Genovo: *De Novo* Assembly for Metagenomes

Jonathan Laserson, Vladimir Jojic, and Daphne Koller*

Department of Computer Science, Stanford University, Stanford CA 94305, USA
koller@cs.stanford.edu

Abstract. Next-generation sequencing technologies produce a large number of noisy reads from the DNA in a sample. Metagenomics and population sequencing aim to recover the genomic sequences of the species in the sample, which could be of high diversity. Methods geared towards single sequence reconstruction are not sensitive enough when applied in this setting. We introduce a generative probabilistic model of read generation from environmental samples and present Genovo, a novel *de novo* sequence assembler that discovers likely sequence reconstructions under the model. A Chinese restaurant process prior accounts for the unknown number of genomes in the sample. Inference is made by applying a series of hill-climbing steps iteratively until convergence. We compare the performance of Genovo to three other short read assembly programs across one synthetic dataset and eight metagenomic datasets created using the 454 platform, the largest of which has 311k reads. Genovo's reconstructions cover more bases and recover more genes than the other methods, and yield a higher assembly score.

1 Introduction

Metagenomics and population sequencing aim to recover the genomic sequences in a genetically diverse environmental sample. Examples of such environments include biomes of narrow systems such as human gut [13], honey bees [8], or corals [23,19] and also larger ecosystems [24,22]. These studies advance our systemic understanding of biological processes and communities. In addition, the recovered sequences can enable the discovery of new species [24] or reveal details of poorly understood processes [26]. Another set of examples include cancer tumor cells [27] and pathogen populations such as HIV viral strains [25], where the genetic diversity is associated with disease progression and impacts the effectiveness of the drug treatment regime. Finally, the genetic structure of microbial populations may yield insight into evolutionary mechanisms such as horizontal gene transfer, and enable determination of genetic islands carrying functional toolkits necessary for survival and pathogenicity [20].

Such studies are made possible through the use of next-generation sequencing technologies, such as the Illumina Genome Analyzer (GA), Roche/454 FLX system, and AB SOLiD system. Compared to older sequencing methods, these

* JL and VJ contributed equally to this work. Correspondence should be addressed to DK.

B. Berger (Ed.): RECOMB 2010, LNBI 6044, pp. 341–356, 2010.

sequencers produce a much larger number of relatively short and noisy reads of the DNA in a sample, at a significantly lower cost.

While there are a few *de novo* assemblers aimed at single sequence reconstruction from short reads [6,29,15,5], there are no such tools designed specifically for metagenomics. The challenges stem from uncertainty about the population's size and composition. Additionally, coverage across species is uneven and affected by the species' frequency in the sample. Analysis of the complete populations requires sensitive methods that can reconstruct sequences even for the low-coverage species. Methods geared towards single sequence reconstruction are not sensitive enough when applied in this setting.

Such single sequence reconstruction tools commonly frame the problem as a search for an Eulerian path in a de Bruijn graph. The nodes of the graph are k-mers, with an edge connecting any two k-mers positioned consecutively on the same read. As mentioned by Chaisson et al. [7], "the Eulerian approach works best for error-free reads and quickly deteriorates as soon as the reads have even a small number of base-calling errors". To cope with this problem, a large computational effort is used to detect and correct read errors before any assembly is done. While this approach is feasible for the ultra-short Illumina reads, the task becomes much harder in 454 reads, as the average read length is above 100 (and can reach 400b) and almost every read has an error. In addition, the error correction usually treats reads with low-frequency k-mers as erroneous and discards them. In metagenomics, this could filter out low-frequency species.

We introduce a generative probabilistic model of read generation from environmental samples and present Genovo, a novel *de novo* sequence assembler that works by discovering likely sequence reconstructions under the model. The model captures the uncertainty about the population structure as well as the noise model of the sequencing technology. A Chinese restaurant process prior accounts for the unknown number of genomes in the sample. To discover likely assemblies we perform a series of deterministic and stochastic hill-climbing moves, based on the iterated conditional modes (ICM) algorithm. As we show, our Bayesian approach offers a better sensitivity for assembly in highly diverse environments.

The accurate and sensitive reconstruction of populations has been tackled in restricted domains, such as HIV sequencing, both experimentally [25] and computationally [16,11,28]. However, these tools require prior information on the population and utilize a reference genome. A Chinese restaurant process, similar to ours, was also used in the recent work of Zagordi et al [28]. However, their approach is applicable only to a very small-scale (10^3) set of reads already aligned to a short reference sequence. Our method uses no prior information, scales up to the order of 10^5 454 reads, and simultaneously performs read multiple alignment, read denoising and *de novo* sequence assembly.

We compare the performance of our algorithm to three state of the art short read assembly programs in terms of the number of GenBank bases covered, the number of amino acids recognized by PFAM profiles, and using a score we developed, which quantifies the quality of a *de novo* assembly using no external information. The comparison is conducted on 8 metagenomic datasets

[20,3,4,8,23,10] and one synthetic dataset. Genovo's reconstructions show better performance across a variety of datasets. Genovo is publicly available online at http://cs.stanford.edu/genovo.

2 Methods

Probabilistic Model

An assembly consists of a list of contigs, and a mapping of each read to a contiguous area in a contig. The contigs are represented each as a list of DNA letters $\{b_{so}\}$, where b_{so} is the letter at position o of contig s. For each read x_i, we have its contig number s_i, and its starting location o_i within the contig. We denote by y_i the alignment (orientation, insertions and deletions) required to match x_i base-for-base with the contig. Bold-face letters, such as $\mathbf{b}$ or $\mathbf{s}$, represent the set of variables of that type. The subscript $-i$ excludes the variable indexed i from the set.

Our probabilistic model can be characterized as a generative process, in which we first construct an unbounded number of contigs (each has unbounded length), then assign place holders for the beginning of reads in a coordinate system of contigs and offsets, and finally copy each read's letters (with some noise) from the place it is mapped to in the contig. Formally, this is defined as follows:

1. Infinitely many letters in infinitely many contigs are sampled uniformly:

$$b_{so} \sim \text{Uniform}(\mathcal{B}) \quad \forall s = 1 \ldots \infty, \forall o = -\infty \ldots \infty$$

   where $\mathcal{B}$ is the alphabet of the bases (typically $\mathcal{B} = \{\text{A,C,G,T}\}$).
2. N empty reads are randomly partitioned between these contigs:

$$\mathbf{s} \sim \text{CRP}(\alpha, N)$$

   We use the Chinese Restaurant Process (CRP) [1] as a prior for the randomized partition. $\text{CRP}(\alpha, N)$ generates a partition of N items by assigning the items to classes incrementally. If the first $i-1$ items are assigned to classes $s_1..s_{i-1}$, then item i joins an existing class with a probability proportional to the number of items already assigned to that class, or it joins a new class with a probability proportional to α. The likelihood of a partition under this construction is invariant to the order of the items, and thus yields the following conditional distribution:

$$p(s_i = s|\mathbf{s}_{-i}) = \frac{1}{N-1+\alpha} \cdot \begin{cases} N_{-i,s} & s \text{ is an existing class} \\ \alpha & s \text{ represents a new class} \end{cases}$$

   Where $N_{-i,s}$ counts the number of items, not including i, that are in class s. The parameter α controls the expected number of classes, which in our case represent contigs. In the appendix we show how to set it correctly.

3. The reads are assigned a starting point o_i within each contig:

$$\begin{aligned} \rho_s &\sim \text{Beta}(1, 1+\beta) && \forall s \text{ that is not empty} \\ o_i &\sim \mathcal{G}(\rho_s) && \forall i = 1..N \end{aligned}$$

We set $\beta = 100$. The distribution $\mathcal{G}$ is a symmetric variation of geometric distribution that includes all the negative integers and is centered at 0. The parameter ρ_s controls the length of the region from which reads are generated:

$$\mathcal{G}(o; \rho) = \begin{cases} 0.5(1-\rho)^{|o_i|}\rho & o \neq 0 \\ \rho & o = 0 \end{cases}$$

4. Each read is assigned a length l_i, and then its letters x_i are copied (with some mismatches) from its contig s_i starting from position o_i and according to the alignment y_i (encoding orientation, insertions and deletions):

$$\begin{aligned} l_i &\sim \mathcal{L} && \forall i = 1..N \\ x_i, y_i &\sim \mathcal{A}(l_i, s_i, o_i, \mathbf{b}, p_{ins}, p_{del}, p_{mis}) && \forall i = 1..N \end{aligned}$$

$\mathcal{L}$ is any arbitrary distribution over read lengths. The distribution $\mathcal{A}$ represents the noise model known for the sequencing technology (454, Illumina, etc.). For example, if each read letter has a p_{mis} probability to be copied incorrectly, and the probabilities for insertions and deletions are p_{ins} and p_{del} respectively, then the log-probability $\log p(x_i, y_i | o_i, s_i, l_i, \mathbf{b})$ of generating a read in the reverse orientation with n_{hit} matches, n_{mis} mismatches, n_{ins} insertions and n_{del} deletions is

$$\log 0.5 + n_{hit} \log(1-p_{mis}) + n_{mis} \log\left(\frac{p_{mis}}{|\mathcal{B}|-1}\right) + n_{ins} \log(p_{ins}) + n_{del} \log(p_{del})$$

assuming an equal chance (0.5) to appear in each orientation and an independent noise model. Given an assembly, we denote the above quantity as $\text{score}^i_{\text{READ}}$, where i is the read index.

This model includes an infinite number of b_{so} variables, which clearly cannot all be represented in the algorithm. The trick is to treat most of these variables as 'unobserved', effectively integrating them out during likelihood computations. The only observed b_{so} letters are those that are supported by reads, i.e. have at least one read letter aligned to location (s, o). Hence, in the algorithm detailed below, if a contig letter loses its read support, it immediately becomes 'unobserved'.

Algorithm

Our algorithm is an instance of the iterated conditional modes (ICM) algorithm [2], which maximizes local conditional probabilities sequentially, in order to reach the MAP solution. Starting from any initial assembly (our initializing assembly

treats each read as occupying its own contig), our algorithm performs a series of hill-climbing moves in the space of assemblies, in an iterative fashion. We run our algorithm until convergence (200-300 iterations), and then we output the assembly that achieved the highest probability thus far. Running the algorithm multiple times with different random seeds showed no significant influence on the resulting assembly. This suggests that while our algorithm has some stochastic elements, the variability of the output is low. We list below the moves used to explore the space:

Consensus Sequence. This type of move performs ICM updates over the (observed) letter variables b_{so}. For each location (s, o), let a^b_{so} be the number of reads in the current assembly that align the letter $b \in \mathcal{B}$ to location (s, o). Since we assumed a uniform prior over the contig letters, we optimize the score by setting $b_{so} = \arg\max_{b\in\mathcal{B}} a^b_{so}$ (ties broken randomly).

Read Mapping. This move performs stochastic ICM updates over the read variables s_i, o_i, y_i. For each read i, we start by removing it completely from the assembly. We choose a new location and alignment for the read (s_i, o_i, y_i) by sampling from the joint posterior $p(s_i = s, o_i = o, y_i = y|x_i, \mathbf{y}_{-i}, \mathbf{s}_{-i}, \mathbf{o}_{-i}, \mathbf{b}, \boldsymbol{\rho})$.

For every potential location (s, o), we first compute y^*_{so}, the best alignment of the read for that location, using the banded Smith-Waterman algorithm (applied to both read orientations):

$$y^*_{so} = \arg\max_y p(x_i, y|s_i = s, o_i = o, \mathbf{b}).$$

This includes locations where the read only partially overlaps with the contig, in which case aligning a read letter to an unobserved contig letter entails a probabilistic price of $\log(|\mathcal{B}|^{-1})$ per letter. We now set s_i, o_i by sampling a location (s, o) from $p(s_i = s, o_i = o, y^*_{so}|\cdot)$:

$$\begin{aligned} p(s_i{=}s, o_i{=}o, y^*_{so}|\cdot) &\propto p(s_i{=}s|\mathbf{s}_{-i})p(o_i{=}o|s_i = s, \rho_s)p(x_i, y^*_{so}|s_i{=}s, o_i{=}o, \mathbf{b}) \\ &\propto N_s \cdot \mathcal{G}(o; \rho_s) \cdot p(x_i, y^*_{so}|s_i = s, o_i = o, \mathbf{b}) \end{aligned}$$

The weights $\{N_s\}$, which are counting the number of reads in each sequence, encourage the read to join large contigs. As dictated by the CRP, we also include the case where s represents an empty contig, in which case we simply replace N_s with α in the formula above. In that case, the $p(x_i, y^*_{so})$ component also simplifies to $l_i \log(|\mathcal{B}|^{-1})$, where l_i is the length of the read. We set $y_i = y^*_{s_i o_i}$.

As bad alignments render most (s, o) combinations extremely unlikely, we significantly speed up the above computation by filtering out combinations with implausible alignments. A very fast computation can detect locations that have at least one 10-mer in common with the read. This weak requirement is enough to filter out all but a few locations, making the optimization process efficient and scalable. A further speedup is achieved by caching common alignments.

Geometric Variables. This step performs ICM updates on the ρ_s variables. Each draw of a location o from $\mathcal{G}(\rho_s)$ can be thought of a set of $|o| + 1$ Bernoulli

trials with $|o|$ failures and one success. Let $\hat{o}_1, \ldots, \hat{o}_{N_s}$ be the offsets of the reads assigned to sequence s. By a known property of the Beta distribution, it follows that $\rho_s|\hat{o}_1, \ldots, \hat{o}_{N_s} \sim \text{Beta}(1 + N_s, 1 + \beta + O_s)$ where $O_s = \sum_{k=1}^{N_s} |\hat{o}_k|$. We set ρ_s to $\frac{N_s}{N_s+\beta+O_s}$, the mode of the above distribution.

Global Moves. The above ICM moves are very local. To speed up convergence, we employ the following set of global moves, each one changes a set of variables at once, and hence takes a larger step in the space of assemblies. **(a) Propose indels.** If at a specific location most reads have an insertion, we propose to delete the corresponding letter in the contig and realign the reads, and accept the proposal if that improves the likelihood. For example, if out of n reads, a reads have an insertion, then after the proposed change those a reads will have one less insertion each, and $n - a$ reads will have a new deletion. We have a similar move for deletions. **(b) Center.** We change the coordinate system of each sequence to maximize the $p(o)$ component of the likelihood. **(c) Merge.** We merge two contigs whose ends overlap, if it improves the likelihood.

Chimeric Reads. Chimeric reads [17] are reads with a prefix and a suffix matching distant locations in the genome. In our algorithm, these rare corrupted reads often find their way to the edge of an assembled contig, thus interfering with the assembly process. To deal with this problem we occasionally (every 5 iterations) disassemble the reads sitting in the edge of a contig, thus allowing other correct reads or contigs to merge with it and increase the likelihood beyond that of the original state. If such a disassembled read was not chimeric, it will reassemble correctly in the next iteration, thus keeping the likelihood the same as before.

Evaluation Metrics

Running on a set of reads, each method outputs the list of contigs that it was able to assemble from the reads. As done in previous studies [6,18], we evaluate only contigs longer than 500bp.

Since for non-simulated data we do not have the actual list of genomes (the 'ground truth') that generated it, exact evaluation of *de novo* assemblies in metagenomic analysis is hard. We utilize three different indicators for the quality of an assembly. For the first indicator, we BLASTed the contigs produced by each method. Our goal was to estimate the number of genome bases that the contigs span. For each dataset, we used the BLAST hits of all the methods to compile a pool of genomes (downloaded from GenBank) that best represent the consensus among the methods. Then, for each method, each base in the pool's genomes received a score indicating the quality of the best alignment covering it (the BLAST alignment score divided by the length of the aligned interval). We were then able to ask the question "How many pool bases were covered with a score greater than x?", and plot it in a graph which we call the *BLAST profile*.

The value of the reconstructed sequences lies in the information they carry about the underlying population, such as is provided by the functional annotation of the contigs. Our second indicator evaluated the assemblies based on this information. We decoded the contigs into protein sequences (in all 6 reading

frames) and annotated these sequences with PFAM profile detection tools [12]. We denote by $\text{score}_{\text{PFAM}}$ the total number of decoded amino acids matched by PFAM profiles.

The above two indicators can be easily biased when exploring environments with sequences that are not yet in these databases, and hence our third indicator is a score that uses no external information and relies solely on the reads' consistency. Given an assembly, denote by S the number of contigs, and by L the total length of all the contigs. We measure the quality of an assembly using the expression

$$\sum_i \text{score}^i_{\text{READ}} - \log(|\mathcal{B}|)L + \log(|\mathcal{B}|)V_0 S.$$

The first term penalizes for read errors and the second for contig length, embodying the trade off required for a good assembly. For example, the first term will be optimized by a naive assembly that lays each read in its own contig (without any changes), but the large number of total bases will incur a severe penalty from the second term. These two terms interact well since they represent probabilities - the first term is the (log) probability for generating each noisy read from the contig bases it aligns to, and the second term is the (log) probability for generating (uniformly) each contig letter. The third term ensures a minimal overlap of V_0 bases between two consecutive reads. To see this, assume two reads have an overlap of V bases. If you split the contig into two at this position, the third term gives you a 'bonus' of $\log(|\mathcal{B}|)V_0$, while the second term penalizes you for $\log(|\mathcal{B}|)V$ for adding V new bases to the assembly. Hence, we will prefer to merge the sequences iff $V > V_0$. We set V_0 to 20.

To be able to compare the above score across different datasets, we normalized it by first subtracting from it the score of a naive assembly that puts each read in its own contig, and then dividing this difference by the total length of all the reads in the dataset. We define score_{denovo} to be this normalized score. See Appendix for another derivation of score_{denovo}, based on our model.

3 Results

While many sequencing techologies are gaining popularity, most of the short-read metagenomic datasets currently available have been sequenced using 454 sequencers (probably due to their longer reads), hence we focus on this technology. We compare the performance of our algorithm to three other tools: Velvet [29], EULER-SR [6] and Newbler, the 454 Life Science *de novo* assembler. Newbler was specifically designed for 454 reads and is provided with the 454 machine. Velvet and EULER-SR were designed for the shorter Illumina reads, but support 454 reads as well and are freely available.

Before testing the methods on the metagenomic datasets, we benchmarked them on a single sequence assembly task. We used run SRR024126 from NCBI short read archive, which contains 110k reads taken from E. coli (length 4.6Mb). Even though Genovo was not optimized for the single sequence assembly task, it performed on par with the other methods, as Table 1 shows.

Table 1. Comparing the methods on a single sequencing task. Contigs were mapped using BLAST to the E. coli reference strand (NC_000913.2). Coverage computed by taking the union of all matching intervals with length > 400b. Identities are exact base matches (i.e. not including gaps and mismatches). Nx is the largest value y such that at least x% of the genome is covered by contigs of length $\geq y$.

	no. contigs	**total contig length(kb)**	**N50 (kb)**	**N90 (kb)**	**coverage (%)**	**identities (%)**
Genovo	129	4693	76.9	25.9	88.4	98.5
Newbler	150	4645	60.4	17.6	88.9	98.5
Velvet	621	4496	10.5	3.6	87.6	98.6
Euler	828	4493	7.6	2.6	86.9	98.6

Table 2. Metagenomic Datasets. Accession numbers starting with 'SRR' refer to NCBI Short Read Archive (http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi).

name (#reads)	**description (source)**
Bee1(19k), Bee2(36k) [8]	Samples from two bee colonies. Data obtained by J. DeRisi Lab.
Coral(40k) [23]	Samples from viral fraction from whole Porites compressa tissue extracts (SRR001078).
Tilapia1(50k), Tilapia2(64k) [10]	Samples from Kent SeeTech Tilapia farm containing microbial (SRR001069) and viral (SRR001066) communities isolated from the gut contents of hybrid striped bass.
Peru(84k) [3]	Marine sediment metagenome from the Peru Margin subseafloor (SRR001326).
Microbes(135k) [4]	Samples from the Rios Mesquites stromatolites in Cuatro Cienagas, Mexico (SRR001043).
Chicken(311k) [20]	Samples of microbiome from chicken cecum. Dataset at http://metagenomics.nmpdr.org, accession 4440283.3
Synthetic(50k)	Metagenomic samples of 13 virus strains, generated using Metasim [21], a 454 simulator. See Appendix for list.

We carried on to compare the methods in a metagenomics setting. The comparison is conducted on 8 datasets from 6 different studies, and one synthetic dataset (see Table 2). Figure 1 compares the different methods across datasets using score_{denovo} (we could not run EULER-SR on Coral). Genovo wins on every dataset, with as high as 366% advantage over the second best method. On the synthetic dataset, Genovo assembled all the reads (100.0%) into 13 contigs, one for each virus. The assemblies returned by the other methods are much more fractured — Euler, Velvet and Newbler returned 33, 47, and 38 contigs, representing only 88%, 36% and 68% of the reads, respectively. The real datasets with highest score_{denovo} were Bee1, Bee2 and Tilapia1. Genovo was able to assemble in large contigs 60%, 80% and 96% of the reads in these datasets, respectively, compared to 30%, 25% and 59% achieved by the second best method. The low score_{denovo} values for the other datasets reflect a low or no overlap between most

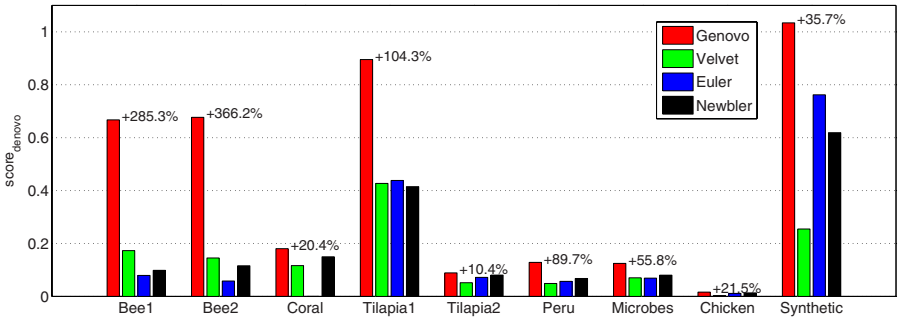

Fig. 1. Comparing the methods based on score_{denovo}. The numbers above the bars represent the improvement (in percentages) between Genovo and the second-best method. To compute score_{denovo}, we had to complete each list of contigs to a full assembly, by mapping each read to the location that explains it best. Reads that did not align well to any location were treated as singletons - aligned perfectly to their own contig.

reads in those datasets. Such reads almost always lead to assemblies with many short contigs, regardless of the method used, which drive the score to 0. An example of such dataset is Chicken — all methods produced assemblies which ignored at least 97% of the reads.

Figure 2 shows the *BLAST profile* for each method, a curve that visualizes the quantity vs. the quality of the contigs (see Methods). On the synthetic dataset, Genovo covered almost all the bases (99.7%) of the 13 viruses. Other methods did poorly: Newbler, Euler and Velvet covered 72.4%, 63.4% and 39.3% of the bases, respectively. As for the real datasets, in Bee1, Bee2, Tilapia2 and Chicken many contigs showed a significant match in BLAST ($E < 10^{-9}$) and the BLAST profiles provide a good indication for the assembly quality. In those cases not only does Genovo discover more bases, but it also produces better quality contigs, since Genovo's profile dominates the other methods even on high thresholds for the alignment quality (except on Tilapia2). These differences could also translate to more species. For example, in Bee1, none of Euler's and Newbler's contigs matched in BLAST to Apis mellifera 18S ribosomal RNA gene, even though Genovo and Velvet had contigs that matched it well. On the other datasets most of the contigs did not show a significant match, and hence the genome pools compiled for those datasets are incomplete in the sense that they do not represent all the genomes in the (unknown) ground truth.

Figure 3 compares the methods in terms of the number of amino acids matched by a protein family, as measured by $\text{score}_{\text{PFAM}}$ (see Methods). In all datasets Genovo has the highest score (with the exception of Bee1, where Newbler wins by 260aa), indicating that Genovo's contigs hold more (and longer) annotated regions. For example, in the highly fractured Chicken dataset, our BLAST and PFAM results are markedly higher: 65% more bases were significantly ($E < 10^{-9}$) matched in BLAST and 36% more amino acids recognized in PFAM compared to the second best method (Newbler). The difference is also qualitative — the

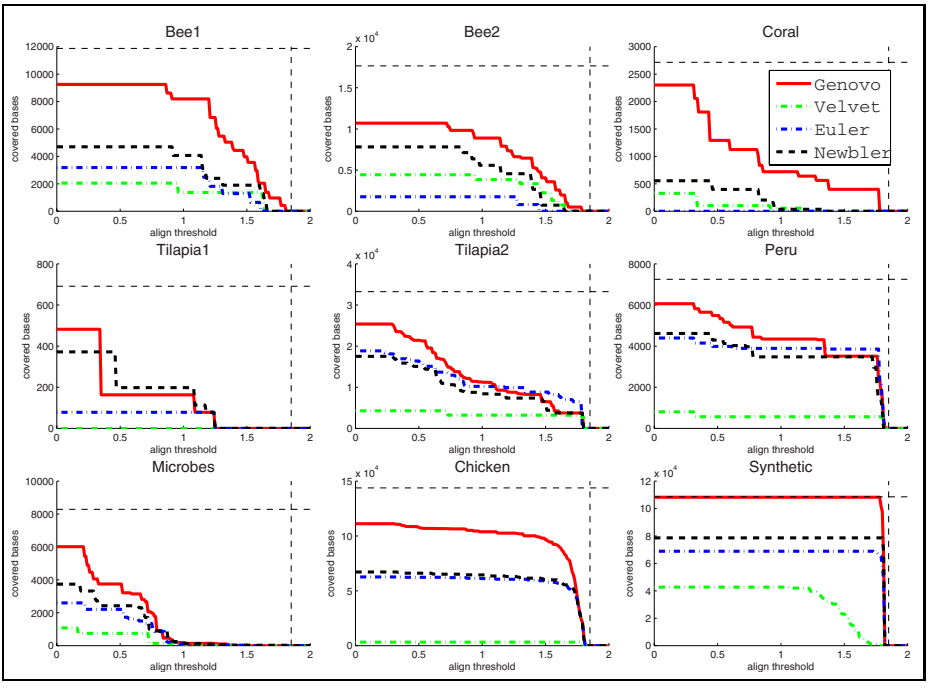

Fig. 2. The *BLAST profiles* of each method across all datasets. For each dataset we compiled a pool of sequences representing the ground truth. For each method, each base in the pool receives a score indicating the quality of the best alignment covering it. The curve shows how many bases received a score higher than the x value. The dashed horizontal line represents the total no. of bases in the pool covered by at least one method. The dashed vertical line represents the alignment quality of an exact match.

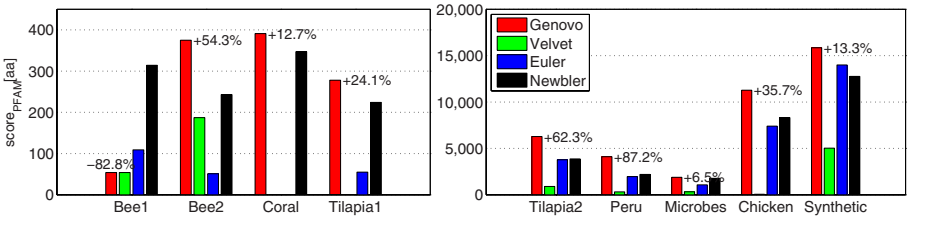

Fig. 3. Comparing the methods based on $\text{score}_{\text{PFAM}}$. The contigs were translated to proteins in all 6 reading frames. $\text{score}_{\text{PFAM}}$ measures how many amino acids were recognized by protein families profilers. Due to the scale difference, results are divided into two figures with the datasets on the right figure having an order of magnitude more annotated amino acids. The numbers above the bars show the change between Genovo and the best of the other methods.

contigs reconstructed by our method were recognized by 84 distinct PFAM families, compared to 67 for Newbler's contigs. It is important to note that in our assembly, the length of matched regions ranged from 54 to1206aa, with average region length ∼289aa. Similar performance on PFAM matching was achieved on the Tilapia2 dataset, where the number of matched families was 47 (compared to Newbler's 33), and the range of matched regions was 60-1137aa. Such long matched regions could not be recovered from a read-level analysis.

The BLAST and PFAM results should not be taken as the ultimate measure of the reconstruction quality, or the dataset quality, since environmental samples may contain uncultured species that are phylogenetically distant from anything sequenced before. An example of such a dataset is Tilapia1, where almost all the contigs did not match significantly, as shown by the BLAST profiles and $\text{score}_{\text{PFAM}}$, even though they had significant coverage (one of our contigs, with no significant BLAST match, had a segment of 3790 bases with a minimal coverage of ×85 and a mean coverage of ×177). Importantly, score_{denovo} does not suffer from the same problems since it is based on the quality of the read data reconstruction, rather than the presence of a ground truth proxy.

4 Discussion

Metagenomic analysis involves samples of poorly understood populations. The sequenced sets of reads approximate that population and can yield information about the distribution of gene functions as well as species. However, due to fluctuations of the genomes' coverage, these distributions may be poorly estimated. Furthermore, a read-level analysis may not be able to detect motifs that span multiple reads. Finally, a detailed analysis of events such as horizontal gene transfer will necessitate obtaining both the transposed elements and the genetic context into which they transposed. All of these concerns, in addition to a desire to obtain sequences for novel species, motivate development of sequence assembly methods aimed at problems of population sequencing.

Uncertainty over the sample composition, read coverage, and noise levels make development of methods for metagenomic sequence assembly a challenging problem. We developed a method for sequence assembly that performs well both on biologically relevant scores (based on BLAST and PFAM matches) and on a score that uses no external information. One advantage of our approach is that our probabilistic model is modular, permitting changes to the noise model without the need to modify the rest of the model. Thus, the extensions to other sequencing methodologies, as they are applied to metagenomic data, should be fairly straightforward. In addition, instead of a uniform prior over the genome letters one can use a prior based on a reference genome. Such prior will boost the model's sensitivity in detecting variants of that genome, which can be useful when sequencing viral populations or transcriptome.

Our algorithm performs deterministic and stochastic hill-climbing moves based on the conditional probabilities derived from our probabilistic model. This

approach is suited for the problem of finding the best assembly. In a setting where the goal is to find multiple alternative reconstructions (alternative splicing, horizontal gene transfer), the same formulas can be used to construct a sampler that comprehensively explores the space according to the MCMC algorithm, and is thus more likely to explore all the modes of the distribution.

The running time required to construct an assembly can range from 15 minutes on a single CPU for a dataset with 40k reads up to a few hours for a dataset with 300k 454 reads, depending not only on the size but also on the complexity of the dataset. Newbler, Velvet and Euler typically provide their results on the order of minutes. Our increase in computational time is compatible with the time spent on a next generation sequencing run and it is worthwhile considering the superior results compared to the other assemblers.

The promise of metagenomic studies lies in their potential to elucidate interactions between members of an ecosystem and their influence on the environment they inhabit. For example, deeper understanding of constituent parts of the microbiota inhabiting humans [9,13,14] as well as their evolution in response to environmental changes, such as presence of antibiotics, will be necessary for targeted drug design. In order to begin answering questions about these populations, systematic *sequence* level analysis is necessary. With the advances of the sequencing technology and increases in the coverage, methods which can explore the space of possible reconstructions will become even more important. The model and method introduced in this paper are well suited to meet these challenges.

Acknowledgements

This material is based upon work supported under a Stanford Graduate Fellowship and a National Science Foundation Grant BDI-0345474.

References

1. Aldous, D.: Exchangeability and related topics. École d'été de probabilités de Saint-Flour, XIII, pp. 1–198 (1983)
2. Besag, J.: On the statistical analysis of dirty pictures. Journal of the Royal Statistical Society, Series B (Methodological) 48(3), 259–302 (1986)
3. Biddle, J.F., Fitz-Gibbon, S., Schuster, S.C., Brenchley, J.E., House, C.H.: Metagenomic signatures of the Peru Margin subseafloor biosphere show a genetically distinct environment. Proc. Natl. Acad. Sci. U.S.A. 105, 10583–10588 (2008)
4. Breitbart, M., Hoare, A., Nitti, A., Siefert, J., Haynes, M., Dinsdale, E., Edwards, R., Souza, V., Rohwer, F., Hollander, D.: Metagenomic and stable isotopic analyses of modern freshwater microbialites in Cuatro Cinegas, Mexico. Environ. Microbiol. 11, 16–34 (2009)
5. Butler, J., Mac Callum, I., Kleber, M., Shlyakhter, I.A., Belmonte, M.K., Lander, E.S., Nusbaum, C., Jaffe, D.B.: ALLPATHS: De novo assembly of whole-genome shotgun microreads. Genome Research 18(5), 810–820 (2008)

6. Chaisson, M.J., Pevzner, P.A.: Short read fragment assembly of bacterial genomes. Genome Research 18(2), 324–330 (2008)
7. Chaisson, M.J.P., Brinza, D., Pevzner, P.A.: De novo fragment assembly with short mate-paired reads: Does the read length matter? Genome Research 19, 336–346 (2009)
8. Cox-Foster, D.L., Conlan, S., Holmes, E.C., Palacios, G., Evans, J.D., Moran, N.A., Quan, P.-L., Briese, T., Hornig, M., Geiser, D.M., Martinson, V., van Engelsdorp, D., Kalkstein, A.L., Drysdale, A., Hui, J., Zhai, J., Cui, L., Hutchison, S.K., Simons, J.F., Egholm, M., Pettis, J.S., Ian Lipkin, W.: A Metagenomic Survey of Microbes in Honey Bee Colony Collapse Disorder. Science 318(5848), 283–287 (2007)
9. Diaz-Torres, M.L., Villedieu, A., Hunt, N., McNab, R., Spratt, D.A., Allan, E., Mullany, P., Wilson, M.: Determining the antibiotic resistance potential of the indigenous oral microbiota of humans using a metagenomic approach. FEMS Microbiol. Lett. 258, 257–262 (2006)
10. Dinsdale, E.A., Edwards, R.A., Hall, D., Angly, F., Breitbart, M., Brulc, J.M., Furlan, M., Desnues, C., Haynes, M., Li, L., McDaniel, L., Moran, M.A., Nelson, K.E., Nilsson, C., Olson, R., Paul, J., Brito, B.R., Ruan, Y., Swan, B.K., Stevens, R., Valentine, D.L., Thurber, R.V., Wegley, L., White, B.A., Rohwer, F.: Functional metagenomic profiling of nine biomes. Nature 452, 629–632 (2008)
11. Eriksson, N., Pachter, L., Mitsuya, Y., Rhee, S.Y., Wang, C., Gharizadeh, B., Ronaghi, M., Shafer, R.W., Beerenwinkel, N.: Viral population estimation using pyrosequencing. PLoS Comput. Biol. 4, e1000074 (2008)
12. Finn, R.D., Tate, J., Mistry, J., Coggill, P.C., Sammut, S.J., Hotz, H.R., Ceric, G., Forslund, K., Eddy, S.R., Sonnhammer, E.L., Bateman, A.: The Pfam protein families database. Nucleic Acids Res. 36, S281–S288 (2008)
13. Gill, S.R., Pop, M., Deboy, R.T., Eckburg, P.B., Turnbaugh, P.J., Samuel, B.S., Gordon, J.I., Relman, D.A., Fraser-Liggett, C.M., Nelson, K.E.: Metagenomic analysis of the human distal gut microbiome. Science 312, 1355–1359 (2006)
14. Grice, E.A., Kong, H.H., Renaud, G., Young, A.C., Bouffard, G.G., Blakesley, R.W., Wolfsberg, T.G., Turner, M.L., Segre, J.A.: A diversity profile of the human skin microbiota. Genome Res. 18, 1043–1050 (2008)
15. Hernandez, D., Franois, P., Farinelli, L., sters, M., Schrenzel, J.: De novo bacterial genome sequencing: Millions of very short reads assembled on a desktop computer. Genome Research 18(5), 802–809 (2008)
16. Jojic, V., Hertz, T., Jojic, N.: Population sequencing using short reads: HIV as a case study. In: Pac. Symp. Biocomput., pp. 114–125 (2008)
17. Lasken, R.S., Stockwell, T.B.: Mechanism of chimera formation during the Multiple Displacement Amplification reaction. BMC Biotechnol. 7, 19 (2007)
18. Margulies, M., Egholm, M., Altman, W.E., Attiya, S., Bader, J.S., Bemben, L.A., Berka, J., Braverman, M.S., Chen, Y.J., Chen, Z., Dewell, S.B., Du, L., Fierro, J.M., Gomes, X.V., Godwin, B.C., He, W., Helgesen, S., Ho, C.H., Ho, C.H., Irzyk, G.P., Jando, S.C., Alenquer, M.L., Jarvie, T.P., Jirage, K.B., Kim, J.B., Knight, J.R., Lanza, J.R., Leamon, J.H., Lefkowitz, S.M., Lei, M., Li, J., Lohman, K.L., Lu, H., Makhijani, V.B., McDade, K.E., McKenna, M.P., Myers, E.W., Nickerson, E., Nobile, J.R., Plant, R., Puc, B.P., Ronan, M.T., Roth, G.T., Sarkis, G.J., Simons, J.F., Simpson, J.W., Srinivasan, M., Tartaro, K.R., Tomasz, A., Vogt, K.A., Volkmer, G.A., Wang, S.H., Wang, Y., Weiner, M.P., Yu, P., Begley, R.F., Rothberg, J.M.: Genome sequencing in microfabricated high-density picolitre reactors. Nature 437, 376–380 (2005)

19. Meyer, E., Aglyamova, G., Wang, S., Buchanan-Carter, J., Abrego, D., Colbourne, J., Willis, B., Matz, M.: Sequencing and de novo analysis of a coral larval transcriptome using 454 gsflx. BMC Genomics 10(1), 219 (2009)
20. Qu, A., Brulc, J.M., Wilson, M.K., Law, B.F., Theoret, J.R., Joens, L.A., Konkel, M.E., Angly, F., Dinsdale, E.A., Edwards, R.A., Nelson, K.E., White, B.A.: Comparative metagenomics reveals host specific metavirulomes and horizontal gene transfer elements in the chicken cecum microbiome. PLoS ONE 3, e2945 (2008)
21. Richter, D.C., Ott, F., Auch, A.F., Schmid, R., Huson, D.H.: Metasim: A sequencing simulator for genomics and metagenomics. PLoS ONE 3(10), e3373 (2008)
22. Tyson, G.W., Chapman, J., Hugenholtz, P., Allen, E.E., Ram, R.J., Richardson, P.M., Solovyev, V.V., Rubin, E.M., Rokhsar, D.S., Banfield, J.F.: Community structure and metabolism through reconstruction of microbial genomes from the environment. Nature 428, 37–43 (2004)
23. Vega Thurber, R.L., Barott, K.L., Hall, D., Liu, H., Rodriguez-Mueller, B., Desnues, C., Edwards, R.A., Haynes, M., Angly, F.E., Wegley, L., Rohwer, F.L.: Metagenomic analysis indicates that stressors induce production of herpes-like viruses in the coral Porites compressa. Proceedings of the National Academy of Sciences 105(47), 18413–18418 (2008)
24. Craig Venter, J., Remington, K., Heidelberg, J.F., Halpern, A.L., Rusch, D., Eisen, J.A., Wu, D., Paulsen, I., Nelson, K.E., Nelson, W., Fouts, D.E., Levy, S., Knap, A.H., Lomas, M.W., Nealson, K., White, O., Peterson, J., Hoffman, J., Parsons, R., Baden-Tillson, H., Pfannkoch, C., Rogers, Y.-H., Smith, H.O.: Environmental Genome Shotgun Sequencing of the Sargasso Sea. Science 304(5667), 66–74 (2004)
25. Wang, C., Mitsuya, Y., Gharizadeh, B., Ronaghi, M., Shafer, R.W.: Characterization of mutation spectra with ultra-deep pyrosequencing: application to HIV-1 drug resistance. Genome Res. 17, 1195–1201 (2007)
26. Warnecke, F., Luginbhl, P., Ivanova, N., Ghassemian, M., Richardson, T.H., Stege, J.T., Cayouette, M., McHardy, A.C., Djordjevic, G., Aboushadi, N., Sorek, R., Tringe, S.G., Podar, M., Martin, H.G., Kunin, V., Dalevi, D., Madejska, J., Kirton, E., Platt, D., Szeto, E., Salamov, A., Barry, K., Mikhailova, N., Kyrpides, N.C., Matson, E.G., Ottesen, E.A., Zhang, X., Hernndez, M., Murillo, C., Acosta, L.G., Rigoutsos, I., Tamayo, G., Green, B.D., Chang, C., Rubin, E.M., Mathur, E.J., Robertson, D.E., Hugenholtz, P., Leadbetter, J.R.: Metagenomic and functional analysis of hindgut microbiota of a wood-feeding higher termite. Nature 450, 560–565 (2007)
27. Warren, R.L., Nelson, B.H., Holt, R.A.: Profiling model T-cell metagenomes with short reads. Bioinformatics 25(4), 458–464 (2009)
28. Zagordi, O., Geyrhofer, L., Roth, V., Beerenwinkel, N.: Deep sequencing of a genetically heterogeneous sample: Local haplotype reconstruction and read error correction. In: Batzoglou, S. (ed.) RECOMB 2009. LNCS, vol. 5541, pp. 271–284. Springer, Heidelberg (2009)
29. Zerbino, D.R., Birney, E.: Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Research 18, 821–829 (2008)

Appendix

Understanding the Likelihood

In order to choose the α parameter correctly, we have to understand our model better. Assume there are N reads and S contigs, with N_s the number of reads in contig s. Our model log-likelihood can be written as

$$\log p(\mathbf{x}, \mathbf{y}|\mathbf{s}, \mathbf{o}, \mathbf{b}) + \log p(\mathbf{b}) + \log p(\mathbf{o}|\mathbf{s}, \boldsymbol{\rho}) + \log p(\mathbf{s})$$

where

$$\begin{aligned}
&\log p(\mathbf{x}, \mathbf{y}|\mathbf{s}, \mathbf{o}, \mathbf{b}) = \sum_i \text{score}^i_{\text{READ}} \\
&\log p(\mathbf{b}) = -\log(|\mathcal{B}|)L \\
&\log p(\mathbf{s}) = \log(\alpha)S + \sum_s \log \Gamma(N_s) + \text{const}(\alpha, N) \\
&\log p(\mathbf{o}|\mathbf{s}, \boldsymbol{\rho}) = \sum_s O_s \log(1-\rho_s) + N_s \log \rho_s + const(N);
\end{aligned}$$

where L is the total length of all the contigs, $O_s = \sum_{i:s_i=s} |o_i|$, and $\Gamma(\cdot)$ is the gamma function. There is an interesting interaction between $\log p(\mathbf{s})$ and $\log p(\mathbf{o})$. To simplify $\log p(s)$ we use the Sterling approximation $\log \Gamma(x) \approx \left(x - \frac{1}{2}\right) \log x - x + \frac{1}{2} \log(2\pi)$:

$$\sum_s \log \Gamma(N_s) \approx \sum_s N_s \log N_s + \frac{1}{2} \log(2\pi) S - \frac{1}{2} \sum_s \log N_s + \text{const}(N)$$

To simplify $\log p(o)$, we will assume there is a roughly uniform coverage across all contigs, with d the average distance between the o_i of two consecutive reads. It follows that contig s is roughly of length $N_s d$. After a centering move, the reads' offsets stretch from $-N_s d/2$ to $N_s d/2$, and we can thus estimate as $O_s = N_s^2 d/4$. When ρ_s is updated, it is set to be

$$\rho_s = \frac{N_s}{N_s + \beta + O_s} = \frac{4}{4 + \frac{\beta}{N_s} + N_s d} \approx \frac{4}{N_s d}$$

(here we assume $N_s >> \beta \geq 1$). Using Taylor approximation:

$$\log(1-\rho_s) \approx -\rho_s - 0.5\rho_s^2 = -\frac{4}{N_s d} - \frac{8}{N_s^2 d^2}$$

Hence:

$$\begin{aligned}
\log p(\mathbf{o}|\mathbf{s}, \boldsymbol{\rho}) &= \sum_s \frac{N_s^2 d}{4} \left(-\frac{4}{N_s d} - \frac{8}{N_s^2 d^2} \right) + \sum_s N_s (\log \frac{4}{d} - \log N_s) \\
&= -\sum_s N_s \log N_s - \frac{2}{d} S + \text{const}(N, d)
\end{aligned}$$

Combining the formulas for $\log p(o)$ and $\log p(s)$, the most dominant term cancels out and we obtain this formula for the log-likelihood (removing constants):

$$\sum_i \text{score}^i_{\text{READ}} - \log(|\mathcal{B}|)L + \left(\log\alpha - \frac{2}{d} + \frac{1}{2}\log(2\pi)\right) S - \frac{1}{2}\sum_s \log N_s$$

As the last term is in effect very weak, this can be seen as an alternative derivation for score_{denovo}.

Consider an assembly that has two contigs with a perfect overlap of V_0 bases. Now consider the assembly obtained by my merging (correctly) the two overlapping contigs. For simplicity, assume both contigs have N_0 reads. The difference in log-likelihood between those two assemblies $\log p(\text{merged}) - \log p(\text{split})$ becomes zero when

$$\log\alpha = \log(|\mathcal{B}|)V_0 + \frac{1}{2}\log\left(\frac{N_o}{4\pi}\right) + \frac{2}{d}$$

We use this formula to tune α appropriately. In the datasets we have, d is always larger than 2, which disables the last term. We want to merge contigs with $N_0 = 10$ reads or more, provided that they have an overlap larger then $V_0 = 20$ bases. Based on this formula, we set $\alpha = 2^{40}$, which experimentally gives better results than other values.

Synthetic Dataset

We used Metasim with the default configuration for 454-250bp reads. The dataset was composed of the following sequences (in parenthesis, number of reads): Acidianus filamentous virus 1 (14505), Akabane virus segment L (4247), Akabane virus segment M (2636), Black queen cell virus (5309), Cactus virus X (3523), Chinese wheat mosaic virus RNA1 (3300), Chinese wheat mosaic virus RNA2 (1649), Cucurbit aphid-borne yellows virus (2183), Equine arteritis virus (4832), Goose paramyxovirus SF02 (4714), Human papillomavirus - 1 (1846), Okra mosaic virus (1016), Pariacoto virus RNA1 (240).

Running Velvet, Euler and Newbler

For *Velvet*, we run `velveth` with k-mer length 21. We run `velvetg` multiple times using 14 values between 1 and 30 for the `-cov_cutoff` parameter. We choose the configuration which maximizes the N50. For *EULER-SR*, we run `Assemble.pl` setting the k-mer length to 25. For *Newbler*, we run `runAssembly` on the fasta file.