TARGETED COMPUTATIONAL APPROACHES FOR MINING FUNCTIONAL ELEMENTS IN METAGENOMES

Yu-Wei Wu

Submitted to the faculty of the University Graduate School

in partial fulfillment of the requirements

for the degree

Doctor of Philosophy

in the School of Informatics and Computing,

Indiana University

August 2012

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Doctoral committee ____

Yuzhen Ye, Ph.D.

____

Haixu Tang, Ph.D.

____

Mehmet Dalkilic, Ph.D.

____

Jerome Busemeyer, Ph.D.

August 14, 2012

To my wife, who initiated this seemingly unfathomable dream

Acknowledgement

The completion of my degree and the success of this dissertation really depend on the help and support of many people. I am most indebted to my advisor, Prof. Yuzhen Ye, for her help, support, and patience throughout this process. Her guidance and insight make this dissertation possible. I am also grateful for Prof. Mehmet Dalkilic and Prof. Haixu Tang, who introduced me the idea of Bioinformatics when I knew very little about this field in my first year in this school, and my committee member, Prof. Jerome Busemeyer, who introduced me some very useful statistical ideas and helped me through this dissertation. I would also like to thank the professors in Indiana University: Dr. Sun Kim taught me to understand the power of machine learning, which helped me throughout the projects that I have ever conducted; Prof. Matthew Hahn taught me the most fundamental ideas of evolutionary biology; and Prof. Esfandiar Haghverdi and Prof. Guilherme Rocha, who made me understand statistics in their helpful courses.

The members of the metagenomics research group, Quan Zhang and Mingjie Wang, support me throughout the Ph.D. years. I am also thankful to all my colleagues in Bioinformatics program, who created a great laboratory environment for helpful discussion. I want to especially thank Linda Hostetter, who helped me a lot when I need help in the school—your help made my life in Indiana University a lot easier.

Last but not least, I am grateful to the support of my family members. Their support makes everything possible. My special thanks go to my wife, who enlightened my path when I was still seeking my future. I cannot thank enough for your love and support.

Yu-Wei Wu

TARGETED COMPUTATIONAL APPROACHES FOR MINING FUNCTIONAL ELEMENTS IN METAGENOMES

Metagenomics enables the genomic study of uncultured microorganisms by directly extracting the genetic material from microbial communities for sequencing. Fueled by the rapid development of Next Generation Sequencing (NGS) technology, metagenomics research has been revolutionizing the field of microbiology, revealing the taxonomic and functional composition of many microbial communities and their impacts on almost every aspect of life on Earth. Analyzing metagenomes (a metagenome is the collection of genomic sequences of an entire microbial community) is challenging: metagenomic sequences are often extremely short and therefore lack genomic contexts needed for annotating functional elements, while whole-metagenome assemblies are often poor because a metagenomic dataset contains reads from many different species. Novel computational approaches are still needed to get the most out of the metagenomes.

In this dissertation, I first developed a binning algorithm (AbundanceBin) for clustering metagenomic sequences into groups, each containing sequences from species of similar abundances. AbundanceBin provides accurate estimations of the abundances of the species in a microbial community and their genome sizes. Application of AbundanceBin prior to assembly results in better assemblies of metagenomes—an outcome crucial to downstream analyses of metagenomic datasets.

In addition, I designed three targeted computational approaches for assembling and annotating protein coding genes and other functional elements from metagenomic sequences. GeneStitch is an approach for gene assembly by connecting gene fragments scattered in different contigs into longer genes with the guidance of reference genes. I also developed two specialized assembly methods: the targeted-assembly method for assembling CRISPRs (Clustered Regularly Interspersed Short Palindromic Repeats), and the constrained-assembly method for retrieving chromosomal integrons. Applications of these methods to the Human Microbiome Project (HMP) datasets show that human microbiomes are extremely dynamic, reflecting the interactions between community members (including bacteria and viruses).

Contents

List of Figures

List of Tables

1. Introduction

Analyzing metagenomic sequences remains a challenging problem due to the complex nature of metagenomes. Traditional sequence analysis approaches, which are designed specifically for single genome sequencing, may not be suitable for annotating metagenomes, each containing sequences sampled from many different species living in the same microbial community. In this thesis I introduced several different methods to alleviate the difficulty of analyzing metagenomic datasets, leading to improved metagenome assemblies and annotations of functional elements (including protein coding genes, CRISPR systems, and integron systems) for downstream analysis.

1.1 Metagenomics

Metagenomics is a science that aims to study entire collections of microbes living in the same environment. Also known as environmental sequencing or community sequencing, metagenomics is able to compensate the drawback of traditional sequencing procedure, where species needs to be cultured before sequencing while the majority of microbes on Earth are unable to grow in petri dishes [1]. The development of metagenomics therefore enables the study of the elusive species on Earth. The first metagenomics research emerged in 1998 [2], which provides a methodology to analyze the soil microbes by direct sequencing. Since then the microbes of many different environments have been studied using metagenomics approaches, including the acid mine drainage [3], ocean [4, 5], soil [6, 7], the sludge [8], the permafrost [9], and even food (e.g., Korean kimchi) [10].

Besides natural environments, animal and human bodies are also the targets of various metagenomic projects. For example, Turnbaugh et al. observed significant differences in the bacterial composition of the gut microbiomes in lean and obese mouse [11]. They also compared the gut microbiomes in lean and obese human twins and made similar discoveries [12]. To understand more about human microbiomes, a larger scale of sequencing effort has been made in Europe (called MetaHIT project), in which 124 human gut samples were collected and analyzed [13]. The MetaHIT project also led to the discovery of three enterotypes of human gut microbiomes [14]. The Human Microbiome Project (HMP), initiated by the NIH Roadmap, enables the sequencing of the microbial communities in several human body sites (including nasal passages, oral cavities, skin, gastrointestinal tract, and urogenital tract) of the latest collection of individuals so far, in order to find the role of these microbes in human health and diseases [15].

1.2 Next Generation Sequencing (NGS)

The Next Generation Sequencing (NGS) technology [16], such as Roche/454 sequencing [17] or Illumine/Solexa [18], plays a very important role in metagenomics research. Compared to traditional Sanger sequencing technology [19], the NGS technology is able to yield many more reads in far less time. It also brings new opportunities. The 1000 genome project, for example, attempts to sequence genomes from individuals around the world and discover all forms of human DNA polymorphism in different populations [20]. Another example is the Genome 10K project, which aims to obtain whole-genome sequences for 10,000 vertebrate species [21]. The massive sequencing effort can only be achieved through NGS technology, which has sharply reduced the cost of sequencing. The cost per megabyte of Traditional Sanger sequencing technology is 100 times more than the Roche/454 sequencing technology, and the Illumina/Solexa sequencing cost is even lower than Roche/454 [22].

The advantages of NGS technology, however, come with a price: NGS sequencing reads are much shorter than Sanger reads: compared to up to 1000 bps per read for Sanger sequencing, the read length for Roche/454 is 400-500 bps and the length for Illumina/Solexa is ~100 bps [22]. Two problems are caused by the relatively short reads. The first problem is that the functional annotation for short reads is not as effective as longer reads; it has been shown that similarity searches of short reads (100 bps) missed 60% to 85% of NCBI BLAST [23] homologs found by using longer reads [24]. The second problem is that shorter reads are much more difficult to assemble, as *de novo* assemblies constructed from short-read data are highly fragmented [25]. As a result the functional annotation or further analysis of the short reads and fragmented assemblies become much more challenging.

1.3 Metagenome annotation

The massive metagenomic data poses great challenges in many areas involving data management and data mining. Since metagenomic samples are retrieved directly from the environment, a metagenomic dataset usually consists of genomic sequences from many (hundreds or even thousands) species in a particular environment. This makes the analysis of metagenomic datasets very difficult because traditional methods for genome annotation do not work well for a mixture of sequences. Furthermore, metagenomics research usually employs NGS technology, which produces short reads that are difficult to analyze.

New computational tools have been developed to address the challenges raised in metagenomics, and most of them are trying to answer one of the two most important questions related to metagenomics: "who is there" and "what do they do." Briefly, the former question is related to deciphering the species composition of a bacteria community, and the latter question is regarded as understanding the functions that the species play individually, and as a whole. In the following two sub-chapters I will review some of the tools and algorithms developed for metagenomics.

1.3.1 Who is there?

The first question, "who is there," is often asked in most metagenomic projects. 16S rRNA gene profiling, or whole genome shotgun (WGS) sequencing of environmental DNA, can be used to study the species composition and diversity of natural bacterial communities. Species composition is often inferred from the resulted 16S rRNA fragments or shotgun sequences by similarity searches. Similarity searches, however, may only offer limited help in understanding the species composition due to the incomplete collection of sequenced bacteria or archaea: the IMG database [26] collects 2780 bacteria genomes and 107 archaea genomes as of January 2012. On the other hand, composition-based binning tools are not limited by the similarities, but may only work for relatively long sequences. I will briefly review the two classes of computational tools (similarity-search based and composition-based) below.

Similarity-search based tools utilize searches of metagenomic sequences against a database of known genes/proteins, with or without phylogeny (often referred to as the phylotyping of metagenomic sequences). MEGAN [27] is a representative similarity-based phylotyping tool, which applies the lowest common ancestor algorithm to assign sequencing reads to taxa based on BLAST results. Phylogenetic analysis of marker genes, including 16S rRNA genes [28], DNA polymerase genes [29], and the 31 marker genes defined by [30], are also applied to determine taxonomic distribution. By employing the marker genes, MLTreeMap [31] and AMPHORA [32], two phylogeny-based phylotyping tools, are developed to estimate the taxonomic distribution of metagenomes. These similarity-based and phylogeny-based tools suffer from a common limitation: The majority of the microbes are still unknown. As a result the analyses based on previous knowledge are very biased.

Composition-based tools attempt to solve the problem by clustering (binning) the metagenomic sequences into different bins (species) without no (or little) prior knowledge of the species inside the metagenomes. These binning tools usually utilize DNA composition information (such as genome G+C content, dinucleotide or *k*-mer frequencies, and synonymous codon usage) that varies among organisms and is generally characteristic of evolutionary lineages [33]. The tools in this category include TETRA [34], MetaClust [35], CompostBin [36], TACOA [37], MetaCluster [38], and a genomic barcode-based method [39] . These tools usually use DNA compositions (such as 4-mer or 6-mer) as a signature of the species. Most composition-based tools achieve a reasonable performance only for long reads (at least 800 bps). This length limitation will be difficult to break because of the local variation of DNA composition [33]. MetaCluster, which employs a different distance metric (Modified Chebychev Distance) to reduce the local variations for 4-mers, is able to bin reads of 500 bps; however reads with 50-150 bps are still out of reach [38].

1.3.2 What do they do?

The second question, "what do they do," is asked because we want to know the roles that the microbes play in the environment, which is usually achieved by assembling the sequencing reads and analyzing the assembled contigs. The most common *de novo* methods to assembling single-genome and metagenome datasets are based on de Bruijn graph [40]. However such assemblers, including EULER [40], Velvet [41], Abyss [42], and SOAPdenovo [43] were all designed for single-genome assembly. When applied to metagenomic datasets, these assemblers often result poor assemblies with short contigs. One of the most important reasons to prevent the assemblers from producing long contigs is the existence of polymorphism of common genomic regions shared by different genomes [44]. Such polymorphisms force the *de novo* assemblers to form new nodes in the de Bruijn graph to represent the differences. As a result, genes are usually fragmented into several contigs, each representing only a part of the genes. Even though specialized methods are proposed to better assemble metagenomic datasets, such as Genovo [45], MAP (Metagenomics Assembly Program) [46], or Meta-IDBA [44], the assembly results are still far from perfect for functional analysis.

Specialized methods are proposed to get functional elements in metagenomes. For example, Ye and Tang [47] developed an ORFome assembly approach, which improved the assembly of genes from metagenomic sequences by isolating reads with predicted ORFs and assemble them. This clearly demonstrates that specialized methods are required for getting specific functional elements (ORFs in this case) in metagenomic samples.

1.4 Overview of proposed methods for metagenomics analysis

In this thesis research, I developed several methods for improving the annotations of functional elements from metagenomics datasets. Firstly, I attempt to improve the whole genome assembly by binning the metagenomic datasets according to the species abundances before the assembly process. The species abundance differences could reduce the effectiveness of assembler since the assembler cannot distinguish whether a contig with low coverage represents a region from a rare species or it is caused by sequencing errors [44]. So if we cluster the species with similar abundance levels together and then assemble each bin separately, we can in principle improve the assembly results. The most challenging part for binning the species is that the species abundance levels for metagenomic datasets are usually unknown, which means that our binning algorithm needs to be un-supervised. I developed an algorithm, AbundanceBin, which is based on an Expectation Maximization (EM) algorithm, to solve this problem by gradually improving guesses of species abundances and the clustering of the reads. AbundanceBin is also able to approximate the species abundance levels and genome sizes, which allow us to take a glimpse of the species composition in the environment. By assembling the bins clustered by AbundanceBin separately and comparing the assembly results before and after applying AbundanceBin, I discovered that application of AbundanceBin prior to assembly improves the assembly of metagenomes.

To achieve better annotation of three types of functional elements, I developed three different targeted computational approaches, each for one type of the functional elements. Firstly I developed a novel method, GeneStitch, to assembling genes, the most important functional elements. GeneStitch is able to traverse the de Bruijn assemblies of a metagenomic dataset, guided by homologous genes using a network matching algorithm, to connect gene fragments scattered in different contigs, and form longer genes. This approach optimizes the utilization of de Bruijn graph representation for a metagenomic dataset by chaining contigs using the information from homologous genes. The idea of "gene-boosted assembly" was firstly employed by [48] for similar purpose (improving the assembly of genes) but using a different methodology (by recruiting reads using similarity search to close the gaps of the assembly). This gene-boosted assembly approach, however, requires that the two species to be very similar (say, 99% identity); but such species may not always be available for assembling a dataset. GeneStitch allows us to use genes of more distantly related species (i.e., species of the same genus or a higher taxonomic level) to improve the assembly of genes in the target (meta-)genome. Furthermore, GeneStitch can be applied to datasets with a mixture of species, improving gene assembly for metagenomic sequences.

I further designed two targeted computational methods for the discovery of two specific types of functional elements: CRISPRs and integrons. CRISPRs (Clustered regularly interspersed short palindromic repeats) together with cas genes are immunity systems of bacteria against viruses and plasmids. CRISPR/Cas systems are found in most archaeal (~90%) and some bacterial (~40%) genomes [49-51], and the CRISPR arrays consist of 24–47 bp direct repeats, separated by unique sequences (spacers) that are acquired from viral or plasmid genomes [52]. CRISPR/Cas defense pathways involve several steps, including integration of viral or plasmid DNA-derived spacers into the CRISPR array, expression of short crRNAs consisting of unique single repeat-spacer units, and interference with invading foreign genomes at both the DNA and RNA levels. By using a novel targeted assembly approach, which employs the uniqueness of direct repeat consensus sequence, I am able to isolate known CRISPRs as well as novel ones from human microbiome samples.

Integrons are genetic elements that acquire and excise gene cassettes from their locus via site-specific recombination. An integron consists of a site-specific tyrosine recombinase (*intI*) gene, a primary recombination site *attI* immediately adjacent to the *intI* gene, and an array of captured gene cassettes encoding accessory functions [53]. Gene cassettes are the minimal units that can be mobilized by the integrase, with each cassette containing one or a very small number of genes and are separated by the recombination site *attC*. There are two types of integrons: chromosomal integrons found in chromosomes and mobile (resistance) integrons found on plasmids. In this research, I focus on chromosomal integrons. Compared to mobile integrons, which often carry only a few antibiotic resistance genes, chromosomal integrons usually carry far more genes of very diverse functions. A novel method, the constrained assembly approach, is developed for the discovery of integron gene cassettes from metagenomic sequences. Application of the constrained assembly approach to the human microbiomes revealed a rich pool of integron gene cassettes associated with the *Treponema denticola* species (an oral spirochete implicated in periodontal disease).

Note two different approaches are devised for discovering CRISPR (the targeted assembly approach) and integron gene cassettes (the constrained assembly approach), considering the difference of the structures of these two systems. As the spacers in CRISPR arrays are significant shorter than NGS reads, we could easily assemble CRISPR arrays using targeted assembly alone, by first collecting reads containing repeats and then assembling the reads using optimized parameters. By contrast, integron spacers (gene cassettes) contain 1–3 genes between the *attC* sites. The lengths of the integron spacers make it very difficult for assemblers to assemble the gene cassettes using the targeted assembly. Constrained assembly is proposed to overcome this limitation, and allows the assembly and characterization of integron gene cassettes. Both applications (the identification of the CRISPR arrays using the targeted assembly approach, and the identification of gene cassettes) demonstrate the importance of directed computational approaches for studies of important functional elements—which are poorly analyzed using generalized computational approaches (such as whole-metagenome assembly)—and that they are essential for the analysis of metagenomic sequences.

2. AbundanceBin: an abundance-based binning algorithm

The abundance differences of bacteria species in metagenomic samples have a large impact on the assembly results: high coverage regions can represent repeats or simply are sampled from genomes of the highly abundant species. We propose to solve this problem by binning the genomic sequences in a metagenome based on the species abundance levels prior to the assembly process. This method, AbundanceBin, attempts to classify the sequences based on the abundance information in an un-supervised manner, given that the species and their compositions in any metagenome are usually unknown beforehand. Since AbundanceBin resolves the abundance level differences of metagenomic datasets, which is one of the main causes for poor assembly of metagenome, we expect that the classification of metagenomic sequences into bins of similar abundances will improve the assembly results. The manuscript of this algorithm was written along with Yuzhen Ye and was published in [54] and [55].

2.1 Rationale

In chapter one I reviewed several binning methods, most of which are based on composition information. However, these methods do not work very well on metagenomic datasets with species abundance level differences—the abundance level differences are very commonly seen in metagenomics (for example, the Acid Mine Drainage project [3] found two dominant species, accompanied by several other rarer species in that environment), and the difference in abundances may affect the classification results for DNA-composition based methods. For example, a weighted PCA was adopted instead of a standard PCA in CompostBin, considering that the within-species variance in the more abundant species might be overwhelming, compared to between-species variance [36]. MetaCluster also reported that the binning accuracy decreased when the abundance ratio increased, especially for closer species [38].

2.2 AbundanceBin algorithm

The AbundanceBin algorithm is built upon an extension of the Lander-Waterman model, [56], which was proposed for characterizing the coverage of each nucleotide position of a genome using a Poisson distribution for single genome sequencing projects. We view the sequencing procedure in metagenomic projects as a mixture of *m* Poisson distributions, *m* being the number of species. The problem is to find the mean values λ_1 to λ_{m}, which are the abundance levels of the species, of these Poisson distributions.

2.2.1 Mixed Poisson distribution

AbundanceBin starts by fitting the genome sequencing procedure to a Poisson distribution. In a random shotgun sequencing process for a single genome, the probability that a read starts from a certain position is $N/(G-L+1)$, where N is the number of reads, G is the genome size, and L is the length of reads. $N/(G-L+1) \approx N/G$, given $G \gg L$. Assume x is a read and a l-tuple (consecutive nucleotide with length l) w belongs to x, The number of occurrences of w is the set of reads follows a Poisson distribution with parameter $\lambda = N(L-l+1)/(G-l+1) \approx NL/G$ in a random sampling process with read length l.

We can also use a similar principle to fit the sampling procedure of a metagenome to a mixed Poisson distribution: the number of occurrences *w* in the set of reads follow a Poisson distribution with parameter $\lambda = N(L - l + 1)/(G - l + 1) \approx NL/G$, but G in this case is the total length of the genomic sequences in the metagenome. Moreover the reads in metagenomic datasets are from species with different abundances. If the abundance of species *i* is *n*, the total number of occurrences of any *l*-tuple *w* in the whole set of reads coming from species *i* should follow a Poisson distribution with parameter $\lambda_i = n\lambda$, due to the additivity of Poisson distribution. So the problem of finding the relative abundance levels of different species is transformed to the modeling of mixed Poisson distribution.

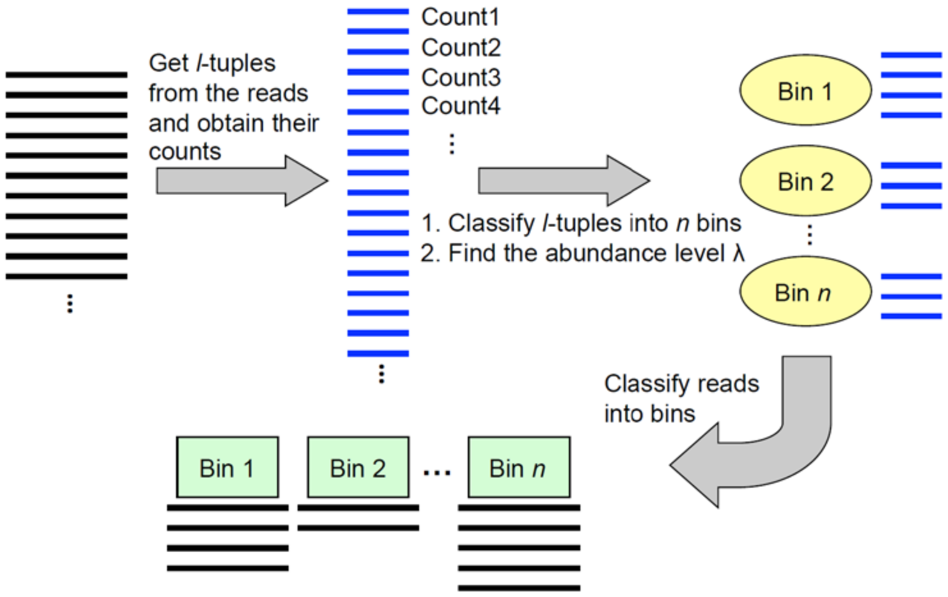

Get *l*-tuples from the reads and obtain their counts
Count1
Count2
Count3
Count4
1. Classify *l*-tuples into *n* bins
2. Find the abundance level λ
Bin 1
Bin 2
Bin *n*
Classify reads into bins
Bin 1
Bin 2
Bin *n*

Figure 1. A schematic illustration of AbundanceBin pipeline.

2.2.2 The binning algorithm

As depicted in Figure 1, the binning algorithm starts by counting l-tuples in all sequencing reads. Denote $x = \{n(w_i)\}$ $(i \in [1, W])$, where $n(w_i)$ is the observed count of tuple i and W is the total number of possible l-tuples. Denote S as the total number of bins. Denote $g = \{g_i\}$ and $\lambda = \{\lambda_i\}$ (for $i \in [1, W]$), where g_i and λ_i are the (collective) genome size and abundance level of bin i, respectively. Denote $\theta = \{S, g, \lambda\}$. Then the goal of the binning algorithm is to optimize the logarithm of the joint probability (likelihood) of obtaining a particular vector of observed l-tuple counts x and the parameter θ, $\log P(x, \theta)$. The hidden variables in the optimization problem are the bin identities of the l-tuples. We use an Expectation-Maximization (EM) algorithm to solve the optimization problem by marginalizing over the hidden variables. The EM steps are as follows.

1. Initialize the total number of bins S, their (collective) genome size g_i, and abundance level λ_i for $i = 1,2, \cdots, S$. We tested various initialization conditions and decide to set the abundance levels to the multiples of 10 (e.g., 1, 10, 20, 30, 40 for five bins) and set the genome sizes to 1,000,000 for all bins.
2. Calculate the probability that the l-tuple w_j $(j = 1,2, \cdots, W$; W is the total number of possible l-tuples) coming from ith species given its count $n(w_j)$. The equation can be derived as follows.

$$\Pr\left(w_j \in s_i \middle| n(w_j)\right) = \frac{Pr\left(n(w_j)|w_j \in s_i\right)Pr\left(w_j \in s_i\right)}{Pr\left(n(w_j)\right)}$$

$$= \frac{Pr\left(n(w_j)|w_j \in s_i\right)Pr\left(w_j \in s_i\right)}{\sum_{m=1}^{S} Pr\left(n(w_j) \in s_m|w_j \in s_m\right)Pr\left(w_j \in s_m\right)}$$

$$= \frac{Pr\left(n(w_j)|w_j \in s_i\right) \cdot \frac{g_i}{G}}{\sum_{m=1}^{S} Pr\left(n(w_j) \in s_m|w_j \in s_m\right) \cdot \frac{g_m}{G}} = \frac{\frac{\lambda_i^{n(w_j)} \cdot e^{-\lambda_i}}{n(w_j)!} \cdot g_i}{\sum_{m=1}^{S} \frac{\lambda_m^{n(w_j)} \cdot e^{-\lambda_m}}{n(w_j)!} \cdot g_m}$$

$$= \frac{g_i}{\sum_{m=1}^{S}\left(\left(\frac{\lambda_m}{\lambda_i}\right)^{n(w_j)} \cdot e^{\lambda_i - \lambda_m} \cdot g_m\right)}$$

$$= \frac{g_i}{\sum_{m=1}^{s} g_m \left(\frac{\lambda_m}{\lambda_i}\right)^{n(w_j)} e^{(\lambda_i - \lambda_m)}}$$

where $Pr\left(w_j \in s_i\right) = \frac{g_i}{G}$ is the prior probability that word *j* is from species *i*, and *G* is the total length of genomic sequences obtained in the metagenomic dataset. The last equation is the result of applying the probability mass function of Poisson distribution into the probability function.

3. Calculate the new values for each g_i and λ_i

$$g_i = \sum_{j=1}^{W} P\left(w_j \in s_i|n(w_j)\right)$$

$$\lambda_i = \frac{\sum_{j=1}^{W} n(w_j)P\left(w_j \in s_i|n(w_j)\right)}{g_i}$$

4. Iterate step 2 and 3 until the parameters converge or the number of runs exceeds a maximum number of runs. The convergence of parameters is defined as

$$\forall\lambda_i\left\{\left|\frac{\lambda_i^{t+1}}{\lambda_i^t}\right|\right\} < 10^{-5} \text{ and } \forall g_i\left\{\left|\frac{g_i^{t+1}}{g_i^t}\right|\right\} < 10^{-5}$$

where $\lambda_i^{(t)}$ and $\lambda_i^{(t)}$ represent the abundance level and genome length of bin *i* at iteration *t* respectively.

Once the EM algorithm converges, we can estimate the probability of a read assigned to a bin by the majority rule based on its *l*-tuple binning results, which is

$$P(r_k \in s_i) = \frac{\prod_{w_j \in r_k} P\left(w_j \in s_i | n(w_j)\right)}{\sum_{s_i \in S} \prod_{w_j \in r_k} P\left(w_j \in s_i | n(w_j)\right)}$$

where r_k is a given read, w_j is the *l*-tuple that belong to r_k, and s_i is any bin. A read will be assigned to the bin with the highest probability among all bins. A read remains unassigned if the highest probability is < 50%.

2.2.3 Lower- and upper-limit of *l*-tuple count

AbundanceBin is able to classify the reads into different bins by using the EM algorithm to extract *l*-tuples and estimate their abundance levels. However, sequencing errors and vector sequences will affect the counting of the l-tuples, which may further have an influence on the accuracy of the binning results. A lower- and upper-limit for *l*-tuple counts is applied as additional parameters when we approximate λ_i and g_i using AbundanceBin. The lower-limit is introduced to deal with sequencing errors, and the upper-limit is introduced to handle *l*-tuples with extremely high counts, such as those from vector sequences or repeats of high copy numbers—this phenomenon has already been utilized for vector sequence removal, as described in [57]. Let the lower-limit be

B_{lower} and the upper-limit be *B_{upper}*. Then the formula for calculating λ_i and g_i is modified to

$$g_i = \sum_{j=1}^{W} P\left(w_j \in s_i | n(w_j)\right), \forall n(w_j) > b_{lower} \wedge n(w_j) < B_{upper}$$

$$\lambda_i = \frac{\sum_{j=1}^{W} n(w_j) P\left(w_j \in s_i | n(w_j)\right)}{g_i}, \forall n(w_j) > b_{lower} \wedge n(w_j) < B_{upper}$$

2.2.4 Detecting the number of bins automatically

The above algorithm, like most un-supervised clustering (binning) algorithms, requires that the number of bins to be assigned before the algorithm can be applied. It may not be realistic, however, since we usually don't know how many species are there in a metagenome. We proposed a recursive binning approach to find the number of bins automatically. This approach is motivated by the observation that reads from genomes with higher abundance levels are better classified than reads from genomes with lower abundance levels. As indicated in Figure 2, the recursive approach starts by binning any dataset into two bins, and further splitting each bin into two bins in a top-down manner. The procedure continues if 1) the predicted abundance values of two bins differ signicantly, i.e., $|\lambda_i - \lambda_j| / \min(\lambda_i, \lambda_j) \geq 1/2$; 2) the predicted genome sizes are larger than a certain threshold (currently set to 400,000, considering that the smallest genomes of living organisms yet found are about 500,000 bps—*Nanoarchaeum equitans* has a genome of 490,885 bps, and *Mycoplasma genitalium* has a genome of 580,073 bps); and 3) the number of reads associated with each bin is larger than a certain threshold proportion (3%) of the total number of reads classified in the parent bin. The recursion stops when the abundance levels predicted by two bins are too close or that the reads assigned to one of the bins are too few—both conditions imply that the bin consists of reads mostly from species of similar abundance levels that they cannot be further separated.

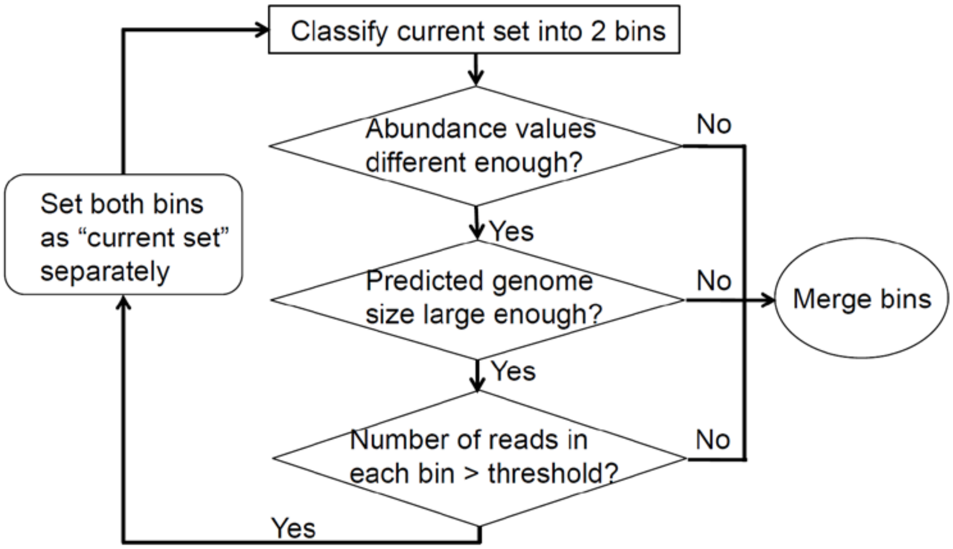

Classify current set into 2 bins
Abundance values different enough?
No
Yes
Predicted genome size large enough?
No
Yes
Number of reads in each bin > threshold?
No
Yes
Merge bins
Set both bins as "current set" separately

Figure 2. The recursive binning approach used to automatically determine the number of bins.

2.2.5 Combination of AbundanceBin and MetaCluster

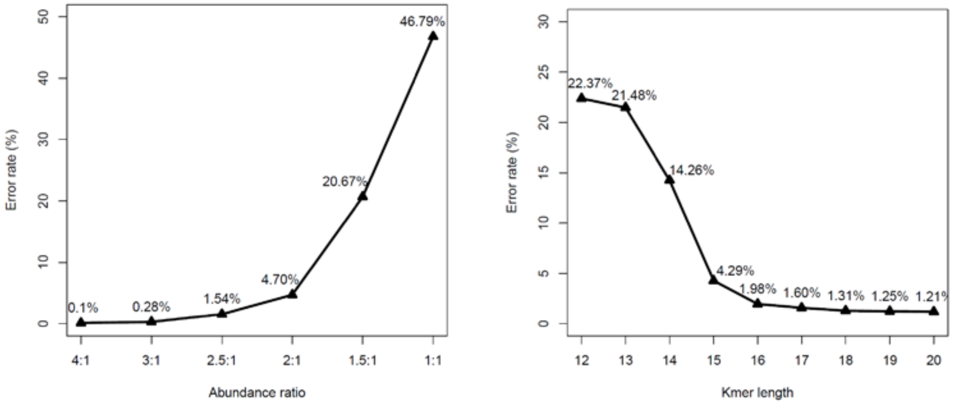
Short reads sampled from species of similar abundances will be classified into the same bin by AbundanceBin. Therefore, these reads can only be further classified into different bins by other binning approaches that utilize species-specific patterns, such as DNA compositions. We combine AbundanceBin and MetaCluster, one of the most recently developed DNA composition-based binning approaches, as follows. Given a metagenomic dataset, AbundanceBin is first used to classify reads into different bins (abundance bins), and then MetaCluster is used to further classify reads in each abundance bin into species bins, each containing reads sampled from a species. We expect that such a two-step approach may achieve higher binning accuracy than using composition-based methods alone, because composition-based methods are less likely to be affected by the different abundance levels of the reads when the reads are classified into different abundance bins in advance. Note in MetaCluster the desired number of bins needs to be defined by prior knowledge, which limits the practical application of our integrated approach. But our proof-of-concept experiments show that AbundanceBin can be used to improve the composition-based binning of reads, especially when the reads are short.

2.3 Results and evaluations of AbundanceBin

AbundanceBin was tested on several datasets, including simulated and real ones, to evaluate its performances. The simulated datasets were generated using MetaSim software [58] with short and very short sequence lengths (400-75 bps). Sequencing errors were also introduced in some of the datasets for benchmarking. The results show that AbundanceBin gives both accurate classification of reads to different bins and precise estimation of the abundances—as well as the genome sizes—in each bin. Note that since these parameters are usually unknown in real metagenomic datasets, we focus on synthetic datasets for benchmarking. AbundanceBin was also applied to the actual AMD dataset, revealing a relatively clear picture of the complexity of the microbial community in that environment, consistent with the analysis reported in [3].

2.3.1 Tests of abundance differences and the length of *l*-tuples

A series of experiments is conducted to test the abundance ranges of species required for accurate binning of reads. The result is demonstrated in Figure 3. Figure 3(a) shows the binning results for simulated short reads sampled from two genomes (*Mycoplasma genitalium* G37 and *Buchnera aphidicola* str. BP) at abundance ratios, 4:1, 3:1, 2.5:1, 2:1, 1.5:1, and 1:1 (with 50,000 simulated reads of ~400 bases for each setting). The classification error rate is low if the abundance ratio is 2.0 (0.1% and 4.7% for ratio 4:1 and 2:1, respectively), but rises dramatically when the abundance ratio drops to 1.5:1 (the error rate is 20.6% for abundance ratio 1.5:1). The results suggest that the abundance ratio needs to reach at least 2:1 for a good classification by AbundanceBin. In addition, different lengths of *l*-tuples are tested on several test cases, including three two-genome cases (one case with species differ in phylum level and the other two cases with species differ in species level) and one three-genome case with species differ in phylum level. The averaged error rates are shown in Figure 3(b). The results show that when *l* drops to 16, the binning performance dropped significantly. The performance improves gradually when *l* increases to 20. Considering the performance on the tested cases, we choose to use *l* = 20 for the following experiments.



(a) (b)

Figure 3. Benchmark results of abundance differences and *l*-tuple lengths. (a) The classification error rates for classifying reads sampled from two genomes versus their abundance differences, and (b) the error rates for different *l*-tuple lengths. These error rates are averaged from four test cases, including three two-genome cases (one test case with species differ in phylum level and the other two test cases with species differ in species level) and one three-genome case.

2.3.2 Binning results of AbundanceBin on synthetic datasets

The results of several synthetic datasets of short reads are summarized in Table 1. Overall AbundaneBin achieves very low error rates for two genomes even under cases that the sequencing reads are very short (75 bps) or that there are errors in the reads (simulated dataset A, C, and E in Table 1). The estimation of genome sizes and genome abundance levels are also very accurate. On the other hand the error rates for binning three genomes are slightly higher (simulated dataset B, D, and F in Table 1). We observe that most of the errors occur in the least abundant bin; but most reads from species with higher abundance levels are correctly classified. The reason may be that reads sampled from higher abundant species fit better to the mixed Poisson distribution than those with lower abundant species—species with lower abundance levels are very difficult to deal with, since the reads sampled from such species are easily diluted among all reads. But AbundanceBin is still able to classify the reads from species of higher abundances correctly for all the tested synthetic metagenomic datasets, including one with reads sampled from 6 different genomes (see Table 2). Note that we also list the normalized error rate for comparison purpose, which is proposed in [36] to estimate the error rate for each bin separately. But we argue that the normalized error rate may not be suitable for datasets with abundance differences since the species with lower abundance levels is very difficult to bin well.

I would like to emphasize here that AbundanceBin can bin reads as short as 75 bases with reasonable classification error rates, as shown in Table 1. As I discussed in previous chapter, binning of very short reads, such as 75 bases, is extremely difficult and cannot be achieved by any of the existing composition based binning approaches, due to the substantial variation in DNA composition within a single genome. AbundanceBin will also give an estimation of the genome size for each bin. As shown in Table 1, for most of the tested cases, the estimated genome sizes are very close to the real ones. Note that AbundanceBin will classify reads from different species of similar abundances into a single bin. In this case, the predicted genome size for that bin is actually the sum of the genome sizes of the species classified into that bin.

Table 1. Tests of AbundanceBin on synthetic metagenomic datasets (A-D without sequencing errors, and E-F with sequencing errors[a])

ID	Spe[b]	Len[c]	Total	Bin	Abundance		Genome size		Error rate
			reads		Real	Predicted	Real	Predicted	(%)
A	2	400 bp	50,000	1	27.23	26.27	580,076	570,859	0.10
				2	6.83	6.49	615,980	614,605	(0.20[d])
B	3	400 bp	50,000	1	24.64	23.78	580,076	568,549	3.10
				2	6.13	6.02	615,980	517,110	(6.64)
				3	1.80	2.39	1,072,950	941,425	
C	2	75 bp	200,000	1	20.47	15.66	580,076	562,584	0.64
				2	5.08	3.92	615,980	608,401	(1.07)
D	3	75 bp	200,000	1	27.60	20.93	580,076	565,859	6.18
				2	6.93	5.99	615,980	368,836	(11.74)
				3	2.07	2.43	1,072,950	1,100,309	
E	2	297 bp	50,000	1	20.21	11.63	580,076	521,168	1.12
				2	5.07	3.01	615,980	945,435	(0.99)
F	3	297 bp	150,000	1	55.48	30.58	580,076	559,395	8.20
				2	13.98	9.60	615,980	341,290	(11.41)
				3	3.50	2.72	1,072,950	3,064,199	

[a]: The average sequencing error rate introduced is 3%, higher than the error rate of recent 454 machines (e.g., the accuracy rate reported in [59] is 99.5%). A 3% sequencing error can reduce the *l*-tuple counts by about half (i.e., about $1 - 0.97^{20} = 0.46$ of expected 20-mers without sequencing errors), which makes accurate estimation of abundance and genome size difficult. [b]: The number of species used in simulating each metagenomic dataset. The genomes used in these tests are *Mycoplasma genitalium* G37, *Buchnera aphidicola* str. BP, and *Chlamydia muridarum* Nigg. The first two genomes are used for the 2 species cases. [c]: The average length of the simulated reads. [d]: Normalized error rates, which calculates the error rate for each bin separately and then take an average of all error rates.

AbundanceBin also works well on binning closely related species (closely related species often have similar genomes, and therefore it is often very difficult to separate reads sampled from closely related species). For the synthetic metagenomic datasets we tested, most reads from species that differ at only the species level can still be classified into correct bins with very low error rates. For examples, for two datasets, the error rates for binning with AbundanceBin are 0.96% and 0.68% for the dataset simulated from the genomes of *Corynebacterium efficiens* YS-314 and *Corynebacterium glutamicum* ATCC 13032, and the dataset simulated from the genomes of *Helicobacter hepaticus* ATCC 51449 and *Helicobacter pylori* 26695 (both sets of genomes only differ at the species level), respectively. These results demonstrate the ability of AbundanceBin to separate short reads from closely related species, even if the species are of the same genus. The only limitation is that AbundanceBin cannot separate reads from two different strains of the same species, but the separation of reads from different strains still remains to be a very difficult scientific problem, and to the best of my knowledge no effective algorithm exists for this task.

2.3.3 Binning results of AbundanceBin on datasets with sequencing errors

As mentioned in Methods, AbundanceBin can be configured to ignore *l*-tuples that only appear once to deal with sequencing errors, considering that those *l*-tuples are likely to be contributed by reads with sequencing errors and that the chance of having reads with sequencing errors at the same position will be extremely low. This may exclude some genuine *l*-tuples, but test results reveal that AbundanceBin achieves even better performance if all *l*-tuples of count 1 are discarded for classifying reads with sequencing errors (data not shown). AbundanceBin achieves slightly worse classification of reads when reads contain sequencing errors, as compared to the classification of simulated reads without sequencing errors (see cases E and F in Table 1). This is expected, given that many spurious *l*-tuples are generated with a 454 sequencing error model. For example, 12,901,691 20-tuples can be found in a dataset of simulated reads from two genomes with sequencing errors (case E in Table 1), 5 times more than the case without error models (2,370,720).

2.3.4 Estimation of the bin numbers

The recursive approach that we developed was used to determine the bin numbers automatically. Evaluations reveal that overall the performances of the recursive binning approach are comparable to the cases with pre-defined bin numbers for test cases from two to six genomes, as shown in Table 2. Overall the performances of the recursive binning approach are comparable to the cases with predefined bin numbers. Figure 4 depicts the recursive binning results of the classification for one of the synthetic metagenomic datasets (which has reads sampled from 6 genomes) into 6 bins of different abundances (with classification error rate = 3.73%), starting with a bin that includes all the reads and ending with 6 bins, each having reads correctly assigned to them. It is interesting that the recursive binning approach achieves even better performance for some cases. A simple explanation to this observation is that the recursive binning strategy may create bigger abundance differences, especially at the beginning of the binning process, and AbundanceBin works better at separating reads from species with greater abundance differences (see Figure 3(a)). We note again that the high abundant bins are classified relatively well. The majority of errors occur in low abundant bins.

Table 2. Comparison of binning performance using the recursive binning approach ("Recursive") versus the binning performance when the total number of bins is given ("Predefined")

Test cases	Error rate (normalized error rate)	
	Predefined	Recursive
3 genomes (no error model; 400 bp)	3.10% (6.64%)	3.24% (7.47%)
3 genomes (no error model; 75 bp)	6.18% (11.74%)	4.84% (9.31%)
3 genomes (454 error model, ~3% error rate; 297 bp)	8.21% (11.41%)	2.29% (4.21%)
4 genomes (no error model; 400bp)	1.12% (5.16%)	2.96% (6.96%)
6 genomes (no error model; 400bp)	2.50% (9.23%)	3.73% (13.07%)

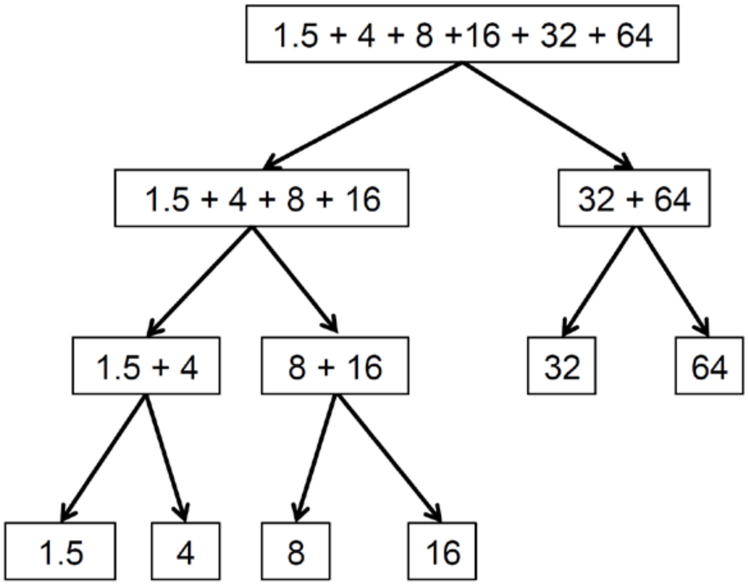


Figure 4. The recursive binning of a read dataset into 6 bins of different abundances. Each box represents a bin with the numbers indicating the abundance of the reads classified to that bin; e.g., the bin on the top has all the reads, which will be divided into two bins, one with reads of abundances 1.5, 4, 8 and 64, and the other bin with reads of abundances 32 and 64.

2.3.5 Binning of Acid Mine Drainage (AMD) dataset

AbundanceBin was also tested on a simulated and real Acid Mine Drainage (AMD) metagenomic dataset. The AMD microbial community was reported to consist of two species of high abundance and three other less abundant species [3]. With the difference of two abundance levels in this environment, it is expected that the algorithm could classify the AMD dataset into two bins.

We first applied AbundanceBin to a synthetic AMD dataset, which we have correct answers to compare with. The synthetic AMD dataset contains 150,000 reads from five genomes, with abundances 4:4:1:1:1. The length of reads is 400bp in average. The recursive binning approach automatically classified the reads into two bins with an error rate of 1.03% (see Figure 5(a)). Note here that each bin has reads sampled from multiple species. A read is considered to be classified correctly if it is classified into the bin of the correct abundance. The binning accuracy dropped only slightly (with an error rate of 2.25%) for the synthetic AMD dataset when sequencing errors are introduced into the dataset.

Next we applied AbundanceBin to reads from the actual AMD dataset (downloaded from NCBI trace archive; 13696_environmental_sequence.007). It successfully classified these reads into exactly two bins (one of high abundance and one of low abundance) using the recursive binning approach (see Figure 5(b)). Note the reads in this dataset have vector sequences, which result in a very small number of *l*-tuples of extremely high abundance (the highest count is 50,720). Two approaches were employed to avoid the influences of the vector sequences: 1) we used the Figaro software package[57] to trim the vector sequences, and 2) we set an upper-limit for the count of all *l*-tuples, ignoring *l*-tuples with counts larger than the upper-limit (200 by default). We also downloaded the sequences of 5 scaffolds of the 5 partial genomes assembled from the AMD dataset, so that we can estimate the classification accuracy of AbundanceBin. The classification error rate of the AMD sequences is ~14.38%. Note this error rate only gives us a rough estimation of the classification accuracy, since only 58% of the AMD reads can be mapped back to the assembled scaffolds based on similarity searches by BLAST—we mapped a read to a scaffold if the read matches the scaffold with BLAST E-value cutoff set to 1e-50, sequence similarity greater than or equal to 95%, and a matched length of at least 70% of the read length. We emphasize that AbundanceBin achieved a much better classification (with an error rate of 1.03%) for the synthetic AMD reads, for which we have correct answers to compare with.

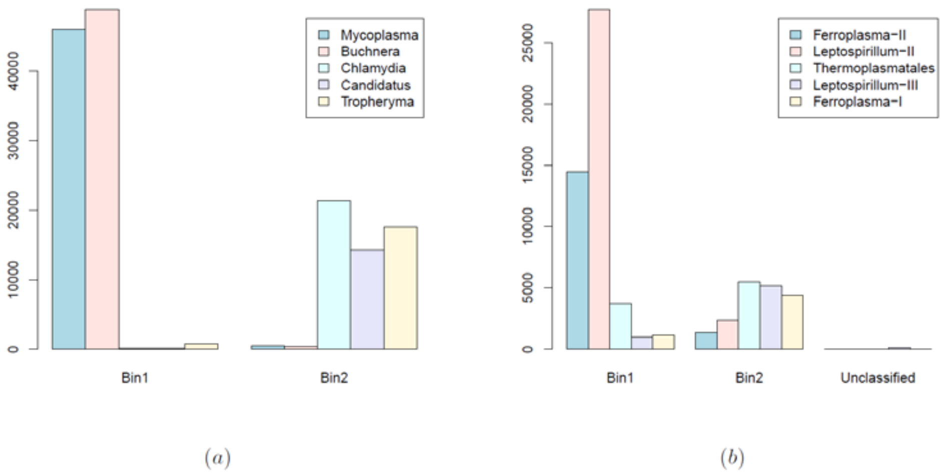


Figure 5. The binning results for a simulated (a), and the actual (b) AMD datasets. The histogram shows the total number of reads from different genomes classified to each bin.

2.3.6 Combination of AbundanceBin and composition-based binning approaches

AbundanceBin can achieve accurate binning of very short metagenomic reads by utilizing abundance differences of the source species of the reads as shown above. However, it cannot be used to separate reads sampled from species of similar abundances. On the other hand, the performance of composition-based binning approaches drops for binning reads with abundance differences. We combine AbundanceBin (an abundance-based binning approach) and MetaCluster (a composition-based binning approach) so that reads of different abundances can first be separated and then reads of similar abundances can be further classified. We apply this methodology to both the synthetic and real AMD datasets. As shown in Figure 6, the classification results of this combined approach are better than those of MetaCluster. For the synthetic AMD dataset with 400bp reads, the error rate of the combined approach is 4.72%, much lower than 26.82% by using MetaCluster alone. Similar trend also exists for the real AMD dataset: the error rate of the combined approach (21.76%) is lower than that of MetaCluster (51.15%).

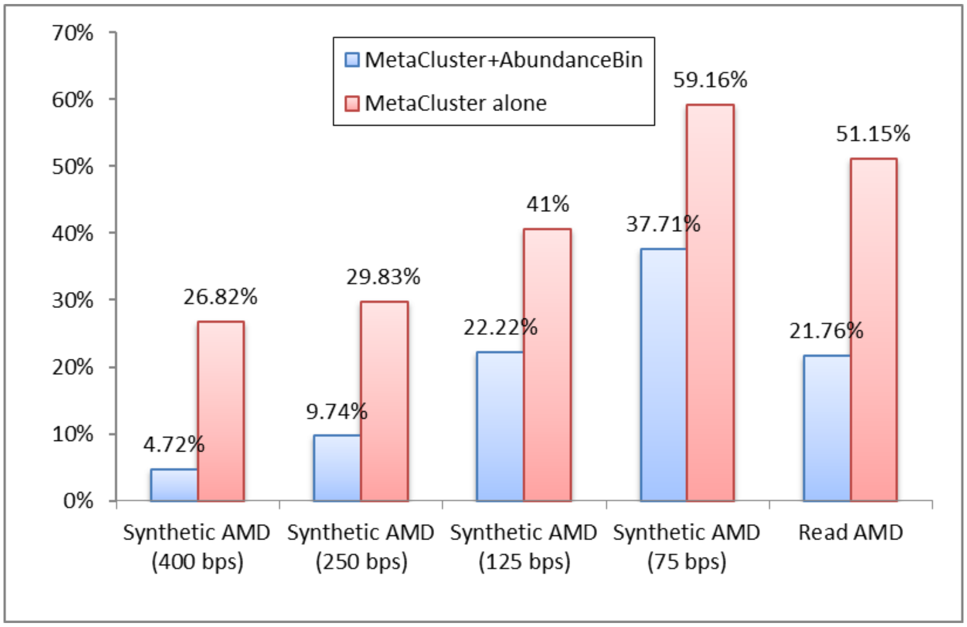


Figure 6. Comparison of error rates of applying AbundanceBin and MetaCluster and applying MetaCluster alone. The datasets include four synthetic AMD datasets and the real AMD dataset.

Current composition-based approaches cannot classify very short reads. To test to what extent AbundantBin can help composition-based methods for binning, we simulated AMD datasets with different read lengths ranging from 75 bp to 400 bp. The binning results are demonstrated in Figure 6. The classification error rates for the combined approach in all test cases are significantly better than MetaCluster. The majority of errors are caused by the inability of MetaCluster to separate very short sequences (due to the local variation of DNA composition patterns). Overall these results demonstrate that a better binning can be achieved to separate metagenomic reads by combining orthogonal information, the abundance differences of the source species and their different composition patterns.

2.3.7 Assembly after binning

Finally, to test whether AbundanceBin helps assembly, we assembled the reads in each bin separately and put the assembly results together using SOAPdenovo [43]. The assembly results for all bins in one dataset are merged together in order to compare them with the assembly without applying AbundanceBin. We use three different metrics (N50, average contig length, and maximum contig length) to compare the assembly results of the simulated and real AMD datasets. The result is shown in Table 3. We found that the N50 and the average contig length are increased for both simulated and real AMD datasets: the N50 increases from 18,286 to 19,567, and the average contig length increases from 2,078 to 2,193. On the other hand the maximum contig length increases in the simulated AMD dataset but decreases in the real AMD dataset. These results suggest that AbundanceBin could in principle improve the assembly as long as most reads are classified correctly. The decrease of the maximum lengths of the real AMD dataset may be caused by misclassification of reads contributing to the longest contig in real metagenomic dataset, which resulted in a break of the longest contig, as indicated in

Figure 5(B); but we still see improvements of N50 and average contig length, indicating that in average the contigs assembled after binning are longer and better. More comprehensive tests and probably adjustments of the parameters involved in the AbundanceBin algorithm and the following assembly of reads in each bin are needed to improve the utilization of AbundanceBin in improving metagenome-assembly.

Table 3. Comparison of assembly results before and after applying AbundanceBin.

Dataset	Before or after AbundanceBin	N50	Average contig len	Max contig len
Simulated AMD dataset	Before	18,286	2,078	177,320
	After	19,567	2,193	373,400
Real AMD dataset	Before	884	411	35,797
	After	897	449	19,818

2.4 Discussion

We have shown that the abundance-based algorithm for binning has the ability to classify short reads from species with different abundances. Our approach has two unique features. First, our method is "unsupervised" (i.e., it doesn't require any prior knowledge for the binning). Second, our method is especially suitable for short reads, as long as the length of reads exceeds the length of the *l*-tuple (e.g., 20). AbundanceBin can in principle be applied to any metagenomic sequences acquired by current NGS, without human interpretation.

Since the initialization conditions of the EM algorithm could have an impact on the convergence, various initialization conditions were tested and we finally decide to set the abundance levels to the multiples of 10 and the genome sizes to 1,000,000 for all species. The advantage of this setting is that the abundance differences are big enough so that each bin will converge to the correct direction. This setting works well for all datasets that we tested, including synthetic datasets and real datasets.

We implemented a simple strategy—excluding *l*-tuples that are counted only once from the abundance estimation—to handle sequencing errors. Tests have showed that AbundanceBin achieved better classification if all *l*-tuples of single count are discarded for the test cases that contain sequencing errors. One potential problem of discarding *l*-tuples of low counts is that some genuine *l*-tuples will be discarded as well, which results in a lower abundance estimation and a worse prediction of genome sizes, especially for the species with low abundance, as shown in Table 1. But we argue that AbundanceBin can still capture the relative abundances of different bins correctly, which is more important than the absolute values. Another potential problem is that reads from low abundant genomes may not be classified when sequencing errors are introduced in the reads. For example, the number of unclassified reads in a two-genome case (metagenomic dataset E in Table 1) is 12, and 389 in a three-genome case (metagenomic dataset F in Table 1). All unclassified reads in both cases belong to the least abundant species, indicating that the abundance values greatly affect the predicted results, especially when sequencing errors are present. We expect that both problems will become less problematic as sequencing coverage is increased, which is possible with massive throughput NGS techniques. As for the abundance ratio required for successful classification, we find that the ratio should be at least 2:1 to obtain an acceptable result. The required ratio, of course, is also affected by several other factors, such as the actual abundance level, the average length of reads, and the sequencing error rate. The tests were intentionally conducted on well-classified datasets, which allow us to follow changes in classification error resulting from abundance differences. Still, other factors besides the abundance ratio must also be considered.

AbundanceBin runs fast, and all the tests shown in the paper were completed within an hour (using single CPU on Intel(R) Xeon(R)@2.00GHz) with moderate memory usage. For example, binning of the synthetic metagenomic dataset A (see Table 1) requires 100MB memory and takes less than two minutes; binning of dataset B requires 150MB memory and also takes less than two minutes. Even for larger dataset such as the synthetic AMD dataset, which contains 150,000 reads, the binning process needs only 300MB memory and takes about seven minutes. Therefore AbundanceBin requires only modest amount of memory unless it is dealing with very large datasets.

Since AbundanceBin employs a unique feature—species abundance levels—to achieve binning of reads, it can be used to assist other tools to analyze metagenomic datasets. To demonstrate this usage, we combine the power of AbundanceBin and MetaCluster to separate datasets with species abundance differences. We apply this methodology on the synthetic and the real AMD dataset, and the results are satisfactory: the error rates of this combined approach are much lower than those of MetaCluster for both tests. These results confirm our hypothesis that, by separating the whole dataset into several sub-dataset, each contains reads with similar abundance level, the composition-based approach can be applied to each sub-dataset, without being influenced by the differences in abundance levels. There are several potential strategies for determining the number of bins for MetaCluster. For example, the dataset can be analyzed by using phylogenetic marker genes for assessing the total number of species as in [30]. We can also test different clustering algorithms that can automatically determine the total number of clusters [34-38]. Our tests show that by integrating different information, we may improve binning accuracy. We further apply AbundanceBin to separate reads into bins of different abundances (coverages), prior to the assembly of metagenomic sequences. The results show that we are able to improve the quality of genome assembly, even when the binning error rate is slightly higher (real AMD dataset in Table 3). By achieving higher binning accuracy and combining AbundanceBin with other composition-based binning algorithms, we hope that we can further improve the assembly quality by using this novel approach.

3. GeneStitch: A Network Matching Algorithm to Gene Assembly

Genes are the basic functional units of a genome and therefore a metagenome. Annotations of genes encoded by a metagenome are important for revealing the functional roles that the microorganisms play as a whole community. For example, the genes retrieved from different depths of the ocean show different distribution composition of functionalities, implying that some genes may be specific to only certain depth levels [60]. However, genes in metagenome assemblies are usually too fragmented to be analyzed as a whole. The algorithm, GeneStitch, was developed to get longer genes from metagenomic assemblies. This manuscript is written along with Mina Rho, Thomas Doak, and Yuzhen Ye, and will appear in [61].

3.1 Rationale

The reason that genes in metagenomic assemblies are very fragmented is that most de Bruijn graph-based assemblers usually produce very tangled graph, especially when sequencing errors exist. This greatly impedes the formation of long contigs, because the branches cannot be resolved. Moreover, *k*-mers from different regions or even from different species may be connected together, which further complicates the structure of the de Bruijn graph. As a result, many short contigs will be reported, which are often insufficient for downstream analysis, such as *ab initio* gene prediction in these short contigs [62], or homology searches of the contigs [24]. For instance, the MetaHIT consortium only considered contigs of length > 500 bp, which represented only 42.7% of the sequencing reads [13].

Salzberg and colleagues proposed a gene-boosted assembly approach to improve assembly quality, which used proteins from reference genomes to recruit sequencing reads to fill in the gaps between contigs [48]. Combining this approach with several other strategies, they successfully produced 76 contigs from 8,627,900 33-bp reads obtained from *P. aeruginosa* PAb1, with the largest contig being 512,638 bps. They also demonstrated that most of the genes in a newly sequenced bacterial strain can be assembled using the genome of another strain of the same species as the reference, using gene-boosted assembly. This approach, however, was only applied to single genome assembly problems. Metagenome assembly is more difficult, because of the presence of homologous genes from multiple species in the same community that may behave like repeats for assemblers. Hence, the success of the approach relies on the utilization of a closely related genome (e.g., the genome of the same species but a different strain), which may not be available in metagenomics, which aims to study un-cultured microbial species in natural habitats.

GeneStitch, an algorithm based on a network-matching algorithm, is developed to infer *gene paths* (sequences of contigs), each of which represents a gene or a gene fragment, in the tangled de Bruijn graph resulted from *de novo* assembly of metagenomic reads. Given a reference gene sequence, GeneStitch searches for a path in the de Bruijn graph that is most similar to the given reference gene. Assuming that the gene paths found by GeneStitch consist of reads most likely sampled from a real gene, we can assemble genes in a metagenomic dataset by using known homologous genes as references. When prior knowledge of the species composition and gene contents of the sequenced metagenome is unavailable, we can use as many reference gene sequences as possible (e.g. the entire set of genes from all available microbial genomes) to guide the inference of gene paths.

One challenge of inferring gene paths is the separation of very similar genes in a metagenome. The gene paths inferred from GeneStitch may overlap substantially with each other, because homologous genes will share identical regions. Instead of attempting to separate these individual genes (with the risk of introducing misassemblies), we propose to merge these paths into *gene graphs*, each of which is a subgraph of the de Bruijn graph that contains reads from the same gene family (homologous genes). We argue that such gene graphs may be considered as single units for downstream analysis of metagenomes, for example, for functional predictions by similarity search.

3.2 GeneStitch algorithm

The inference of *gene paths* from a de Bruijn graph can be formulated as a problem of aligning the graph against a set of reference genes, aiming to derive—in the graph—paths of sequence blocks (or contigs) that are most similar to the reference genes; each path represents a gene or a gene fragment that contains shorter gene fragments. Computationally, this problem is equivalent to the network matching algorithm, which is used to find the best alignment between a graph and a sequence, or between two graphs, and has been applied on computational biology, such as the spliced alignment algorithm for gene prediction considers all potential exon predictions [63] or protein sequence alignment considering all potential secondary structure prediction [64]. The network matching problem can be solved efficiently by a dynamic programming algorithm that searches for the set of connected blocks with the highest similarity to the reference sequence, without exploring all possible paths through the blocks (which would be exponential in the number of blocks).

3.2.1 Network matching algorithm for gene assembly

Consider a set of contigs $(C_1, \cdots, C_n)$ and a de Bruijn graph *G*[1], in which each node represents a contig, and a directed edge is connected between two nodes if these two contigs share $k - 1$ nucleotides (k is a pre-defined number, e.g., k = 30). Our goal is to find the optimal local alignment between the contigs (sequence blocks) and a reference sequence $T = t_1 \cdots t_m$, as illustrated in Figure 7.

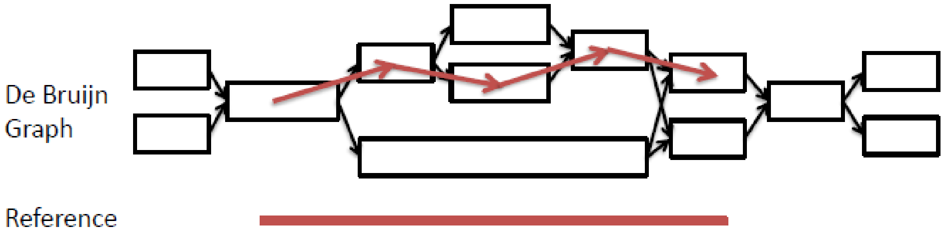


Figure 7. Alignment between a de Bruijn graph and a reference sequence. Blocks in the de Bruijn graph represent nodes, and black arrowheads are the directed edges connecting nodes that overlaps by *k*-1 mers. Typically an assembler based on de Bruijn graph will report the nodes as contigs. Red arrowheads constitute that path of the nodes that best aligns to the reference sequence derived from the network matching algorithm.

[1] Throughout this chapter, we consider the de Bruijn graph in which each simple path (a maximal directed path in the graph, in that all internal vertices have one incoming and one outgoing edge) is collapsed into a single node.

The network matching problem can be solved using a dynamic programming algorithm in polynomial time. Let $S(i,j,k)$ be the optimal alignment score between all possible paths ending at position i of contig k in the input de Bruijn graph and the prefix ending at position j (i.e., $t_1 t_2 \cdots t_j$) of the input reference sequence. For each contig C_k, we denote its first letter as $first(k)$, and its last letter as $last(k)$. A path in the de Bruijn graph can start from any contig and contain at least one contig, but must strictly follow the de Bruijn graph structure, where two contigs C_l and C_k can be connected only if a directed edge goes from C_l to C_k (denoted by $C_l \rightarrow C_k$). Let $E(k) = \{l: C_l \rightarrow C_k\}$ be the set of contigs that are connected to contig k by a directed edge. Our network matching algorithm first computes a dynamic programming matrix to record the optimal alignment scores for1 $\leq i \leq last(k)$, $1 \leq j \leq m$, and $1 \leq k \leq n$ (n is the total number of contigs). $S(i,j,k)$ can be computed recursively as

$$S(i,j,k) = \max \begin{cases} S(i-1,j-1,k) + g(i_k,j) & if\ i \neq 1 \\ \max\limits_{l \in E(k)} S(last(l),j-1,l) + g(i_k,j) & if\ i = 1 \\ I(i,j,k) & \\ D(i,j,k) & \end{cases}$$

where $i = 1$ indicates it is the first nucleotide in contig k, and $g(i_k,j)$ is the scoring function of matching the nucleotide at position *i* in contig *k* and the nucleotide at position *j* of the input reference sequence: $g(i_k,j) = \Delta match$ if the two nucleotides are the same; otherwise $g(i_k,j) = \Delta mismatch$ ($\Delta match$ and $\Delta mismatch$ are two preset parameters). $I(i,j,k)$ and $D(i,j,k)$ are the optimal alignment scores between the paths of the de Bruijn graph (ending at position *i* in contig *k*) and the prefix of input reference sequence (ending at position j), ending with insertion and deletion in the alignment, respectively. The recursive definitions of $I(i,j,k)$ and $D(i,j,k)$ are as follows:

$$I(i,j,k) = \max\begin{cases} S(i-1,j,k) + \Delta g\_open & if\ i \neq 1 \\ I(i-1,j,k) + \Delta g\_ext & if\ i \neq 1 \\ \max\limits_{l \in E(k)} S(last(l),j,l) + \Delta g\_open & if\ i = 1 \\ \max\limits_{l \in E(k)} I(last(l),j,l) + \Delta g\_ext & if\ i = 1 \end{cases}$$

$$D(i,j,k) = max\begin{cases} S(i,j-1,k) + \Delta g\_open \\ I(i,j-1,k) + \Delta g\_ext \end{cases}$$

where Δg_open and Δg_ext are affine penalties [65] for opening and extending gaps, respectively.

The dynamic programming matrix is initialized as

$$\begin{cases} S(i,0,k) = 0 \\ S(0,j,k) = 0 \\ I(i,0,k) = 0 \\ I(0,j,k) = 0 \\ D(i,0,k) = 0 \\ D(0,j,k) = 0 \end{cases}$$

for all i,j, and k.

Once we are done filling in the matrix, we will use a traceback procedure to find the best alignment between the de Bruijn graph and the reference sequence. We first find the maximum score in the dynamic programming matrix and then trace back from that corresponding cell until reach 0 to find the path of the contigs (which we call a gene path) that leads to the best alignment. We also retrieve the gene sequence by concatenating the nucleotide sequences of the contigs in the path. Note that since two nodes connected by an edge in a de Bruijn graph are overlapped by $k-1$ nucleotides, we need to exclude one redundant copy of the $k-1$ nucleotide sequences when retrieving the gene sequence.

We note that GeneStitch does not explicitly consider the cycles that may be found in de Bruijn graphs, in order to employ an efficient dynamic programming algorithm to solve the network matching problem: GeneStitch will traverse (randomly) through one of the cyclic paths (if present). In our tests, GeneStitch rarely encounters such cases, as gene sequences typically don't contain repeats.

3.2.2 Speedup of GeneStitch

The algorithm for GeneStitch described above aligns the reference sequence against the entire de Bruijn graph. The amount of time required for this process is linearly correlated to the number of nodes (representing contigs) in the graph and the lengths of the contigs. Accordingly, we implement two strategies to speed up the network matching procedure, given that a single gene will only span a small portion of the graph.

The first strategy is to employ a similarity-based approach to constrain the search space in the de Bruijn graph for each reference gene sequence. First, we use BLAST to search all nodes (i.e., contigs) of the de Bruijn graph against the reference sequences with a relatively-high E-value cutoff (currently set to 0.1). For each reference sequence, the node with the best alignment score will be used as the starting node to recruit more inbound and outbound nodes with BLAST hits. Considering that short contigs may be missed by the similarity search process [24], we allow the recruiting process to extend an additional *N* layers of inbound and outbound nodes without BLAST hits (*N* is set to 5). This process is repeated until no more nodes can be recruited. The included nodes (and the edges that connect them) —instead of the whole graph—then serve as the input graph for the network matching process.

The second strategy is to exclude intact genes found in the input contigs. We use FragGeneScan [66] to predict fragmented genes as well as intact genes in all contigs, and then remove intact genes (defined as the predicted gene fragments that do not include the first or the last nucleotide of any contig) from the contigs prior to the network matching process, retaining only fragmented genes and intergenic regions adjacent to them. This pre-processing step greatly speeds the network-matching process.

3.2.3 Construction of gene graphs

Gene paths—each representing a (fragmented) gene—inferred from a de Bruijn graph using homologous reference genes by the network matching algorithm described above may overlap with each other. These paths can be merged into a *gene graph* that represents a collection of homologous genes in a compact fashion.

To make sure that we generate gene graphs that consist of only homologous genes, three empirical criteria are applied when finding gene paths in the de Bruijn graph: a) the optimal score of the alignment between the gene path and the reference gene is ≥ 50 (score threshold), b) the identity of the alignment is ≥ 60% (identify threshold), and c) the alignment covers at least 40% of the length of the reference sequence (gene coverage threshold). The identity threshold is set to 60%, since genes may not be very similar at the nucleotide level, especially if the reference genes are obtained from not-so-closely-related species. Two gene paths sharing at least one contig are merged into a gene graph if the reference sequences used to infer the gene paths are highly similar (i.e., with identity ≥ 70%). We will further compare the merged gene graphs with other gene paths or gene graphs and merge them if they contain genes inferred from very similar reference genes. This merging process is performed between any two gene graphs until all pairs of graphs have been checked. Once the merging is completed, we will select—for each gene graph—its composite gene path with the highest network-matching alignment score as its representative sequence.

3.2.4 Extention of the gene graphs

The network matching algorithm and the subsequent merging steps may leave out gene segments from the constructed gene graphs that are not sufficiently similar to the reference sequences. To make gene graphs complete, we will extend each gene graph by recruiting the inbound and outbound nodes of its contigs if they share similarities with the contigs already included in the graph. This process is repeated until no more nodes can be added. The algorithm is given as follows.

```
for all contigs in gene graph P do
  \\Check inbound nodes
  Listed = inbound nodes ∈ P
  Not Listed = inbound nodes ∉ P
  for each node m1 ∈ Not Listed do
    if identity(m1, any node ∈ Listed) > Identity Threshold then
      Add the node into the gene graph
    end if
  end for

  \\Check outbound nodes
  Listed = outbound nodes ∈ P
  Not Listed = outbound nodes ∉ P
  for each node m2 ∈ Not Listed do
    if identity(m2, any node ∈ Listed) > Identity Threshold then
      Add the node into the gene graph
    end if
  end for
end for
```

Currently we set the identity threshold to 70% so that only very similar inbound and outbound contigs will be recruited into the gene graph.

3.2.5 Datasets and tools used in evaluation

We implemented our algorithm in C++ and tested our program (named GeneStitch) on simulated datasets for a single genome and a dataset for an artificial microbial community.

We produced three test datasets of sequencing depths 6X, 13X, and 20X from the *Escherichia coli* str. K-12 substr. MG1655 genome (NC_000913) using Metasim software [58]. We used the 80bp error model downloaded from the Metasim website to simulate Illumina reads of 80bp with a 1% error rate. Genes from the *Escherichia coli* HS (NC_009800), *Escherichia fergusonni* (NC_011740), and *Salmonella enterica* (NC_003198) were used as the references for GeneStitch.

The community dataset comprises sequencing reads obtained from an artificial microbial community with ten mixed lab-cultured species [67]. The main reason we chose this dataset (of 454 sequencing reads) as our test case is that we can directly evaluate the quality of the assembled genes because the genes and genomes of the species in the community are already known. Among the 10 species, nine are either bacterial or archaeal, and one is a eukaryotic species (*Saccharomyces cerevisiae* S288C). We use genes from nine species as the reference gene sets, which are different at the species level (or higher level if species level is not available) compared to the bacteria or archaea species in the mock dataset. Table 4 lists the species we chose. We do not test on the eukaryotic genome because eukaryotic genes contain an intron-exon structure that our method is not currently designed for. To check for misassembly, we map assembled genes against the source genomes, using bwasw, provided by the BWA package [68]. A gene is considered to be misassembled if it cannot be mapped, or maps to two or more locations in the genomes.

Table 4. The list of species contained in the mock dataset, and corresponding species used as references in GeneStitch.

Species in mock dataset	Reference species
NC_002662 *Lactococcus lactis* subsp. lactis Il1403	NC_012984 *Lactobacillus plantarum* JDM1 (genus)[a]
NC_008527 *Lactococcus lactis* subsp. cremoris SK11	NC_014724 Lactobacillus amylovorus GRL 1112 (order)
NC_008525 *Pediococcus pentosaceus* ATCC 25745	NC_008529 *Lactobacillus delbrueckii* subsp. *bulgaricus* ATCC BAA-365 (family)
NC 010999 *Lactobacillus casei* BL23	NC_014106 *Lactobacillus crispatus* ST1 (genus)
NC_008497 *Lactobacillus brevis* ATCC 367	NC_009513 *Lactobacillus reuteri* DSM 20016 (genus)
NC 008700 *Shewanella amazonensis* SB2B	NC_014012 *Shewanella violacea* DSS12 (genus)
NC_008095 *Myxococcus xanthus* DK 1622	NC_011891 *Anaeromyxobacter dehalogenans* 2CP-1 (family)
NC_008578 *Acidothermus cellulolyticus* 11B	NC_014666 *Frankia* sp. EuI1c (order)
NC_002607 *Halobacterium* sp. NRC-1	NC_013967 *Haloferax volcanii* DS2 (family)

[a]: The taxonomic ranks in the parentheses indicate the lowest common taxonomy level shared between the reference species and the species in the mock dataset.

3.3 Evaluations of GeneStitch

3.3.1 GeneStitch improves gene assembly

We first test our algorithm on datasets simulated from only one genome (*E. coli* K-12) to show that reference genes from closely-related (*E. coli* HS and *E. fergusonni*) or more distantly-related species (*S. enterica*) can be used to improve gene assembly. We evaluate the performance of GeneStitch by both *gene coverage*, and the number of *complete genes* assembled. The gene coverage is defined as the average percentage of the annotated genes (in length) that are covered by the assemblies (e.g., gene coverage of 100% means that full-length genes are assembled). An assembled gene is considered complete if it covers at least 90% of the actual gene, sharing at least 98% sequence identity.

The results are summarized in Table 5. Since GeneStitch is designed for assembling fragmented genes, we isolate the fragmented genes from the contigs either from the initial assembly, or after various GeneStitch treatments, and calculate the gene coverage for them (the statistics of all genes are also given). For all datasets, GeneStitch significantly improves the completeness of assembled genes as compared to initial assembly's genes (with higher gene coverage), and the number of complete genes, especially for the datasets with lower sequencing depths (6X or 13X). For example, for the dataset with 13X sequencing depth, SOAPdenovo alone assembled 2320 complete genes, and GeneStitch assembled 1097 more (i.e., a 47% improvement). Improvement is also observed, although less significant, for the dataset with 20X sequencing depth (which can already be assembled fairly well by SOAPdenovo with a gene coverage—for all genes in contigs—of 81%). These results demonstrate the ability of GeneStitch to link fragmented genes together and form longer genes.

Table 5. A summary of the GeneStitch results for *E.coli* K-12 at 6X, 13X, and 20X sequencing depths.

Sequencing depth	Reference	Genes/fragments[a]	Gene coverage[b]	Complete genes[c]	Complete gene ratio[d]	Misassembly rate
6X	-[e]	13947 (14149)[f]	26% (28%)[f]	572	14%	-
	E. coli HS	5343	62%	+461	25%[g]	0.3%
	E. fergusonni	4473	62%	+384	23%[g]	0.5%
	S. enterica	3917	62%	+330	22%[g]	0.2%
13X	-[e]	6642 (9158)[f]	33% (50%)[f]	2320	56%	-
	E. coli HS	4189	77%	+1097	82%[g]	0.2%
	E. fergusonni	3495	77%	+974	79%[g]	0.3%
	S. enterica	3038	77%	+858	77%[g]	0.1%
20X	-[e]	1904 (3491)[f]	45% (81%)[f]	3264	79%	-
	E. coli HS	1628	83%	+448	90%[g]	0.4%
	E. fergusonni	1276	83%	+401	88%[g]	0.3%
	S. enterica	1068	83%	+345	87%[g]	0.6%

[a]: This column specifies the number of gene fragments in assembled contigs (the first row for each section) or the number of genes assembled by GeneStitch. [b]: Gene coverage reflects the completeness of assembled genes; a small value indicates that assembled genes are highly fragmented. [c]: This column lists the assembled genes or genes in contigs (the first row for each section) that are complete or almost complete (at least 90% of the entire length) as compared to the real genes. Additional complete gene numbers assembled by GeneStitch are highlighted by a '+' sign. [d]: This column lists the ratio of completely assembled genes versus all annotated genes in the *E. coli* K-12 genome. [e]: This row lists the assembly results before applying GeneStitch. [f]: The two numbers indicate the statistics of fragmented genes and all genes (within parentheses) in contigs. See text for details. [g]: The ratio is calculated over all complete genes, including the ones assembled by SOAPdenovo and GeneStitch.

Another observation is that the improvement introduced by GeneStitch decreases with the taxonomic distances of the reference species, which is not surprising. Our tests, however, show that even when using distantly related species (e.g, *S. enterica*) as references, GeneStitch improved the quality of gene assembly. Overall these results demonstrate the power of GeneStitch, in which fragmented genes split into different contigs are assembled into longer gene fragments even if we use reference species of different genera (e.g., target species *E.coli* K-12 vs reference *S. enterica*).

We also examined the potential for misassembly in the assembled gene sequences by mapping the assembled genes against the *E. coli* K-12 genome. The proportions of misassembled sequences are very low for all three test datasets, indicating that GeneStitch introduces few misassemblies into single genome assemblies.

3.3.2 GeneStitch successfully identifies genes in a metagenomic dataset

We next tested GeneStitch with the artificial community dataset. Since the sequencing depth of the 454 dataset is not very high (2.86X) and contains a eukaryote organism, we also simulated a dataset with higher depth (9X) that included only the prokaryotic species from the dataset. Results are shown in Figure 8. Similar to the single genome cases, the gene coverage ratio for both the simulated and real metagenomic datasets increases (shown in Figure 8(A)), suggesting that GeneStitch is capable of assembling longer genes from the metagenomes. An intriguing observation is that even though there are fewer genes assembled from the real sequence dataset (8,283 genes) as compared to the simulated dataset (22,331 genes), the gene coverage ratio of the assembled genes in the real dataset is actually higher after treatment with GeneStitch (71% vs 52%).

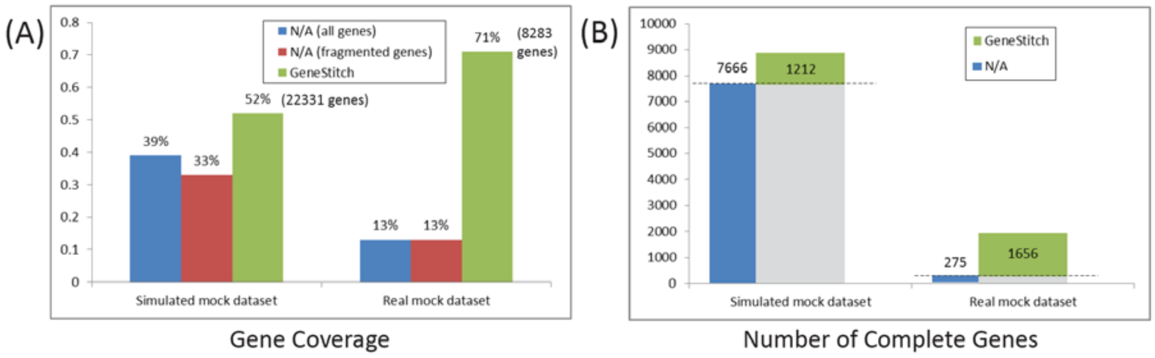


Figure 8. Improvement of gene assembly by GeneStitch for the simulated and real community datasets, as evaluated by gene coverage (A) and the number of complete genes (B).

The number of complete genes, as demonstrated in Figure 8(B), also suggests that GeneStitch has the ability to produce complete genes from metagenomes. Besides the already complete genes in the contigs, GeneStitch is able to build 1,212 and 1,656 more complete genes from gene fragments. From the real dataset, GeneStitch assembled more than five times more complete genes than those in contigs! The reason that the number of complete genes assembled for the simulated dataset is less than that for the real dataset is that many complete genes are already well assembled for the simulated data due to its higher sequencing depth. On the other hand, the genes in the real dataset are mostly fragmented and are then recovered using GeneStitch. Nevertheless, the number of assembled genes for the simulated dataset (22,331 genes) is still higher than the real dataset (8,283), suggesting that higher sequencing depth is still needed for ideal gene assemblies.

The misassembly rates for the genes assembled from the metagenomes are higher than those for single genomes. In total, 1,109 genes (4.97%) and 165 genes (1.99%) are probably misassembled for the simulated and real dataset, respectively. Further analysis reveals that the majority of these genes (832 out of 1,109 genes for simulated dataset and 37 out of 165 genes for real dataset) can be mapped to exactly two homologous genes in the community: for example, an assembled gene may consist of segments from two homologous genes and produce a chimeric sequence. Considering that these cases are sometimes unavoidable for metagenome assembly (and we call them "minor" misassembles), especially when very similar genes from different species exist in the sample (there are two strains of *Lactococcus lactis*, namely *L. lactis* cremoris IL1403 and *L. lactis* cremoris SK11, exist in the mock dataset), the "severely" misassembly rate is only 1.24% and 1.55% for the simulated and real datasets.

Below we present two cases from the real community dataset, to demonstrate how we find the gene graph from the assembled de Bruijn graph.

3.3.3 Example gene graph #1

The first example demonstrates how a gene path can be inferred from a connected component in the de Bruijn graph with 17 nodes. Only one gene annotated as beta glucosidase, YP_812362 from the species *Lactobacillus delbrueckii* subsp. *bulgaricus* ATCC BAA-365, passes the threshold values and is detected in this example. The result is shown in Figure 9: the path with similarity to the reference gene contains seven nodes; no nodes can be further recruited into this connected graph, thus only seven nodes (contigs) covered by the path represent the gene graph, and the sequences in this path constitute the representative gene for this gene graph.

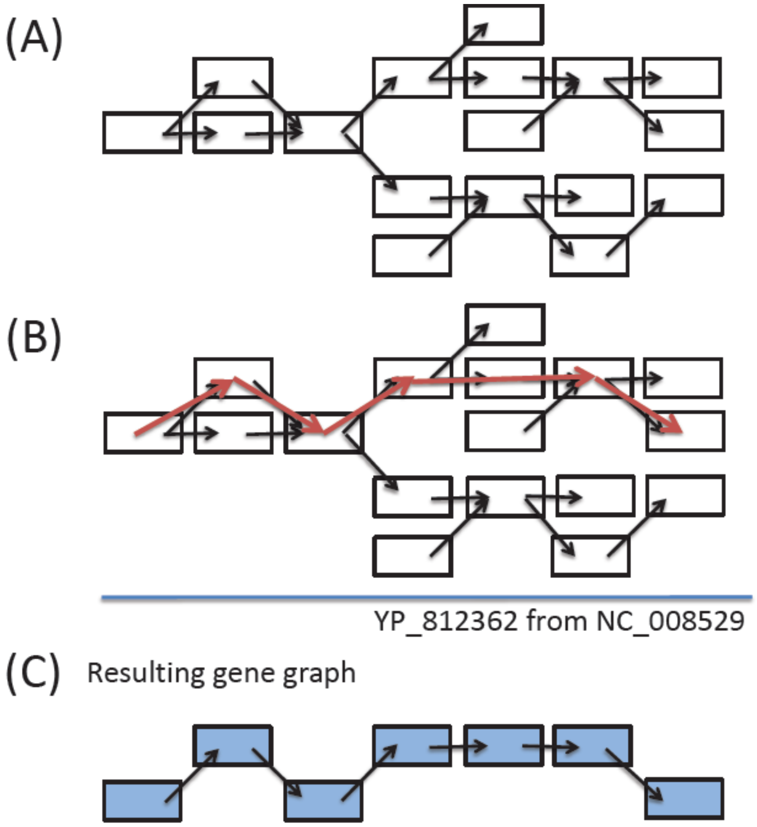


Figure 9. An example demonstrating the inference of a gene path from a connected component in the de Bruijn graph. The reference gene recruited by BLAST in this example is YP_812362. (A) In total, 17 nodes are present in this connected component. (B) The path found by GeneStitch using the reference gene. (C) The gene path.

3.3.4 Example gene graph #2

This example demonstrates how we infer gene graphs by merging paths (or gene graphs). Figure 10(A) shows a connected component of the de Bruijn graph. Two reference genes, YP_003601430 from *Lactobacillus amylovorus* GRL 1112 and YP_004031707 from

Lactobacillus crispatus ST1, can be recruited as reference genes to this graph. The identity between these two genes is 76%. From Figure 10(B) one can observe that the paths are very similar—only one branching node is different. Since the identity of the two reference genes is higher than the threshold (default 70%; see section 3.2.3) and the two graphs are overlapping, these two graphs are merged into one gene graph, as shown in Figure 10(C). The first assembled sequence, which has a higher score value (as well as a higher identity), is selected as the representative gene for this gene graph.

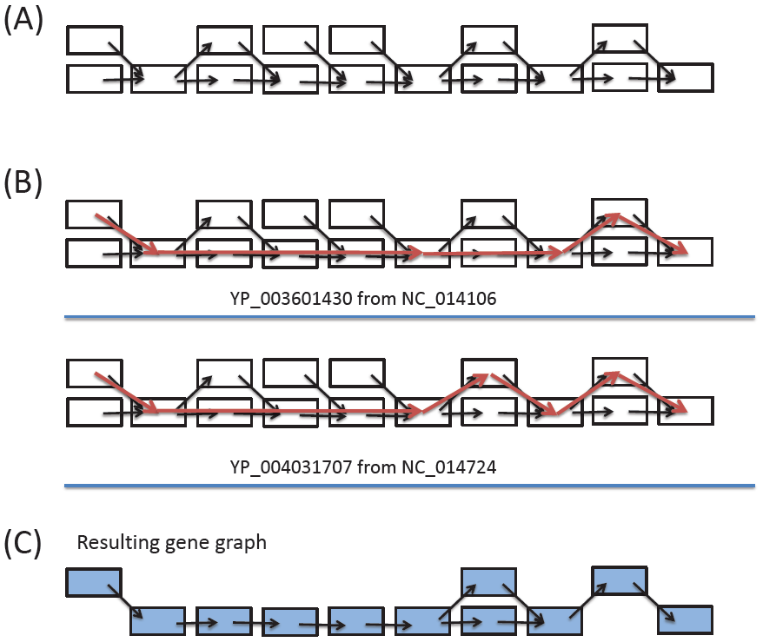


Figure 10. An example demonstrating the construction of a gene graph by merging gene paths. (A) Only 19 nodes are shown in this figure for clarity (the actual component is larger). (B) Two paths are found by GeneStitch, using YP_003601430 and YP_004031707 as the reference genes. (C) The two paths are merged into a gene graph.

3.4 GeneStitchPro: using protein level similarity for matching

The network matching algorithm used in GeneStitch aims to optimize the chaining of contigs by aligning the network of contigs and reference genes at the nucleotide level. Considering that protein sequences are typically more conserved as compared to nucleotide sequences for protein-coding genes among related species, we extended the network matching algorithm to consider the similarity between the network of contigs and the reference gene at protein level, enabling the utilization of more distant homologs as reference genes. After testing the modified algorithm, which is named GeneStitchPro, we found that the algorithm works significantly better than the original GeneStitch, especially when the reference species are only distantly-related to the actual species in the dataset.

3.4.1 GeneStitchPro Algorithm

The workflow of GeneStitchPro is similar to the original GeneStitch. The key difference is that we now match the contigs against amino acid sequences instead of nucleotide sequences. By using the codon table for bacteria (Table 11 of the genetic codes collected in http://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi?mode=c) to translate nucleotide triplets into corresponding amino acids in the mapping process on the fly, we can calculate the optimal alignment score $S(i, j, k)$ between all possible paths ending at position *i* of contig *k* in the input de Bruijn graph and the prefix ending at position *j* (i.e., $t_1 t_2 \cdots t_j$) of the input reference protein sequence shown in Appendix 2. Briefly, GeneStitchPro tries to convert nucleotide triplets into amino acids and then compare the amino acid sequences using pre-defined amino acid substitution matrix. Note that this method is very different to the old school method, which translates all contigs into six-frame amino acid sequences and compare the sequences against the references. The main reason is that the traditional method cannot consider the combination of cross-border contigs, in which one codon may start at one contig and end at another. Similar to GeneStitch, GeneStitchPro utilizes a dynamic programming approach to calculate all combinations in linear time in order to find the nucleotide sequences whose translations are most similar to the reference amino acid sequences.

3.4.2 Evaluation of GeneStitchPro

We compare the GeneStitchPro results, including the one-genome dataset and the mock dataset, against the results of GeneStitch, in order to demonstrate that GeneStitchPro is more effective when the reference sequences are more distantly related. For the one genome case, the *E. coli* K-12 datasets (with 6X, 13X, and 20X sequencing depths) are used again to find the genes. Besides the three reference species (*E. coli* HS, *E. fergusonni*, and *S. enterica*), we added one more reference species, *P. aeruginosa*, which only shares the same class with *E. coli* K-12 in phylogeny, to find the genes. The results clearly indicate that GeneStitchPro works much better than GeneStitch, especially in the cases where *S. enterica* and *P. aeruginosa* are served as reference species. We again evaluate the results in terms of gene coverage and the number of complete genes. The results are shown in Figure 11. One could observe that the gene coverage, as shown in Figure 11(A), shows that GeneStitchPro slightly outperforms GeneStitch, especially when the reference species is distantly related (*S. enterica* and *P. aeruginosa*). On the other hand, GeneStitchPro is apparently much better than GeneStitch at getting complete genes, as shown in Figure 11(B). For example, in the 13X dataset, GeneStitchPro assembled 751 complete genes, which is almost four times more than that from GeneStitch when we use *P. aeruginosa* as reference species. Similar results can also be observed in 6X and 20 X datasets. The results clearly indicate that GeneStitchPro is very powerful in assembling genes using distantly-related species as references.

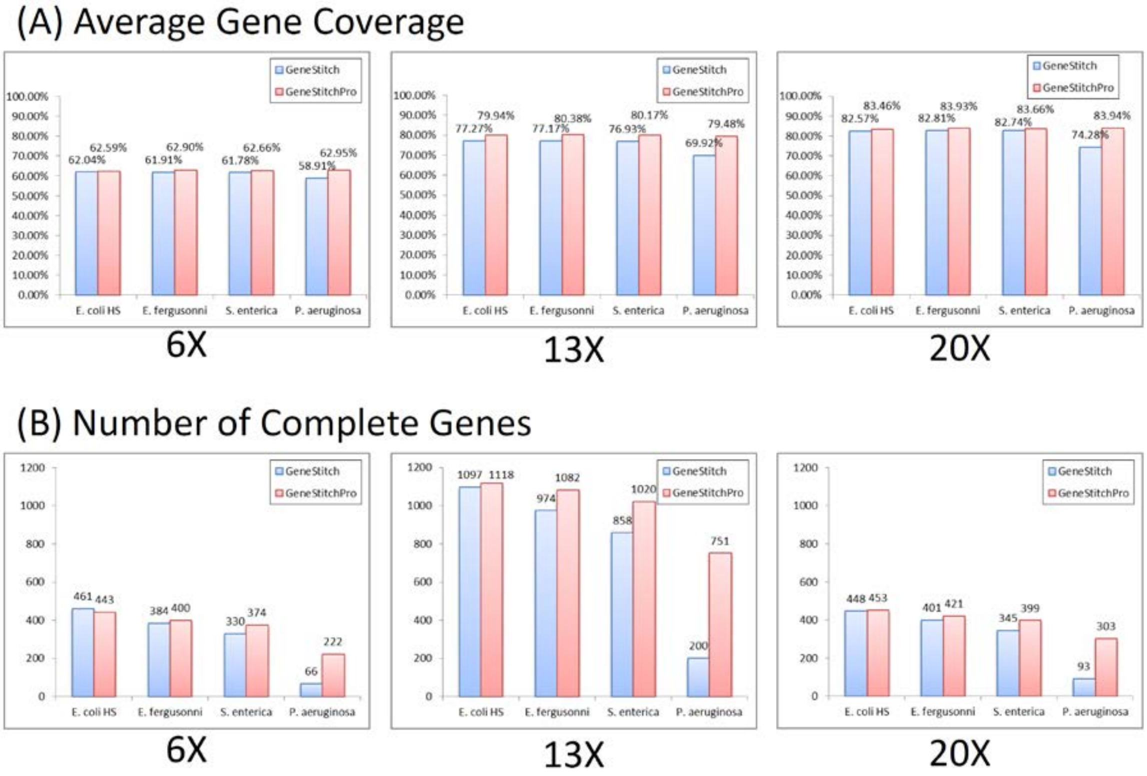


Figure 11. Comparing the performances of GeneStitch and GeneStitchPro. (A) The average gene converage of the three one-genome datasets with 6X, 13X, and 20X sequencing depths. (B) The number of complete genes of the three datasets.

We also tested GeneStitchPro on the simulated and real mock dataset and again found that GeneStitchPro performs better than GeneStitch in both average gene coverage and the number of complete genes. The results are shown in Figure 12. One can observe that GeneStitchPro indeed outperforms GeneStitch in every aspect. For example, GeneStitchPro retrieves 2997 complete genes in simulated mock dataset, which is more than twice as compared to GeneStitch. Similar improvements can also be observed in real mock dataset. These results again exemplify the function of GeneStitchPro, which compares the contigs against amino acid sequences.

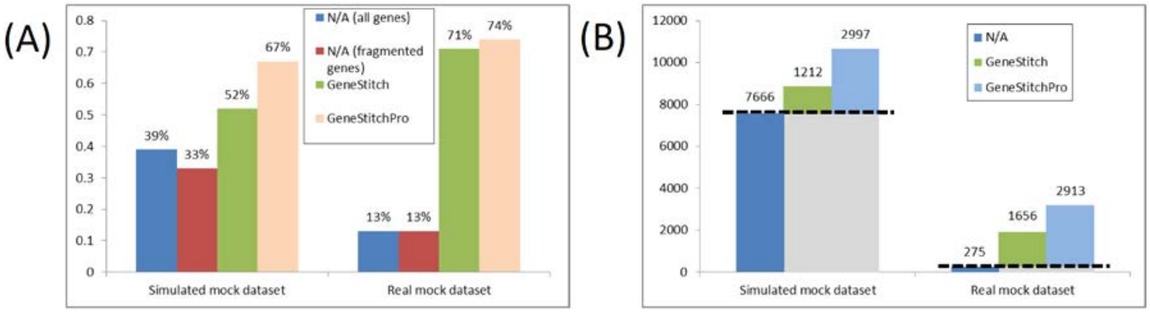


Figure 12. Comparison of the improvements of gene assembly by GeneStitch and GeneStitchPro for the simulated and real community datasets, as evaluated by gene coverage (A) and the number of complete genes (B).

3.5 Discussion

We present GeneStitch, which is based on a network matching algorithm, for inferring gene paths and gene graphs from the tangled de Bruijn graphs that result from assembly of metagenomic sequences. If we have prior knowledge of the taxonomic composition of a metagenomic dataset (e.g., through 16S rRNA gene profiling [69]), or taxonomic analysis using shotgun sequences [70]), we can use genes from the most closely related species available as references for GeneStitch, considering that GeneStitch benefits more by using the most similar gene sequences as the reference. However, in principle, we can use a general dataset of genes (e.g, microbial genes in the NCBI NR (non-redundant) dataset) as reference genes in GeneStitch, if we have no prior knowledge of the taxonomic composition of a metagenomic sample.

For all tests that we performed, the application of GeneStitch greatly improves the assembly of genes, resulting in complete or nearly complete genes. The assembly of complete gene sequences is important because traditional metagenome sequencing projects are largely limited by the length of contigs and scaffolds, and small contigs are often difficult (if not possible) to use for subsequent functional analysis. We believe that our approach will increase the amount of information that can be gleaned from past and future genome and metagenome projects, by providing longer genes for analysis. We note that GeneStitch is able to improve the gene assembly even when only distantly related species are available as references, and when sequence depth is modest. This capability is especially important because sequenced bacterial or archaeal genomes are still limited and very closely-related species (such as different strain of the same species) are not always available. GeneStitch greatly broadens the choice of reference species for gene annotation in metagenomic assemblies.

Our approach can be conceived as a gene predictor that works with de Bruijn graphs for assembly, instead of linear sequences. In this sense, GeneStitch is fundamentally different from current gene predictors including FragGeneScan [66] and GLIMMER [71]. Note that gene paths are fundamentally different from the directed acyclic graphs used to represent exons (as nodes) and their connectivity (the edges) in predictors for eukaryotic genes [72]. We have also proposed a novel concept, the gene graph, to represent a collection of homologous genes in a metagenomic dataset. A gene graph may not include all similar (or homologous) genes in a metagenomic dataset, because we set the identity threshold to a relatively high value (e.g. 70%) in the process of constructing gene graphs. But it is not our goal to build comprehensive gene graphs; instead, we want to assemble metagenomic sequences into separate genes as long as we have strong evidence the assembled genes contain no misassemblies. We note that GeneStitch cannot help with the assembly of novel genes that lack similarity with known genes.

Although the gene graph is used to represent the cases where gene paths overlap with each other—a non-conventional way of representing genes—we argue that gene graphs can be considered as single units for downstream functional analysis of metagenomes. For example, we can attempt to get all real genes from the gene graphs by walking all potential paths in the gene graphs and select those supported by reads. This approach is used by the Trinity assembler to find all spliced isoforms and transcripts of recently duplicated genes from transcriptomes [73]. Another application would be functional prediction: we can search an unknown gene against all gene graphs and determine which gene graph is most similar to this gene, in order to determine its function.

To further improve the performance of gene assembly using more distantly-related reference species, we devised GeneStitchPro, which aligns the contigs against the amino acid sequences since the protein sequences are more conserved than nucleotide sequences. Also based on a modified network matching algorithm, GeneStitchPro is capable of assembling more complete genes. For example, when we use *P. aeruginosa* as reference species, which only shares the same class of the target species, GeneStitchPro is able to assemble more than three times of complete genes as compared to GeneStitch. This and other examples clearly demonstrate the power of using amino acid sequences as references to retrieve more intact genes.

Notably, other strategies have been used to improve metagenome assembly, for examples: by merging assemblies from different assemblers or using the same assembler but with various parameter settings [74]; by recruiting reads to fill in gaps between contigs using tblastn searches against reference genes as in the gene-boosted assembly approach [48]; and by assembling potential protein-coding reads at the peptide level as in the ORFome assembly approach [47]. GeneStitch and GeneStitchPro utilize similarity between the genes included in a metagenomic dataset and reference genes available in a novel way, and uses the matches between the de Bruijn graph assembly and the reference genes to improve the gene assembly. In principle, GeneStitch (or GeneStitchPro) and other strategies to improve assembly can be combined to further improve the assembly of metagenomes.

3.6 Future improvements

We proposed gene graphs to represent collections of gene families, in which each gene graph represents a family of genes after applying GeneStitch or GeneStitchPro and other steps described above; however it is still very challenging to retrieve the actual gene sequences from the gene graph. We propose to apply the idea described in [73] to find actual gene sequences *de novo*. Briefly, Grabherr and colleagues attempt to assemble full-length transcriptome sequences from RNA-Seq data without a reference genome. Their assembler, Trinity, checks the support of the connection joints between contigs by reads information. The connection of two contigs are said to be "supported" if the joint is covered by reads. By using a similar idea, we could in principle find the *de facto* connection joints for each gene graph and construct genes using this information.

4. Targeted assembly approach to CRISPR discovery

CRISPR/Cas systems are a widespread class class of adaptive immunity systems, which bacteria and archaea mobilize against foreign DNA, including phage and conjugative plasmids [52, 75]. CRISPR elements, however, are very difficult to be extracted from any metagenome. We devised a targeted assembly approach to get the CRISPR arrays (regions in CRISPR/Cas systems that contain arrays of repeats and spacers between the repeats) from metagenomes. Moreover we also analyzed the array content to compare the differences between the metagenomes. This manuscript was written along with Mina Rho, Haixu Tang, Thomas Doak, and Yuzhen Ye and was published in [76].

4.1 Rationale

The CRISPR systems are found in most archaeal (~90%) and bacteria (~40%) genomes [49-51]. The CRISPR array consists of 24-47 bp direct repeats, separated by unique sequences (spacers) that are acquired from viral or plasmid genomes [77]. Figure 13 illustrates the structure and function of the CRISPR/Cas system. This system works as follows: when foreign DNAs invade, the CRISPR RNAs (crRNAs), which are used to silence foreign nucleic acids in a sequence-specific manner, are expressed and used to interfere with invading genomes at both the DNA and RNA levels, by mechanisms that are not fully understood yet [78-80].

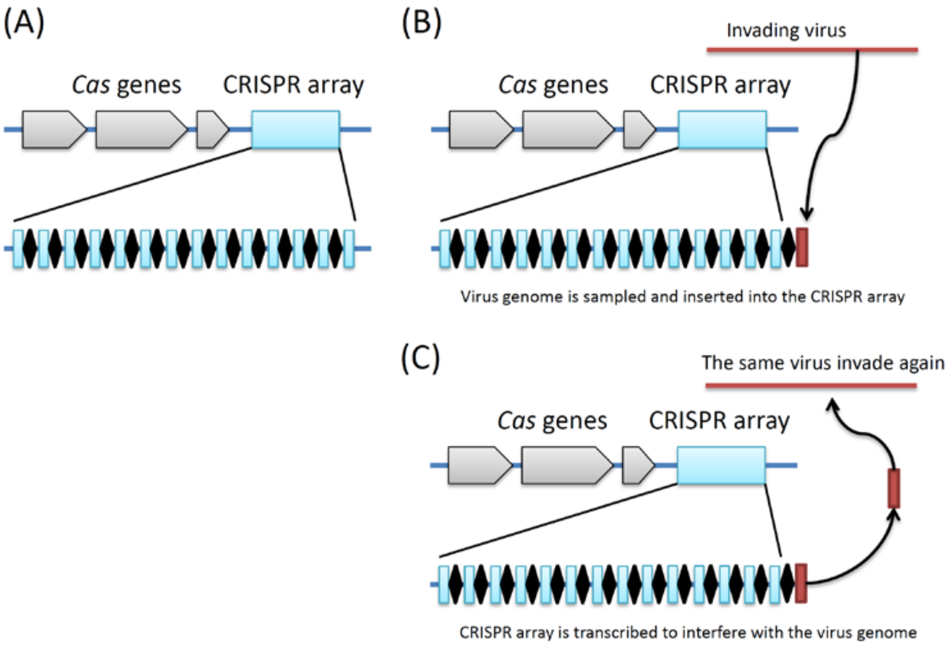

(A)
Cas genes
CRISPR array
(B)
Invading virus
Cas genes
CRISPR array
Virus genome is sampled and inserted into the CRISPR array
(C)
The same virus invade again
Cas genes
CRISPR array
CRISPR array is transcribed to interfere with the virus genome

Figure 13. (A) Illustration of CRISPR/Cas system. The direct repeats are represented by black diamonds. (B) The CRISPR/Cas system samples spacers from the invading viruses or plasmids and store the sequence in the array. (C) When the same viruses or plasmids invade again, the CRISPR array will be transcribed, and specific CRISPR spacer will bind to and interfere with the invading sequences. This figure is adapted from [75] and [81].

CRISPR loci can change very rapidly as a result of the interaction between viruses (plasmids) and bacteria: several metagenomic studies investigating host-virus dynamics has shown that CRISPR loci evolve in response to viral predation and that CRISPR spacer content and sequential order provide both historically and geographically insights [82-85]—especially, epidemiology. A recent study of streptococcal CRISPRs from human saliva using the conserved streptococcal repeat sequence for priming revealed substantial spacer sequence diversity within and between subjects over time [86], which reflected the dynamics of infectious agents in the human mouth. With the release of more than 700 metagenomic datasets from the Human Microbiome Project [15], we can explore the distribution and diversity of many more CRISPRs as well as discover new ones across different body sites.

Whole genome assembly may be the most common way to get any functional elements from any sequencing dataset, including CRISPRs; however it is very difficult to assemble metagenomic reads into contigs containing CRISPRs because of their repetitive structure. All existing tools, including CRISPRFinder [87], CRT [88], PILER-CR [89], and CRISPI [90], are also able to identity CRISPRs only in the contig levels. Therefore we need a specialized assembly method to get more complete CRISPRs from metagenomes. Targeted assembly approach, a variant of the ORFome assembly approach [47], is used to collect all reads with CRISPR repeats and assemble these reads into CRISPR contigs. This approach is also extended to novel CRISPRs or new CRISPR variants, which are not seen in the reference genomes, in order to get a more comprehensive identification of the CRISPR systems across the human samples.

4.2 Assembly of CRISPR arrays

4.2.1 Extraction of CRISPR repeats

The targeted assembly of CRISPRs starts from the identification of CRISPR repeats. The repeats are identified using both *de novo* method and similarity-based method. The *de novo* method, metaCRT, is modified from CRT [88], which first detecting repeats that are separated by a similar distance and then check other CRISPR specific requirements (e.g. the spacer needs to be non-repeating and of similar sizes). The similarity-based method, CRISPRAlign, on the other hand, identifies CRISPRs in a target sequence that contains repeats similar to a given query CRISPR. CRISPRAlign works by first detecting CRISPR repeats in the target sequence (and its reverse complement) that are similar enough to query CRISPR and then check other requirements as in metaCRT.

By using metaCRT and CRISPRAlign, we prepared a list of known CRISPR repeats (identified from complete/draft bacterial genomes in the IMG database [26]) as well as novel CRISPR repeats (identified from the whole-genome assemblies (PGA) of the HMP database). Known CRISPRs were first identified from the bacterial genomes (or drafts) collected in the IMG dataset (version 3.3) [91], using metaCRT. We then selected a subset of the identified CRISPRs that meet the following requirements: direct repeats are of length 24–40 bps; there are a minimum of 4 copies of the direct repeats; and the individual repeats differ by at most one nucleotide from the repeat consensus sequence, on average. The parameters were chosen to minimize false CRISPRs, considering that a CRISPR array typically contains 27 repeats, with an average repeat length of 32 base pairs [88]. We only kept CRISPRs that can be found in at least one of the whole-metagenome assemblies, using CRISPRAlign. We further reduced the number of candidate CRISPRs by only keeping those that share at most 90% sequence identity along their repeats by CD-HIT [92], as there are CRISPRs that share very similar repeats, and our targeted assembly strategy can recover the CRISPRs with slight repeat differences. To avoid including a repeat and its reverse complete (metaCRT does not consider the orientation for the repeats) in the non-redundant list, we included reverse complement sequences of the CRISPR repeats in the clustering process. Therefore, a repeat would be classified into two clusters by CD-HIT (the reverse complete of the repeat would be classified into a different cluster), one of which was removed to reduce redundancy. After clustering we collect a set of non-redundant CRISPR repeats, including 64 known and 86 novel ones, for further assembly of the CRISPR contigs. The detailed information for these CRISPRs (repeat sequences, and their resources) is provided in Supplementary Table 1 of [76].

4.2.2 Targeted Assembly approach

To assemble the CRISPR contigs, the collected non-redundant CRISPR repeats are used to search against the sequencing reads using BLASTN, in order to collect the reads that are similar to the repeat sequence, as shown in Figure 14. In order to make the similarity search tolerant to sequencing errors and genomic variations that are observed among the multiple copies of a CRISPR repeat (in one CRISPR locus or between different CRISPR loci), we allowed three mismatches over the entire CRISPR repeat sequence: we retained only the reads that are aligned with the entire CRISPR repeat sequence with a maximum of three mismatches. With these reads containing CRISPR repeat sequences, we ran SOAPdenovo [43] with *k*-mers of 45 bps, which are sufficiently long to assemble reads with the repetitive sequences found in CRISPRs. In general, whole-metagenome contigs are assembled using shorter *k*-mers (for example, 21-23 bps in MetaHit [13] and 25 bps in HMP project [15]), as longer *k*-mers often fragment assemblies into shorter contigs. After the CRISPR contigs are assembled, the exact boundaries of the repeats and the spacers are predicted using CRISPRAlign.

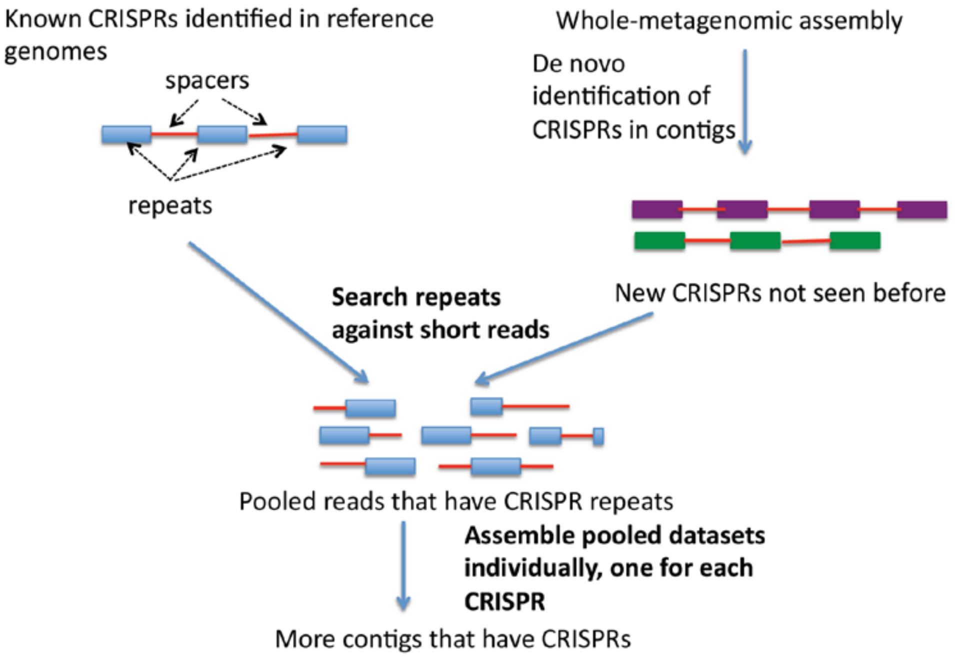

Known CRISPRs identified in reference genomes
spacers
repeats
Whole-metagenomic assembly
De novo identification of CRISPRs in contigs
New CRISPRs not seen before
Search repeats against short reads
Pooled reads that have CRISPR repeats
Assemble pooled datasets individually, one for each CRISPR
More contigs that have CRISPRs

Figure 14. A diagram of the targeted assembly approach for CRISPRs

4.2.3 Datasets used in CRISPR identification

The targeted-assembly approach was applied on the dataset Human Microbiome Illumina WGS Reads (HMIGWS) Build 1.0 available at http://hmpdacc.org/HMIWGS, and the whole-metagenome assemblies from the HMP consortium (http://www.hmpdacc.org/).

4.3 Results and evaluations

We identified and selected 64 known CRISPRs—including the streptococcal CRISPR—from complete (or draft) bacterial genomes and 86 novel CRISPRs from the 751 HMP whole-metagenome assemblies using metaCRT and CRISPRAlign. In order to test the effectiveness of the targeted-assembly approach, short reads from six reference genomes (*Azospirillum* B510, *Streptococcus mutans* NN2025, *Deferribacter desulfuricans* SSM1, *Dehalococcoides* GT, *Erwinia amylovora* ATCC 49946, and *Escherichia coli* K12 MG1655) are simulated using MetaSim software. We then apply our method to assemble the 10 known CRISPRs in the genome. All 54 contigs assembled by the targeted-assembly approach match perfectly to known CRISPRs in the reference genome, suggesting that our approach is quite effective and precise in getting CRISPRs from sequencing reads.

4.3.1 Targeted assembly approach improves the characterization of CRISPRs

We first apply this approach on Human Microbiome Project (HMP) datasets to identify the 64 known CRISPRs and find that this approach greatly improves the detection of CRISPR elements, as illustrated in Table 6. Two improvements are achieved using our approach. First, the targeted assembly approach identifies known CRISPRs in more human microbiome datasets, as compared to the annotation of CRISPRs using whole-metagenome assemblies. Second, targeted assembly resulted in longer CRISPR arrays, from which we can extract many more diverse spacers for analyzing the evolution of the CRISPRs and other purposes. Below we discuss several examples to show the effectiveness of the targeted assembly approach in identification of known CRISPRs.

The first example is the CRISPR Smuta36, which is conserved in streptococcal species such as *Streptococcus mutans* [86] and can be find only in 38 out of 751 datasets using contigs from whole genome assembly. By using the targeted-assembly approach, however, we are able to find this CRISPR in 386 datasets, which is ten times more than using the whole genome assembly and is consistent with the distribution of *Streptococcus* across body sites, as shown in Table 6. Most of the 386 datasets are from oral samples:

120 out of 128 supragingival plaques (94%), 128 out of 135 tongue dorsum samples (95%), and 97 out of 121 buccal mucosa samples (80%). On the other hand, CRISPR SmutaL36 was only found in a small proportion of samples from other body locations, where streptococcus rarely exists.

The other two examples listed in Table 6 include GhaemL36 and SRS018394L37. CRISPR GhaemL36 was initially identified from the genome of Gemella haemolysans ATCC 10379 using metaCRT. Targeted assembly further identified instances of this CRISPR in 258 oral-associated samples. The longest contig—of 3121 bases—was assembled from the SRS019071 dataset. This CRISPR array has even more repeats (48 repeats; i.e., 47 spacers) than the CRISPR array in the Gemella haemolysans reference genome, which has 29 repeats. CRISPR SRS018394L37 (currently not yet associated with a host genome) was initially identified from the whole-metagenome assembly of SRS018394, but targeted assembly reveals the presence of this CRISPR in 238 oral-associated microbiomes. The contig that was assembled in SRS049389 is the longest one (2014 bps), which contains 25 spacers.

Table 6. Comparison of CRISPR identification using whole-metagenome assembly and targeted assembly.

CRISPR	Sample datasets	Whole-metagenome assembly		Targeted assembly		
		Spacers (max)	Spacers (total)	Short reads	Spacers (max)	Spacers (total)
SmutaL36 (386 [a] vs 38 [b])	SRS017025 (plaque)	1 [c]	1 [d]	1078 [e]	26	76
	SRS011086 (tongue)	1	2	4018	24	78
GhaemL36 (257 versus 9)	SRS019071 (tongue)	0	0	1718	47	21
	SRS014124 (tongue)	3	3	490	21	58
SRS018394L37 (238 versus 39)	SRS049389 (tongue)	0	0	5778	25	492
	SRS049318 (plaque)	1	1	1463	38	134

[a]: the total number of samples that have streptococcal CRISPRs identified if using targeted assembly, and [b] if using whole-metagenome assembly; [c]: the total number of spacers found in the longest CRISPR locus found in the given dataset; [d]: the total number of spacers found in all contigs assembled from the given dataset; [e]: the total number of sequences that contain the repeats of a given CRISPR, *i.e.*, the recruited reads used for targeted assembly.

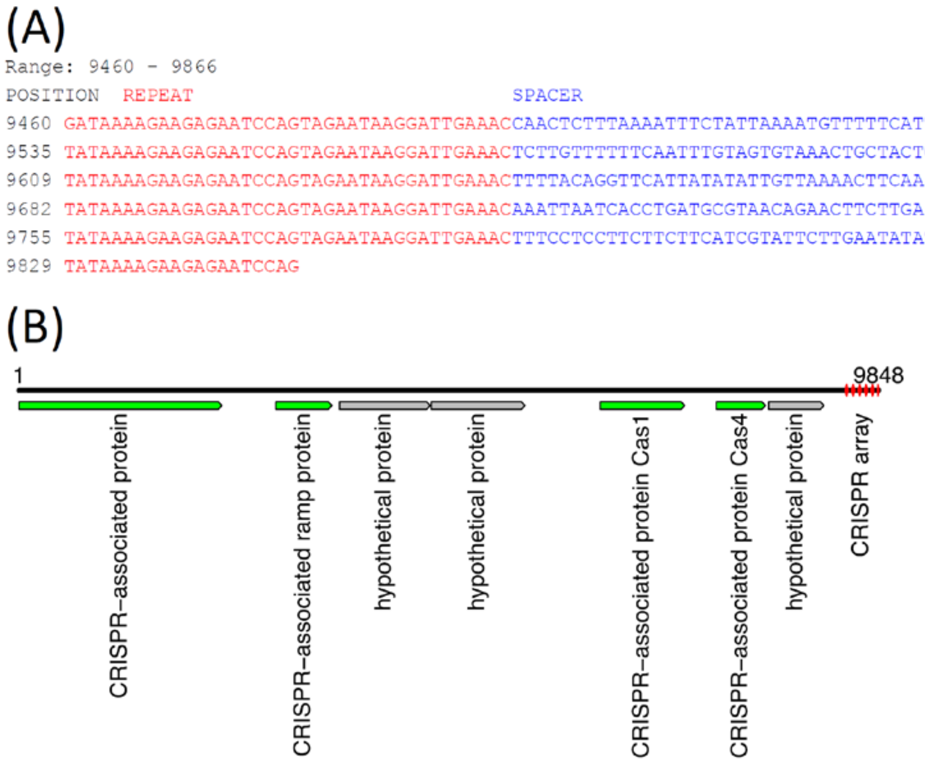
4.3.2 Novel CRISPRs are identified in human microbiome samples

Besides known CRISPRs, novel CRISPRs are also discovered to fuel further targeted assemblies. By using the program metaCRT, which we modified from CRT, we find the CRISPR loci based on their structure patterns. Overall we found 80 novel CRISPR loci in metagenomic samples. Table 7 lists a few examples of the novel CRISPRs that we identified. See Supplementary table 1 of [76] for the detailed list.

Table 7. Selected novel CRISPR loci.

CRISPR ID	HMP sample ID Consensus sequence of the CRISPR repeats
SRS012279L38	SRS012279 (dataset from a tongue dorsum sample) TATAAAAGAAGAGAATCCAGTAGAATAAGGATTGAAAC
SRS018394L37	SRS018394 (dataset from a supragingival plaque sample) GTATTGAAGGTCATCCATTTATAACAAGGTTTAAAAC
SRS023604L36	SRS023604 (dataset from a posterior fornix sample) GTTTGAGAGTAGTGTAATTTATGAAGGTACTAAAAC

Below we discuss two examples of two novel CRISPRs. The first example, CRISPR SRS012279L38, was identified from a whole-metagenome assembly contig of dataset SRS012279 (derived from a tongue dorsum sample; see Figure 15(A)). The identified CRISPR contig has 6 copies of a 38-bp, and BLASTX search of this contig against the nr protein database revealed proteins next to the CRISPR array that are similar to cas genes from other genomes, including *Leptotrichia buccalis* DSM 1135 (NC_013192, an anaerobic, gram-negative species, which is a constituent of normal oral flora [93] and *Fusobacterium mortiferum* ATCC 9817 (see Figure 15(B)). In addition, similarity searches revealed a single identical copy of this repeat in the genome of *Leptotrichia buccalis* DSM 1135 (from 1166729 to 1166764; *de novo* CRISPR prediction shows that this genome has several CRISPR arrays, including an array that has 84 copies of a 29-bp repeat, but none of the CRISPRs have the same repeat sequence as SRS012279L38). All evidence (similar cas genes, and an identical region in the genome) suggest that the SRS012279L38 CRISPR we found in the human microbiomes could have evolved from *Leptotrichia buccalis* or a related species.


(A)
(B)
1
9848
CRISPR–associated protein
CRISPR–associated ramp protein
hypothetical protein
hypothetical protein
CRISPR–associated protein Cas1
CRISPR–associated protein Cas4
hypothetical protein
CRISPR array

```
Range: 9460 - 9866
POSITION  REPEAT                                SPACER
9460 GATAAAAGAAGAGAATCCAGTAGAATAAGGATTGAAACCAACTCTTTAAAATTTCTATTAAAATGTTTTTCATTG
9535 TATAAAAGAAGAGAATCCAGTAGAATAAGGATTGAAACTCTTGTTTTTTCAATTTGTAGTGTAAACTGCTACTC
9609 TATAAAAGAAGAGAATCCAGTAGAATAAGGATTGAAACTTTTACAGGTTCATTATATATTGTTAAAACTTCAA
9682 TATAAAAGAAGAGAATCCAGTAGAATAAGGATTGAAACAAATTAATCACCTGATGCGTAACAGAACTTCTTGA
9755 TATAAAAGAAGAGAATCCAGTAGAATAAGGATTGAAACTTTCCTCCTTCTTCTTCATCGTATTCTTGAATATAT
9829 TATAAAAGAAGAGAATCCAG
```

Figure 15. A potentially novel CRISPR array identified in a contig (9848 bases) from sample SRS012279. This CRISPR array has 6 copies of the repeat (repeat sequences shown in red font and spacer shown in blue (A). (B) shows our annotation of this contig, in which the CRISPR array is highlighted in red. The annotations are based on BLAST search results; for example, the predicted CRISPR-associated Cas1 is similar to the Cas1 protein identified in Leptotrichia buccalis C-1013-b (accession ID: YP_003163976), with 60% sequence identify and 80% sequence similarity.

Targeted assembly of this novel CRISPR (SRS012279L38) in HMP datasets resulted in 278 contigs from 97 datasets, confirming the presence of this CRISPR in human microbiomes. In particular, the CRISPR fragments (407 bps) identified from the whole-metagenome assembly of SRS012279 were assembled into a longer CRISPR contig (890 bps) by targeted assembly. A total of 14 unique but related repeat sequences were identified from 278 CRISPR contigs, and two of them (which differ at 3 positions) are dominant, constituting 71% of the repeats in the CRISPR contigs. Notably, all the repeats could be clustered into a single consensus sequence with an identity threshold of 88%. By contrast, the spacer sequences are very diverse across different samples. For example, we obtained a total of 352 unique spacer sequences, which were clustered into 342 consensus sequences with an identity threshold of 80%. Among 352 unique spacers, 114 spacer sequences were shared by multiple samples—a single spacer was shared by at most eight samples.

The second example is CRISPR SRS023604L36, initially identified in a whole-metagenome assembly contig of dataset SRS023604 (derived from posterior fornix), which has 5 copies of a 36 bp. Our targeted assembly of this CRISPR across all HMP metagenomic datasets revealed further instances of this CRISPR in several other datasets, including two from stool, and two from posterior fornix. Moreover, the CRISPR contig was assembled into a longer contig of 778 bps containing 12 copies of the CRISPR repeat. BLAST search of the CRISPR repeat against the nr database did not reveal any significant hits.

In most cases we have tested, targeted assembly dramatically improves the identification of both known or novel CRISPRs in the HMP datasets: for 142 CRISPRs (out of 150), targeted assembly resulted in CRISPR identification in more HMP samples as compared to using whole-metagenome assemblies, and for 36 CRISPRs, targeted assembly identified instances of the corresponding CRISPR in at least 10 times more datasets. See Supplementary Table 1 in [76] for more details. It suggests that specifically designed assembly approaches, such as the targeted assembly approach for CRISPR assembly presented here, are important for the characterization of functionally important repetitive elements that otherwise may be poorly assembled in a whole-metagenome assembly (which tends to be confused by repeats), and such a comprehensive identification is important for achieving an unbiased distribution of these functional elements across different body sites among individuals.

4.3.3 Diverse distribution of CRISPRs across human body sites and individuals

Overall, the distributions of CRISPRs are largely body-site specific (see Figure 16; the name of CRISPR and the number of samples in which the CRISPR was found are listed in Supplementary Table 2 of [76]). For example, CRISPRs AhydrL30 and BcoprL32 are only found in stool samples (see Table 8). Exceptions include two CRISPRs that were found from both a significant number of gut- and oral-associated samples: Neis_t014_L28 were found in 51 gut samples and 92 oral-associated samples; FalocL36 identified from *Filifactor alocis* ATCC 35896 were found in 63 gut samples and 72 oral-associated samples, including 50 tongue dorsum samples (see Table 8).

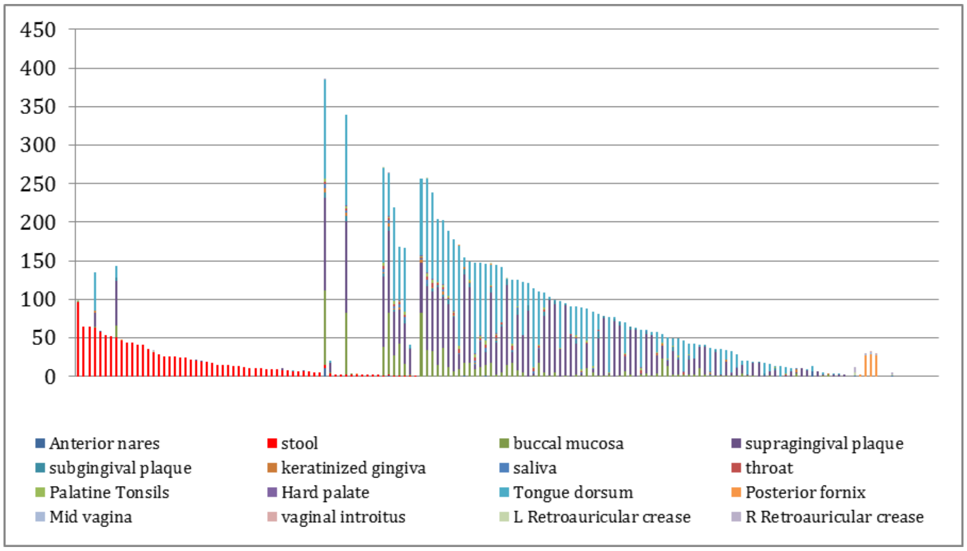


Figure 16. Distribution of CRISPRs across body sites. In this figure, x-axis represents 150 CRISPRs and y-axis represents the total number of samples in which instances of each of the CRISPR are found. Note that there are roughly one third as many stool samples as oral samples, probably explaining the apparently smaller number of CRISPRs in the gut microbiome.

Table 8. Distribution of selected CRISPRs across body sites.

CRISPR	anterior nares (94)	stool (148)	oral			posterior fornix (61)	Skin	
			buccal mucosa (121)	Supra-gingival plaque (128)	tongue dorsum (135)		L-[a] (9)	R-[a] (18)
SmutaL36	11	4	97	120	128	0	0	1
AhydrL30	0	53	0	0	0	0	0	0
BcoprL32	0	65	0	0	0	0	0	0
FalocL36	0	63	1	18	50	0	0	0
Neis_t014_L28	0	**51**	15	58	**15**	0	0	0
Neis_t014_L36	0	**0**	37	66	**82**	0	0	0
PacneL29	1	**0**	0	0	0	0	4	7

[a]: L: L-Retroauricular crease; R: R-Retroauricular crease. Note that not all body sites are listed in this table.

The first 50 CRISPRs shown in Figure 16 are mainly found in stool samples. AshahL36, which was initially identified from *Alistipes shahii* WAL 8301, was found in more than half of gut-related samples (96 out of 147 samples). On the other hand, 99 CRISPRs are mainly found in oral samples, in particular, tongue dorsum, supragingival plaque, and buccal mucosa. We found 5 CRISPRs that exist in more than half of the oral-associated samples (out of 417): SmutaL36, KoralL32 from Kingella oralis ATCC 51147, Veil_sp3_1_44_L36 and Veil_sp3_1_44_L35 from Veillonella sp. 3_1_44, and SoralL35 from Streptococcus oralis ATTC 35037. 4 CRISPRs are mostly found in vaginal samples (AlactL29, LjensL36, LjassL36, and LcrisL29). 1 CRISPR is skin-specific (PacneL29), found mainly in skin samples, such as the retroauricular crease.

4.3.4 The CRISPR spacers are very diverse

The CRISPRs that we identified in human microbiome samples (with 751 samples from 104 healthy individuals) shows substantial sequence diversity in their spacers among subjects. The CRISPRs that we identified in human microbiomes exhibited substantial sequence diversity in their spacers among subjects. Targeted assembly of the streptococcal CRISPRs (SmutaL36) in HMP datasets resulted in a total of 15,662 spacers identified from 386 samples, among which 7,815 were unique spacers (clustering of the spacers at 80% identify resulted in a non-redundant collection of 7,436 sequences). See Supplementary Fig. 2 in [76] for the sharing of the spacers in streptococcal CRISPRs among all individuals, which shows several large clusters of spacers that are shared by multiple individuals (for clarity, we only keep spacers that were shared by more than eight samples in this figure). In particular, the most common spacer is shared by 25 individuals (in 32 samples).

More importantly, we could check the sharing of CRISPR spacers across different body sites and sub-body sites (*e.g.*, multiple oral sites) using HMP datasets (Pride *et al.* examined streptococcal CRISPRs in saliva samples from 4 individuals [86]). Figure 17 shows the space sharing among 6 selected individuals, each of whom has multiple samples with identified streptococcal CRISPRs from multiple body sites. By examining the distribution of the spacers across samples, we observed that samples re-sampled from the same individual and oral site shared the most spacers, different oral sites from the same individual shared significantly fewer, while different individuals had almost no common spacers, indicating the impact of subtle niche differences and histories on the evolution of CRISPRs. Our observation is largely consistent with the conclusion from Pride *et al.* [86]. But our study showed that different samples from the same oral site of the same person, even samples collected many months apart, could still share a significant number of spacers (*e.g.*, the supragingival plaque samples from individual 1 in visit 1 and visit 2, with 238 days between the two visits, and the tongue dorsum samples from individual 5 in visit 1 and visit 3, with 336 days between the two visits; as shown in Figure 17). Our study also showed that although the different oral sites of the same individual share similar spacers, this sharing (*e.g.*, between the supragingival plaque sample and the buccal mucosa sample for individual 1) is minimal, as compared to the spacer sharing between samples collected in different visits but from the same oral site (*e.g.*, between the supragingival plaque samples from visit 1 and visit 2 for individual 1).

Finally, our study shows that the spacer turnover varies among individuals—for the 6 selected individuals, individual 3 shows significantly higher turnover of the spacers between visits, as compared to other individuals.

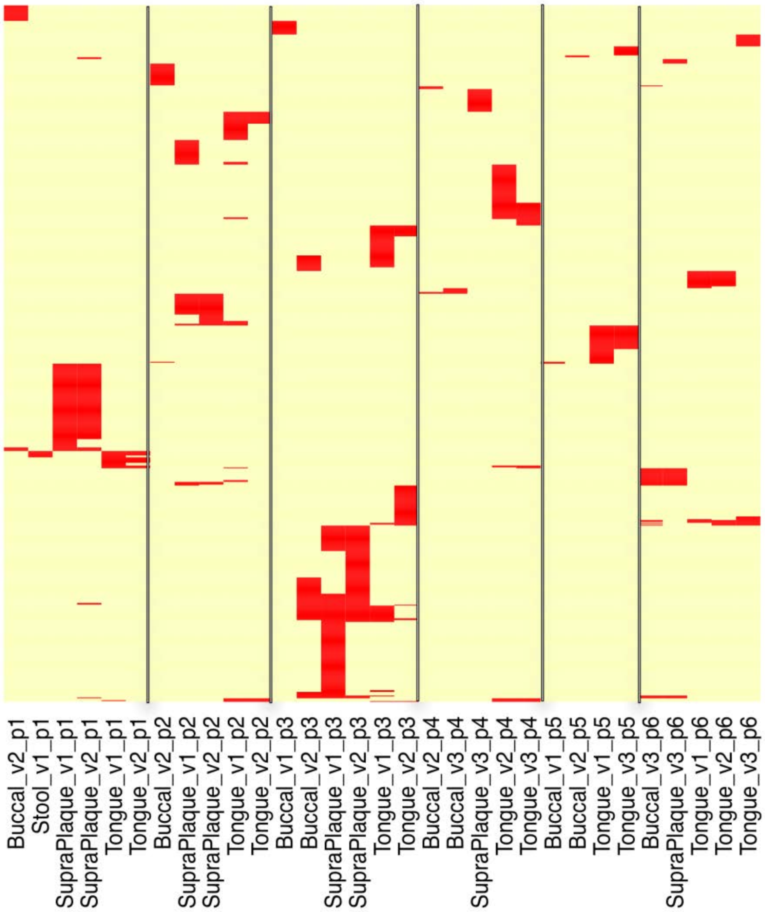


Figure 17. Sharing of streptococcal spacers among samples from 6 individuals. In this map, rows are the 761 spacers (clustered at 98% identify) identified in one or more of these 6 individuals, and the columns are samples (e.g., Stool_v1_p1 means a sample from stool of individual 1, in visit 1; tongue_v2_p1 indicates dataset from tongue, individual 1, in visit 2). Buccal stands for buccal mucosa, and SupraPlaque stands for supragingival plaque. The red lines indicate the presence of spacers in each of the samples. Multiple lines in the same row represent a spacer that is shared by multiple samples.

4.4 Discussion

We have applied a targeted assembly approach to CRISPR identification, to characterize CRISPRs across body sites in different individuals. Our studies show that an effective approach—such as our targeted assembly approach—is important for a comprehensive (thus less biased) estimation of the distribution of CRISPRs across body sites and individuals, and their dynamics. Note that in this study, we only focused on CRISPRs identified in eubacterial genomes, since archaea are rare in human microbiomes. Also for the sake of simplicity, we derived a non-redundant list of CRISPRs based on the similarity of the CRISPR repeats, and detailed targeted assembly was only applied to the non-redundant CRISPRs.

Although many CRISPR arrays will be missed by whole-metagenome assembly, we show that whole-metagenome assemblies are useful for finding novel CRISPRs (as de novo prediction of CRISPRs relies on sequence features of CRISPRs that do not exist in short reads). Once seeding CRISPRs are identified from whole-metagenome assemblies, we can go back to the original short read datasets, and pursue a comprehensive characterization of the CRISPRs, using the targeted assembly approach. Also, we did not fully utilize the presence of cas genes for identification of novel CRISPRs in our study, since in many cases we could identify arrays of repeats, but not their associated cas genes. A future direction is to combine targeted assembly of CRISPRs and whole-metagenome assembly, aiming to achieve even better assembly of functional elements that contain repetitive regions.

While the immediate utility of this study is to provide more complete inventories of CRISPR loci in human microbiomes, and indicate the usefulness of CRISPR repeats as phylogenetic markers, we look forward to being able to utilize the spacer sequences to understand human and human microbiome biology better, utilizing the metadata associated with the HMP datasets. This awaits more complete sampling of individuals over time, and of known relationships; and a far better characterization of bacteriophage and other selfish genetic elements in the human biome (our inventory of spacers is a standard against which phage and plasmid collections can be judged).

5. Constrained Assembly Approach to the Discovery of Integron Gene Cassettes

The targeted assembly approach that we discussed in chapter four is very effective in assembling CRISPR arrays, which consist of spacer sequences bounded by direct repeats. This approach is effective since the CRISPR spacer sequences are usually very short. For example, the average length of spacers detected in 51 complete *Escherichia* and *Salmonella* genomes is ~32 bps [94]. For elements with longer spacer regions such as integrons, however, the targeted assembly approach can no longer generate complete spacer contigs. To assemble the integron sequences, we designed another approach named *constrained assembly approach* to assemble this and other similar elements. This manuscript was written along with Mina Rho, Thomas Doak, and Yuzhen Ye, and was published in [95].

5.1 Rationale

Integrons are genetic elements that acquire and excise gene cassettes from their locus via site-specific recombination. The first integron, which is discovered in 1980s as the source of antibiotic resistant determinants [96], has been named resistant integron, or mobile integron, as they are often found in plasmids or associated with transposons. Another type of integron, the chromosomal integrons, were discovered in 1998 from examination of *Vibrio cholerae* genome [97]. Although they have similar structures, the two types of integrons (mobile integrons and chromosomal integrons) have different evolutionary histories, and differ in that the mobile integrons usually carry relatively few genes (predominantly antibiotic genes) while chromosomal integrons often carry far more genes of very diverse functions [98].

Integron consist of: a site-specific tyrosine recombinase (*intI*) gene, the primary recombination site *attI* immediately adjacent to the *intI* gene, and an array of captured gene cassettes encoding accessory functions [53]. Gene cassettes are the minimal units that can be mobilized by the integrase, with each cassette containing one or a very small number of genes [99] and are separated by a recombination site *attC*. Aggregation of different gene cassettes results in variable gene cassette arrays. The number of gene cassettes in integrons can reach several hundred; for example, the total length of the gene cassette pool from merely five *Vibrio* chromosomal integrons is equivalent to a small genome [100].

PCR with degenerate primers targeting the conserved regions of *attC* sites has recovered novel integrase genes and hundreds of diverse gene cassettes from various environments, including soil, sediment, biomass, or water habitats [101-103]. Rowe-Magnus et al. employed a three-plasmid genetic strategy to recover integron genes, using the integrase to bind integron *attC* sites [100]. These methods, which utilized the conserved nature of integron recombination sites, revealed a very dynamic integron gene repertoire and suggested that the gene cassette pool is likely to be limitless [104], while at the same time we do not know of work identifying the sources of integron genes.

A different approach, the constrained assembly approach, is designed to discovering chromosomal integrons in human-associated microbial communities, using shotgun metagenomic sequences of the human microbiomes. Human bodies are complex ecological systems, in which various microbial organisms and viruses interact with each other, and with human hosts. The MetaHit project has established a human gut microbial gene catalogue [13], and defined three enterotypes of human gut microbiomes [14]. The Human Microbiome Project (HMP) [15] has resulted in > 700 datasets of shotgun metagenomic sequence (http://www.hmpdacc.org/), from which we can learn the compositions and functions of human-associated microbial communities.

Our approach to integron discovery builds upon two novel computational methods: a targeted assembly approach for identifying the *attC* sites associated with chromosomal integrons (the repeats) in reads; and a constrained assembly approach for identifying the gene cassettes, which first greedily retrieves potential paths in the de Bruijn graph [40, 105] for a metagenomic dataset, constrained to contigs containing the *attC* sites, and then selects the paths that most likely represent cassette genes. We will demonstrate that such specialized computational tools are important for a comprehensive characterization of metagenomic functional elements that contain repeats (such as the *attC* sites in the integron gene cassettes), as these repetitive regions are extremely difficult to assemble using a whole-metagenome assembly strategy.

In this study we focus on the identification and characterization of integrons associated with *Treponema* species implicated in periodontal disease [106, 107] in the HMP datasets, using our integron discovery system. *T. denticola* genome contains a chromosomal integron with 45 gene cassettes [99], and it was the only human-associated bacterial species that harbors chromosomal integrons [53]. We also discover that the draft assemblies of two HMP reference genomes: *T. vincentii* and *T. phagedenis* (http://www.hmpdacc.org/HMRGD/) contain integron *attC* sites similar to *T. denticola* and possibly harbor integrons. We do not find integrons in other *Treponema* species, including *T. pallidum* SS14 uid58977 [108], *T. pallidum* Nichols uid57585 [109], *T. primitia* ZAS-2, and *T. azotonutricium* ZAS-9 [110]. From the HMP datasets we identify 826 integron gene cassettes that are related to the *Treponema* species, providing a gene cassette pool with 598 non-redundant genes. With these newly identified gene cassettes, we are able to compare the gene cassettes from different human subjects, and study the dynamics of the integron gene cassettes in their natural environments (i.e., human bodies), providing a first survey of integron-containing *Treponema* species and their integrons in a normal human population.

5.2 Assembly of integron gene cassettes

5.2.1 Selecting representative repeat sequences for *Treponema denticola* chromosomal integrons

Eight distinct sequences were selected to represent the integron *attC* repeats in the *T. denticola* genome (the complete genomes of the other two integron-containing *Treponema* species, *T. vincentii* and *T. phagedenis*, are not available), given that not all the repeats are identical (see Figure 18). The pairwise sequence similarity between these eight sequences ranges from 77% to 44%, and all the *attC* sites in *T. denticola* can be aligned to at least one of the representative sequences with > 85% sequence identify. Once the representative sequences are selected, we are able to identify new *attC* sites using similarity searches, instead of looking for features of integron recombination sites as in [103]. One advantage of using similarity searches is that we can recover degenerate sites that may lack some typical characteristics of integron recombination sites.

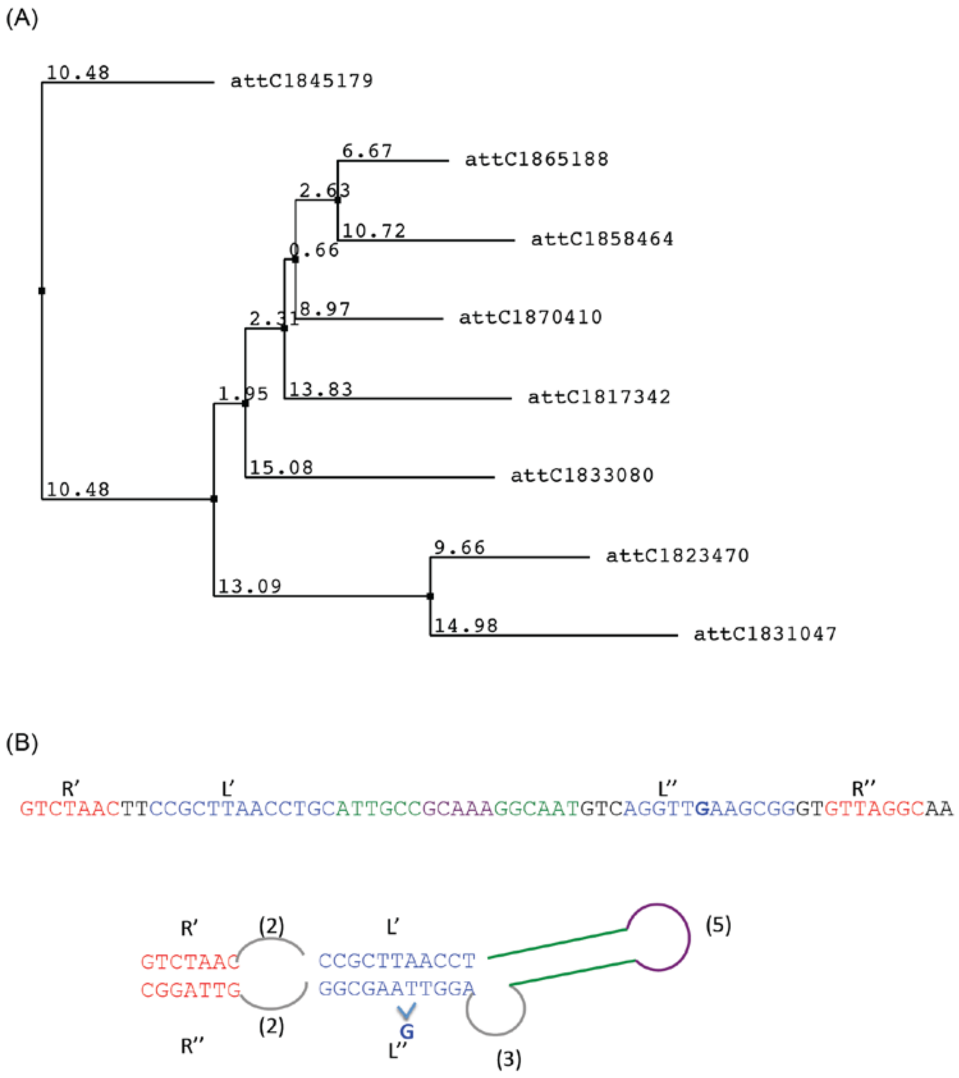


Figure 18. (A) The NJ-tree of the eight representative sequences of the *T. denticola* chromosomal integron recombination sites. The sequences are named by the starting position of the sites in the genome. The multiple alignment was prepared using ClustalW [111], and the NJ-tree was prepared using the jalview tool [112]. (B) The predicted structure of one of the representative sequences, attC1870410, which has the typical structure of an integron recombination site, with two stems and one conserved unpaired G. The structure was predicted by RNAscf

[113], software that performs simultaneous alignment and folding of RNAs, using the eight representative sequences as input.

5.2.2 Targeted assembly approach to identify integron *attC* sites

The targeted assembly approach was developed to characterize CRISPR arrays from shotgun metagenomic sequences [76] and was employed here to identify and assemble the integron *attC* sites.

1. Searching for reads that contain *attC* sites (with identity > 70% and covering > 50% of at least one of the representative *attC* sequences) using BLAST [23]. For paired-end reads, if one of a pair qualifies, both reads for the pair are included.
2. Assembling the retrieved short reads using SOAPdenovo [43]. We used *k*-mers of 31 bp, which were sufficiently long to assemble reads with the repetitive sequences found in the integrons; by contrast, whole-metagenome assembly generally uses shorter k-mers (for example, 21-23 bps in MetaHit [13] and 25 bps in HMP project [15]).

5.2.3 Constrained assembly approach to retrieve integron gene cassettes

A second approach, constrained assembly, was used to assemble integron gene cassettes from metagenomic shotgun reads. Since integron cassettes consist of genes that are much longer than the read length (~100 bp for the current Illumina technology), and the *attC* sites behave like repeats that confuse (meta-)genome assemblers, it is extremely difficult to obtain gene sequences using either a whole-genome-assembly method, or the targeted-assembly approach (which is good for assembly of repeats, but does not assemble very far beyond the repeats). As the integron cassettes are bounded by two *attC* repeats, we took advantage of this structure and devised a novel way to retrieve the cassette genes by traversing in the assembly graph, constrained by the edges (contigs) that contain the *attC* sites. To avoid introducing artificial integron genes, we further applied several criteria to select paths that are most likely to present genuine gene cassettes. The constrained assembly approach consists of the following steps (see Figure 19):

1. Assembling all shotgun reads in a metagenomic sequence dataset—along with the contigs constructed by the targeted assembly approach, which may contain more complete *attC* sites as compared to shotgun reads—using SOAPdenovo [43] with *k*=39 (see below for the selection of *k*-mer parameter using simulated datasets), producing both contigs and the assembly graph (a *de Bruijn* graph) [40] (see Figure 19(B)).
2. Searching for *attC* sites in contigs using BLAST (with an identity threshold of 70% and coverage threshold of 50%), and tagging contigs with *attC* sites to be used as constraints to constrain the next step.
3. Extracting paths that start from one tagged contig and end at another tagged contig using a depth-first search algorithm, and assembling the sequences for each path; the maximum length from one integron *attC* site to another *attC* site is set to 5000 bp (see Figure 19(C)).
4. Checking the support of each assembled sequence by mapping the reads and read pairs onto the assembled sequences using BWA [68]. We consider that a traverse

between two contigs is valid if the flanking regions of the connection (of *l* bp at both sides; *l* is set to 15) are supported by at least one read or read pair, and an assembled sequence is considered to be supported only if all the traverses involved are supported by reads (see Figure 19(D)).

5. Predicting the genes in each assembled sequence using FragGeneScan [66], with error model turned off. We require that the maximum gene number between any two integron *attC* sites is 3, considering that most integron cassettes contain 1-3 genes [99] (see Figure 19(E)).

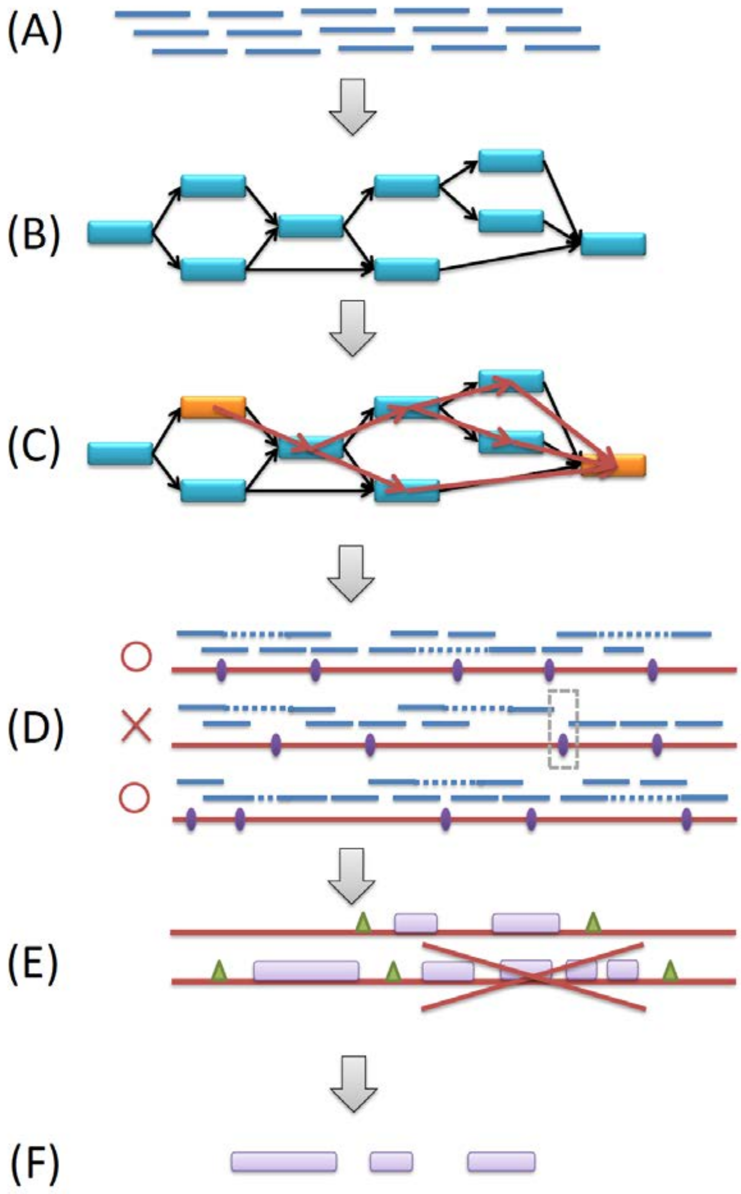


Figure 19. A diagram of the constrained assembly of integron gene cassettes.

5.2.4 Validation of constrained assembly using simulation

We simulated three metagenomic datasets by sampling reads at different coverage (10X, 20X and 31X) from nine *Treponema* genomes (or genome drafts) using MetaSim [58] with the Illumina 80bp error model of error rate ~1% provided by the authors (http://ab.inf.uni-tuebingen.de/software/metasim/errormodel-80bp.mconf). The species include: *T. denticola* ATTC 35405 (NC_002967), *T. azotonutricium* ZAS-9 (NC_015577), *T. primitia* ZAS-2 (NC_015578), *T. pallidum* subsp. *pallidum* SS14 (NC_010741), *T. pallidum* subsp. *pallidum* str. Nichols (NC_000919), *T. succinifaciens* DSM 2489 (NC_015385), *T. denticola* str. F0402 (downloaded from http://www.broadinstitute.org/), *T. vincentii* (http://hmpdacc.org), and *T. phagedenis* (http://hmpdacc.org). We tested different *k*-mer parameters for the constrained assembly approach using these simulated datasets, and the results show that *k*=39 resulted in the most integron genes for all the datasets, as illustrated in Figure 20. The 31X dataset contains 4,499,532 paired-end reads and 500,468 singleton reads. 73 integron genes were identified from this dataset by our constrained assembly approach: 37 genes from *T. denticola* ATCC 35405, 27 genes from *T. denticola* str. F0402, seven genes from *T. vincentii*, and two genes from *T. phagedenis*. We mapped these genes back to the genomes and confirmed that 1) all the genes were correctly assembled (error rate is 0%), and 2) all the genes were mapped to the big integron, or the degenerate, small integron region in the genomes. In addition, we did not find any genes in the *Treponema* species that do not harbor integrons. All suggest that our constrained approach is reliable even when reads from closely related species are present.

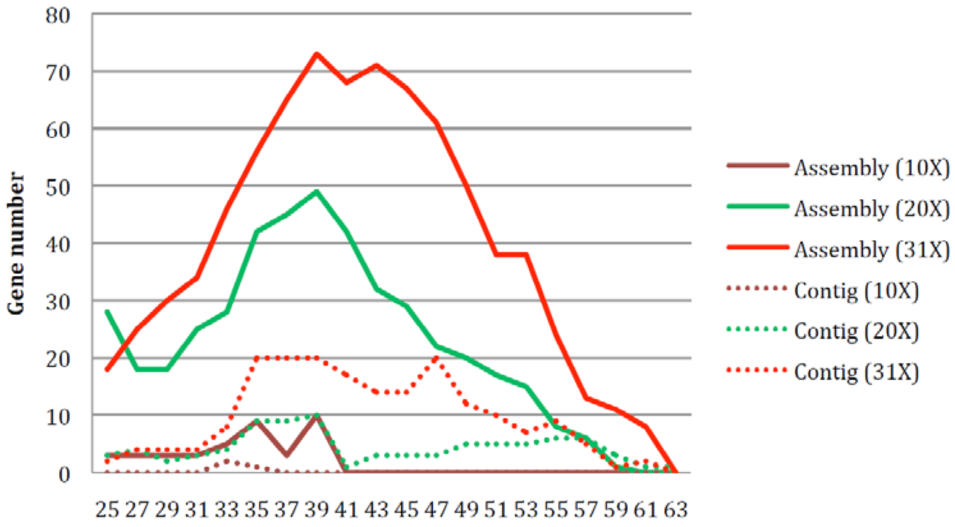


Figure 20. The number of integron genes discovered using different *k*-mer settings. The x-axis lists the *k*-mers, while the y-axis shows the total number of genes assembled. We generated three datasets of different coverage (10X, 20X, 31X) and applied our constrained assembly method to these datasets. Lines indicate the gene numbers found, and dashed-lines are the number of genes that are identified solely at the contig level (i.e. genes on the contigs that are bounded between two integron recombination sites).

5.2.5 Functional annotation of identified gene cassettes

We downloaded all protein sequences from the eggNOG v2.0 database [114], and retrieved the sequences with COG annotation [115]. MUSCLE [116] was used to generate a multiple alignment for each COG family, and the HMM builder from the HMMER3 package [117] was then applied to build a HMM for each COG. HMMER searches (by hmmscan from the HMMER3 package) were used to annotate the predicted integron gene cassettes, with an E-value cutoff of 0.001. For a gene with COG hits, we recorded the best non-overlapped results, so that if a gene encodes multiple domains with distinct functions, all the functions will be reported.

5.2.6 Identification of potential source species of gene cassettes

We used MEGAN [27] to identify the possible source species of the identified gene cassettes. We searched the genes against the NCBI NR database (as of September 2011) using BLASTP and applied the MEGAN software to analyze the similarity search results. Since the average length of the genes is 506 bp, we set the minimum score threshold to 100, as suggested by MEGAN's authors for longer reads.

5.2.7 The HMP datasets

We used the Human Microbiome Illumina WGS Reads (HMIGWS) Build 1.0, and the whole-metagenome assemblies (PGAs) from the HMP consortium (http://www.hmpdacc.org/). There are 757 total metagenomic samples from 103 subjects (individuals). The reference genomes were also downloaded from this website.

5.3 Results and evaluations

5.3.1 The *T. denticola* integron *attC* sites are unique to *Treponema* species

BLAST searches using the eight representative *attC* sequences against the NCBI nucleotide collection (NT) and the genome database (chromosomes) with default settings only hit *Treponema* genomes. Using an identity threshold of 70% and coverage threshold of 50%, 64 *attC* sites were found in the *T. denticola* ATCC 35405 genome, of which 45 are located within the chromosomal integron (1,817,049-1,874,294) identified by [99]. We also found two additional *attC* sites downstream of the integron region, suggesting that the integron may be even larger and contain more genes. The *attC* site located immediately downstream of the previously-reported integron location is more degenerative (barely passes the coverage threshold), but the site further downstream is more complete, and we believe these two *attC* sites are genuine. In addition, we found 7 *attC* sites outside the big integron region (for example, there is an *attC* site located between 300,167 and 300,227, which shares 98% sequential identify with the *attC* site within the integron array between 1,870,410 and 1,870,474). Furthermore a degraded *IntI* gene exists between 302,289 and 302,350, suggesting that a degraded, small integron may exist in this region of the genome. We also discovered integron sequences in *T. denticola* F0402 (sequence downloaded from http://www.broadinstitute.org/). While the integrase genes (*intI*) are very similar between these two strains (with 95% identity), the integron gene cassettes are quite different—only ten integron genes are shared between these two strains, as shown in Figure 21.

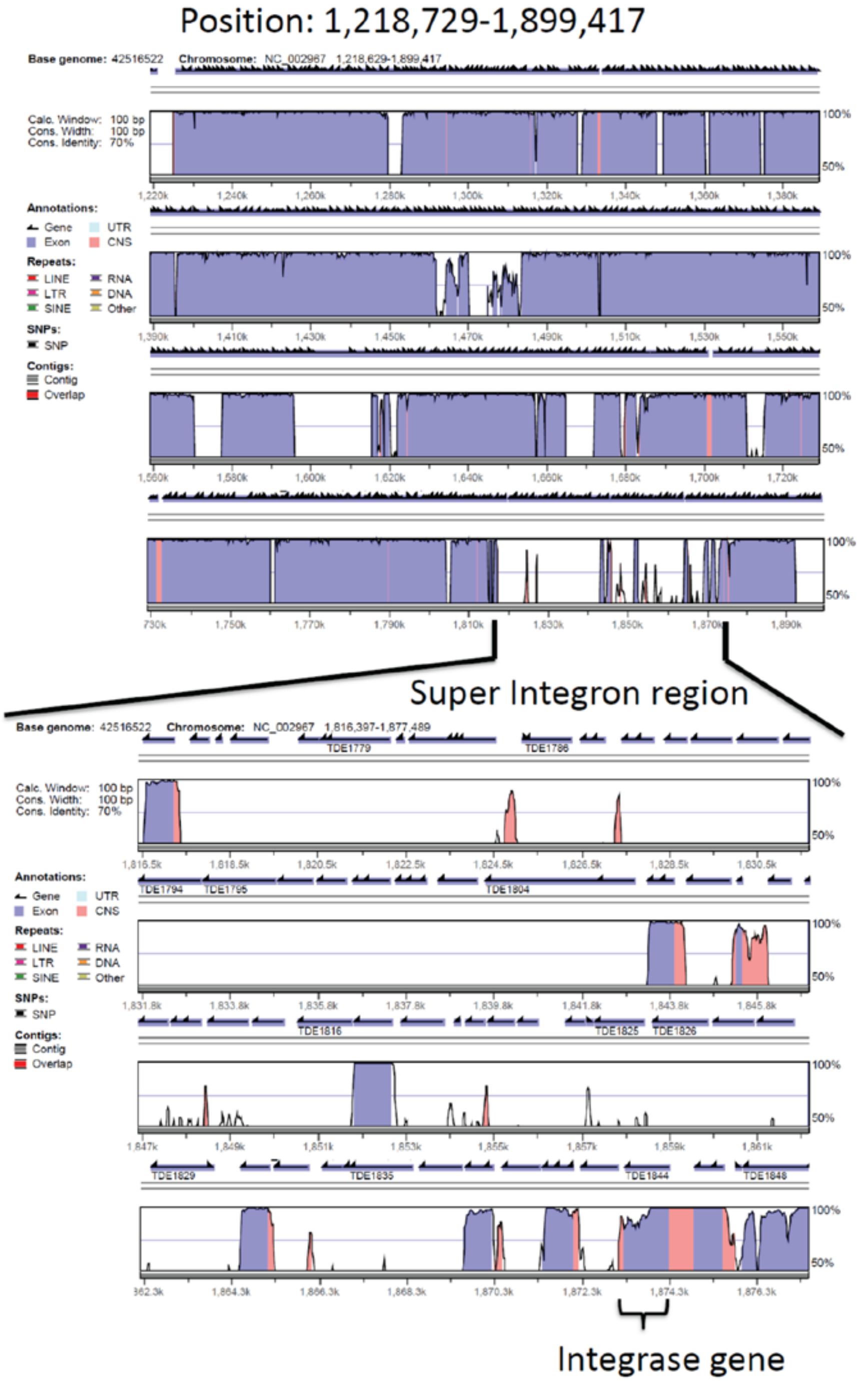


Figure 21. Mapping result of *Treponema denticola* F0402 contig ADEC01000014 to the *T. denticola* ATCC 35405 genome. This plot is generated using RankVISTA web service [118].

Instances of the *T. denticola attC* sites were also found in the draft assemblies of two human microbiome reference genomes (as of July 2011): *T. vencentii* ATCC 35580 and *T. phagedenis* F0421. A total of 16 *attC* sites were found in five contigs of *T. vencentii* ATCC 35580, and 6 *attC* sites were found in three contigs of *T. phagedenis* F0421. We further checked the *T. vencentii* and *T. phagedenis* genomes for features indicative of integrons. In both genomes, there are gene cassettes flanked by *attC* sequences: we identified one gene in a *T. phagedenis* contig, and 12 genes from three contigs of *T. vincentii*. One of the *T. vincentii* contigs exhibits a very clear integron structure, as shown in Figure 22. None of the 12 genes identified in *T. vincentii* share significant similarity with the integron genes of the *T. denticola* integron, suggesting that the gene cassettes of the two integron loci have undergone substantial changes since these two species diverged. We also searched the *T. vencentii* and *T. phagedenis* genomes using the *T. denticola intI* gene and detected a significant (sequence similarity=86%) and long *intI* (953 bp) gene on the *T. vincentii* contig ACYH1000073, which is demonstrated in Figure 22. Together with the recombination sites and the gene cassettes, this region contains all elements required for an integron.

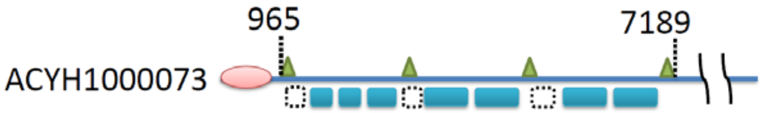


Figure 22. The predicted integron recombination sites and genes in the contig ACYH1000073 of *T. vincentii*. Triangles are recombination sites, rectangles represent the integron genes, and the oval is the *IntI* gene. We use solid rectangles to represent the genes that pass our integron

gene discovery threshold, and dashed rectangles are open reading frames that do not meet the criteria.

5.3.2 Detecting the existence of the integron-containing *Treponema* species in human samples

We identified integron *attC* sites in 300 of >700 HMP samples, using targeted assembly. The body sites that have identified integrons are summarized in Table 9. Most samples with integrons are oral-related (including hard palate, supragingival plaque, saliva, tongue dorsum, subgingival plaque, throat, buccal mucosa, and attached/keratinized gingiva sites), whereas non-oral samples, including stool and vagina, do not contain integron *attC* recombination sites (repeats). It suggests that a high proportion of oral samples contain the *Treponema* species implicated in dental diseases, implying that these pathogens are ubiquitous among people. The existence of *Treponema* species implicated in dental diseases in most normal human individuals (though of low abundances) is also supported by mapping the sequencing reads onto the available compete genomes (or drafts) of the three integron-containing *Treponema* species (*T. denticola*, *T. vincentii*, and *T. phagedenis*) (See mapping results in Figure 23). We found only rare samples from nose (anterior nares) and ear (retroauricular crease) with integron repeats.

Table 9. Summary of the HMP samples with identified *T. denticola* integron *attC* sites.

Location	Samples with *attC* sites	Total number of samples	% of samples with *attC* sites
Hard palate	1	1	100%
Supragingival plaque	98	128	77%
Saliva	5	5	100%
Tongue dorsum	109	136	80%
Vaginal introitus	0	3	0%
Stool	0	150	0%
Mid vagina	0	2	0%
Subgingival plaque	8	8	100%
Throat	6	7	86%
Posterior fornix	0	62	0%
Anterior nares	2	94	2%
Buccal mucosa	60	122	49%
R Retroauricular crease	2	18	11%
L Retroauricular crease	0	9	0%
Palatine Tonsils	6	6	100%
Attached/Keratinized gingiva	3	6	50%

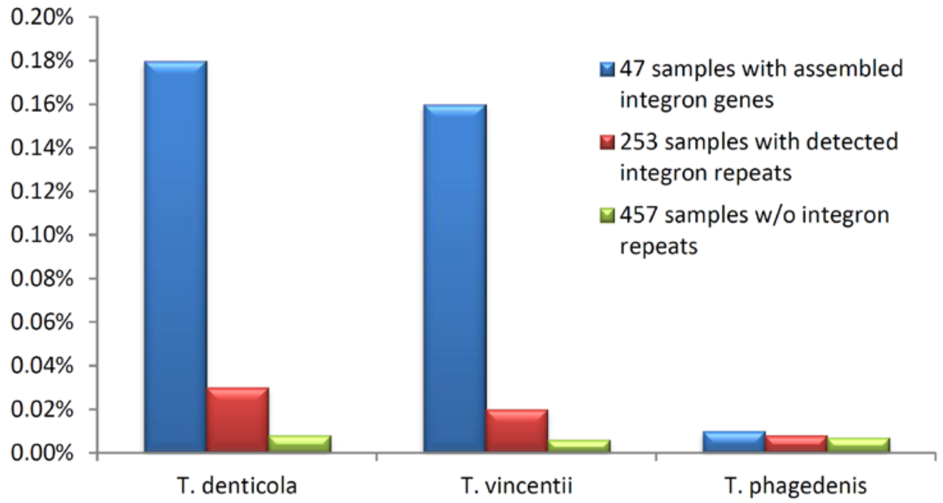


Figure 23. Comparison of the average abundances of the three integron-containing *Treponema* species in the HMP samples of different categories: samples with assembled integron genes (shown in blue), samples with detectable recombination sites (but no integron genes are assembled; shown in red), and samples without recombination sites detected (in green). The abundances in each HMP sample were estimated by mapping paired-end shotgun sequences of the HMP datasets onto the genomes (or genome drafts), by BWA [68]. Both reads in a pair are counted if at least one read maps to the genomes. Reads that map to common regions of genomes from different species are considered for all corresponding species in the estimation of the abundances. This chart confirms the existence of these *Treponema* species in the HMP datasets, with *T. denticola* and *T.vincintii* being more abundant in the samples. The mapping results are consistent with the results of the identification of *attC* sites and the integron gene cassettes: the samples with integron gene cassettes identified have the most *T. denticola* and *T. vincintii*, and the samples without recombination sites identified have the lowest presence of these species. This figure also suggests that the integron genes we identified are more likely to be from *T. denticola* and *T. vincintii*.

The 300 samples containing *attC* sites resulted in 85 out of 103 individuals having an identified infection of *Treponema* species (82.5%; between 1 and 15 samples per individual). This number is consistent with a previous report that disease associated with *T. denticola* occurs in 80% of adults, at some time in their lives [107].

We checked the size of each oral sample (as measured by the total bases), and found that oral samples with identified integron *attC* sites are significantly larger than samples without *attC* sites (Welch's t-test, Z=4.63, degree of freedom=230, p<0.001). This is expected; as the *Treponema* species implicated in dental disease are not abundant in oral sites of healthy individuals (see Figure 23), and will be difficult to detect when sequencing is shallow. Thus the 80% prevalence may be a conservative estimate.

5.3.3 Detecting integron genes in HMP whole-metagenome assembly

We first identified integrons in the contigs from the whole-metagenome assemblies of human metagenomes by looking for genes flanked by *attC* sites. 741 *attC* sites were detected in the whole-metagenome assemblies, but most contigs carry only one *attC* site. As a result, we only found 66 non-redundant (at 97% identify cutoff) genes from 25 samples: 17 are from supragingival plaque, six are from tongue dorsum, and two samples are from subgingival plaque. The sample distribution shows that we can indeed find integron genes associated with *Treponema* species (and hence demonstrate the existence of these oral pathogens) in mouth-related samples.

Figure 24 shows an example from contig SRS049318_LANL_scaffold_118938, with two *attC* sites at 176-226 and 817-877 bps. FragGeneScan predicted one protein-coding gene between the two sites, and similarity search of this predicted protein against the NCBI

NR database revealed similarity to a hypothetical protein in the *T. denticola* genome; and to a HNH nuclease domain (SUPERFAMILY ID, cl00083) [119]. HNH endonuclease features 11 conserved residues, and all are conserved in the predicted protein.

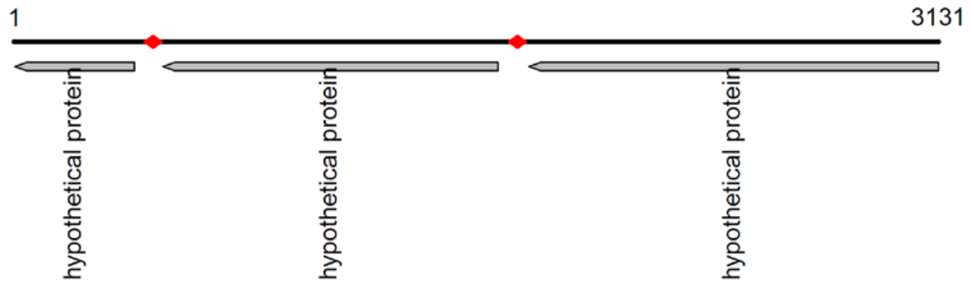


Figure 24. Annotation of a contig from sample SRS022602 (SRS022602_Baylor_scaffold_118781) of 3131 bp. Red diamonds indicate the two repeats identified in this contig with similarity to the *attC* sites in the *T. denticola* chromosomal integron, and the three gray boxes indicate the predicted genes. The first gene (1-407) shares 46% sequence identify and 66% similarity along 97% of the gene with a protein (YP_001868417.1) from the *Nostoc punctiforme* PCC 73102 genome (a nitrogen-fixing cyanobacterium). The second gene (503-1639) shares 31% identify (53% similarity) along 99% of the gene with a protein (ADE86468.1) from *Rhodobacter capsulatus* SB 1003 (a purple, nonsulfur photosynthetic bacterium). The third gene (1743-3131) shares 24% identify and 45% similarity, covering 88% of the gene, with a protein (ZP_04160697.1) from *Bacillus mycoides* Rock3-17 (a Gram-positive, non-motile soil bacterium); this gene also shares 24% sequence identify and 46% similarity (covering 65% of the gene) with a protein (YP_002158281.1, Nuclease-related domain family protein, NERD) from *Vibrio fischeri* MJ11 [120].

5.3.4 Using constrained assembly approach to detect more integron gene cassettes in HMP samples

Using the whole-metagenome assemblies, we were able to retrieve only 66 integron-associated genes (see above). Application of our constrained assembly approach to the HMP data sets led to the identification of 794 genes in 47 samples. After combining both predictions and keeping only unique genes for each sample, we derived a total of 826 unique genes (598 97% non-redundant). The detailed comparison between the results generated by the constrained assembly approach and that obtained from the whole metagenome assembly is listed in Table 10, which shows that the constrained assembly approach is able to discover far more genes for most of the individual HMP samples. The distribution of sample locations and the number of genes in each location are listed in Table 11. We identified genes in 24 supragingival plaque samples, 19 tongue dorsum samples, and 4 subgingival samples. The proportion of samples with gene cassettes identified using the constrained assembly approach is still low—compared with samples with identified *attC* sites (300)—due to the low abundance of the *Treponema* species in many samples (see Figure 23). But we can still utilize the *attC* sites (taking advantage of the multiple copies of the *attC* sites) to identify *T. denticola* or related species in those samples, demonstrating the power of using unique repeats to trace rare species. We note that mapping shotgun sequences onto the known reference genomes (or drafts) of *Treponema* species can be used to identify the existence of these species in the HMP samples, but such a mapping cannot be effectively used to identify the integron gene cassettes due to the dynamic nature of the integron genes (*e.g.*, the two *T. denticola* strains only share 10 cassette genes; see above).

Table 10. Identified integron gene numbers for each sample using constrained assembly approach (CONST) and whole metagenome assembly (WHOLE).

Sample-ID	CONST	WHOLE
SRS011115	13	0
SRS011126	12	0
SRS011152	2	1
SRS011255	0	2
SRS013533	28	1
SRS013705	17	2
SRS013836	2	0
SRS013950	36	1
SRS014470	4	0
SRS014476	37	7
SRS014477	42	0
SRS014573	45	0
SRS014578	8	8
SRS014691	5	0
SRS015215	19	11
SRS015434	17	0
SRS016331	64	0
SRS017209	10	1
SRS017227	0	5
SRS017691	2	0
SRS018157	13	0
SRS018739	43	4
SRS019029	3	0
SRS019071	1	0
SRS022143	2	0
SRS022149	35	1
SRS022602	8	1
SRS023595	66	3
SRS024441	32	1
SRS024561	0	1
SRS042643	29	0
SRS045313	3	0
SRS047113	8	1
SRS047634	11	0
SRS049318	42	8
SRS049389	18	0
SRS050244	5	0
SRS050669	1	0
SRS051930	12	1
SRS055378	8	5
SRS055401	2	0
SRS057205	1	0
SRS058053	2	1
SRS058808	18	0
SRS062544	20	1
SRS063215	0	5
SRS063603	51	6
SRS063932	11	9
SRS063999	4	0
SRS064774	2	0
SRS075404	12	0

Table 11. Breakdown of the samples that have identified *T. denticola* integron gene cassettes into body locations.

	# of samples	# of genes	# of genes without COG hits
Supragingival plaque	24	457	252
Tongue dorsum	19	283	203
Subgingival plaque	4	86	46

Similarly, among the 300 samples with detected *attC* sites, the samples with gene cassettes assembled by constrained assembly were significantly larger than those with no identified genes (Welch's t-test, Z=4.42, degree of freedom=68, p<0.001). This can also explain why we did not find gene cassettes in samples from buccal mocusa—the buccal mocusa samples are significantly smaller than other oral datasets (Welch's t-test, Z=25.28, degree of freedom=388, p<<0.001), partially caused by a large contamination of human DNAs in the buccal mocusa samples.

5.3.5 The majority of integron gene cassettes are of unknown function

We annotated the predicted cassette genes using similarity-searches. Among the 826 genes, 501 cannot be assigned to a COG family (see Table 11): ~60% are un-assigned. Of the remaining genes, ~60% are assigned to COG categories R (general function prediction only) and S (function unknown). Combining these two categories, 85% of the 826 genes are of unknown function: the proportion is even higher than reported for other integrons (it was reported that 75% of the cassette pool associated with Vibrionales genomes corresponds to genes with undefined functions [53, 98]).

To analyze genes with identified functions, we clustered the genes within each location (at 97% identity) to see how many genes are unique to distinct locations. The functional category L (replication, recombination, and repair) is the majority among all functional categories (25%); genes associated with category D (cell cycle control, cell division, chromosome partitioning), K (transcription), N (cell motility), and T (signal transduction mechanisms) are also elevated among all functional categories, with 12%, 11%, 13%, and 13% of the genes with known functions, respectively, as shown in Table 12. Integron genes with these functions have been reported previously: for example, category L and category T are among the most prevalent functions reported by [53]. Genes in other categories, such as genes predicted to be part of the toxin/antitoxin system in category D, DNA-methyltransferase in category K, and methyl-accepting chemotaxis protein in category N, were also reported by [53]. This again demonstrates that our results are consistent with the previous findings of gene functions encoded by chromosomal integrons.

Table 12. The COG functional category distributions of the integron gene cassettes identified in different human body locations.

COG Functional Categories[1]	Supragingival plaque	Tongue dorsum	Subgingival plaque
[C] Energy production and conversion	1	0	0
[D] Cell cycle control, cell division, chromosome parititioning	8 (11)[2]	3	1
[E] Amino acid transport and metabolism	2 (4)	1	0
[F] Nucleotide transport and metabolism	1	0	0
[G] Carbohydrate transport and metabolism	1	0	4
[H] Coenzyme transport and metabolism	1	1 (8)	0
[I] Lipid transport and metabolism	1 (3)	0	0
[J] Translation, ribosomal structure and biogenesis	3	0	1
[K] Transcription	9	4	1
[L] Replication, recombination and repair	10 (12)	10 (14)	5 (6)
[M] Cell wall/membrane/envelope biogenesis	3 (4)	2 (3)	0
[N] Cell motility	8 (10)	2 (5)	1
[O] Posttranslational modification, protein turnover, chaperones	0	1	0
[P] Inorganic ion transport and metabolism	0	1	0
[R] General function prediction only	*45 (67)*	*13 (14)*	*7 (8)*
[S] Function unknown	*59 (72)*	*18 (21)*	*14 (15)*
[T] Signal transduction mechanisms	6 (10)	3	2 (3)
[U] Intracellular trafficking, secretion, and vesicular transport	2 (3)	1	0
[V] Defense mechanisms	2	3 (4)	1

[1]: the functional categories (including [A] RNA processing and modification, [B] Chromatin structure and dynamics, [Q] Secondary metabolites biosynthesis, transport and catabolism, [W] Extracellular structures, [Y] Nuclear structure, and [Z] Cytoskeleton) that have no gene cassettes are not listed in the table.

[2]: Number of genes is obtained by clustering the genes at a 97% identity threshold for each functional category within each location. Numbers within parentheses indicate the number of genes before clustering.

We further compared the predicted genes found in the HMP datasets against the genes in the *T. denticola* chromosomal integron (located at 1,817,049-1,874,294 on NC_002967, as reported by [99]) using BLAST with an E-value cutoff of 0.001. We found that of the 826 genes, 192 (23%) hit to the genome's integron genes. We also found that of the 70 integron genes identified in the *T. denticola* genome, 39 (56%) genes had homologs in the 826 genes retrieved from the human samples. In other words, about 44% of integron genes in the complete genome were missing from our broad survey of human samples. This clearly demonstrates that the *T. denticola* integron is undergoing an active process of cassette insertion and excision.

5.3.6 Tracing origins of integron gene cassettes

To infer the potential origins of the integron gene cassettes associated with Treponema species, we applied MEGAN [27] to analyze all the gene cassettes identified in the HMP samples. The MEGAN taxonomic assignments of the gene cassettes are summarized in Figure 25. A total of 365 (44%) genes cannot be assigned to any taxon. Among the genes (461) assigned to a taxon, 152 (18%) are assigned to *T. denticola* (at the specie level), 47 (6%) genes are assigned to *T. vincentii*, and 262 genes are likely originated from other species: 117 (14%) genes from other spirochete species, and 145 (17%) genes from non-spirochete species.

Table 13 lists the detailed list of candidate donor species, and the annotations of the potential donor genes in these species. Here we show two examples: the first example is 14 genes assigned to the order *Clostridia*, which was first discovered in soil, but also appears in human microbiomes [121, 122]; and the other example is 25 genes assigned to *Spirochaeta caldaria*, a thermophilic bacterium [123].

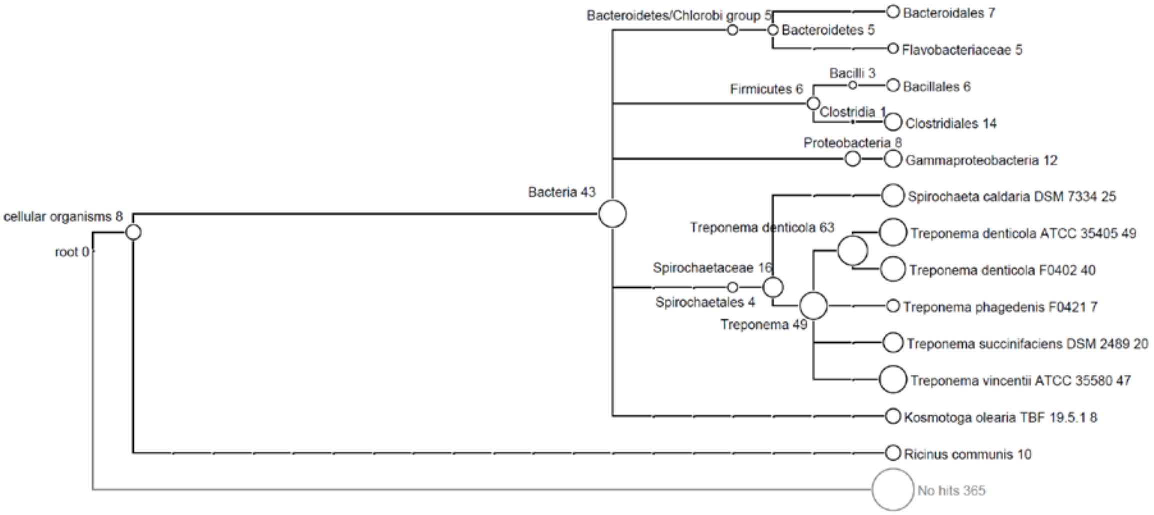


Figure 25. Taxonomic assignments of the integron genes by MEGAN. The numbers following clade names are the number of genes assigned to that taxonomic rank, not including the genes assigned to the taxa below that rank (for example, there are 63 genes assigned to *T. denticola* species, 49 genes assigned to strain ATCC 35405, and 40 genes assigned to strain F0402; in total, 138 genes can be assigned to the *T. denticola* species).

Table 13. Functions of genes related to species other than *T. denticola* or *T. vincentii*

Species	Gene functions	Number of genes[1]
Bacillales		
	Hydrolase	2 (3)
	Hypothetical	2 (3)
Bacteroidetes		
	Hypothetical	5 (6)
	DNA-cytosine methyltranferase	1 (1)
Clostridiales		
	Hypothetical	3 (4)
	D-alanine-D-alanine ligase	1 (2)
	Type II restriction enzyme *HphI*	1 (1)
	Acetyltransferase (GNAT) family	1 (1)
	Toxon-antitoxin system, antitoxin component, XRE family	1 (1)
	Hydrolase, NUDIX family	1 (1)
	Toxon-antitoxin system, toxin component, *Txe*/*Yoe* family	1 (1)
	ABC transporter, ATP-binding protein	1 (2)
	Toxon-antitoxin system, toxin component, *RelE* family	1 (1)
Flavobacteriaceae		
	Hypothetical transmembrane protein	1 (3)
	Hypothetical	1 (1)
	FRG domain protein	1 (1)
Gammaproteobacteria		
	Hypothetical membrane protein	1 (3)
	Type II restriction enzyme	1 (1)

	BanI	
	DNA (cytosine-5-)-methyltransferase	1 (1)
	Hypothetical	7 (7)
Kosmotoga olearia		
	Methyltransferase type 11	1 (8)
Ricinus communis		
	Hypothetical protein	1 (10)
Spirochaeta caldaria DSM 7334		
	toxin-antitoxin system, toxin component, PIN family (PilT domain)	1 (4)
	Prevent-host-death family	1 (1)
	Hypothetical	4 (20)
Treponema phagedenis F0421		
	Restriction endonuclease	3 (3)
	Hypothetical	4 (4)
Treponema succinifaciens DSM 2489		
	XRE family transcriptional regulator	1 (1)
	Plasmid maintenance system killer	1 (1)
	hypothetical	8 (13)
	Transcriptional modulator of *MazE*/toxin, *MazF*	1 (5)

[1]: The numbers indicate the unique gene numbers by clustering the genes using a 97% identity threshold. Number of genes before clustering is shown within parentheses.

5.3.7 Most integron genes are unique to samples and individuals

In order to characterize the cassette genes shared among different samples, we clustered genes from different samples using CD-HIT [92], with an identity cutoff of 70% at the amino acid level and then mapped the clustered genes to samples. Figure 26 clearly shows that gene sharing among samples is minimal. Most of the genes uniquely belong to only one sample—only 84 genes are shared between exactly two samples and 63 genes are shared among three or more samples. This finding is consistent with the findings from [124] that integron genes from 12 *Vibrio* isolates share only a very small number (< 10%) of genes. The HMP cohort contains individuals who were sampled at multiple body sites and visits, enabling us to compare the sharing of the integron cassette genes within and across individuals. The list of samples from the same individual is detailed in Table 14. We calculated the proportion of shared genes between any two samples and found that samples from the same individual tend to share more genes than samples from different individuals: the average proportion of gene shared between samples from the same individual is 13%, and the average proportion of genes shared between samples from different individuals is slightly lower: 8%. Note again that the result is consistent with the report that *Vibrio* isolates share < 10% of their integron genes. Our results indicate that even within an individual, there is strong population subdivision between *Treponema* species collected at different sites.

The functions of the shared genes also vary, and the majority of them are still of unknown function: for the 84 genes shared between two samples, 56 genes cannot be assigned to any COG function, and 19 are assigned to unknown function (category R or S). Similarly, for the 63 genes shared by three or more samples, 30 genes do not hit to any COG function and 17 genes hit to unknown functions. Overall, the percentage of shared genes with an unknown function is 83%. This number is similar to the proportion for all 826 genes. Furthermore, the number of genes in category L (replication, recombination, and repair) is again the highest among all categories with known functions. These numbers hint that the genes shared among two or more samples are sampled from all integron genes, without any preference for genes of certain functions.

Table 14. List of samples with predicted integron genes that belong to the same individuals.

Individual ID	Sample IDs
158499257	SRS022602,SRS011152
159571453	SRS024441,SRS013836
763577454	SRS014477,SRS014476,SRS014470
764143897	SRS015215,SRS051930
160158126	SRS047634,SRS018157
675950834	SRS050244,SRS055401
638754422	SRS022149,SRS022143
159814214	SRS047113,SRS050669,SRS017209
764083206	SRS019071,SRS015434
763961826	SRS014691,SRS019029
158479027	SRS011126,SRS011115
763840445	SRS063999,SRS014578,SRS014573
765701615	SRS058808,SRS049389

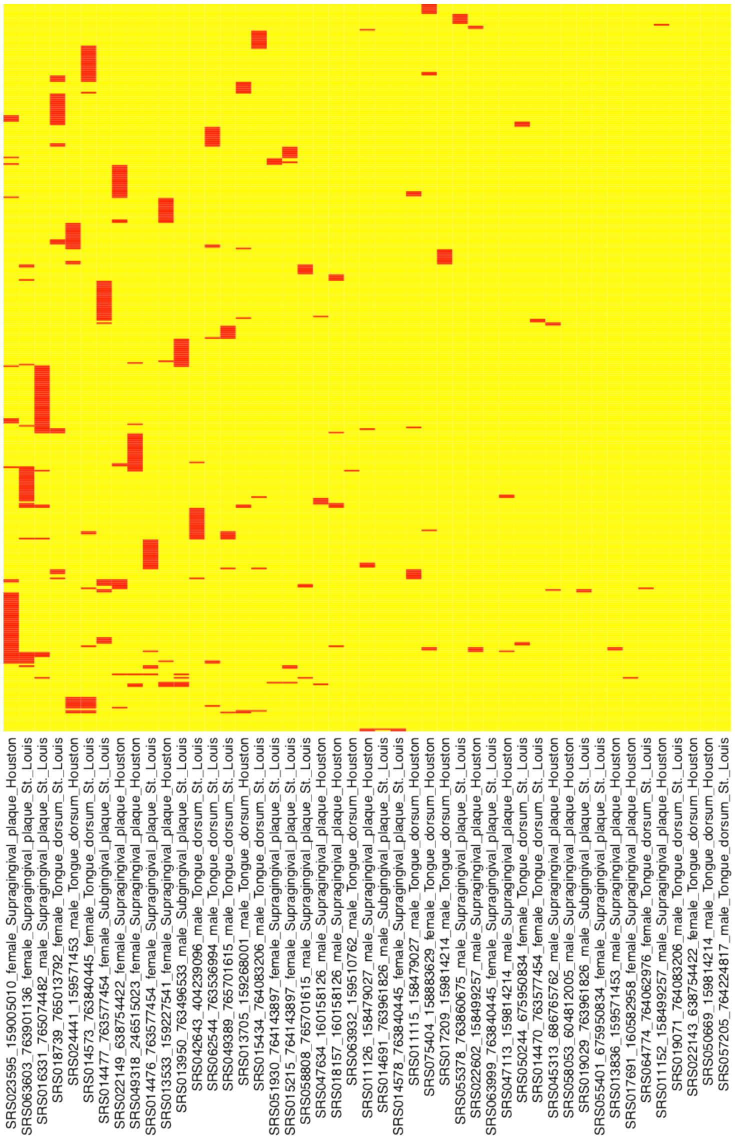


Figure 26. Sharing of gene cassettes among the samples. In this map, columns are the samples and rows are the genes found in the integron gene cassettes, clustered at 70% sequence identify at the amino acid level (by CD-HIT). A red cell means that the corresponding gene exists in the corresponding sample. The naming convention for the samples is SRS-ID_individual-ID_female/male_body-site_location. Note some samples are from the same individual (with the same individual-ID).

5.4 Discussion

To assemble integron gene cassettes, we designed a novel method to trace the de Bruijn assembly graph and then extract sequences bounded by contigs that contain *attC* sites. Assembly approaches based on de Bruijn graphs typically report the sequences of the edges (i.e., contigs) while discarding the connections between contigs embedded in the graph—the ambiguous connections between contigs may be difficult to resolve if no further information can be applied [25]. Our novel constrained assembly approach to integron gene cassettes enables us to traverse between the contigs in the de Bruijn graph by applying further information learned from the integron structures. The effect is enormous, as we obtained 826 genes *de novo* using this approach, compared to only 66 genes in the whole-metagenome assembly contigs.

Our integron gene discovery pipeline includes two validation steps (step 4 and step 5): only genes encoded by the sequences that are supported by reads mapping (step 4) and contain 1-3 genes (step 5) will be reported as candidate integron genes. For the HMP datasets, only 22% of sequences passed the first validation process, and 56% genes passed the second. We did not observe any misassembled integron genes when we applied the pipeline to the simulated datasets. We cannot completely exclude the possibility of having misassemblies in the real HMP datasets, considering that the prediction of the integron genes may be affected by reads from unknown species. Also our method may miss some integron genes due to the heterogeneity of *attC* sites of the *Treponema* species in the real samples.

Our targeted assembly and constrained assembly approaches can in principle be applied to any metagenome containing integron system. Given the *attC* sites, we are able to detect species with the corresponding integrons and generate integron gene cassettes. For example, the coral-mucus-associated *Vibrio* integrons [124] can be used to detect this coral pathogen in ocean samples, such as the Sargasso Sea metagenomic samples [5]. By analyzing integron genes we can help to understand how this species evolves and co-exists with coral. We can also analyze genes from different sites (or depths) of the ocean and understand how bacteria in these sites interact with the outer environment. Even if species with integrons are of low abundance, we can still detect their existence in metagenomic samples, as in the case of *T. denticola*.

Note that our targeted assembly (used in this work to characterize the integron *attC* sites) was developed to characterize CRISPR arrays in metagenomic samples, as described in Chapter 4 and in [76]. CRISPR/Cas systems are a widespread class of adaptive immunity systems that bacteria and archaea mobilize against foreign nucleic acids; the CRISPR arrays contain repeats, and short spacers that are likely derived from viral genomes or plasmids. Because the spacers in CRISPR arrays are significantly shorter than Illumina reads, we could easily assemble CRISPR arrays using targeted assembly alone, by first collecting reads containing repeats and then assembling the reads using optimized parameters. By contrast, integron spacers (cassettes) contain 1–3 genes between the *attC* sites, so it is hard to assemble the gene cassettes using targeted assembly alone. The constrained assembly approach was developed to overcome this limitation, and allows the assembly and characterization of integron gene cassettes. Both applications (the identification of the CRISPR arrays using the targeted assembly approach, and the identification of integron gene cassettes) demonstrate the importance of directed computational approaches for studies of important functional elements—which are poorly analyzed using generalized computational approaches (such as whole-metagenome assembly)—and that they are essential for the analysis of metagenomic sequences.

6. Research Summary

In this thesis I developed several different methods to improve the mining and annotation of functional elements in metagenomic datasets. Since the discovery of functional elements usually starts from whole (meta-)genome assemblies, my first attempt was to improve metagenome assemblies by first binning metagenomic datasets, to separate reads sampled from species of different abundances. I developed AbundanceBin, which pioneered the abundance-based binning approaches and can be used alone to reveal the structure of a microbial community, or combined with assemblers to improve metagenome assemblies.

More specialized methods were also developed, each for a type of functional elements to complement the approach to whole metagenome assembly. The first type of functional elements that we focused is genes. I developed GeneStitch, which is based on a network matching algorithm, to improve gene assembly from metagenomic sequences. GeneStitch is able to connect and assemble genes scattered in many different contigs into longer and more complete ones with the help of reference genes. Tests of GeneStitch revealed that it is capable of generating more complete genes or longer genes on top of the metagenomic assembly results. Such an improvement is important, as it has been shown that short gene fragments are difficult to annotate.

Besides genes, I also developed methods to improve the characterization of two more special types of functional elements (CRISPRs and integrons) from metagenomic sequences. CRISPRs play an important role in the immune system of bacteria and archaea. The targeted assembly method that I developed is very effective in retrieving CRISPRs from metagenomic sequences, which allows us to draw a more comprehensive picture of the CRISPR systems in bacteria and their dynamics in human microbiomes. As important agents of bacterial evolution, integrons are genetic elements capable of acquiring, rearranging and expressing genes contained in gene cassettes, bounded by direct repeats. The constrained assembly approach that I developed targets integron genes, utilizing the fact that integron genes are bounded by direct repeats. Application of the integron discovery system to the HMP datasets significantly enriched the gene pool of chromosomal integrons. Both applications (the identification of the CRISPR arrays using the targeted assembly approach, and the identification of integron gene cassettes) demonstrate the importance of directed computational approaches for studies of important functional elements and that they are essential for the analysis of metagenomic sequences.

7. Appendix

The symbols are exactly the same as those defined in section 3.2.1 Network matching algorithm for gene assembly. Let $S(i,j,k)$ be the optimal alignment score between all possible paths ending at position i of contig k in the input de Bruijn graph and the prefix ending at position j (i.e., $t_1 t_2 \cdots t_j$) of the input reference sequence. For each contig C_k, we denote its first letter as $first(k)$, and its last letter as $last(k)$. The network matching algorithm computes a dynamic programming matrix to record the optimal alignment scores for$1 \leq i \leq last(k)$, $1 \leq j \leq m$, and $1 \leq k \leq n$ (n is the total number of contigs). $S(i,j,k)$ can be computed recursively as

$$S(i,j,k)$$

$$= max\left\{\begin{array}{ll}
S(i-3,j-1,k)+sim(aa([i-2,k],[i-1,k],[i,k]),j) & ,if\ i>3\\
\underset{l\ precedes\ k}{\max}\ S(last(l),j-1,l)+sim(aa([1,k],[2,k],[3,k]),j) & ,if\ i=3\\
\underset{l\ precedes\ k}{\max}\left.\begin{cases} S(last(l)-1,j-1,l)+sim(aa([last(l),l],[1,k],[2,k]),j) & ,if\ last(l)>1\\ \underset{m\ precedes\ l}{\max}\{S(last(m),j-1,m)+sim(aa([1,l],[1,k],[2,k]),j) & ,if\ last(l)=1\end{cases}\right\} & ,if\ i=2\\
\underset{l\ precedes\ k}{\max}\left.\begin{cases} S(last(l)-2,j-1,l)+sim(aa([last(l)-1,l],[last(l),l],[1,k]),j) & ,if\ last(l)>2\\ \underset{m\ precedes\ l}{\max}\ S(last(m),j-1,m)+sim(aa([1,l],[2,l],[1,k]),j) & ,if\ last(l)=2\\ \underset{m\ precedes\ l}{\max}\left.\begin{cases} S(last(m)-1,j-1,m)+sim(aa([last(m),m],[1,l],[1,k]),j) & ,if\ last(m)>1\\ \underset{n\ precedes\ m}{\max}\ S(last(n),j-1,n)+sim(aa([last[m],m],[1,l],[1,k]),j) & ,if\ last(m)=1\end{cases}\right\} & ,if\ last(l)=1\end{cases}\right\} & ,if\ i=1\\
I(i,j,k) & \\
D(i,j,k) &
\end{array}\right.$$

where i is used to indicates the position of nucleotide in contig k. The symbol $aa([i,p],[j,q],[k,r])$ represents the translated amino acid from the codon triplet, which is composed of nucleotide at position i of contig p, nucleotide at position j of contig q, and nucleotide at position k of contig r; $sim(m,n)$ indicates the BLOSUM62 score between amino acid m and n. $I(i,j,k)$ and $D(i,j,k)$ are the optimal alignment scores between the paths of the de Bruijn graph (ending at position i in contig k) and the prefix of input reference sequence (ending at position j), ending with insertion and deletion in the alignment, respectively. The recursive definitions of $I(i,j,k)$ and $D(i,j,k)$ are as follows:

$$I(i,j,k) = max\begin{cases} \begin{array}{l} S(i-3,j,k)+\Delta g\_open \\ I(i-3,j,k)+\Delta g\_ext \end{array} & ,if\ i>3 \\ \max\limits_{l\ precedes\ k}\left\{\begin{array}{l} S(last(l),j,l)+\Delta g\_open \\ I(last(l),j,l)+\Delta g\_ext \end{array}\right\} & ,if\ i=3 \\ \max\limits_{l\ precedes\ k}\left\{\begin{array}{ll} \begin{array}{l} S(last(l)-1,j,l)+\Delta g\_open \\ I(last(l)-1,j,l)+\Delta g\_ext \end{array} & ,if\ last(l)>1 \\ \max\limits_{m\ precedes\ l}\left\{\begin{array}{l} S(last(m),j,m)+\Delta g\_open \\ I(last(m),j,m)+\Delta g\_ext \end{array}\right\} & ,if\ last(l)=1 \end{array}\right\} & ,if\ i=2 \\ \max\limits_{l\ precedes\ k}\left\{\begin{array}{ll} \begin{array}{l} S(last(l)-2,j,l)+\Delta g\_open \\ I(last(l)-2,j,l)+\Delta g\_ext \end{array} & ,if\ last(l)>2 \\ \max\limits_{m\ precedes\ l}\left\{\begin{array}{l} S(last(m),j,m)+\Delta g\_open \\ I(last(m),j,m)+\Delta g\_ext \end{array}\right\} & ,if\ last(l)=2 \\ \max\limits_{m\ precedes\ l}\left\{\begin{array}{ll} \begin{array}{l} S(last(m)-1,j,m)+\Delta g\_open \\ I(last(m)-1,j,m)+\Delta g\_ext \end{array} & ,if\ last(m)>1 \\ \max\limits_{n\ precedes\ m}\left\{\begin{array}{l} S(last(n),j,n)+\Delta g\_open \\ I(last(n),j,n)+\Delta g\_ext \end{array}\right\} & ,if\ last(m)=1 \end{array}\right\} & ,if\ last(l)=1 \end{array}\right\} & ,if\ i=1 \end{cases}$$

$$D(i,j,k) = max\begin{cases} S(i,j-1,k) + \Delta g\_open \\ \\ D(i,j-1,k) + \Delta g\_ext \end{cases}$$

where Δg_open and Δg_ext are affine penalties [65] for opening and extending gaps, respectively.

8. References

1. Schloss, P.D. and J. Handelsman, *Metagenomics for studying unculturable microorganisms: cutting the Gordian knot.* Genome Biol, 2005. **6**(8): p. 229.
2. Handelsman, J., et al., *Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products.* Chem Biol, 1998. **5**(10): p. R245-9.
3. Tyson, G.W., et al., *Community structure and metabolism through reconstruction of microbial genomes from the environment.* Nature, 2004. **428**(6978): p. 37-43.
4. DeLong, E.F., et al., *Community genomics among stratified microbial assemblages in the ocean's interior.* Science, 2006. **311**(5760): p. 496-503.
5. Venter, J.C., et al., *Environmental genome shotgun sequencing of the Sargasso Sea.* Science, 2004. **304**(5667): p. 66-74.
6. Fierer, N. and R.B. Jackson, *The diversity and biogeography of soil bacterial communities.* Proc Natl Acad Sci U S A, 2006. **103**(3): p. 626-31.
7. Tringe, S.G., et al., *Comparative metagenomics of microbial communities.* Science, 2005. **308**(5721): p. 554-7.
8. Garcia Martin, H., et al., *Metagenomic analysis of two enhanced biological phosphorus removal (EBPR) sludge communities.* Nat Biotechnol, 2006. **24**(10): p. 1263-9.
9. Yergeau, E., et al., *The functional potential of high Arctic permafrost revealed by metagenomic sequencing, qPCR and microarray analyses.* ISME J, 2010. **4**(9): p. 1206-14.
10. Jung, J.Y., et al., *Metagenomic analysis of kimchi, a traditional Korean fermented food.* Appl Environ Microbiol, 2011. **77**(7): p. 2264-74.
11. Turnbaugh, P.J., et al., *An obesity-associated gut microbiome with increased capacity for energy harvest.* Nature, 2006. **444**(7122): p. 1027-31.

12. Turnbaugh, P.J., et al., *A core gut microbiome in obese and lean twins.* Nature, 2009. **457**(7228): p. 480-4.

13. Qin, J., et al., *A human gut microbial gene catalogue established by metagenomic sequencing.* Nature, 2010. **464**(7285): p. 59-65.

14. Arumugam, M., et al., *Enterotypes of the human gut microbiome.* Nature, 2011. **473**(7346): p. 174-80.

15. Peterson, J., et al., *The NIH Human Microbiome Project.* Genome Res, 2009. **19**(12): p. 2317-23.

16. Hutchison, C.A., 3rd, *DNA sequencing: bench to bedside and beyond.* Nucleic Acids Res, 2007. **35**(18): p. 6227-37.

17. Margulies, M., et al., *Genome sequencing in microfabricated high-density picolitre reactors.* Nature, 2005. **437**(7057): p. 376-80.

18. Bentley, D.R., *Whole-genome re-sequencing.* Curr Opin Genet Dev, 2006. **16**(6): p. 545-52.

19. Sanger, F., S. Nicklen, and A.R. Coulson, *DNA sequencing with chain-terminating inhibitors.* Proc Natl Acad Sci U S A, 1977. **74**(12): p. 5463-7.

20. The_1000_Genomes_Project_Consortium, *A map of human genome variation from population-scale sequencing.* Nature, 2010. **467**(7319): p. 1061-73.

21. Genome_10K_Community_of_Scientists, *Genome 10K: a proposal to obtain whole-genome sequence for 10,000 vertebrate species.* J Hered, 2009. **100**(6): p. 659-74.

22. de Magalhaes, J.P., C.E. Finch, and G. Janssens, *Next-generation sequencing in aging research: emerging applications, problems, pitfalls and possible solutions.* Ageing Res Rev, 2010. **9**(3): p. 315-23.

23. Altschul, S.F., et al., *Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.* Nucleic Acids Res, 1997. **25**(17): p. 3389-402.

24. Wommack, K.E., J. Bhavsar, and J. Ravel, *Metagenomics: read length matters.* Appl Environ Microbiol, 2008. **74**(5): p. 1453-63.

25. Pop, M., *Genome assembly reborn: recent computational challenges.* Brief Bioinform, 2009. **10**(4): p. 354-66.

26. Markowitz, V.M., et al., *The integrated microbial genomes system: an expanding comparative analysis resource.* Nucleic Acids Res, 2010. **38**(Database issue): p. D382-90.

27. Huson, D.H., et al., *MEGAN analysis of metagenomic data.* Genome Res, 2007. **17**(3): p. 377-86.

28. Chakravorty, S., et al., *A detailed analysis of 16S ribosomal RNA gene segments for the diagnosis of pathogenic bacteria.* J Microbiol Methods, 2007. **69**(2): p. 330-9.

29. Monier, A., J.M. Claverie, and H. Ogata, *Taxonomic distribution of large DNA viruses in the sea.* Genome Biol, 2008. **9**(7): p. R106.

30. Ciccarelli, F.D., et al., *Toward automatic reconstruction of a highly resolved tree of life.* Science, 2006. **311**(5765): p. 1283-7.

31. von Mering, C., et al., *Quantitative phylogenetic assessment of microbial communities in diverse environments.* Science, 2007. **315**(5815): p. 1126-30.

32. Wu, M. and J.A. Eisen, *A simple, fast, and accurate method of phylogenomic inference.* Genome Biol, 2008. **9**(10): p. R151.

33. Bentley, S.D. and J. Parkhill, *Comparative genomic structure of prokaryotes.* Annu Rev Genet, 2004. **38**: p. 771-92.

34. Teeling, H., et al., *TETRA: a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in DNA sequences.* BMC Bioinformatics, 2004. **5**: p. 163.

35. Woyke, T., et al., *Symbiosis insights through metagenomic analysis of a microbial consortium.* Nature, 2006. **443**(7114): p. 950-5.

36. Chatterji, S., et al., *CompostBin: A DNA composition-based algorithm for binning environmental shotgun reads.* The 12th Annual International Conference on Research in Computational Molecular Biology, RECOMB 2008, 2008: p. 17-28.

37. Diaz, N.N., et al., *TACOA: taxonomic classification of environmental genomic fragments using a kernelized nearest neighbor approach.* BMC Bioinformatics, 2009. **10**: p. 56.

38. Yang, B., et al., *Unsupervised binning of environmental genomic fragments based on an error robust selection of l-mers.* BMC Bioinformatics, 2010. **11 Suppl 2**: p. S5.

39. Zhou, F., V. Olman, and Y. Xu, *Barcodes for genomes and applications.* BMC Bioinformatics, 2008. **9**: p. 546.

40. Pevzner, P.A., H. Tang, and M.S. Waterman, *An Eulerian path approach to DNA fragment assembly.* Proc Natl Acad Sci U S A, 2001. **98**(17): p. 9748-53.

41. Zerbino, D.R. and E. Birney, *Velvet: algorithms for de novo short read assembly using de Bruijn graphs.* Genome Res, 2008. **18**(5): p. 821-9.

42. Simpson, J.T., et al., *ABySS: a parallel assembler for short read sequence data.* Genome Res, 2009. **19**(6): p. 1117-23.

43. Li, R., et al., *De novo assembly of human genomes with massively parallel short read sequencing.* Genome Res, 2010. **20**(2): p. 265-72.

44. Peng, Y., et al., *Meta-IDBA: a de Novo assembler for metagenomic data.* Bioinformatics, 2011. **27**(13): p. i94-101.

45. Laserson, J., V. Jojic, and D. Koller, *Genovo: de novo assembly for metagenomes.* J Comput Biol, 2011. **18**(3): p. 429-43.

46. Lai, B., et al., *A de novo metagenomic assembly program for shotgun DNA reads.* Bioinformatics, 2012. **28**(11): p. 1455-62.

47. Ye, Y. and H. Tang, *An ORFome assembly approach to metagenomics sequences analysis.* J Bioinform Comput Biol, 2009. **7**(3): p. 455-71.

48. Salzberg, S.L., et al., *Gene-boosted assembly of a novel bacterial genome from very short reads.* PLoS Comput Biol, 2008. **4**(9): p. e1000186.

49. Jansen, R., et al., *Identification of genes that are associated with DNA repeats in prokaryotes.* Mol Microbiol, 2002. **43**(6): p. 1565-75.

50. Sorek, R., V. Kunin, and P. Hugenholtz, *CRISPR--a widespread system that provides acquired resistance against phages in bacteria and archaea.* Nat Rev Microbiol, 2008. **6**(3): p. 181-6.

51. van der Oost, J., et al., *CRISPR-based adaptive and heritable immunity in prokaryotes.* Trends Biochem Sci, 2009. **34**(8): p. 401-7.

52. Barrangou, R., et al., *CRISPR provides acquired resistance against viruses in prokaryotes.* Science, 2007. **315**(5819): p. 1709-12.

53. Cambray, G., A.M. Guerout, and D. Mazel, *Integrons.* Annu Rev Genet, 2010. **44**: p. 141-66.

54. Wu, Y.W. and Y. Ye, *A novel abundance-based algorithm for binning metagenomic sequences using l-tuples.* Lecture Notes in Computer Science (RECOMB 2010), 2010. **6044**: p. 535-549.

55. Wu, Y.W. and Y. Ye, *A novel abundance-based algorithm for binning metagenomic sequences using l-tuples.* J Comput Biol, 2011. **18**(3): p. 523-34.

56. Lander, E.S. and M.S. Waterman, *Genomic mapping by fingerprinting random clones: a mathematical analysis.* Genomics, 1988. **2**(3): p. 231-9.

57. White, J.R., et al., *Figaro: a novel statistical method for vector sequence removal.* Bioinformatics, 2008. **24**(4): p. 462-7.

58. Richter, D.C., et al., *MetaSim: a sequencing simulator for genomics and metagenomics.* PLoS One, 2008. **3**(10): p. e3373.

59. Huse, S.M., et al., *Accuracy and quality of massively parallel DNA pyrosequencing.* Genome Biol, 2007. **8**(7): p. R143.

60. DeLong, E.F., *Microbial community genomics in the ocean.* Nat Rev Microbiol, 2005. **3**(6): p. 459-69.

61. Wu, Y.W., et al., *Stitching Gene Fragments with a Network Matching Algorithm Improves Gene Assembly for Metagenomics.* accepted by European Conference on Computational Biology 2012 (ECCB 2012), 2012.

62. Hoff, K.J., *The effect of sequencing errors on metagenomic gene prediction.* BMC Genomics, 2009. **10**: p. 520.

63. Gelfand, M.S., A.A. Mironov, and P.A. Pevzner, *Gene recognition via spliced sequence alignment.* Proc Natl Acad Sci U S A, 1996. **93**(17): p. 9061-6.

64. Ye, Y., et al., *A segment alignment approach to protein comparison.* Bioinformatics, 2003. **19**(6): p. 742-9.

65. Vingron, M. and M.S. Waterman, *Sequence alignment and penalty choice. Review of concepts, case studies and implications.* J Mol Biol, 1994. **235**(1): p. 1-12.

66. Rho, M., H. Tang, and Y. Ye, *FragGeneScan: predicting genes in short and error-prone reads.* Nucleic Acids Res, 2010. **38**(20): p. e191.

67. Morgan, J.L., A.E. Darling, and J.A. Eisen, *Metagenomic sequencing of an in vitro-simulated microbial community.* PLoS One, 2010. **5**(4): p. e10209.

68. Li, H. and R. Durbin, *Fast and accurate short read alignment with Burrows-Wheeler transform.* Bioinformatics, 2009. **25**(14): p. 1754-60.

69. Hamady, M., et al., *Error-correcting barcoded primers for pyrosequencing hundreds of samples in multiplex.* Nat Methods, 2008. **5**(3): p. 235-7.

70. Gerlach, W. and J. Stoye, *Taxonomic classification of metagenomic shotgun sequences with CARMA3.* Nucleic Acids Res, 2011. **39**(14): p. e91.

71. Delcher, A.L., et al., *Improved microbial gene identification with GLIMMER.* Nucleic Acids Res, 1999. **27**(23): p. 4636-41.

72. Mathe, C., et al., *Current methods of gene prediction, their strengths and weaknesses.* Nucleic Acids Res, 2002. **30**(19): p. 4103-17.

73. Grabherr, M.G., et al., *Full-length transcriptome assembly from RNA-Seq data without a reference genome.* Nat Biotechnol, 2011. **29**(7): p. 644-52.

74. Zimin, A.V., et al., *Assembly reconciliation.* Bioinformatics, 2008. **24**(1): p. 42-5.

75. Horvath, P. and R. Barrangou, *CRISPR/Cas, the immune system of bacteria and archaea.* Science, 2010. **327**(5962): p. 167-70.

76. Rho, M., et al., *Diverse CRISPRs Evolving in Human Microbiomes.* PLoS Genet, 2012. **8**(6): p. e1002441.

77. Grissa, I., G. Vergnaud, and C. Pourcel, *The CRISPRdb database and tools to display CRISPRs and to generate dictionaries of spacers and repeats.* BMC Bioinformatics, 2007. **8**: p. 172.

78. Deltcheva, E., et al., *CRISPR RNA maturation by trans-encoded small RNA and host factor RNase III.* Nature, 2011. **471**(7340): p. 602-7.

79. Deveau, H., et al., *Phage response to CRISPR-encoded resistance in Streptococcus thermophilus.* J Bacteriol, 2008. **190**(4): p. 1390-400.

80. Deveau, H., J.E. Garneau, and S. Moineau, *CRISPR/Cas system and its role in phage-bacteria interactions.* Annu Rev Microbiol, 2010. **64**: p. 475-93.

81. Marraffini, L.A. and E.J. Sontheimer, *CRISPR interference: RNA-directed adaptive immunity in bacteria and archaea.* Nat Rev Genet, 2010. **11**(3): p. 181-90.

82. Andersson, A.F. and J.F. Banfield, *Virus population dynamics and acquired virus resistance in natural microbial communities.* Science, 2008. **320**(5879): p. 1047-50.

83. Heidelberg, J.F., et al., *Germ warfare in a microbial mat community: CRISPRs provide insights into the co-evolution of host and viral genomes.* PLoS One, 2009. **4**(1): p. e4169.

84. Held, N.L. and R.J. Whitaker, *Viral biogeography revealed by signatures in Sulfolobus islandicus genomes.* Environ Microbiol, 2009. **11**(2): p. 457-66.

85. Kunin, V., et al., *A bacterial metapopulation adapts locally to phage predation despite global dispersal.* Genome Res, 2008. **18**(2): p. 293-7.

86. Pride, D.T., et al., *Analysis of streptococcal CRISPRs from human saliva reveals substantial sequence diversity within and between subjects over time.* Genome Res, 2011. **21**(1): p. 126-36.

87. Grissa, I., G. Vergnaud, and C. Pourcel, *CRISPRFinder: a web tool to identify clustered regularly interspaced short palindromic repeats.* Nucleic Acids Res, 2007. **35**(Web Server issue): p. W52-7.

88. Bland, C., et al., *CRISPR recognition tool (CRT): a tool for automatic detection of clustered regularly interspaced palindromic repeats.* BMC Bioinformatics, 2007. **8**: p. 209.

89. Edgar, R.C., *PILER-CR: fast and accurate identification of CRISPR repeats.* BMC Bioinformatics, 2007. **8**: p. 18.

90. Rousseau, C., et al., *CRISPI: a CRISPR interactive database.* Bioinformatics, 2009. **25**(24): p. 3317-8.

91. Markowitz, V.M., et al., *IMG: the Integrated Microbial Genomes database and comparative analysis system.* Nucleic Acids Res, 2012. **40**(Database issue): p. D115-22.

92. Li, W. and A. Godzik, *Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences.* Bioinformatics, 2006. **22**(13): p. 1658-9.

93. Bhally, H.S., et al., *Leptotrichia buccalis bacteremia in two patients with acute myelogenous leukemia.* Anaerobe, 2005. **11**(6): p. 350-3.

94. Touchon, M. and E.P. Rocha, *The small, slow and specialized CRISPR and anti-CRISPR of Escherichia and Salmonella.* PLoS One, 2010. **5**(6): p. e11126.

95. Wu, Y.W., et al., *Oral spirochetes implicated in dental diseases are widespread in normal human subjects and carry extremely diverse integron gene cassettes.* Appl Environ Microbiol, 2012. **78**(15): p. 5288-96.

96. Stokes, H.W. and R.M. Hall, *A novel family of potentially mobile DNA elements encoding site-specific gene-integration functions: integrons.* Mol Microbiol, 1989. **3**(12): p. 1669-83.

97. Mazel, D., et al., *A distinctive class of integron in the Vibrio cholerae genome.* Science, 1998. **280**(5363): p. 605-8.

98. Boucher, Y., et al., *Integrons: mobilizable platforms that promote genetic diversity in bacteria.* Trends Microbiol, 2007. **15**(7): p. 301-9.

99. Coleman, N., et al., *An unusual integron in Treponema denticola.* Microbiology, 2004. **150**(Pt 11): p. 3524-6.

100. Rowe-Magnus, D.A., *Integrase-directed recovery of functional genes from genomic libraries.* Nucleic Acids Res, 2009. **37**(17): p. e118.

101. Nield, B.S., et al., *Recovery of new integron classes from environmental DNA.* FEMS Microbiol Lett, 2001. **195**(1): p. 59-65.

102. Rodriguez-Minguela, C.M., et al., *Worldwide prevalence of class 2 integrases outside the clinical setting is associated with human impact.* Appl Environ Microbiol, 2009. **75**(15): p. 5100-10.

103. Stokes, H.W., et al., *Structure and function of 59-base element recombination sites associated with mobile gene cassettes.* Mol Microbiol, 1997. **26**(4): p. 731-45.

104. Rowe-Magnus, D.A., et al., *The evolutionary history of chromosomal super-integrons provides an ancestry for multiresistant integrons.* Proc Natl Acad Sci U S A, 2001. **98**(2): p. 652-7.

105. Compeau, P.E., P.A. Pevzner, and G. Tesler, *How to apply de Bruijn graphs to genome assembly.* Nat Biotechnol, 2011. **29**(11): p. 987-91.

106. Dashper, S.G., et al., *Virulence factors of the oral spirochete Treponema denticola.* J Dent Res, 2011. **90**(6): p. 691-703.

107. Seshadri, R., et al., *Comparison of the genome of the oral pathogen Treponema denticola with other spirochete genomes.* Proc Natl Acad Sci U S A, 2004. **101**(15): p. 5646-51.

108. Matejkova, P., et al., *Complete genome sequence of Treponema pallidum ssp. pallidum strain SS14 determined with oligonucleotide arrays.* BMC Microbiol, 2008. **8**: p. 76.

109. Fraser, C.M., et al., *Complete genome sequence of Treponema pallidum, the syphilis spirochete.* Science, 1998. **281**(5375): p. 375-88.

110. Graber, J.R., J.R. Leadbetter, and J.A. Breznak, *Description of Treponema azotonutricium sp. nov. and Treponema primitia sp. nov., the first spirochetes isolated from termite guts.* Appl Environ Microbiol, 2004. **70**(3): p. 1315-20.

111. Thompson, J.D., T.J. Gibson, and D.G. Higgins, *Multiple sequence alignment using ClustalW and ClustalX.* Curr Protoc Bioinformatics, 2002. **Chapter 2**: p. Unit 2 3.

112. Waterhouse, A.M., et al., *Jalview Version 2--a multiple sequence alignment editor and analysis workbench.* Bioinformatics, 2009. **25**(9): p. 1189-91.

113. Bafna, V., H. Tang, and S. Zhang, *Consensus folding of unaligned RNA sequences revisited.* J Comput Biol, 2006. **13**(2): p. 283-95.

114. Muller, J., et al., *eggNOG v2.0: extending the evolutionary genealogy of genes with enhanced non-supervised orthologous groups, species and functional annotations.* Nucleic Acids Res, 2010. **38**(Database issue): p. D190-5.

115. Tatusov, R.L., et al., *The COG database: an updated version includes eukaryotes.* BMC Bioinformatics, 2003. **4**: p. 41.

116. Edgar, R.C., *MUSCLE: multiple sequence alignment with high accuracy and high throughput.* Nucleic Acids Res, 2004. **32**(5): p. 1792-7.

117. Eddy, S.R., *A new generation of homology search tools based on probabilistic inference.* Genome Inform, 2009. **23**(1): p. 205-11.

118. Frazer, K.A., et al., *VISTA: computational tools for comparative genomics.* Nucleic Acids Res, 2004. **32**(Web Server issue): p. W273-9.

119. Wilson, D., et al., *SUPERFAMILY--sophisticated comparative genomics, data mining, visualization and phylogeny.* Nucleic Acids Res, 2009. **37**(Database issue): p. D380-6.

120. Mandel, M.J., et al., *A single regulatory gene is sufficient to alter bacterial host range.* Nature, 2009. **458**(7235): p. 215-8.

121. Hattori, M. and T.D. Taylor, *The human intestinal microbiome: a new frontier of human biology.* DNA Res, 2009. **16**(1): p. 1-12.

122. Sanada, I. and S. Nishida, *Isolation of Clostridium Tetani from Soil.* J Bacteriol, 1965. **89**: p. 626-9.

123. Pohlschroeder, M., S. Leschine, and E. Canale-Parola, *Spirochaeta caldaria sp. nov., a thermophilic bacterium that enhances cellulose degradation by Clostridium thermocellum.* Archives of Microbiology, 1994. **161**: p. 17-24.

124. Koenig, J.E., et al., *Coral-mucus-associated Vibrio integrons in the Great Barrier Reef: genomic hotspots for environmental adaptation.* ISME J, 2011. **5**(6): p. 962-72.

Wu, Yu-Wei

School of Informatics and Computing
Indiana University

Education

09/2007 - 08/2012

Ph.D., Bioinformatics, School of Informatics and Computing, Indiana University
Minor: Statistics

09/1998 – 06/2000

M.S., Computer Engineering, National Tsing Hua University, Taiwan

09/1994 – 06/1998

B.S., Computer Science, TungHai University, Taiwan

Research and Working Experience

08/2008 - 08/2012	**Research Associate**, School of Informatics and Computing, Indiana University, Bloomington
09/2007 - 05/2008	**Associate Instructor for INFO-I308 "Information Representation"**, Indiana University, Bloomington
03/2005 - 06/2007	**Research Assistant**, Institute of Information Science, Academia Sinica, Taiwan.
01/2004 - 03/2005	**Software Engineer**, Network and Multimedia Institute, Institute for Information Industry, Taiwan.
09/2000 - 12/2003	**Software Engineer**, Embedded Systems Laboratory, Institute for Information Industry, Taiwan.
10/1998 - 06/1999	**Teaching Assistant for "Introduction to Computing"**, National Tsing-Hua University, Taiwan.

Publications

1. **Yu-Wei Wu**, Mina Rho, and Yuzhen Ye, "Inference of genes and gene graphs from de Bruijn graph assembly of metagenomes by network matching algorithm", accepted by 11th European Conference on Computational Biology (ECCB 2012).

2. **Yu-Wei Wu**, Mina Rho, Thomas Doak, and Yuzhen Ye "Oral spirochetes implicated in dental diseases are widespread in normal human subjects and carry extremely diverse integron gene cassettes", *Applied and Environmental Microbiology,* 78(15): 5288-5296, 2012.
3. Mina Rho, **Yu-Wei Wu**, Haixu Tang, Thomas Doak, and Yuzhen Ye "Diverse CRISPRs evolving in human microbiomes", *PLoS Genetics*, 8(6): e1002441, 2012.
4. **Yu-Wei Wu** and Yuzhen Ye, "A novel abundance-based algorithm for binning metagenomic sequences using *l*-tuples", *Journal of Computational Biology,* 18(3): 523-534, 2011.
5. **Yu-Wei Wu** and Yuzhen Ye, "A novel abundance-based algorithm for binning metagenomic sequences using *l*-tuples", *Lecture Notes in Computer Science,* 6044: 535-549, RECOMB 2010, 2010.
6. Michael Lee Salmans, Shu-Miaw Chaw, Ching-Ping Lin, Arthur Chun-Chieh Shih, **Yu-Wei Wu** and Michael R. Mulligan, "Editing site analysis in a gymnosperm mitochondrial genome reveals similarities with angiosperm mitochondrial genomes", *Current Genetics,* 56(5): 439-446, 2010.
7. Shu-Miaw Chaw, Arthur Chun-Chieh Shih, Daryi Wang, **Yu-Wei Wu**, Shu-Mei Liu and The-Yuan Chou, "The Mitochondrial Genome of the Gymnosperm *Cycas taitungensis* Contains a Novel Family of Short Interspersed Elements, Bpu sequences, and Abundant RNA Editing Sites", *Molecular Biology and Evolution,* 25(3): 603-615, 2008.
8. Daryi Wang, **Yu-Wei Wu**, Arthur Chun-Chieh Shih, Chung-Shien Wu, Ya-Nan Wang, Shu-Miaw Chaw, "Transfer of chloroplast genomic DNA to mitochondrial genome occurred at least 300 million years ago," *Molecular Biology and Evolution* 24(9): 2040-2048, 2007.
9. Arthur Chun-Chieh Shih, D.T. Lee, Chin-Lin Peng , and **Yu-Wei Wu**, "Phylo-mLogo: An interactive multiple-logo visualization tool for large-number sequence alignments", *BMC Bioinformatics* 8: 63, 10 pages, 2007.
10. Arthur C-C Shih, D T Lee, Laurent Lin, Chin-Lin Peng, Shiang-Heng Chen, **Yu-Wei Wu**, Chun-Yi Wong, Meng-Yuan Chou, Tze-Chang Shiao and Mu-Fen Hsieh.

"SinicView: A visualization environment for comparisons of multiple nucleotide sequence alignment tools", *BMC Bioinformatics*, 7: 103, 15 pages, 2006.

Posters

1. **Yu-Wei Wu** and Yuzhen Ye, "A novel abundance-based algorithm for binning metagenomic sequences using *l*-tuples" in 2010 *Industry Collaboration Workshop on Life Sciences Informatics*, Indiana University.
2. Arthur Chun-Chieh Shih, D.T. Lee, Chin-Lin Peng, and **Yu-Wei Wu**, "PhylomLogo: An interactive multiple-logo visualization tool for large-number sequence alignments," in 2006 *IEEE Computational Systems Bioinformatics Conference*.

Talks

1. "Metagenomics: challenges and opportunities to analyzing the mostly-unknown species in the environment", Institute of Computer Science, Academia Sinica, Taiwan, May 23 2011.
2. "Metagenomics: challenges and opportunities to analyzing the mostly-unknown species in the environment", Chung Cheng University, Taiwan, May 27 2011.
3. Guest lecturer, INFO-I690, Advanced Seminar I: Bioinformatics for Microbial Genomics and Metagenomics, Indiana University, Bloomington, April 20 2010.

Conference Presentation

1. " A novel abundance-based algorithm for binning metagenomic sequences using *l*-tuples", Fourteenth Internation Conference on Research in Computational Molecular Biology 2010 (RECOMB 2010), Lisbon, Portugal

Award

1. Travel Fellowship, 11th European Conference on Computational Biology, 2012.
2. Travel Grant, Taiwan Ministry of Education, 2010.

Patent

1. An effective method to reduce network load when transferring XML documents (with W. C. Sun), I237774, 8/11/2005 – 12/22/2022, Taiwan, R.O.C.

Certificate

1. Sun Certified Java Programmer (SCJP), 2001.

Technical Skills

- Languages: C, C++, C#, Java, Visual Basic, SQL, R, Perl, Stata
- Platform: Windows, Linux/Unix, Computer clusters

Software Development

- **NMGene**: A tool to infer long and unbroken genes from de Bruijn graph assembly of metagenomes by matching the graph against reference genes using network matching algorithm.
- **Integron gene finder**: A tool to find the integron genes from the metagenomic sequencing reads given the integron recombination sites.
- **AbundanceBin**: An abundance-based binning tool for classifying metagenomic reads.
- **Zinc Finger Binding Site Predictor**: A C#-based tool for graphically display the potential zinc finger binding sites given an input sequence.
- **Phylo-mLogo**: a multiple-logo alignment visualization tool, which allows the user to visualize the global profile of the whole multiple sequence alignment and to hierarchically visualize homologous logos of each clade simultaneously.
- **SinicView**: A Java-based environment alignment visualization tool, which allow users to efficiently evaluate and compare assorted alignment results obtained by different tools.