New York 10065, USA;* [2] *Institute for Computational Biomedicine, Department of Physiology and Biophysics, Weill Cornell Medicine of Cornell University, New York, New York 10065, USA;* [3] *The Feil Family Brain and Mind Research Institute, Weill Cornell Medicine, New York, New York 10065, USA;* [4] *Englander Institute for Precision Medicine, The Meyer Cancer Center, Weill Cornell Medicine, New York, New York 10065, USA*

Emerging Linked-Read technologies (aka read cloud or barcoded short-reads) have revived interest in short-read technology as a viable approach to understand large-scale structures in genomes and metagenomes. Linked-Read technologies, such as the 10x Chromium system, use a microfluidic system and a specialized set of 3′ barcodes (aka UIDs) to tag short DNA reads sourced from the same long fragment of DNA; subsequently, the tagged reads are sequenced on standard short-read platforms. This approach results in interesting compromises. Each long fragment of DNA is only sparsely covered by reads, no information about the ordering of reads from the same fragment is preserved, and 3′ barcodes match reads from roughly 2–20 long fragments of DNA. However, compared to long-read technologies, the cost per base to sequence is far lower, far less input DNA is required, and the per base error rate is that of Illumina short-reads. In this paper, we formally describe a particular algorithmic issue common to Linked-Read technology: the deconvolution of reads with a single 3′ barcode into clusters that represent single long fragments of DNA. We introduce Minerva, a graph-based algorithm that approximately solves the barcode deconvolution problem for metagenomic data (where reference genomes may be incomplete or unavailable). Additionally, we develop two demonstrations where the deconvolution of barcoded reads improves downstream results, improving the specificity of taxonomic assignments and of *k*-mer-based clustering. To the best of our knowledge, we are the first to address the problem of barcode deconvolution in metagenomics.

[Supplemental material is available for this article.]

Recently, long-read sequencing technologies (e.g., Pacific Biosciences, Oxford Nanopore) have become commercially available. These techniques promise the ability to improve de novo assembly (Jain et al. 2018), particularly in metagenomics (Frank et al. 2016). While these technologies offer much longer reads than standard short-read sequencing, their base pair error rates are substantially higher than short reads (10%–15% error vs. 0.3%). More important, long-read technologies have substantially higher costs, lower throughput, and require large amounts of DNA, or PCR amplification, as input. Currently, this makes long reads impractical for large-scale screening of whole genome or metagenome samples and most low-input clinical samples.

As an alternative, low-cost and low-input (~1 ng) DNA library preparation techniques using microfluidic 3′ barcoding methods have recently emerged (e.g., Moleculo/Illumina, 10x Genomics) that address these shortcomings. With these new technologies, input DNA is sheared into long fragments of ~10–100 kbp. After shearing, a 3′ barcode is ligated to short reads from the fragments such that short reads from the same fragment share the same 3′ barcode (note that the 3′ barcode is unrelated to the standard 5′ barcode used for sample multiplexing). Finally, the short reads are sequenced using industry standard sequencing technologies (e.g., Illumina HiSeq). This process is commonly referred to as Linked-Read sequencing. Linked-Reads offer additional *long-range* information over standard short reads. We refer to the set of reads that share a 3′ barcode as a read cloud. For a more detailed explanation of the process of Linked-Read sequencing we refer the reader to Zheng et al. (2016).

Reads with matching 3′ barcodes are more likely to have emerged from the same fragment of DNA than two randomly sampled reads. However, each fragment of DNA is only fractionally covered by reads. This increases the amount of long-range information obtained from a given experiment but makes it impossible to assemble reads from a single barcode into a contiguous stretch of sequence. This trade-off has been used recently to phase large-scale somatic structural variations (Greer et al. 2017; Spies et al. 2017).

State-of-the-art Linked-Read sequencing systems use the same 3′ barcode to label reads from several fragments of DNA. Existing systems hone the order of 10^6 3′ barcodes; loading multiple fragments of DNA into the same 3′ barcodes is critical for high-throughput experiments. In particular, in our work using the 10x Genomics system, we observed that there were 2–20 long fragments of DNA per 3′ barcode (Supplemental Fig. S1) and that 3′ barcodes with more reads tended to have more fragments. This can complicate downstream applications. In the absence of other information, it is difficult to distinguish the random assortment of reads into a 3′ barcode from an actual structural variation or a different source genome.

Corresponding author: imh2003@med.cornell.edu

Article published online before print. Article, supplemental material, and publication date are at http://www.genome.org/cgi/doi/10.1101/gr.235499.118. Freely available online through the *Genome Research* Open Access option.

To address this critical issue, we define the barcode deconvolution problem. Briefly, each group of reads that share a 3′ barcode has an unobserved set of fragments from which each read was drawn. The barcode deconvolution problem is the problem of assigning each read with a given 3′ barcode to a group such that every read in the group came from the same fragment and so there is only one group per fragment. We note that a fragment assignment is stricter than genomic assignment. Each read from the same fragment necessarily came from the same genome, but it is possible to have multiple fragments from the same genome whose reads share the same 3′ barcode.

Linked-Reads provide significantly more information about the proximity of short reads than standard short-read sequencing. With the exception of very common or repetitive sequences, the co-occurrence of particular sequences across several 3′ barcodes provides evidence that the co-occurring sequences were drawn from the same underlying DNA molecules.

Linked-Reads have several potential benefits for metagenomic research compared to standard short reads. Linked-Reads carry information about long stretches of sequence. In principle, this information can be used to improve taxonomic classification of reads, improve the assembly of microbial genomes, identify horizontally transferred sequences, quantify the genetic structure of low-abundance organisms, and catalog intra-sample genetic structural variants. In the near term, algorithms for analyzing short-read sequences can be used on Linked-Read data without modification which makes Linked-Reads a practical choice for many studies.

Compared to long-read sequencing, Linked-Reads can be used to sequence samples far more deeply for the same amount of money and can accept much smaller amounts of input DNA. This is important for metagenomics; even at the same read depth, Linked-Reads may be more useful for studying low-abundance organisms because Linked-Reads span a much longer stretch of a genome for the number of bases sequenced (i.e., very high physical coverage) and could be used to resolve microbial structural variation.

In this paper, we address the barcode deconvolution problem, a fundamental problem of using Linked-Reads for metagenomics. We show that addressing the barcode deconvolution problem improves downstream results for two demonstration applications.

We formally define the barcode deconvolution problem for a single 3′ barcode. We note that our solution requires information from multiple 3′ barcodes but that this is not necessary to state the barcode deconvolution problem generally.

As input, we are given a set of n reads from the same read cloud. Each read has the same 3′ barcode and an unobserved class that represents the fragment from which the read was drawn. For a given 3′ barcode with n reads drawn from f fragments, we have $\vec{R} = \langle r_1, \ldots, r_n \rangle$ where r_i represents the unobserved class for read i, and $\vec{F}$, the set of possible fragment classes $[1, f]$.

A solution to the barcode deconvolution problem for a single read cloud would be a function mapping a set of read classes to fragment classes

$$D{:}\vec{R} \mapsto \vec{F}$$

such that the function produces the same value for reads from the same fragment for all n reads in $\vec{R}$

$$D(i) = D(j), \forall_j r_i = r_j, \forall_i \in 1{:}|\vec{R}|.$$

A solution to the barcode deconvolution problem for a set of read clouds would be a map from each read cloud to a function which solves the barcode deconvolution problem for that read cloud.

When a reference genome is available, the barcode deconvolution problem is relatively trivial, so long as major structural variants are absent. All individual reads from the same read cloud can be mapped to the reference genome using any good read alignment method. Reads from the same fragment will tend to be clustered near one another on the reference genome; there is little chance that reads from different fragments in the same read cloud will be proximal, unless the reference genome is very small. However, if a reference genome is not available, is small, or structural variation is present, read mapping may not provide a good solution to the barcode deconvolution problem. All of these conditions are common in metagenomics.

To the best of our knowledge, we are the first to formally describe the barcode deconvolution problem.

We have developed a novel method, Minerva, that explicitly uses information from sequence overlap between read clouds to approximately solve the barcode deconvolution problem for metagenomic samples. Our approach was inspired by topic modeling in Natural Language Processing (NLP) which studies methods to find groups of co-occurring words in text. We demonstrate how our technique can be effectively applied to real metagenomic Linked-Read data and improve analysis for two example use cases.

We present our solution to the barcode deconvolution problem in detail. We also develop a probabilistic generative model justifying key assumptions of our procedure. We also report our negative results—models that we tested but that performed poorly (Supplemental Materials).

Results

Algorithm overview

We have developed Minerva, an algorithm that approximately solves the barcode deconvolution problem for metagenomics. Minerva works by matching reads from the same read cloud that share *k*-mers with reads from other read clouds. This algorithm processes each read cloud individually by building a sparse graph between reads and other read clouds, converting that graph into a graph between reads, and clustering that graph. This method is discussed in detail.

Primary data sets

We tested Minerva using two primary real data sets from two microbial mock communities. The first community (Data set 1) contained five bacterial species: *Escherichia coli*, *Enterobacter cloacae*, *Micrococcus luteus*, *Pseudomonas antarctica*, and *Staphylococcus epidermidis*. The second community (Data set 2) contained eight bacterial species and two fungi: *Bacillus subtilis*, *Cryptococcus neoformans*, *Enterococcus faecalis*, *E. coli*, *Lactobacillus fermentum*, *Listeria monocytogenes*, *Pseudomonas aeruginosa*, *Saccharomyces cerevisiae*, *Salmonella enterica*, and *Staphylococcus aureus*. The relative abundance of each species in each data set is listed in Table 1.

We elected to use mock communities over simulated data in order to provide as realistic a data set as possible. All species in the mock communities had well-characterized genomes and make taxonomic assignment easy. The mock communities chosen are standard microbial positive controls as noted by Mason et al. (2017).

Roughly 1 ng of high molecular weight (HMW) DNA was extracted from each sample. The HMW DNA was processed using a

Table 1. **Taxa detail: Relative abundance is based on read counts and is not adjusted for genome size**

Taxa	Ref. genome size (Mb)	Rel. abund. Data set 1	Rel. abund. Data set 2
Escherichia coli	5.4	29.39	31.54
Enterobacter cloacae	5.7	31.37	n/a
Micrococcus luteus	2.5	12.19	n/a
Pseudomonas antarctica	6.7	11.48	n/a
Staphylococcus epidermidis	2.6	15.57	n/a
Bacillus subtilis	3.9	n/a	3.23
Lactobacillus fermentum	1.9	n/a	12.82
Listeria monocytogenes	3.0	n/a	3.64
Pseudomonas aeruginosa	6.8	n/a	14.70
Salmonella enterica	4.8	n/a	28.95
Staphylococcus aureus	2.9	n/a	1.50
Enterococcus faecalis	3.0	n/a	3.50
Cryptococcus neoformans	18.9	n/a	0.05
Saccharomyces cerevisiae	19.1	n/a	0.016

10x Chromium instrument, and we prepared sequencing libraries. Each library was sequenced on an Illumina HiSeq with 2 × 150 paired-end reads. Roughly 20 million reads were generated for each sample; for testing, we selected 10 million reads from each while ensuring that we only selected complete barcodes. Both samples showed some evidence of human contamination; reads that mapped to the human genome were not removed from the samples (but were not used to generate statistics on barcode purity) since some amount of human DNA is typical in metagenomic samples. In both samples, reads were distributed over 3×10^6 barcodes.

We used *Long Ranger BASIC* to attach barcodes to reads and perform error correction on barcodes (https://support.10xgenomics.com/genome-exome/software/pipelines/latest/advanced/other-pipelines). Both samples have a similar number of reads per barcode. Sample 2 had more species represented in each barcode, on average, though not necessarily more fragments, since fragments can originate from the same genome. Statistics about the data sets are summarized in Table 2.

We determined the actual fragment of origin for each read by mapping reads to the source genomes and clustering positions in case multiple fragments from the same genome were present in the same read cloud.

Runtime and performance

Minerva's runtime performance largely depends on two parameters: *K*, the size of the *k*-mers used to match reads; and anchor dropout, the minimum size of the read cloud being deconvolved. We list the total runtime and RAM usage for Minerva (Table 3) on both of our test data sets with different parameters. We note that our implementation of Minerva is single-threaded but that the algorithm itself is trivially parallelizable across 3′ barcodes.

Minerva approximately solves the barcode deconvolution problem

Minerva was able to identify subgroups in read clouds that largely corresponded to individual fragments of DNA. We term these subgroups 'enhanced read clouds.' We measured the quality of each enhanced read cloud using two metrics: Shannon entropy index $H = \sum p_i \log p_i$, and purity $P = \max(\vec{p})$, where p_i indicates the proportion of an enhanced read cloud that belongs to each fragment. These values are shown in Figure 1 as compared to read clouds which were not enhanced ('standard' read clouds). In general, Minerva produced a large number of perfect ($P = 1$, $H = 0$) enhanced read clouds.

We also tested whether the quality of the enhanced read clouds changed with the number of reads in the read cloud. We found a small inverse relationship between read cloud size and purity but established that our previous results were not being inflated by a large number of very small enhanced read clouds (Supplemental Fig. S2). Note that enhanced read clouds of size 1 would be trivially perfect and are always excluded from results.

In testing, we found three parameters that seemed to have the most effect on Minerva's performance. The number of links required between reads to form a cluster (*eps*), the *k*-mer size used to make minimizing *k*-mers (*K*), and the maximum allowed frequency of each read (*maxk*). In Supplemental Figure S3, we show how these parameters affect Minerva's performance under three different metrics: mean enhanced barcode purity, mean enhanced barcode size, and total reads clustered, large, pure, and complete clusters being the ideal. We found that Minerva's parameters could be used to tune performance between very large and very pure enhanced barcodes depending on the downstream application.

Enhanced read clouds can be clustered into meaningful groups

After deconvolving barcodes into enhanced read clouds, it is useful to group enhanced read clouds that likely came from the same genome. This is essentially a clustering problem. Initially, we explored graph-based approaches similar to our algorithm for read cloud deconvolution. These algorithms relied on the assumption that elements being clustered would have small numbers of distinguishing elements and a relatively high a priori probability of originating from the same cluster. When dealing with individual barcodes, these assumptions proved reasonable; faced with the complexity of a full data set, these assumptions became inaccurate, and graph-based algorithms performed poorly.

With relatively little structure in the data that could be known a priori, we turned to topic modeling algorithms to discover implicit genetic structures in our data. Latent Dirichlet allocation (LDA) is a classic model in Natural Language Processing (Blei et al. 2003). LDA is a generative model that assumes data was created using a certain well-defined, stochastic process. Training the model consists of finding parameters that make it more likely that the observed data would be generated using the given stochastic process; typically, this is done with Gibbs sampling.

Typically, LDA is used to analyze corpora of natural language. Natural language corpora are organized into documents (e.g., emails or book chapters) that consist of words. The base version of LDA does not consider what order words in a document occur, just how often each word occurs in a given document; this is referred to as a bag-of-words model. Formally, documents are modeled as a sparse vector over a large vocabulary of words where

Table 2. **Data set properties**

	Data set 1	Data set 2
Number of read pairs	10^7	10^7
Number of species	5	10
Mean read cloud richness	2.74	5.79
Mean read cloud size	7.399	7.515
Barcode N50 size	11	11
Barcode N90 size	4	4

Table 3. Runtime performance

Data set	*K*	Anchor dropout	Runtime (hr)	RAM (GB)
Data set 1	20	50	1.48	93
Data set 1	30	50	1.71	163
Data set 1	20	30	12.84	92
Data set 1	30	30	15.5	163
Data set 2	20	50	3.27	115
Data set 2	30	50	3.66	200
Data set 2	20	30	36.69	115
Data set 2	30	30	40	200

entries represent the number of times a word occurs in the document. LDA maps document from a high dimensional word-space to a lower dimensional topic-space. In NLP, topics typically have intuitive interpretations as thematically consistent units. A key advantage of LDA is that it can distinguish synonyms based on context (i.e., a river bank vs. a financial bank); this may be useful for classifying conserved motifs.

We used LDA to cluster read clouds (represented as sets of *k*-mers). Each topic generated by LDA was considered to be a single cluster.

We used LDA to project enhanced and standard read clouds into a lower dimensional space. We treated each read cloud as a document containing minimum sparse *k*-mers as words. We removed *k*-mers that occurred far more often than average in a process similar to removing stop-words in NLP. We ran LDA with hyperparameter optimization on our read cloud documents and clustered to obtain a topic vector for each read cloud using the implementation LDA in MALLET (http://mallet.cs.umass.edu). Using X-Means, we clustered the topic vectors representing read clouds into discrete groups.

With standard read clouds, LDA essentially cannot distinguish any structure; with enhanced read clouds, LDA can generate clusters that are less diverse. The clusterings are compared in Figure 2. This could be used to improve assemblies by clustering similar reads and reducing spurious connections. Note that we denote chromosomes rather than genomes in the figure since our process does not attempt to link chromosomes from the same organisms.

Enhanced read clouds improve short-read taxonomic assignment

We observed that reads from a single linked fragment could be classified using any short-read taxonomic classifier. These classifiers, however, often have trade-offs between recall and precision. Enhanced read clouds can be used to improve recall of a short-read classifier without harming precision.

Many of the reads classified by short-read taxonomic classifiers cannot be assigned to low taxonomic ranks. However, all reads from the same fragment of DNA must all have the same taxonomic rank. Read clouds can be used to promote unspecific taxonomic assignments. Any read with a taxonomic rank that is an antecedent of a lower taxonomic rank in the same read cloud can be promoted to the lower rank, provided there are no conflicts with other ranks in the same cloud. Enhanced read clouds reduce the risk of conflicting ranks and make it more likely that reads can be promoted.

We used Minerva to improve the specificity of short-read taxonomic assignments obtained from Kraken, a popular pseudo-alignment-based tool (Wood and Salzberg 2014). We selected Kraken because it was found to have good precision but relatively poor recall in a study by McIntyre et al. (2017).

Using the technique described above, we were able to rescue a large number of reads from unspecific taxonomic assignments. We rescued reads using both enhanced read clouds and standard read clouds. In every case, rescue with enhanced read clouds matched or outperformed rescue with standard read clouds. All cases where rescue with enhanced read clouds outperformed standard read clouds for Data set 1 are shown in Table 4. All observed taxonomic assignments were correct after promotion. Without enhanced barcodes, many annotations cannot be rescued or are incorrect.

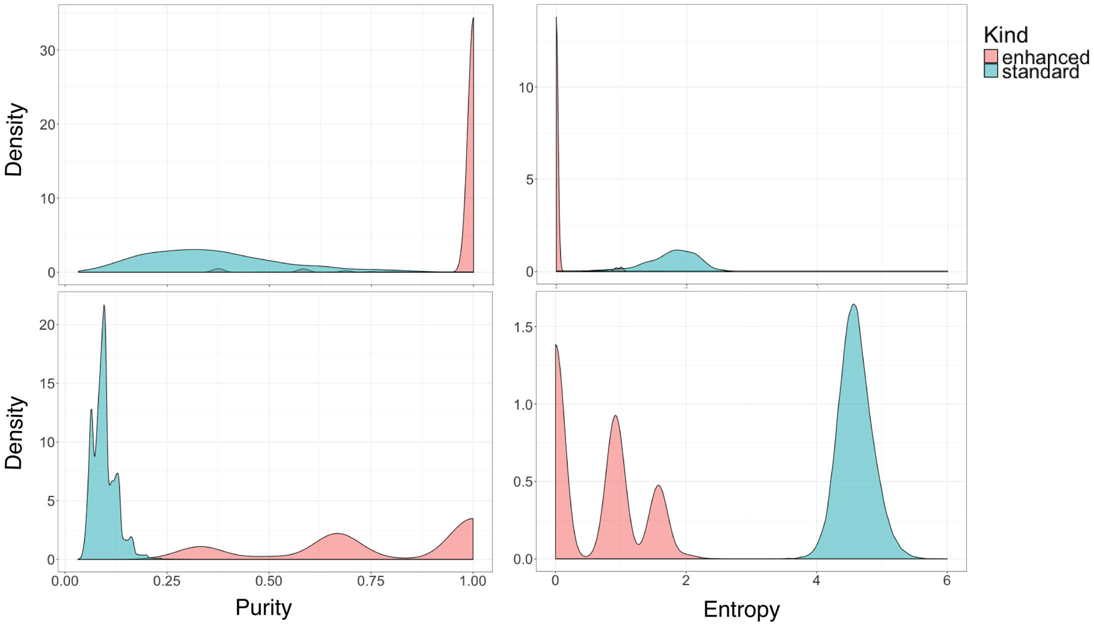

Figure 1. Clockwise from *top, left*: (1) Purity in Data set 1 for enhanced and 3′ barcodes; (2) Shannon index in Data set 1 for enhanced and 3′ barcodes; (3) Shannon index in Data set 2 for enhanced and 3′ barcodes; (4) purity in Data set 2 for enhanced and 3′ barcodes.

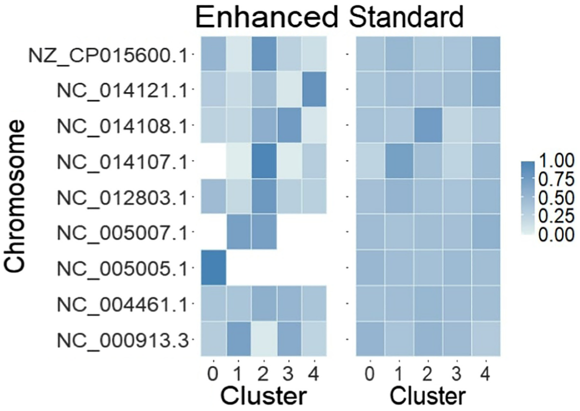

Figure 2. Abundance of different chromosomes across clusters as assigned by Latent Dirichlet allocation (LDA). Enhanced read clouds dramatically improve LDA's ability to distinguish structure in Data set 1. This figure uses the same deconvolution as Figure 1.

Discussion

We have introduced Minerva, a graph-based algorithm, to provide a solution to the barcode deconvolution problem. By design, Minerva provides conservative solutions to barcode deconvolution for metagenomics and uses essentially no information (except *k*-mer overlaps) about the sequences being clustered. We note that it will be beneficial to test Minerva on more complex communities. As such, Minerva is a relatively pure demonstration of how information can be extracted from Linked-Reads. With some modification, the algorithms underlying Minerva may even be useful for detecting structural variations and other genetic structures in the human genome.

However, the current version of Minerva could be enhanced by leveraging a number of practical sequence features, such as known taxonomic assignment, GC content, tetramer frequency, or motifs. These have been shown to be good indicators of lineage in metagenomics and could be easily incorporated to improve Minerva's clusterings. In particular, taxonomic assignments could be incorporated into Minerva to evaluate barcode deconvolution, since there is no a priori reason to think reads with a known taxonomic classification would be deconvolved more effectively than reads that could not be classified.

The current version of Minerva provides reasonable performance but still represents a potential bottleneck for workflows using Linked-Reads. A large performance issue is Minerva's routine to calculate the size of an intersection between two sets that is naïve and exact. Jain et al. (2018) has shown that bloom filters can be effectively used to speed up the calculation of set intersection in biology with acceptable errors. Future versions of Minerva could employ similar techniques to improve performance. Minerva uses the same parameters to process every barcode; however, the nature of Linked-Read sequencing provides a rich source of information that could be used to optimize model parameters for deconvolving individual barcodes. This would require a more thorough mathematical model of Linked–Reads, which we leave to a future work. Similarly, external sequence annotation could be incorporated as a practical approach to setting parameters for individual barcodes, though it is unlikely that such a technique would generalize to nonmicrobial applications.

Of particular interest to us is the possibility of using Minerva to directly improve downstream applications. For simple applications, Minerva may be used with a single set of parameters to produce a deconvolution that meets certain requirements. For applications built to take advantage of barcode deconvolution, Minerva could be run with multiple parameters to produce increasingly strict tiers of enhancement. This may be particularly important for de Bruijn graph (DBG) assembly. DBG assembly typically relies on effectively trimming and finding paths through a de Bruijn graph. Multiple tiers of linkage between reads could be used to inform trimming or pathfinding programs about likely paths and spurious connections. This could likely be modeled either as an information theory or probabilistic approach depending on the situation and assembler.

Overall, we believe that Minerva is an important step toward building techniques designed to take advantage of Linked-Reads. Linked-Reads have the potential to dramatically improve detection of large genetic structures without dramatically increasing sequencing costs, while taking advantage of existing techniques to process short reads.

Methods

We have developed a graph-based algorithm to subdivide reads from the same read cloud into groups that, ideally, solve the barcode deconvolution problem.

Table 4. Taxonomic promotion

Original rank	Promoted rank	Enhanced	Standard	Difference	Ratio
Bacteria	*Enterobacter cloacae*	3	2	1	1.5
Proteobacteria	*Enterobacter cloacae*	24	17	7	1.41
Gammaproteobacteria	*Enterobacter cloacae*	21	13	8	1.62
Enterobacterales	*Enterobacter cloacae*	87	72	15	1.21
Enterobacteriaceae	*Enterobacter cloacae*	765	642	123	1.19
Bacteria	*Escherichia coli*	9	6	3	1.5
Proteobacteria	*Escherichia coli*	8	7	1	1.14
Enterobacterales	*Escherichia coli*	17	13	4	1.31
Enterobacteriaceae	*Escherichia coli*	9221	7846	1375	1.18
Escherichia	*Escherichia coli*	201	198	3	1.02
Gammaproteobacteria	*Pseudomonas antarctica*	3	2	1	1.5
Pseudomonas	*Pseudomonas antarctica*	256	200	56	1.28

The number of reads which could be promoted using standard or enhanced read clouds in a deconvolution of Data set 1. This figure uses the same deconvolution as Figure 1. Cases where enhanced read clouds did not outperform standard read clouds are omitted; there are no cases where standard outperformed enhanced.

The core intuition behind our approach is that reads from the same read cloud from the same fragment will tend to overlap with similar sets of reads from other read clouds. Critically, if the total genome length in a sample is large enough, a pair of read clouds is unlikely to contain reads from more than one overlapping genomic region. In what follows, we justify this statement.

Mathematical justification of the model

We have developed a simple model to justify our statement that reads from the same fragment will overlap with reads from similar sets of read clouds. This model is similar to empirical results and can be used to inform the parameters used for deconvolution.

First, we develop a model for drawing fragments of DNA from genomes in a metagenomic sample. For simplicity, we model each microbial genome G_i in a metagenome G as a discrete collection of exactly N_g fragments $\overrightarrow{F_i}$, where i is an index numbering each genome in the metagenome. N_g is the same for all genomes. The probability of selecting a given fragment $F_{i,j}$ (where individual fragments are indexed by j) from a given microbial genome G_i is given by a uniform distribution. We model the probability of selecting a given genome as a geometric distribution; this choice is motivated by observations of real microbial communities that tend to be dominated by 1–2 species with a long tail of lower abundance species.

The probability of selecting a particular fragment $F_{i,j}$ given that we are drawing fragments from genome G_i is

$$P(F = F_{i,j}|G_i) = \frac{1}{N_g} \forall i \in 1{:}|G|.$$

For simplicity, we assume the abundance of genomes G_1, G_2, ... is sorted in descending order by their index. The probability of selecting a given genome G_i is

$$P(G = G_i) = \frac{1}{2^i}.$$

This gives us the probability of drawing a single given fragment $F_{i,j}$ without a given genome.

$$P(F_{i,j}) = \frac{1}{N_g \times 2^i}.$$

The probability that two fragments $F_{w,x}$, $F_{y,z}$ are the same given that their genomes G_w, G_y are the same is

$$P(F_{w,x} = F_{y,z}|G_w = G_y) = \frac{1}{N_g}.$$

The probability that two genomes G_w, G_y are the same is given below. In real communities, this is an approximation that improves as the total number of species increases.

$$P(G_w = G_y) = \lim_{|G|\to\infty} \sum_{i=1}^{|G|} \frac{1}{2^{2i}} = \frac{1}{2}.$$

Let p_f be the probability that two fragments $F_{w,x}$, $F_{y,z}$ are the same without conditioning on a given genome. We have

$$p_f = P(F_{w,x} = F_{y,z}) = \frac{1}{2N_g}.$$

Second, we develop a generative model for assembling a read cloud from a set of fragments. We model each read cloud as a selection of N_f fragments drawn from the set of all possible fragments. We refer to the set of fragments in a given read cloud as R_i. For simplicity, we do handle the case where two read clouds both contain multiple fragments from the same class; this case is very unlikely with parameters relevant to our scenario (1 in 25,000 with the parameters given below).

Let $X(k)$ be the probability that two read clouds R_i and R_j, both with N_f fragments, share exactly k fragments. In other words, any fragment in R_i overlaps with at least one fragment in R_j and vice versa.

We have

$$X(0) = P(|R_i \cap R_j| = 0) = (1 - p_f)^{N_f^2}.$$

This is simply because none of the N_f^2 possible pairs of fragments (i.e., one in R_i and one in R_j) overlap.

We also have

$$\begin{aligned} X(1) &= P(|R_i \cap R_j| = 1) \\ &= 2N_f(1 - (1 - p_f)^{N_f})(1 - p_f)^{N_f(N_f-1)} - N_f^2 p_f (1 - p_f)^{N_f^2-1}. \end{aligned}$$

Here, exactly one fragment in R_i overlaps with one or more fragments in R_j or vice versa. While it is extremely unlikely that we observe overlap of a fragment in R_i with more than one fragment in R_j, we handle this case in our equations because this is allowed in our approximate generative model. We have $2N_f$ possibilities to select a fragment in either R_i or R_j. This fragment must overlap with at least one fragment in the other read cloud (i.e., the term $1 - (1 - p_f)^{N_f}$). No other pair of fragments must overlap (i.e., the term $(1 - p_f)^{N_f \cdot (N_f-1)}$) and because we double-counted cases where exactly one fragment in R_i overlaps with exactly one in R_j we subtracted the term $N_f^2 p_f (1 - p_f)^{N_f^2-1}$.

The probability that two read clouds share more than one fragment is

$$X(\geq 2) = P(|R_i \cap R_j| > 1) = 1 - X(0) - X(1).$$

We choose reasonable, conservative (compared to our observed data) values for all parameters $N_f = 5$, $N_g = 100$, $|G| = 10$ and obtain the following estimates

$$p_f = \frac{1}{200},$$

$$X(0) \approx 0.8822,\ X(1) \approx 0.113,\ X(\geq 2) \approx 0.0048,$$

$$\frac{X(1)}{X(\geq 2)} > 23.$$

We find that it is about 23 times more likely to have exactly one overlapping fragment between two read clouds than multiple overlapping fragments in our mathematical model. We verified this through simulation and obtained a similar ratio of 1 to 40 (the discrepancy occurs because of how our simulation samples the geometric distribution). This is true even with conservative parameters chosen to minimize the ratio. This is important because it means we are likely to avoid a large number of spurious connections between genomic regions that could lead to poor deconvolution. However, this model does not account for the fact that individual fragments may have similar sequences, which is a major source of noise for Minerva. To reduce this noise, we use the parameters of this model to justify removing any overlaps that occur far more often than expected.

On average, each fragment in a data set is only fractionally covered at a rate of C_r (with a read depth of 1). While the precise coverage might vary between fragments, this parameter can be used to estimate the size of overlaps between fragments and their expected sparsity. Two long fragments would be expected to overlap at C_r^2 points in their overlap. In cases where fragments overlap much more frequently than C_r^2 over their lengths, it can be inferred that the sequence present is too repetitive or common to be useful for deconvolution.

These facts are used in Minerva to filter connections between repetitive regions, restrict overlaps to regions of a certain length, and to heuristically filter comparisons between read clouds unlikely to have significant overlap. This carries practical performance benefits and reduces errors.

A graph-based algorithm for barcode deconvolution

Summary

We have developed a graph-based algorithm that effectively deconvolves the reads within a given read cloud. The model constructs a bipartite graph between all reads with a given read cloud and all other read clouds. Reads have an edge to a read cloud if they are found to contain a *k*-mer that is specific to exactly one read in the foreign read cloud. Once the bipartite graph is constructed, read clouds and reads with too many or too few edges (by user-supplied parameters) are removed. The filtered bipartite graph is used to construct an adjacency matrix between reads, and the matrix is clustered into groups of reads. This algorithm is $O(n^2)$ over the number of read clouds, though we note that the number of read clouds is a constant for each specific technology that could be used.

The specific steps in our algorithm are as follows:

1. Read clouds are parsed; read clouds below a certain size (dropout) are dropped. Each read in each read cloud is parsed into a set of minimum sparse *k*-mers.
2. Each read cloud above a certain size (anchor dropout) is compared to all other read clouds. The read cloud being compared is called the ‘anchor.’
3. A bipartite graph is constructed between the reads in the anchor and all other read clouds based on *k*-mer overlap.
4. The bipartite graph is reduced to a graph between reads in the anchor.
5. The read graph is broken into discrete clusters which are output as solutions to the barcode deconvolution problem.

The model

Initially, each read cloud in a given data set is parsed into a set of minimizing *k*-mers (Fig. 3, part 1). Global counts for *k*-mers are retained. Once parsing is complete, *k*-mers that occur exactly once or many times more than the average (10 times more, by default) are discarded. Singleton *k*-mers cannot occur in more than one barcode and *k*-mers that are too common tend to create false positives (these *k*-mers appear to originate from low complexity or conserved regions). This process is analogous to removing stop words in Natural Language Processing applications. A map of *k*-mers to reads is retained for each read cloud.

After parsing, the set of reads in a given read cloud is compared to every other read cloud (Fig. 3, part 2). Comparisons between read clouds that share too many *k*-mers are discarded as these likely represent low complexity or evolutionarily conserved regions as opposed to real overlaps. Comparisons between read clouds that share too few *k*-mers are also rejected to improve performance. The intersection of the *k*-mer sets between the given read clouds and all read clouds that passed filtering is calculated.

A bipartite graph is constructed by creating nodes for every read in the read cloud being processed and every read cloud that was not filtered out (Fig. 3, part 3). Edges are only added between read-nodes (left nodes) and read cloud-nodes (right nodes). An edge is drawn between a read-node and a read cloud-node if, and only if, the read shares a *k*-mer with the given read cloud. This is a fast proxy measure for read overlap. Finally, any read cloud-node with degree above a given threshold is discarded.

Each bipartite graph representing the reads from a given read cloud is given a final round of filtering where reads that matched too many foreign read clouds are removed based on a user-supplied threshold. The filtered bipartite graph is converted to an adjacency matrix of reads where the similarity between reads is equivalent to the number of read clouds with which both reads overlapped (Fig. 3, part 4). This adjacency matrix is converted to a binary matrix by setting all values below a user-supplied threshold to zero and all remaining values to one (Fig. 3, part 5).

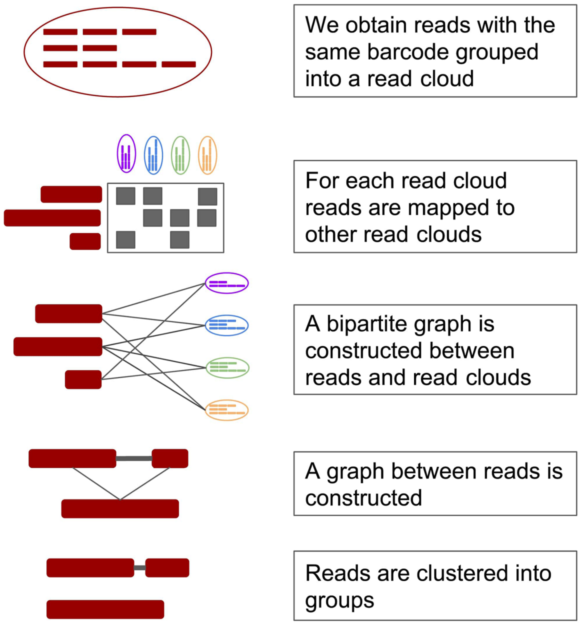
We obtain reads with the same barcode grouped into a read cloud
For each read cloud reads are mapped to other read clouds
A bipartite graph is constructed between reads and read clouds
A graph between reads is constructed
Reads are clustered into groups

Figure 3. Processing steps for a single read cloud. From *top*: (1) Fragments are sequenced and tagged with 3′ barcodes. (2) Reads in a given read cloud are mapped to reads in other read clouds using minimizing *k*-mers. (3) A bipartite graph between reads and other read clouds is constructed. (4) A graph between reads that map to the same read clouds is constructed. (5) Reads are clustered into groups.

All connected components in the binary matrix are found. Connected components consisting of single reads are discarded; the remaining components define clusters. This process is analogous to DBSCAN (Ester et al. 1996) for graphs.

Information Theory bounds on barcode deconvolution

We note that the barcode deconvolution problem on the graph-based model we have described is analogous to the community recovery problem (Girvan and Newman 2002) in Information Theory. In particular, 3′ barcodes provide linkage information between pairs of reads. We use this linkage information to construct a graph between the reads being deconvolved with the expectation that reads from the same fragment will have a better chance of being linked than reads from different fragments. Formally, we say that two reads are connected with probability p if they are from the same fragment and probability q if they are from different fragments. Termed differently, p is the true positive rate while q is the false positive rate.

For clarity, we note that this model is distinct from the model we developed previously to justify why overlaps between read clouds were likely to be useful for deconvolution.

If we make a simplifying assumption that all fragments in our read cloud produce equal numbers of reads, we can use the formula determined by Hajek et al. (2016) to determine the minimum connectivity of linking 3′ barcodes necessary to deconvolve our reads. We define the number of reads per fragment as N_r/N_f, where N_r is the total number of reads in a read cloud and N_f is the number of fragments in the given read cloud.

For the community recovery problem, Hajek et al. (2016) have provided a lower bound on the size of graph that can be accurately clustered given values of p and q, regardless of the algorithm used. If a graph is smaller than this threshold, it is unlikely that it will be possible to distinguish clusters from spurious edges. This boundary requires us to assume that all fragments with the same 3′ barcode produced equal numbers of reads. Using the definitions above this definition, we can apply the following inequality to read cloud deconvolution:

$$\frac{N_r}{\log N_r}(\sqrt{p}-\sqrt{q})^2 > N_f.$$

Using the model developed previously and a simulation, we estimate the maximum true positive rate p to be 0.998 and we estimate the minimum false positive rate q to be $p/15 = 0.067$. We note that these values do not account for multiple sources of error, notably sequence homology, and should be interpreted as a best case scenario. Using these values, we can reduce the previous equation

$$0.549\frac{N_r}{\log N_r} > N_f.$$

If a barcode deconvolution graph does not meet this inequality, it is unlikely that it will be possible to accurately reconstruct all clusters. More generally, this formula can be used to estimate the minimum number of reads and maximum error rates that can lead to effective barcode deconvolutions. In principle, this inequality should apply to all barcode deconvolution algorithms that can be formulated as a graph. However, different algorithms may have different values of p and q. We also note that the above formula is based on asymptotic behavior for graphs with thousands of nodes. We observed that typical deconvolution graphs in our model have fewer than 50 nodes.

Minimum sparse hashing

Minerva frequently tests whether pairs of reads overlap. Many solutions to finding overlaps between reads exist, such as sequence clustering algorithms, sequence aligners, and k-mer matching. These techniques typically make trade-offs between overall performance and error rates. Since Minerva is meant to be relatively fast and can tolerate some errors, we elected to use a minimal sparse hash of k-mers to match read pairs. This technique reduces the number of unique k-mers Minerva uses to find overlaps, which reduces runtime and RAM usage.

Minimum sparse hashing was originally developed independently for biological sequence search and Natural Language document search (Schleimer et al. 2003; Marçais et al. 2017) (in a Natural Language search, the technique is referred to as winnowing). While the original application of this technique in biology defined minimization as the lexicographic minimum of a set of sequences, we use a uniform random hash function to determine the minimal sequence in a set. This is a common practical enhancement recently detailed by Orenstein et al. (2016).

Minimum sparse hashing for sequences takes three parameters—a length k, a window size w, and a hash function h. Given a set K of n, $n \geq w$ k-mers, the min-sparse hash computes the hash h of each k-mer, then selects the k-mer with the smallest numerical hash from each consecutive set of w k-mers in K. The final set of minimizers is the unique set of k-mers generated, W. Each consecutive window shares $w-1$ k-mers, so there is a good chance that each window shares the same minimum with its predecessor. Formally, $W = \{\min(h(k)\forall k \in K_{(i,i+w)})\forall i \in 0{:}(n-w)\}$. This algorithm guarantees that any pair of reads with an exact overlap of at least $w+k-1$ bases will share at least one minimum sparse k-mer while drastically reducing the number of k-mers which must be stored in memory (Fig. 4). In certain implementations, minimum sparse k-mers may also improve performance by allowing a k-mer that can be stored in a single 64-bit cell ($k \leq 32$) of memory to represent a longer sequence.

Minimum sparse k-mers are prone to false positives when presented with similar, but not identical, runs of w bases in read pairs. We measured this phenomenon by comparing all k-mers of length w from pairs of reads that share a minimum sparse hash. Figure 4 shows the minimum hamming distance for windows of length w between reads that share a min-sparse hash. When k is larger, the average hamming distance is smaller, though outliers persist. Small values of k produce many distant false positives. Raising k from 20 to 30 ($w = 40$) improved accuracy and precision to the point where false positives could be controlled using downstream techniques.

The mathematics that underlie minimum sparse hashing may also be used to efficiently approximate the overlap between sets, another important operation for Minerva. We did not use this technique in our current implementation of Minerva but plan to explore this for later versions.

Data access

All raw sequencing reads from this study have been submitted to the NCBI BioProject (https://www.ncbi.nlm.nih.gov/bioproject) under accession number PRJNA505182. All code from this study is available in the Supplemental Materials and at https://github.com/dcdanko/minerva_barcode_deconvolution.

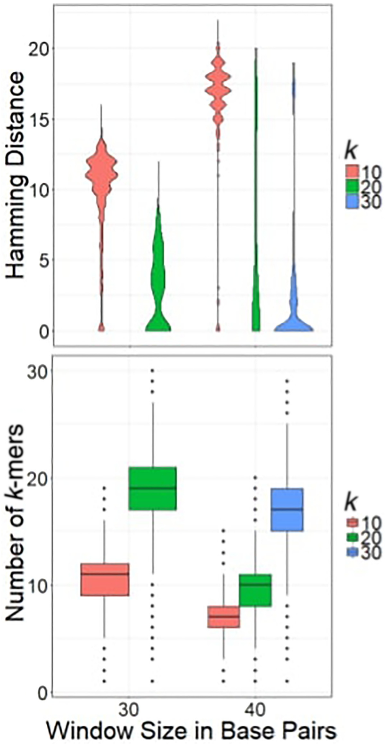

Figure 4. *Top:* hamming distance between windows that share minimizing k-mers, using various parameters. *Bottom*: number of representative minimizing k-mers per read.

Acknowledgments

We thank Victoria Popic and her team at Illumina, Inc. for fruitful discussions on the problem and her help in benchmarking. We sincerely thank 10x Genomics, Inc. for providing us with valuable real sequencing data sets used in this study for evaluations. We also thank Stephen Williams of 10x Genomics, Inc. for coordinating the 10x Metagenomics consortium (a small group of laboratories committed to studying the role of Linked-Reads in metagenomics) in which our team participates. D.C.D. and D.M. were supported by the Tri-Institutional Training Program in Computational Biology and Medicine (CBM) funded by National Institutes of Health (NIH) grant 1T32GM083937. This work was also supported by start-up funds (Weill Cornell Medicine) to I.H. and a National Science Foundation (NSF) grant under award number IIS-1840275 to I.H. and C.M.

References

Blei DM, Ng AY, Jordan MI. 2003. Latent Dirichlet allocation. *J Mach Learn Res* **3:** 993–1022.

Ester M, Kriegel H-P, Sander J, Xu X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the second international conference on knowledge discovery anddata mining*, pp. 226–231, Portland, OR.

Frank JA, Pan Y, Tooming-Klunderud A, Eijsink VGH, McHardy AC, Nederbragt AJ, Pope PB. 2016. Improved metagenome assemblies and taxonomic binning using long-read circular consensus sequence data. *Sci Rep* **6:** 25373. doi:10.1038/srep25373

Girvan M, Newman MEJ. 2002. Community structure in social and biological networks. *Proc Natl Acad Sci* **99:** 7821–7826. doi:10.1073/pnas.122653799

Greer SU, Nadauld LD, Lau BT, Chen J, Wood-Bouwens C, Ford JM, Kuo CJ, Ji HP. 2017. Linked read sequencing resolves complex genomic rearrangements in gastric cancer metastases. *Genome Med* **9:** 57. doi:10.1186/s13073-017-0447-8

Hajek B, Wu Y, Xu J. 2016. Achieving exact cluster recovery threshold via semidefinite programming: extensions. *IEEE Trans Inf Theory* **62:** 5918–5937. doi:10.1109/TIT.2016.2594812

Jain M, Koren S, Miga KH, Quick J, Rand AC, Sasani TA, Tyson JR, Beggs AD, Dilthey AT, Fiddes IT, et al. 2018. Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nat Biotechnol* **36:** 338–345. doi:10.1038/nbt.4060

Marçais G, Pellow D, Bork D, Orenstein Y, Shamir R, Kingsford C. 2017. Improving the performance of minimizers and winnowing schemes. *Bioinformatics* **33:** i110–i117. doi:10.1093/bioinformatics/btx235

Mason CE, Afshinnekoo E, Tighe S, Wu S, Levy S. 2017. International standards for genomes, transcriptomes, and metagenomes. *J Biomol Tech* **28:** 8–18. doi:10.7171/jbt.17-2801-006

McIntyre ABR, Ounit R, Afshinnekoo E, Prill RJ, Hénaff E, Alexander N, Minot SS, Danko D, Foox J, Ahsanuddin S, et al. 2017. Comprehensive benchmarking and ensemble approaches for metagenomic classifiers. *Genome Biol* **18:** 182. doi:10.1186/s13059-017-1299-7

Orenstein Y, Pellow D, Marçais G, Shamir R, Kingsford C. 2016. Compact universal k-mer hitting sets. In *Algorithms in bioinformatics, WABI 2016, lecture notes in computer science* (ed. Frith M, et al.), Vol. 9838, pp. 257–268. Springer, New York.

Schleimer S, Wilkerson DS, Aiken A. 2003. Winnowing: local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD international conference on management of data - SIGMOD'03*, pp. 76–85, San Diego, CA.

Spies N, Weng Z, Bishara A, McDaniel J, Catoe D, Zook JM, Salit M, West RB, Batzoglou S, Sidow A. 2017. Genome-wide reconstruction of complex structural variants using read clouds. *Nat Methods* **14:** 915–920. doi:10.1038/nmeth.4366

Wood DE, Salzberg SL. 2014. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biol* **15:** R46. doi:10.1186/gb-2014-15-3-r46

Zheng GXY, Lau BT, Schnall-Levin M, Jarosz M, Bell JM, Hindson CM, Kyriazopoulou-Panagiotopoulou S, Masquelier DA, Merrill L, Terry JM, et al. 2016. Haplotyping germline and cancer genomes with high-throughput linked-read sequencing. *Nat Biotechnol* **34:** 303–311. doi:10.1038/nbt.3432

Received January 31, 2018; accepted in revised form November 26, 2018.