# ALGORITHM DEVELOPMENT FOR NEXT GENERATION SEQUENCING-BASED METAGENOME ANALYSIS

A Thesis
Presented to
The Academic Faculty

by

Andrey O. Kislyuk

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Bioinformatics

School of Biology
Georgia Institute of Technology
December 2010

# ALGORITHM DEVELOPMENT FOR NEXT GENERATION SEQUENCING-BASED METAGENOME ANALYSIS

Approved by:


Professor Joshua S. Weitz, Advisor
School of Biology
*Georgia Institute of Technology*

Professor I. King Jordan, Co-Advisor
School of Biology
*Georgia Institute of Technology*

Professor Nicholas H. Bergman
Department of Homeland Security
National Biodefense Analysis and Countermeasures Center
Adjunct Professor, School of Biology
*Georgia Institute of Technology*

Professor Yury Chernoff
School of Biology
*Georgia Institute of Technology*

Professor David A. Bader
School of Computational Science and Engineering
*Georgia Institute of Technology*



Date Approved: 13 August 2010

*To the ghost in the machine*

# ACKNOWLEDGEMENTS

I am deeply grateful to my advisor, Joshua Weitz, for his invaluable guidance, support, and encouragement. I would like to thank Inna Dubchak and Michael Brudno, who introduced me to computational biology, and King Jordan, my co-advisor, for their support and mentorship. I am also grateful to Nicholas Bergman for his support and advice; and I am thankful to Stephen Turner for his support, mentorship, and the opportunities he has given me. It has been a privilege to work with these brilliant and talented mentors.

I want to thank my collaborators and co-workers, especially Alexandre Lomsadze, Alex Mitrophanov, Alex Poliakov, Alla Lapidus, Scott Sammons, Russell Neches, Jonathan Eisen, Jonathan Dushoff, Bart Haegeman, and Mark Borodovsky, who have all contributed to my understanding of the science and practice of bioinformatics and computational biology. I would also like to thank Lee Katz, Sonia Agrawal, Matthew Hagen, Andrew Conley, Pushkala Jayaraman, Viswateja Nelakuditi, Jay Humphrey, Dhwani Govil, Raydel Mair, Kathleen Tatti, Maria Tondella, Brian Harcourt, Leonard Mayer, and Srijak Bhatnagar, who have contributed to the work that became part of this thesis.

I am thankful to my friends and instructors at Georgia Tech and Berkeley; to Nils Onsager for his dedicated vision and instruction; and to Madison Park for her support. Finally, I am thankful to my family for their love and support.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# SUMMARY

We present research on the design, development and application of algorithms for DNA sequence analysis, with a focus on environmental DNA (metagenomes). We present an overview and primer on algorithm development for bioinformatics of metagenomes; work on frameshift detection in DNA sequencing data; work on a computational pipeline for the analysis of bacterial genomes; work on phylogenetic clustering of metagenomic fragments; and work on estimation of bacterial genome plasticity and diversity.

# CHAPTER I

# INTRODUCTION

Since the development of the first DNA sequencing methods by Sanger, Maxam and Gilbert in the 1970s, the pace of discovery in biology has increased dramatically. The level of biological complexity that can be characterized is steadily rising. In the past decade, sequencing of DNA isolated directly from an environment (instead of a clonal colony of cells) has been added to the arsenal of tools available to biologists. The combined genomes of a community sharing an environment are known as the metagenome, and metagenomics is the science of analyzing these genomes together. Challenges arise from the fact that metagenomes cannot yet be sequenced with the fidelity available with single, isolate genomes.

The rate of progress in the design of computers is famously characterized by Moore's law: the number of transistors that can be placed on an integrated circuit doubles every two years, and the computational power of processors grows correspondingly. This continuous growth has supported dramatic progress in many areas, including biology. For example, computational infrastructure for the human genome project was designed with hardware unavailable at the time, but expected to arrive according to Moore's law.

Since the advent of second generation DNA sequencing technologies, the number of DNA bases produced by one sequencing instrument has increased at a pace which exceeds Moore's law. Combined with the continual increase in the number of DNA sequencing facilities – driven by their increased affordability – the total size of biological databases is growing at a speed far beyond that. For example, in one of the first metagenome sequencing projects, the acid mine drainage dataset was sequenced

in 2004 and yielded 80 MB of raw sequence data. In 2010, sequencing of the clinically relevant human gut metagenome yielded over 500 GB of raw sequence data. At the same time, limitations in the speed at which transistors can operate have constrained the performance of a single processor thread, and necessitated parallelization of computer programs. If a decade ago a computer workstation usually contained a single processor, modern workstations contain 2 to 16 general-purpose processor cores and hundreds of smaller, specialized ones.

Despite the increased affordability and massive throughput, second generation sequencing technologies suffer from a reduced average read length compared to first generation, Sanger or capillary sequencing. While this shortcoming will be addressed in third generation technologies soon to enter production use, the large installed user base and continued use of second generation machines will necessitate the ability to analyze these data for many years. Simultaneously, third generation sequencing technologies will offer the ability to observe single-molecule interactions, which will expand the horizons of biological methods yet again. In metagenomes, the improvements will allow higher fidelity, longer fragments to be sequenced.

All of these factors lead to the need to design, update and improve algorithms used for genome and metagenome analysis. In this thesis, we contribute several improvements to the algorithms and software used for metagenome and isolate genome analysis. The tools developed for isolate genomes are also applicable to metagenome data, except for the genome fluidity estimation procedure, which will be adapted to applications on metagenome data in a future work.

# CHAPTER II

# ALGORITHM DESIGN IN METAGENOMICS: A PRIMER

This primer is targeted toward beginning graduate students in bioinformatics. Its purpose is to contribute to an understanding of algorithm design and development for metagenomics, present the tools and methods used in bioinformatics and computational biology of metagenomes, and give a brief overview of up-to-date developments in computational metagenomics.

A metagenome is the combined set of genomes of a community of organisms sharing a particular environment. The community may consist of a variety of bacteria, viruses, microscopic eukaryotes, or microorganisms in conjunction with their host.

The earliest examples of metagenome DNA sequencing are surveys of environmental 16S (small ribosomal subunit) DNA [178]. These are covered in more detail in section 2.2.3. In the past five years, the increased affordability and throughput of second generation DNA sequencing has led to a great increase in the number of shotgun sequencing projects of unamplified or whole-genome amplified metagenomic DNA. The data from many of these projects are available in GenBank [19], others are seen only on centralized metagenomic analysis portals, most importantly CAMERA [149], IMG/M [113], and MG-RAST [15].

Shotgun DNA sequencing allows the recovery of a fraction of a metagenome in fragments of 40 to 2000 bases, or paired DNA fragments spaced at up to about 20Kb. Since the advent of DNA sequencing, a sophisticated set of tools has been developed for the analysis of complete or nearly complete genomes [96, 142, 182]. These tools rely on highly accurate finished DNA sequence, consisting of many overlapping reads covering the same locus of a given genome (at least 5x for Sanger sequencing, and

higher for second generation sequencing). When forced to operate on short, numerous, low coverage and therefore error-prone DNA fragments, these tools suffer from decreased performance. Many of the algorithms can be adjusted to this increased error rate, while others have to be replaced by different approaches or cannot be applied until third-generation sequencing technologies offer longer read lengths and higher accuracies.

Most early whole-genome sequencing projects concentrated on the sequencing of cultivated isolates implicated in previously characterized diseases or previously selected model organisms. This was appropriate for a number of reasons: such projects are directly medically relevant, scientific inferences about genotype-phenotype connections are much more tractable in isolates, previous knowledge about model organisms could be applied, etc.

For the purpose of answering questions such as "what is the total diversity of biological function in a given natural environment?", "what is the ecological community structure of organisms in a given environment?", "what is the evolutionary history of microbes in an environment, considering horizontal gene transfer?", and "how diverse and plastic are the genomes of bacteria of the same species in a natural community?", the sequencing of cultivated isolates has a number of shortcomings. First, most of these isolates are selected by their anthropic interest. Second, only isolates which could be cultivated are sequenced, leaving microbes with complex or unknown nutritional requirements unsequenced. Third, cultivated isolates undergo bottleneck effects which obscure the amount of diversity and plasticity of their genomes in the environment; in effect, the mere fact of using an isolate insulates the observer from detecting quasispecies states and horizontal gene transfer effects which can occur in the wild. Direct sequencing of metagenomes can address all of these drawbacks, while creating new challenges in the process.

Comparative analysis of genomes is the most powerful way of elucidating biological function of DNA sequences. With metagenomes, comparative analysis can be performed between or within samples: between, for spatial or temporal diversity, and within, for strain-level diversity or comparative abundance of strains or quasispecies within an environment.

Current metagenomic data are characterized by incompleteness, low coverage, and high error rate compared to isolate data, which makes them much harder to analyze. We will first describe the analysis tasks in metagenomics in light of a hypothetical situation where these shortcomings are mitigated, and then cover each area in more detail to explain the mitigation strategies and challenges. The full process is outlined in Figure 1.

## *2.1 Overview of metagenome analysis*

While future metagenomic analyses will include experimental techniques such as cell sorting [190] and longer read lengths [55], currently most metagenomic data appears as short, relatively error-prone contigs (contiguous fragments which consist of multiple reads) and single reads of size ranging from 50 bp (single microreads) to a few Kbp (large assembled contigs from abundant metagenome constituents). Gene annotation in such DNA is feasible [77, 192] but problematic given the possibility of frameshifts [81] in the coding region. A number of practical strategies are used to mitigate this. Two recent frameshift detection algorithms, one covered in Chapter 3 of this thesis, the other published by Antonov et al. [14], can be used to annotate likely frameshifts and speculatively correct them in silico or otherwise consider them. Quality values emitted by the sequencer for each base in the read or assembled contig can be used to filter possible frameshift or low-fidelity locations. The two approaches can be integrated together, although we are not aware of such an implementation.

Cell sorting techniques offer the ability to sequence amplified DNA from one cell

at a time [184] or unamplified DNA from clonal populations of cells. Currently, the sequencing of a large number of individual cells' genomes is still out of reach due to technological limitations, but this problem will be overcome with further throughput enhancements. Without cell sorting, assembly of metagenome reads must be configured to account for the danger of chimeras – DNA fragments which overlap and align together but come from different strains or species. In well-conserved genes such as the 16S small ribosomal subunit gene, long overlaps with perfect alignment may be produced by reads from different hosts, which may later be diagnosed by incongruent alignment patterns against a rRNA database (e.g. [48]), but may go undiagnosed and lead to erroneous inferences.

The final stage of a whole-genome sequencing project is the process of finishing – joining the gaps which remain between assembled reads and ensuring that every base in the genome has been sequenced a minimum number of times and the level of consensus between different reads covering that base is high (polishing). This task is currently close to impossible on metagenomes, where average coverage is low, gaps are abundant and long, and total diversity of the sample is unknown. Even if a particular metagenome constituent can be assembled to the level where finishing is possible, the process of gap closing requires targeted amplification of the regions containing the gaps, which may be impossible in a metagenomic sample.

Biological inferences from a microorganism's genome are usually made by comparing its genes against a database of genes, proteins and protein domains with known functions, then reconstructing metabolic networks using known connections between these genes. This task is much harder when the genome data is incomplete and no certainty exists that it came from the same strain. With gaps in the reconstructed gene network, only partial inferences or certain types of inferences can be made. However, different types of inferences are possible: metabolic networks can be recovered from the metagenome as a whole, without regard to which host each constituent gene

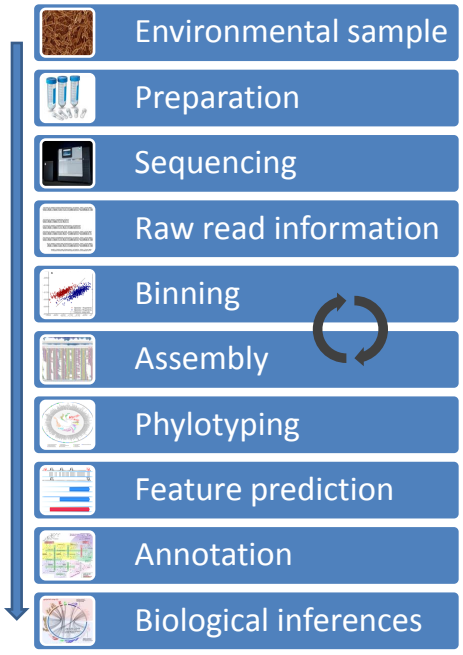



**Figure 1:** The metagenome analysis workflow.

came from, uncovering mutualism or other relationships between the constituents. Similarly, comparative analysis can be done over different environmental conditions and time series without regard to the hosts that the fragments belong to.

## *2.2 Major computational tasks in metagenome analysis*

### 2.2.1 Assembly

After DNA is sequenced, the first step in traditional genome analysis is sequence assembly. The output of this stage is a set of DNA fragment sequences (contigs) with associated quality scores for each position (as estimated by the sequencing machine on the basis of signal strength for each read and the level of consensus between multiple reads) and linkages between contigs (scaffolds) derived from mate pair or paired end reads. If the sequencing run had sufficient performance and coverage of the material, the contigs cover a large majority of the input genetic material (usually 95-99% of bacterial isolates and similar percentages of BACs or similar constructs into which

large genomes are partitioned for sequencing). However, there remain gaps where the input sequence is not covered by the contigs, and the full ordering of the contigs is not always certain. Gaps may also arise from positions with uncertainty due to repetitive regions with no reads spanning their entire length. These gaps and regions of low quality are processed semi-manually in a stage called finishing, where targeted PCR reactions are designed to span the gaps, or more sequencing is performed, sometimes using a combination of technologies.

This standard sequence assembly protocol is not fully applicable to metagenomic data. Many researchers do apply standard assembly algorithms to metagenome samples. This may be appropriate for situations where constituent genomes in the metagenome have dissimilar sequence content, and the intrinsic sequencing error rate is low, but when these conditions are not met, this approach risks the assembly of chimeras (combinations of reads from different clonal populations). At the same time, the relatively low average coverage of the metagenome results in a higher sequencing error rate than seen in isolate sequencing projects. Single-read sequencing errors can be subdivided into 3 types: insertions, deletions, and substitutions. Substitutions of single nucleotides can lead to incorrect knowledge about the identity of an amino acid in a protein encoded by the locus involved, modify the conformance to the expected sequence of a regulatory element, or cause a read to be erroneously considered as evidence for a new variant of this locus. Rarely, they can also introduce an erroneous stop codon that will break up an open reading frame encoded at the locus. Insertions and deletions of nucleotides can cause frameshifts in ORFs encoded at the locus, resulting in two ORFs with incomplete genes or one truncated ORF (the other one being too short to plausibly contain a gene), or they can lead to the same consequences as substitutions. Substitutions, insertions and deletions are also referred to as miscalls, overcalls and undercalls, respectively.

Sequencer and assembly outputs contain quality values (QVs) for each position,

which estimate the confidence in the nucleotide called at the position using the signal strength and error model of the sequencer. These values are normally used for assembly, where they help in calling the correct consensus nucleotide at each position using discordant information from multiple reads. They can also be used to mask or disregard positions in single reads of a metagenome, but high QVs are not a guarantee of an error-free read. Also, some error types, like undercalls or overcalls at DNA homopolymer positions (repeats of a single nucleotide), are not evenly distributed in the sequence due to error biases in the sequencing instrument. Assembly correction algorithms such as frameshift detection can be used to mitigate insertions and deletions (see Chapter 3), but again perfect accuracy cannot be expected. Thus many analyses based on the accuracy and full length of gene models, detection of polymorphism, etc. must be performed with an understanding of how the significance levels of their findings can be evaluated.

To reduce the likelihood of chimeric contigs, metagenomic binning can be applied before assembly [175, 62, 183]. Binning is the task of separating reads or contigs in the metagenome by their phylogenetic or functional origin. A trade-off must be considered when applying metagenomic binning and assembly. Binning methods are highly sensitive to the length and error rate of the read being sequenced, with binning accuracy and power increasing geometrically with length of the read, so assembly is desirable before binning; but assembly before binning is prone to chimeric results. One solution is to try using only high confidence settings or low error rate binning algorithms at this stage. Assembly can then be run on sequences which are predicted to come from the same phyla, but strain-level or even higher level variation may still result in chimeric contigs (although strain-level chimeras may in some cases be preferable to shorter, less informative contigs). A more granular, possibly less accurate binner can then be run on the assembled data. In large metagenomes, this process can be repeated iteratively, applying progressively more sensitive assembly

and binning thresholds until no more progress is seen in either step.

If the composition of the metagenome is known at least partially, for instance as a result of initial binning, multiple reference sequences can be supplied to the assembler for mapping [144]. Reads in the metagenome are then partitioned into two groups: those which accurately map to the reference sequences and those which do not, and the unmapped reads are independently assembled de novo into contigs of novel origin. However, this approach runs the risk of supplying references which are similar but not identical to the strains present in the metagenome, leading to incorrect assemblies; and it cannot be applied to constituent species whose genomes have not been seen before or for which no authoritative reference sequences exist.

Another approach to the problem of intertwined assembly and binning is to integrate the two processes. In addition to aligning reads and deriving chains of overlapping reads or k-mers, an assembly algorithm would also cluster reads and use binner-like probabilistic or heuristic methods to determine which reads should not be overlapped together due to suspicion of coming from multiple sources.

Finally, similar to the step of removing highly over-represented k-mers as one of the first steps in isolate genome assembly, metagenomic assembly may need to discount regions which align to sequences that are highly repetitive in other genomes, or are highly evolutionarily constrained, since they may provide paths from one source's reads to another's, resulting in chimeras.

Because different metagenomes can have very different depth of coverage, strain- and species-level diversity, enrichment bias, and instrument error profiles, the best way to assess assembler performance is to model these different parameters in synthetic datasets and check the quality of resulting assembly, then adjust assembler parameters and heuristics [139].

### 2.2.2 Binning

Metagenomic binning algorithms cluster or assign labels to genomic fragments collected from a metagenome. Strictly speaking, the task of binning refers to clustering of reads into distinct groups, and phylogenetic classification or categorization refers to assignment of phylogenetic labels in conjunction to or separately from binning. Less often, binning and categorization is done using functional criteria, for example by homology with a gene with known metabolic function, regardless of the diversity of strains from which the matches came from.

Machine learning algorithms are classified into three categories: supervised, semi-supervised, and unsupervised. Supervised algorithms use a set of labeled training data to build their model, then apply it to unlabeled input data. Semi-supervised algorithms use both labeled and unlabeled data to build the model. Unsupervised algorithms use no labeled data and build their model directly from the unlabeled input data.

Generally, supervised or semi-supervised algorithms can assign phylogenetic labels while binning, using labels that were given with their training data, and unsupervised algorithms cannot assign labels – they must be assigned using a post-processing step or with another algorithm.

On the feature space level, binning algorithms can be subdivided as homology based (those which use nucleotide or translated protein alignment to a database of reference sequences) and composition-based (those which use statistical patterns in distributions of short subsequences).

When a metagenome constituent is closely related to a previously sequenced isolate, homology-based methods offer the highest power for its detection and classification [29, 83]. However, when reads in the metagenome sample come from an uncharacterized species, and no close homologs exist in the database, homology-based methods cannot provide any meaningful information. This situation creates a duality

of applications, and combination-based methods [29] have been created to address it.

As demonstrated in [29], methods based on a combination of homology-based and composition-based binning have superior performance to methods based on just one type of binning. This is because supervised composition-based methods can recover relationships between sequences when sequence database coverage is insufficient to produce a homology-based match. Similarly, we hypothesize that unsupervised composition-based methods can enhance performance where supervised composition-based methods fail to produce a close match, by clustering unclassified or incompletely classified reads into putative operational taxonomic units. We cover an example implementation of this approach at the end of the next section.

Regular BLAST alignments with tabulation of top-scoring hits can be used for binning and work acceptably well when the species of all constituents of the metagenome are characterized. When metagenome constituents have a moderate level of divergence from genomes in the database, MEGAN [83] provides a significant improvement by assessing confidence of assignment through last common ancestor determination. This yields an estimate of the most detailed classification that can be derived with confidence from the phylogenetic tree given the reference genomes available.

In addition to BLAST and MEGAN, profile hidden Markov model-based protein family alignment [91, 148] has also been proposed as another type of homology-based binning.

A large variety of algorithms has been applied in the task of composition-based binning. So far, uses of support vector machines [115], HMM and IMM-derived statistics [168, 29], Markov chain Monte Carlo simulations (see Chapter 5), seeded growing self-organized maps [34], principal component analysis with spectral clustering [36], and k-nearest neighbor clustering [50] have been published. Further discussion of issues in metagenomic binning is given in Chapter 5.

### 2.2.3 Phylotyping and phylogeny reconstruction

The earliest methods in computational DNA sequence analysis [181] are applications of multiple alignments of the small subunit (16S) rRNA-coding gene to phylogeny reconstruction. Multiple sequence alignment and subsequent tree-building methods work best on well-conserved sequences with known patterns of selective pressure at each position. In addition to 16S rRNA for bacterial phylogenies, other well-conserved genes such as Rho and HSP70 in prokaryotes [73] and cytochrome c in eukaryotes have been identified in efforts to increase the resolution and confidence of the phylogenies. A set of genes present in almost all sequenced bacterial species has been identified [186] for this purpose. Further, where widespread marker genes provide insufficient resolution for strain-level phylogenies, multi-locus sequence typing (MLST) [107] uses concatenated interior segments of sets of up to ten genes, customized for strain typing within each species. When sequencing sexually reproducing organisms' genomes, or mixes of clonal populations of asexual organisms, haplotype or strain-level diversity analysis is now routinely performed by analyzing repeat clusters or single-nucleotide polymorphisms in the assembly. Finally, whole-genome alignment based methods of phylogeny reconstruction attempt to use alignments of as many components of the genomes as possible. On the other hand, the possibility of horizontal gene transfer (HGT) in unicellular organisms confounds this analysis, making per-gene phylogenies differ from one another in a given pair of genomes. This phenomenon can be addressed in two ways: one, the construction of reticulate trees (phylogenetic networks) to approximate the amount of HGT between the genomes; and two, the identification of genomic regions from a putative core genome, operationally defined in this context as a set of constitutive genes shared between members of the tree, such as the 16S or MLST genes above but possibly broader in scope.

Most levels of analysis described above can be applied to metagenomic data as a form of diversity estimation. Due to low coverage of most loci, and correspondingly

```
Reference   GCAGCTACAATCTGAGGCTCAGCTCCATCCCCGGAACGATGCC-GCGCAAGGATATTGAAA

Reads    >  GCAGCTACAATCTGAAGCTCTGCTACATCCC
         >  GCAGCTACAATCTGAAGCTCTGCTACATCCCCGGAACGATGCCCG
         >  GCAGCTACAATCTGAAGCTCAGCTACATC-CCGGAACGATGCC-GCGCAAGGATATTG
         <  GCAGCTACAATCTGAGGCTCTGCTACATCCCCGGAACGATGCC-GCGCAAGGATATTGAAA
         <       TACAATCTGAAGCTCAGCTACATCCCCGGAACGATGCC-GCGCAAGGATATTGAAA
         <            TCTGAAGCTCAGCTACATCCCCGGAACGATGCC-GCGCAAGGATATTG-AA
                           *    !   ^    *            *             ?
```

**Figure 2:** Example consensus determination problem in shotgun DNA sequence. Each row in the "Reads" pane represents a shotgun sequencing read mapped against a reference genome. (*) Likely sequencing errors. (!) Likely single-nucleotide polymorphism within sample. (^) Likely single-nucleotide polymorphism with respect to reference. (?) Ambiguous case.

high error rate in the reported sequence, this analysis must take into consideration the error model or quality values of the sequencing instrument. Many metagenome sequencing projects conduct 16S rRNA surveys (based on selective amplification of only the 16S genes) to estimate overall diversity. This can be done either as a pilot stage before or in parallel to whole genome amplification-based sequencing, or on its own when the aim is only to estimate total diversity in the sample.

#### *2.2.3.1 Example: Consensus base calling with phylotyping (variant calling).*

Consider a sequencer error model in which the probability of erroneous report with an insertion of base X is $k_X\epsilon$, the probability of deletion is $kD_X\epsilon$, and the probability of substitution of base $X$ for $Y$ is $kS_{XY}\epsilon$, where $\epsilon$ is the quality value reported by the base caller (itself a function of a signal processing pipeline which gauges signal-to-noise ratio in the read) and $k$ are coefficients derived by observing error characteristics of the sequencer in multiple alignments of its reads to known reference. The base caller is a program which interprets signal values from the sequencer and predicts (calls) the most likely nucleobases which produced the signals. (Sequencer error rates are also context-dependent, however we will omit this consideration for simplicity; the base caller may also have already taken this dependency into account when computing the

quality value. For a hypothetical example of this situation, see Figure 2.) The coefficients above will first be used in multiple sequence alignment of the reads. Assume after the alignment a total of 8 reads cover the position where the base call is being made, with quality values $Q = [q_1, \ldots q_8] = [40, 32, 30, 34, 35, 28, 4, 37]$ (where the expectation of the base being wrong is defined as $10^{-\frac{q}{10}}$) and the base predictions of the reads at the position are $p = [\texttt{A, A, T, A, T, G, T, T}]$. Then the probability of an $A$ being one of the variants is estimated as

$$P_A = \prod_{i \in [1,8]} P(r_i \in H_A) k S_{p_i A} \tag{1}$$

To estimate the most likely combination of haplotypes, all combinations of assignments of reads to haplotypes need to be evaluated. However, this task is $O(n^2)$ in the number of reads at the position, so a more efficient solution is used. Most possible combinations of haplotypes (e.g., those whose nucleotides are not already present in the alignment column) can be immediately discarded because their probability will never rise above the threshold for calling.

#### *2.2.3.2 Example*

. We now provide an example design of a phylogenetically guided ensemble decision tree classifier. This approach can be adapted to use multiple instances of the same type of binary classifier, such as SVM, or to use combined outputs of instances of multiple classifiers (this technique is referred to as stacking or blending).

The guiding principle behind this classification scheme will be to recursively subdivide the input sequences into sets by phylogenetic origin by running a binary classifier at each node in the phylogenetic guide tree from the root down. The phylogenetic guide tree is a subset of the full phylogenetic tree of all sequenced genomes, retrieved by mining the metagenome for gene sequences usable as opportunistic phylogenetic

markers. We define opportunistic phylogenetic markers as sequences which can provide with high confidence the assignment of the read they belong to in the overall phylogenetic tree.

The recursive subdivision of the dataset according to the tree structure is done in an effort to avoid introducing conflicting data from models in the other parts of the tree and to avoid "confusing" online self-training or semi-supervised algorithms by not running diverse data through them, which helps to avoid overfitting (also, this serves to obtain a performance advantage compared to schemes which run all reads through classifiers for all nodes). However, this design also increases vulnerability to misclassification since any read directed down a wrong branch by any one classifier will end up misclassified and may confound online models. This can be countered by checking the consistency of the classification and blending the outputs of different classifiers together.

The algorithm will proceed as follows. First, we will use a BLASTN search (TBLASTX could also be used) using the metagenome reads as the query and the non-redundant database (nr), possibly restricted to microorganisms or prokaryotes, depending on our expectations of constituents in the metagenome. Then, we will construct a phylogeny of species containing the hits in the database in a manner similar to MEGAN, but using an automatically built complete tree of life like the one built in AMPHORA, as opposed to a tree based on the NCBI taxonomy database. We will gauge the uniqueness of the top-scoring hit for each metagenome fragment and if the next best hit is not sufficiently less likely (using a heuristic cutoff), the species containing that hit will also be included in the tree.

Next, we will use the resulting phylogenetic guide tree and associated complete genomes to provide training sets for the binary classifiers that will be instantiated at each internal node of the tree. For example, the root of the tree has two branches which subdivide it into two sets of nodes. The binary classifier is trained on data

from one set labeled with label 1, and data from the other labeled with label 2. The performance of the classifier is then checked using cross-validation on the training sets fragmented with length distributions similar to those seen in the metagenome, and if no convergence occurs, the node is highlighted as unreliable or collapsed with a nearby node for lower-granularity classification.

More than one binary classifier can be used at each node, and non-binary classifiers can be used in binary mode. To combine outputs of the classifiers together, we can use a Bayesian belief network trained together with the classifiers, e.g., given the outputs of three classifiers, gsom1, svm1, and mcmc1, the combiner may use a function

$$P(r_x \in b_1) = \prod_{clas=gsom1,svm1,mcmc1} P(r_x \in b_1 | o_{clas}) P(o_{clas}) \tag{2}$$

where $r_x$ is read with index $x$, $b_1$ is the branch with index 1, and $o_{clas}$ is the output of classifier $clas$.

Moreover, classifiers can be excluded if their performance is consistently low at a certain node (or they achieve no convergence in training).

Note that $o_{clas}$ may be either a categorical or continuous variable, where a non-categorical value is the output of the likelihood function used by the classifier $clas$. Using this function, we can estimate the confidence of assignment for every metagenome read at every node. If the confidence is below a heuristic cutoff, we can stop subsequent classification and report assignment only down to the current node, reflecting a more coarse phylogenetic assignment.

### 2.2.4 Metabolic pathway reconstruction

Many metagenome sequencing projects have a goal of recovering the metabolic pathways present in constituents of the metagenome as a way to functionally characterize the community whose metagenome is being analyzed, e.g. [17]. In fact, the combined

analysis of DNA from many hosts, together with an enrichment or amplification strategy that can reflect the relative abundance of DNA coding for the metabolic process of interest (or transcriptome analysis), offers power beyond what sequencing of isolates can provide.

Current projects usually focus on subdivision of genetic material according to GO or EC term assignment based on homology, a form of functional binning. These hits can then be used to map onto a known metabolic network using, for example, the KEGG database [87] and highlight a particular pathway. Coverage and relative abundance of pathway components (both in terms of number of genes covered and the coverage of individual genes by reads) is taken as indication of relative abundance of metabolic activity. cDNA and EST studies (metatranscriptome sequencing) are also employed for this purpose [70, 16, 61, 65, 66].

### 2.2.5 Gene prediction and annotation

Protein-coding gene prediction is a key step in any genome analysis pipeline. This task is made much harder in metagenome assemblies since fragments may contain ORFs truncated from one or both sides and low coverage makes frameshift errors more likely. Gene predictors developed with the expectation of low error rate sequence can still be used with considerable success [192], but three updates are desirable. First, gene prediction models which incorporate frameshift detection, as opposed to using a post-processing step for frameshift prediction, can increase sensitivity and overall accuracy on short fragments. Second, models which can fit the statistical model of the coding frame accurately on very small amounts of training data can also increase sensitivity. Models which can take into account long k-mer statistics, such as the IMMs used in Glimmer [46], are more suitable for this task. Third, metagenomic gene prediction algorithms need to be tuned to call genes in truncated ORFs.

### 2.2.6 Technology advancement

Many metagenome sample analysis methods in use today will change and be replaced by other methods as the technology progresses. As mentioned in the introduction, cell sorting followed by amplification-free sequencing with very long read lengths will eventually become the method of choice for all genomic sequence analysis, but the progress toward this goal will be gradual and may take a decade or more. In the meantime, many techniques like the ones covered here will need to be employed to deal with the imperfect data.

Strobe read based analysis [141] is a new technology that extends the concept of paired reads and mate pairs to sequencing of multiple subreads from single contiguous fragments of DNA, potentially up to tens of kilobases in total length. This works by allowing a single polymerase molecule to sequence a long segment of DNA and observing it at staggered time intervals to mitigate the photodamage effects of continuous observation. This technique is very useful for repeat region traversal and scaffolding of low-complexity regions; it also offers a big advantage for metagenomes where it can serve as a scaffolding tool to aid binning of metagenomic fragments. In the long term, single-molecule sequencing of very long stretches of DNA is feasible, since it has been shown in vivo that a single polymerase can replicate the entire multi-megabase genome in some species. This will eliminate the need for binning as it is performed now, since very long reads with very long overlaps will allow easy assembly of clonal or almost identical populations and precise diversity analysis.

## *2.3 Algorithmic techniques*

Next, we will outline the paradigms prevalent in algorithm design for metagenome sequence analysis and note some specific implementations and considerations.

### 2.3.1 Feature selection

The task of elucidation of biological function from DNA, RNA and protein sequences lends itself to applications of machine learning algorithms and probabilistic modeling techniques. Many types of machine learning algorithms and probabilistic modeling techniques are applicable to biological sequences. In many cases, however, one must first decide the feature space on which the algorithms will operate, and the feature selection process becomes key to the algorithms' performance. For example, nucleotide k-mer statistics are used extensively in both gene finding and metagenomic binning. Sequence GC content is a k-mer statistic of first order, and it is widely known that distributions of nucleotide triplets (corresponding to codons when aligned with a protein coding frame) contain information usable for both of these tasks. Beyond that, meaningful over- and underabundances of nucleotide subsequences can be observed at much higher lengths [46]. However, any attempt to infer expected distributions of raw k-mers of length $k > 5$ runs into a shortage of data, because the length of an average bacterial genome is on the order of 5e6 nucleotides.

To avoid the shortage of data, we can use feature selection frameworks that select a subset of all k-mers that are over- or underrepresented in the data and are present in sufficient quantity to make their frequency estimate reliable. One such framework is principal component analysis (PCA), which selects linear combinations of features to explain variance in the sample; another is independent component analysis, which recovers coefficients of a linear combination of independent factors assumed to govern the process. More generally, a diversity of techniques can be used for nonlinear dimensionality reduction of the feature space. For example, the interpolated Markov model framework used in [46] effectively searches through the feature space of all possible gapped k-mer motifs of length up to 12 with 3 wildcards by default and selects those motifs with the best mutual information with the position of interest as features for the protein coding sequence model.

The interpolated Markov model is a modification of the hidden Markov model (HMM), widely used in DNA sequence alignment and feature prediction because the one-dimensional DNA sequence lends itself naturally to Markov models. When used for gene prediction, HMMs must either be modified to work as 3-periodic Markov models, to properly model the statistical distributions at the 3 codon positions, or must emit one symbol per nucleotide triplet. HMMs belong to the family of dynamic Bayesian networks and are particularly useful because of the efficient dynamic programming algorithms (Viterbi, forward-backward, and Baum-Welch algorithms) that exist for computing the most likely parameters of the model given the data and scoring the data according to the model (i.e., training and evaluation of the model). A more general type of dynamic Bayesian network algorithm is the conditional random field (CRF), which relaxes the uniformity constraints of HMMs and allows more flexible probability models, but loses the ability to use efficient training algorithms available for HMMs.

While unsupervised or semi-supervised machine learning algorithms will struggle with the curse of dimensionality (a term describing the exponential increase in the size of the search space with linear increase in the number of dimensions of data – usually equivalent to the number of features), many supervised algorithms are designed to work with high-dimensional data and select the relevant dimensions, i.e. they contain embedded feature selection algorithms. For example, artificial neural network training algorithms can be used to select relevant features from the input feature set. Support vector machines produce a coordinate transformation and dimension ranking as part of their model that can also be used for feature selection. For categorical data on which a topology and a distance metric cannot be naturally established, such as nucleotide k-mers, a random or annealed coordinate space reduction followed by use of information criteria such as AIC or BIC is possible. Alternatively, when using regression frameworks, a vector of regressor variables can be used to establish a

topology on the space; the topology selection itself can be done through a random search guided by a minimum mutual information or maximum variance criterion.

Given categorical data, and especially with small or unavailable training sets, self-training clustering algorithms (also known as density estimators) can be used to reveal patterns in data. We discuss one application of clustering in Chapter 5.

### 2.3.2 Randomized and approximation algorithms

Randomized algorithms employ a random search through the space of possible model parameters. Randomized algorithms are used extensively in computational biology. The Markov Chain Monte Carlo family of algorithms is particularly well adaptable to DNA sequence data. One such algorithm is Gibbs sampling, which finds the optimal joint distribution of the parameters of the model given the data by varying one parameter at a time, iterating through all parameters repetitively. Another algorithm is Metropolis-Hastings, a more general strategy similar to Gibbs sampling, which allows changing (perturbing) all of the model parameters at once.

The expectation-maximization algorithm is another model estimation technique, used when only the general model structure is known but no estimates of the parameters can be given. Its structure is similar to that of the Gibbs sampling algorithm, but without randomization at each step; only at random restarts of the algorithm. In each iteration, EM first determines the probability distribution of assignments of models to data, then re-estimates parameters of the models given the assignments.

Approximation algorithms avoid directly searching for the optimal solution to the problem at hand, which is usually NP-complete or NP-hard, but instead look for a solution guaranteed to satisfy an approximation guarantee and to have a bounded difference from the optimal solution. For example, algorithms on string overlap graphs used in sequence assembly solve Hamiltonian path or Eulerian path problems, but yield approximate solutions only.

### 2.3.3 String processing

A family of string processing algorithms is widely applied to the problem of fast non-exact sequence mapping and assembly which is at the core of shotgun genome sequencing assemblers and mappers. The naive problem of comparing all pairs of sequences against each other to find their overlaps is $O(n^2)$ in the number of sequences (where the comparisons themselves are pairwise sequence alignment problems) and $O(n^2)$ in the space required to store the matches, but if an index of short subsequences (k-mers, normally lengths of 8-32 nucleotides are used) is first constructed for all reads, the complexity is reduced to $O(n \log n)$ in time and $O(n)$ in space. This index is usually stored in a data structure called a suffix array. The overlap finding problem is then solved by finding co-occurrences of k-mers within pairs of reads. This requires perfect matches of a minimal length, which constrains the sensitivity to inexact matches somewhat but can usually be adjusted to obtain sensitivity beyond that required not to miss any matches.

A number of other techniques are commonly used to prevent the suffix array from consuming too much memory. Highly overrepresented k-mers are indicative of repetitive regions, and are not useful for overlap detection since other information must be used to distinguish true overlaps from repeats of a common subsequence. Such k-mers can be pruned or filtered from the index. Compressed suffix arrays use neighbor functions and adaptive coding to reduce the space requirement. The Burrows-Wheeler transform is used to permute characters in the reads into a pattern-grouping, more easily compressible string which retains the positional substring information for matching [31, 102] . The Ferragina-Manzini transform achieves even better theoretical results by combining the BWT and suffix array construction processes [154]. Locality improvements to suffix arrays are possible [155]. These allow even huge arrays which must be stored on disk to be accessed in a more linear manner, reducing seek-related penalties on rotational disks. While these algorithms are of most importance in de

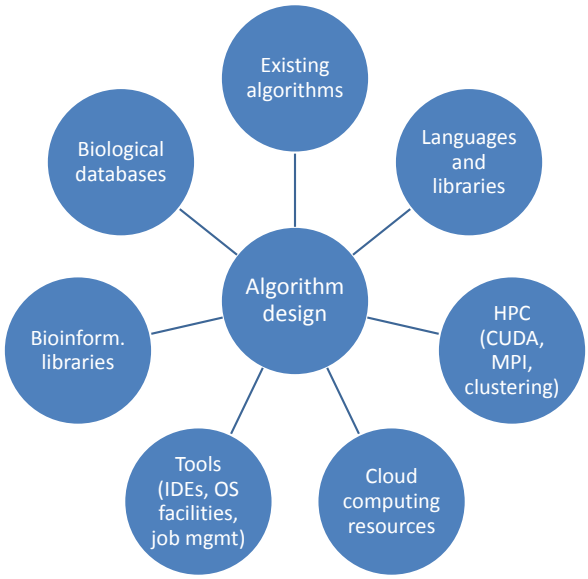



**Figure 3:** The metagenomics algorithm development toolkit.

novo assembly of large genomes using short reads, they are applicable to all shotgun sequencing datasets, in particular metagenome datasets.

Metabolic and regulatory pathway modeling is a well-developed field in isolate genome analysis; module-finding techniques [167, 84, 158, 125, 72, 52, 166, 101] [103] and network alignment techniques [75, 131, 41, 191, 180] are used to analyze protein-protein interaction networks and metabolic networks, which are constructed experimentally, predictively by homology, or by de novo predictions of molecular interactions. These models can be applied directly to metagenome data with the understanding that the pathway may span multiple organisms and the predictions may have reduced power compared to isolate genomes, in particular because of missing data which may lead to incomplete reconstruction of the pathway. In particular, a popular type of analysis evaluates gene abundance with the expectation that overrepresented genes are responsible for dominant metabolic processes; this is more biologically justified when analyzing environmental mRNA samples (metatranscriptomes).

## 2.4 Toolkit

Development of bioinformatics algorithms is much easier and faster if an appropriate set of tools is used. Accumulation of knowledge about existing software and algorithms is a time-consuming task, and it is therefore important to take advantage of software repositories, review articles, up-to-date texts and online community resources. Some components of the toolkit are illustrated in Figure 3; we cover them in more detail below.

The choice of programming language is one of the first questions facing a developer. Historically, Perl has been very popular in bioinformatics, due to its sophisticated string handling capabilities and other features. However, since it is an interpreted language, Perl is not suitable for high performance implementations of algorithms. Therefore a common pattern has been to prototype software in Perl and then write the high performance implementation in C or C++, often with a Perl wrapper for auxiliary tasks such as option parsing, I/O format conversion, thread and resource setup and teardown, Web interfaces, etc. Other languages that have been popular for bioinformatics programming include Python and Java. Correspondingly, popular general-purpose bioinformatics libraries exist for these languages: BioPerl [161], BioPython [42] and BioJava.

Description of the basic computer science skills necessary for algorithm design is outside the scope of this review, but a number of textbooks [8, 43] can provide key reference material. Beyond the core concepts covered in these texts, important aspects of commodity hardware architecture must be taken into account to maximize algorithm performance potential.

Many bioinformatics workflows are easily parallelizable. Two levels of parallel computing are available in commodity hardware. All modern workstation and server CPUs contain multiple cores, meaning that more than one processor is available. Taking advantage of parallelization is often as easy as dividing the input data into equal

parts using a simple workload manager and launching multiple instances of the worker thread. More sophisticated approaches can include a dispatch thread feeding data to worker threads, or a fully integrated multi-threaded program using a popular threading and parallelization API such as POSIX Threads, MPI or OpenMP. Also, new parallel programming frameworks are emerging [4, 6, 11]. These frameworks manage the complexity associated with shared data structures and thread communication in parallel programming, reducing the likelihood of hard-to-diagnose concurrency bugs.

The second common level of parallelization is seen in cluster computing. Commodity server hardware, usually running the Linux operating system, is organized into clusters of multiple nodes. A common hardware configuration in 2010 included 10 or more compute nodes with 2 CPU sockets each, providing up to 16 cores per node and 1 to 2 GB of RAM per core. The clusters are usually driven by a head node, which provides networked storage, scheduler and workload manager software. This configuration provides 160 cores and, with well-parallelizable workloads, a corresponding speed-up compared to single-core or single-node execution.

Another, new level of massive parallelization is seen in the field of general-purpose graphical processing unit (GPGPU) computing. This technology, advanced by NVIDIA under the name CUDA (Compute Unified Device Architecture), uses hundreds of small compute cores on a chip that is normally dedicated to high performance 3D graphics. While the cores are less powerful and have more limitations than CPU cores, there are many of them (128 to 512 cores in a common configuration) and they share massive bandwidth to their memory. A special software development kit, including a custom C/C++ compiler and parallel programming libraries, is necessary to compile code for this platform. Some bioinformatics tasks can achieve 50x speedup compared to CPU execution [172], and the theoretical maximum speedup is higher.

Finally, an important option for the infrastructure for computationally intensive tasks is cloud computing. This option allows the developer to rent resources on a

vendor-supplied compute infrastructure, customize this setup according to the requirements, and dynamically scale it up and down with load. This can remove the time and resource requirements for maintaining one's own computing infrastructure. This technology can be coupled with virtual machines, which are partitions of a physical computer set up to provide independent instances of the operating system to different users of the computer. From a bioinformatics software engineering standpoint, a virtual machine that is uploaded to the cloud and launched there allows one to configure the computing environment once and then deploy it remotely anywhere without worrying about potential incompatibilities and dependencies of the software being deployed on the remote OS. However, an important caveat with cloud computing is that large datasets such as raw sequencing data must first be transferred over the network to the cloud resource, and if the network connection is slow or expensive, it becomes the rate limiting step in either speed or cost effectiveness of analysis.

### 2.4.1 Testing and validation

The development of accurate machine learning and inference algorithms relies on the availability of "gold standard" data to train, test and validate the algorithms. With metagenomes in particular, this data has been hard to obtain, because no affordable methods exist to separate metagenome constituents for individual sequencing or ensure their complete sequencing. Therefore, most efforts to create metagenomic test sets have focused on the estimation of sampling parameters inherent in DNA amplification and shotgun sequencing, and application of these parameters to isolate genome sequencing data in artificial mixing scenarios. One effort [114] constructs artificial metagenomes from individual reads of isolate genome projects. Another effort [139] simulates metagenome samples from completed isolate genomes. A new project [121] experimentally quantifies the amplification bias and sequencing parameters of the metagenome of an artificial community whose constituents are known and have

been previously sequenced.

### 2.4.2 Conclusion

The amount and scope of data to be analyzed in the fields of bioinformatics and computational biology is increasing at a much faster pace than the number of available experts and new graduates. This presents an interesting challenge of knowledge management and productivity improvement. Fittingly, this is more true in metagenomics than in isolate genomics, since the estimated number of distinct species in various environments is measured in thousands, and the number of data points in time and spatial surveys of metagenomes is steadily rising. We hope that knowledge of the methods described in this chapter will enable the reader to successfully analyze such datasets and create tools which can be useful to the larger scientific community.

**Table 1:** Task-approach matrix: metagenome analysis tasks.

| Major tasks | Common approaches | Algorithmic techniques | Citations |
|---|---|---|---|
| Assembly | Overlap-Layout-Consensus; De Bruijn graphs | BW transform; FM transform; Suffix arrays; Index compression | [119, 118, 116, 106, 102, 154] |
| Binning | Pairwise alignment K-mer frequency spaces | Classifier algorithms | [29, 83, 148, 115, 34], Chapter 5 |
| Phylotyping and phylogeny reconstruction | Multiple alignment; Evolutionary modeling; Marker selection | Tree-building methods; Multiple alignment methods | [186, 30, 56] |
| Metabolic pathway reconstruction | Pairwise alignment; Comparison to known networks/pathways; Functional assignment | Network motif search; Graph/network alignment | [167, 84, 158, 125, 72, 52, 166, 101] [103] [75, 131, 41, 191, 180] |
| Gene prediction and annotation | Pairwise alignment; Self-training gene prediction | HMM, IMM; Motif search; Codon bias modeling | [46, 24, 192, 176, 108] |
| Technology advancement | Single-molecule, Strobe, Single-cell, Amplification-free, DNA modification sequencing | | [55, 141] |

# CHAPTER III

# FRAMESHIFT DETECTION

Next generation sequencing technologies accelerate genome sequence data acquisition, but introduce a higher risk of sequencing errors. Frameshift detection can reduce the overall cost of microbial genome finishing in whole-genome sequencing projects and decrease the error rate in metagenomic sequences. We have developed a combination of ab initio and alignment-based algorithms for frameshift detection, which can aid sequencing quality control. This unsupervised algorithm focuses on discrimination of frameshifts caused by sequencing errors from frameshifts that occur due to overlaps of adjacent genes located in the same DNA strand. An evaluation of the method's accuracy showed that its performance is comparable with the performance of the earlier developed program FrameD.

The rest of this chapter is based on published work which first appeared in the following article:

A. Kislyuk, A. Lomsadze, A. L. Lapidus, and M. Borodovsky, "Frameshift detection in prokaryotic genomic sequences." International Journal of Bioinformatics Research and Applications, vol. 5, no. 4, pp. 458-477, 2009.

## *3.1 Introduction*

Progress in DNA sequencing technology has revolutionized biology. Over the past decade, the ever increasing use of Sanger instruments has resulted in an unprecedented explosion of available genomic information. Yet as next generation sequencing techniques such as 454 pyrosequencing [111] and Solexa/Illumina [20] enter production, an even more massive influx of sequenced data is anticipated. Frameshifts - changes of reading frame in protein-coding genes - can be classified by origin into natural and

artificial. Natural frameshifts occur in pseudogenes, in programmed frameshift locations [120]. Artificial frameshifts are caused by sequencing and assembling errors that may occur even in high X coverage sequencing (errors of length not divisible by 3). Early detection of frameshifts related to sequence errors could improve the quality of the assembly process, decrease requirements on the sequencing coverage, and, thus, reduce the cost of sequencing. Two general approaches have been used to detect frameshift errors: ab initio (intrinsic) algorithms [57, 124, 145] and extrinsic algorithms based on protein similarity search [162, 133, 40, 71, 129, 25]. In the beginning of the 1990s, when sequencing with high X coverage required significant expenses, the computational frameshift detection attracted considerable interest. The pioneer paper on frameshift detection [162] introduced three crucial elements of a general method: alignment of gene products to known proteins, protein coding frame prediction based on known codon usage pattern, and identification of nucleotide patters associated with error locations. Development of the extrinsic approach included the initial heuristic program DETECT using 3-frame translations of potentially frameshifted sequence in protein database searches [133]; introduction of frameshift dependent scoring matrices for protein sequence alignment algorithm [40]; refinement of translated DNA to protein alignment techniques to detect both frameshifts within codons and between codons [129]; implementation of dynamic programming algorithm for correct alignment of the protein translation of DNA in three frames to a homologous protein [71, 25, 26]. Another major approach, ab initio frameshift prediction, progressed from utilization of k-tuple frequencies to identify the frame of genetic code along the genomic sequence [57], to using posterior probabilities of the reading frames determined by the GeneMark program [124], to including frameshift states into the HMM-based gene prediction algorithm [145]. The ab initio method presented here is designed to extract information on a possible frameshift from the values of posteriori probabilities of protein coding frames in a given genomic position. These values are generated

by the GeneMark program [28]. Earlier the GeneMark coding potentials were used for frameshift detection in the Bacillus subtilis genome project by [124] and, as we describe in details below, in the extension of the GeneMark algorithm (J. McIninch, unpublished). In the tool developed by Medigue et al. the posterior probabilities computed by the GeneMark program were processed heuristically by a hierarchical decision making. The tool performed with 54.4% Specificity (Sp) while its Sensitivity (Sn) was not assessed. Currently this tool is not available. Performance of the tool developed by Mcininch has not been evaluated in terms of Sn and Sp. Given the renewed interest in frameshift detection we have explored once again the potential of the approach based on the analysis of the posterior probabilities. We designed and implemented an ab initio frameshift finder in combination with the post-processing of predicted frameshifts using information derived at the protein level. A frameshift error in a prokaryotic protein-coding region (which is necessarily a part of an ORF, defined here as a nucleotide sequence of length divisible by 3, delimited by two stop codons) results in a split of the ORF into two overlapping ORFs. If long enough, the protein coding parts of these ORFs are detected as genes. A critical task for frameshift finding is to distinguish the ORF overlaps caused by frameshifts from natural overlaps of adjacent genes carried genetic code in the same DNA strand but in different reading frames (Figure 4). For example, over 30% of Escherichia coli genes overlap each other (with about 15% being 1 or 4 nucleotides long). Majority of the overlaps occur between genes located in the same strand. Presence of a Ribosomal Binding Site (RBS) exhibiting a conserved motif could help identify genuine overlapping genes. However, genes internal to an operon may not possess pronounced RBS motifs while genes possessing leaderless mRNA even do not have a sequence upstream to a start codon for a ribosome to bind.

## *3.2 Materials*

### 3.2.1 Sequences with artificial sequencing errors

For performance evaluation of the new tool, we selected complete genomes of the following species of varying G+C composition: *Anaeromyxobacter dehalogenans* 2CP-C (GenBank accession no. `NC_007760.1`), *Bacillus subtilis* subsp. Subtilis str. 168 (`NC_000964.2`), *Clavibacter michiganensis* subsp. michiganensis NCPPB 382 (`NC_009480.1`), *Clostridium botulinum* F str. Langeland (`NC_009699.1`), *E. coli* K12 (`NC_000913.2`), *Frankia* sp. EAN1pec (`NC_007777.1`), *Fusobacterium nucleatum* subsp. nucleatum ATCC 25586 (`NC_003454.1`), *Haemophilius influenzae* Rd KW20 (`NC_000907.1`), *Lactobacillus reuteri* F275 (`NC_009513.1`), *Methanocorpusculum labreanum Z* (`NC_008942.1`), *Mycoplasma mycoides* subsp. mycoides SC str. PG1 (`NC_005364.2`), *Shewanella loihica* PV-4 (`NC_009092.1`), and *Shewanella putrefaciens* CN-32 (`NC_009438.1`). Indels were introduced into protein-coding regions as annotated in GenBank. The species were subdivided into high, low, and medium G+C composition groups. We have added indels at random with the rate between 0.02 and 0.5 per Kbp (the highest error rate that we have observed in raw sequencing data was 0.25 errors/Kbp).

### 3.2.2 Sequences with 454 pyrosequencing errors

To give the algorithm yet another test, we used 454 pyrosequencing pre-production data from the DOE Joint Genome Institute microbial finishing pipeline. We selected Methanococcus aeolicus Nankai-3, Shewanella putrefaciens CN-32, and Pseudomonas putida sp. F1 as representatives of low, medium, and high G+C ranges, respectively. These sequence data consisting of 100-3000 contigs (per genome) were mapped by MegaBLAST to the genomic sequence of the same species produced by the Sanger instruments. We assumed that the finished Sanger sequence had no errors. Based on these alignments, we detected the errors in 454 pyrosequencing, classified them by type and determined their distribution.

### 3.2.3 Protein sequences

Some predicted frameshifts could be verified on protein level. For protein sequence similarity search we used a database compiled from protein translations of genes predicted by GeneMarkS [24] in 313 bacterial genomes (a database maintained by Wenhan Zhu).

## *3.3 Methods*

We chose to work with GeneMark rather than GeneMark.hmm [105] for the following reasons. GeneMark.hmm uses the Viterbi algorithm [92] to determine the maximum likelihood parse of genomic sequence into protein-coding and non-coding regions. Frameshifts contradict the "genomic grammar" wired into the HMM, thus, the frameshift detection would require a significant change of the underlying HMM and in the GeneMark.hmm algorithm.

On the other hand, the GeneMark algorithm could be viewed as an approximation of an a posteriori decoding algorithm for an HMM consisting of six coding states (corresponding to six coding frames) and one non-coding state. The "approximated" posterior decoding algorithm computes a posteriori probability of a hidden state (e.g. coding in a particular frame or non-coding) for a rather short sequence segment assuming that only one type of a hidden state is underlying the observed short sequence. This algorithm requires an additional routine to process the posterior probabilities and determine the whole likely sequence of hidden states. The possibility of detection of "jumps" between the hidden states (the frames) of HMM underlying the GeneMark algorithm (though this HMM is introduced retrospectively) suits our goals.

Parameters of the Markov chain models used in the algorithm are estimated by the self-training program GeneMarkS [24]. This program performs well for long genomic sequences (longer that 100 Kb). If the frameshift finder has to run on a sequence contig with length insufficient for self-training, the algorithm uses one of heuristic

models, which parameters are precomputed for possible values of G+C content [23].

We have implemented a method of scanning the posterior probabilities determined by the GeneMark algorithm (see Appendix for details). The method is designed to identify a characteristic "between-frames-jump" of the coding potential, expected to appear near a frameshift position. After finding all the candidate positions, the scanning algorithm reports them to a classifier algorithm (whose parameters are described in the Appendix) to identify the predicted frameshift positions from the reported candidates.

To train the classifier, we used genomes of five bacterial species: A. dehalogenans 2CP-C, C. michiganensis, C. botulinum, E. coli, and S. putrefaciens. Assuming that these genomes sequenced with high X coverage by Sanger instruments have a vanishingly low number of sequence errors, we generated artificial frameshifts in protein-coding regions, then selected all the regions with pairs of adjacent gene overlapping each other and satisfying (see Appendix) a condition

$$\left(P\left(M_{F_A}^{\mathrm{COD}}|Seq_{f-W\ldots f}\right) > 0.5,\ P\left(M_{F_B}^{\mathrm{COD}}|Seq_{f+1\ldots f+W}\right) > 0.5\right) \tag{3}$$

The vast majority of sequences with introduced frameshifts as well as a number of gene overlap regions satisfy this condition. Increasing the probability threshold beyond 0.5 does not significantly increase specificity, while it negatively impacts sensitivity. We have trained the classifier on these two sets and determined parameters for the three types of models: models for genomes with low, medium, and high G+C content. Then we have assessed the accuracy of the classifier via cross-validation.

Upon application of the classifier we have observed (compare "C: ab initio" with "B: no classifier" columns of Table 2) a decrease in false positive predictions (increase in specificity) but not a decrease in false negative predictions (increase in sensitivity). Therefore, as expected, the classifier works as a filter, i.e. a mechanism to reject some predictions made in the first step, the analysis of coding potentials; the classifier

application does not add new predictions.

In our experiments with several genomes the ab initio frameshift finder with classifier off has detected 59% to 81% of all frameshifts while with classifier on 51% to 69% of frameshifts were detected (Table 2). At the same time with classifier off 32% to 72% of the predictions were correct, while 37% to 85% of predictions were correct with classifier on.

### 3.3.1 Verification by protein sequence alignment

Additional improvement of specificity could be achieved by a subsequent analysis of the DNA sequences with predicted frameshifts on the protein level; thus, we have implemented a protein alignment-based frameshift verification algorithm. While the alignment approach could be used independently for frameshift finding or the outputs of the two algorithms can be combined on an equal basis, we chose to implement protein alignment as a post-processing step.

This step starts with using a conceptual protein translation of the ORF with predicted frameshift as a query in the BLASTP program [12] for search of the statistically significantly similar protein sequences in a protein database (see Materials). The proteins and protein alignments found by the search can provide positive or negative evidence for the frameshifts predicted by the ab initio algorithm. The alignments are analyzed for the presence of one of the four possible scenarios (Figure 5).

"Bridge": An alignment with high coverage ($> 85\%$) and significant identity ($> 50\%$) of the conceptual translation (query) to a single protein (target) is admitted as a positive evidence for the frameshift.

"Local bridge": A near-perfect ($> 90\%$ identity) alignment of the translation of the nucleotide region $(f_{MAX} - w, f_{MAX} + w)$ enclosing the frameshift to a target protein is admitted as a positive evidence as well.

"Broken bridge": A high-coverage (> 85%) and significant identity (> 50%) alignment of the translated upstream and downstream ORFs to separate proteins in the database is admitted as a negative evidence for the frameshift.

"Half bridge": A high-coverage (> 85%) and significant identity (> 50%) alignment of only one of the translated upstream or downstream ORF is also admitted as a negative evidence for the frameshift.

This type of verification, as was already mentioned, increases Sp while Sn may decrease (Table 2). For some species, particularly those that are distantly related to the majority of the species with genomes sequenced, similarity searches produce fewer number of hits in the protein database; thus, with little information derived from database searches almost no improvement had occurred; for other species, improvement in Sp by as much as 30% was observed along with some decrease in Sn.

We compared the tool's performance with the performance of FrameD program [145]. Sequences with artificial sequencing errors were submitted to the FrameD web server for model generation. These models were used in a local copy of the FrameD program to predict frameshifts in these sequences. Sn and Sp values were computed in the same way as above.

The performance of a frameshift detection method based solely on protein sequence comparison depends on the evolutionary distance of a given genome to genomes already sequenced. Also, for a particular gene, presence or absence of sequenced homologs will influence the "local" performance of the extrinsic method to detect a frameshift in this gene. Given this consideration, we have decided not to conduct the comparison of performance with the algorithms of purely extrinsic type. It is rather obvious that an ab initio method will perform better for those genomes and genes that are lacking, on average or individually, the extrinsic references, while an extrinsic method will perform better for genomes and genes possessing, on average or individually, the extrinsic references.

## *3.4 Results*

The results are summarized in Tables 2 and 3. In Table 2, six types of predictions were considered: A/ prediction by the "prior design" algorithm; B/ prediction by the algorithm performing coding potentials analysis (with the classifier part of the ab initio algorithm off); C/ full ab initio prediction (coding potential analysis followed with application of the classifier algorithm); D/ full ab initio prediction followed by protein database search and alignment with rejection of putative frameshifts with negative evidence ("Broken bridge" or "Half bridge"); E/ ab initio prediction followed by the database search and alignment and retaining only those predictions that have a positive evidence ("Bridge" or "Local bridge"); F/ prediction by the FrameD program [145].

In the pure ab initio prediction, Sn varied between 51% and 69%, while Sp was observed between 37% and 85%. At this stage, one could observe that the performance of the FrameD program in terms of (Sn+Sp)/2, is higher for the species with medium and high G+C content (Table 2) e.g. for E. coli by 8% and for M. tuberculosis by 18%. Addition of the alignment verification steps produced the following results i/ rejecting negative evidence moved Sp up to the 51% to 95% range while Sn decreased slightly and stayed in the range 50% to 66%; ii/ retaining only predictions with positive evidence produced further increase in Sp to the range of values from 79% to 100% while Sn decreased noticeably to take the values in the range from 27% to 56%. Notably, in the case of S. loihica the program identified 56% of real frameshifts with no false predictions.

Additionally, the three 454-pyrosequenced genomes *M. aeolicus*, *S. putrefaciens* and *P. putida F1* with low, medium and high G+C ranges respectively, were aligned to the genomes of the same species sequenced by the Sanger instruments (see Materials). Locations of insertions/deletions recognized as 454 pyrosequencing errors were recorded. Five types of frameshift prediction methods were used (Table 3): types B

and C as in Table 2, as well as types C*, D* and E* which are the types C, D and E in Table 2 augmented by an additional analysis for presence of homopolymers (see Methods). Notably, Sn was reduced in S. putrefaciens and P. putida compared to predictions in sequences with synthetic frameshifts; this apparently has occurred due to a large number of disjoint contigs and that produced an increase of the fraction of the 454 sequencing errors in the flanking sequences. The errors in the close vicinity of sequence ends are not detectable by the algorithm, thus the Sn decreases. Specificity is reduced in detecting 454 sequencing errors in cases B, C, C*, and D* for *P. putida* as compared to sequences with synthetic frameshifts. This result could be related by the less accurate estimation of the algorithm parameters for a genome split into a large number of contigs (over 3000). Finally, the homopolymer related corrections did not make any significant effect on the accuracy of the algorithm. Still, the results show that the method works with 454-pyrosequenced genomic sequences with about the same accuracy as with sequences carrying artificial frameshifts.

Changing the values of the algorithm parameters (mentioned in Appendix and listed in Table 4) generated the results plotted as curves of Sn from Sp dependence in detecting the 454 pyrosequencing errors. Technically, the results form a cloud of points in the Sn vs. Sp plane. Plotting a convex envelope around the cloud resulted in the curve shown in Figure 6. The points denoted by stars in Figure 6 correspond to the largest values of (Sn+Sp)/2 for a given species. Homopolymer detection was enabled in these computations, with a presence of a homopolymer longer than 5 nt required for frameshift prediction. Notably, homopolymer sequences (which may cause indel errors in sequencing) are underrepresented in protein-coding regions compared to intergenic regions Figure 7.

Detection rates for computationally generated and empirically observed frameshifts were nearly identical for high and low G+C content; in medium G+C content genomes the frameshift detection rates were lower in 454-generated sequences than in sequences

with synthetic frameshifts. We observed (Table 5) that the total number of predicted frameshifts in a sequence is frequently close to the total number of real frameshifts. This numerical fact indicates that the number of false predictions is close to the number of false negative predictions (the number of real but not detected frameshifts). We observed that a sizable fraction of mispredictions occurred in the locations of actual gene overlaps; on the other hand a significant fraction of undetected frameshifts appeared near the start and end of a gene.

## *3.5 Discussion*

The main impediment to highly accurate frameshift detection in protein-coding sequences is the difficulty of distinguishing frameshifts changing the frame of genetic code in a single gene from two adjacent overlapping genes located in the same strand. The combination of intrinsic and extrinsic methods presented here is a promising approach, allowing for frameshift detection at a distance as little as 60 nt from 5' or 3' ends of a gene.

The sequencing errors that occur too close to the 5' and 3' ends of a gene are often not detectable; this limitation is difficult to overcome by any method based on the statistical analysis of protein-coding regions.

The method described here is also applicable to metagenomic sequences. In metagenomic studies, single reads are a frequent case. Thus, the frameshift prediction in metagenomic fragments is not less important than in studies of isolated genomes. Still, the fact that the frameshifts are more difficult to predict in flanking regions of genes (both complete and incomplete) may reduce the effect of frameshift finding in short metagenomic fragments.

A smaller average length of contigs in an unfinished assembly is frequently associated with low assembly quality. Unfinished assembly is likely to contain low coverage fragments with higher probability of sequencing and misassembly errors; the error

rate can reach as high as 0.5 errors/Kbp. Interestingly, some predicted frameshifts can indicate assembly errors where the protein coding regions of genes unrelated but located in the same strand have happened to be joint together with a frameshift by an assembly algorithm. Detection of these errors as assembly errors at the stage of assembly can improve the quality of finished sequence. Note that assembly errors may produce partial genes (genes without start or stop codons ot both) as well as himeric sequences with partial genes in direct and reverse DNA strands adjacent to each other. We assumed that these errors are very rare and did not consider them.

During our analysis, we tuned up the parameters of the ab initio algorithm to produce about equal values of Sn and Sp. However, the techniques used in our method can be adjusted to obtain other desirable combinations of the Sn and Sp values (Figure 8). Thus, the output can be adjusted with regard to the need of a particular project which can be a preference for low rate of false negative or low rate of false positive errors or balanced rate of both types of errors.

One of the assumptions made in this work is an independent random distribution of sequencing errors. This assumption underlies the random model used for frameshifts generation. However, this model is not fully supported by experimental data as it can be shown by a comparison of the earliest E. coli genome versions (GenBank accession numbers AE000111-510) to the latest genomic sequence of E. coli (U00096.2). In this small dataset, fewer than 50 errors, the errors are tightly clustered into stretches of about 100nt.

The software package for frameshift detection is available for researchers on our website [2].

## *3.6 Technical details*

Technical details of the algorithm implementations follow. First, we describe the previously implemented, but unpublished algorithm, the predecessor of the current

algorithm.

### 3.6.1 Prior design

We introduce the GeneMark algorithm parameters [28], the scanning window length w, the step size s, the coding threshold COD_THR, and the non-coding threshold NON_THR. (Default values: w=96, s=12, COD_THR=0.5, NON_THR=0.4.)

- Set the window counter $c$ to 0.
- For each frame $F_i$ of the 6 frames, for each position index pos $= 0, s, 2s, 3s \ldots$,
  - Increment $c$ by 1 if the window $w$ starting at $pos$ has coding potential CP ( see details below) in frame $F_i$ above COD_THR.
  - Reset $c$ to 0 if $c$ is smaller than $w/s$ and two adjacent windows $w$ starting at $pos, pos + s$ have CPs in frame $F_i$ below NON_THR.
  - If $c$ is larger than $w/s$ and the window at $pos$ has CP in frame $F_i$ smaller than NON_THR,
    - *and* The average CP in all windows within the region $[pos, pos + 2w]$ in frame $F_i$ is smaller than NON_THR,
    - *and* no stop codon exists in frame $F_i$ in the fragment $[pos + w/s, pos + w]$,
    - *and* in either of the two frames $F_j$, $F_k$ collinear to $F_i$, average CP over all windows within $[pos - 2w, pos]$ is smaller than NON_THR and average CP over all windows within $[pos, pos + 2w]$ is larger than COD_THR,
  - Then mark the position $pos + w/s$ as a predicted frameshift, reset $c$ to 0, and move ahead by length $w$.

This algorithm takes into account the coding potential over about 200 nt in each direction from the putative frameshift position. It requires the coding potential in the upstream frame to decrease, while requiring the coding potential in the downstream frame to rise. This initial algorithm produced better Sp than Sn, it has high speed and relative simplicity (Table 2). Still, it is not highly sensitive, with Sn below 40% in genomic sequences with error rates between 0.1 and 0.4 frameshifts/Kbp. Therefore, our goal was to increase both the Sn and Sp of the algorithm while retaining the same type of inut data, the coding potentials generated by GeneMark. Eventually, we observed that in genomes with extremely high G+C content the old algorithm performs in Sn terms on par with the new design. Therefore, for high G+C genomes (>60% G+C), we enabled the old algorithm and added its output (removing redundant predictions) to the output of the new algorithm.

### 3.6.2 New design

The input sequence is scanned in six frames for ORFs with length above a minimal length, min_orflen. Any pair of ORFs located in the same strand and overlapping by at least min_orf_overlap is taken into consideration. Values of the min_orflen and min_orf_overlap parameters are given in Table 6.

In analysis of 454 sequenced genomes a presence of a homopolymer in the vicinity of fmax served as an additional evidence. The error statistics for 454 sequencing suggests a minimum length of 5, as an informative one for error detection (data not shown). Therefore, the sequence was scanned for up to 20 nt in both directions from $f_{max}$. If a homopolymer longer than 4nt was not found, the ORF overlap was excluded from further consideration.

The order of the Markov chain model was selected based on the volume of sequence available. Parameters of the model of protein-coding sequence were calculated by a self-training program GeneMarkS [24]. For a given fragment of length w Bayesian

a posteriori probabilities of genetic code to appear in one of the six frames (coding potentials) as well as an a posteriori probability that a given fragment is non-coding were calculated as follows:

$$P(\mathrm{Seq}_{N..N+w} \mid M^{\mathrm{COD}}_{F_K})$$

$$= p^0_{F_K}(n_N)\, p^1_{F_{K+1}}(n_{N+1} \mid n_N) \ldots \prod_{X=N+\mathrm{ORD}}^{N+w} p^{\mathrm{ORD}}_{F_{K+X \bmod 3}}(n_X \mid n_{X-ORD} \ldots n_{X-1})$$

$$P(M^{\mathrm{COD}}_{F_K} \mid \mathrm{Seq}_{N..N+w})$$

$$= \frac{P(M^{\mathrm{COD}}_{F_K}) P(\mathrm{Seq}_{N..N+w} \mid M^{\mathrm{COD}}_{F_K})}{P(M^{\mathrm{NONC}}) P(\mathrm{Seq}_{N..N+w} \mid M^{\mathrm{NONC}}) \prod_{J=-3,-2,-1,1,2,3} P(M^{\mathrm{COD}}_{F_J}) P(\mathrm{Seq}_{N..N+w} \mid M^{\mathrm{COD}}_{F_J})}.$$

Here, $w$ the window size is 72nt long for G+C $> 60\%$, otherwise 96 nt, $Seq_{N..N+w}$ is the nucleotide sequence in a given window, $n_X$ is a nucleotide in position $x$ , $M^{COD}_{F_K}$ is the protein coding region model for frame $K$, $M^{NONC}$ is the non-coding model, $p^{ORD}_{F_K}(n_X|n_Y)$ is the probability of appearance of nucleotide $n_X$ after a string $n_Y$ defined by the Markov chain of order $ORD$, and $P(M^{COD}_{F_K}$ are prior probabilities [28]. The product of posterior probabilities of carrying genetic code in different frames for the fragments located upstream and downstream from position $f$ is considered as a measure of the likelihood of a possible frameshift in position $f$:

$$P(FS@f) = P(M^{\mathrm{COD}}_{F_A} \mid \mathrm{Seq}_{f-W..f}) P(M^{\mathrm{COD}}_{F_B} \mid \mathrm{Seq}_{f+1..f+w})$$

(here $F_A$ and $F_B$ are the upstream and downstream coding frames; see Figure 8).

The overlapping region of an ORF pair ($[S_2, E_1]$, Figure 8) is scanned to find the maximum $f_{MAX} = argmax_{f\in[S_2,E_1]} P(FS@f)$. The following parameters are then recorded for the point $f_{max}$:

$$P(M^{\mathrm{COD}}_{F_A} \mid \mathrm{Seq}_{f-w..f})$$

$$P(M^{\mathrm{COD}}_{F_B} \mid \mathrm{Seq}_{f+1..f+w}).$$

Distance from fmax down to the stop codon of the upstream ORF.

The length of putative gene overlap. This parameter can be negative, which denotes that no overlap of is seen, but a gap of corresponding number of nucleotides is seen instead between the upstream stop codon and the putative start of the downstream gene.

Maximum of the score of potential RBS motif located at up to 20 nt distance upstream of the most likely start codon in the downstream ORF [24].

This set of parameters (attributes) is then transferred to a machine learning classifier, trained on a set of examples of two types (frameshifts and gene overlaps), to classify the point $f_{max}$ as a frameshift or gene overlap. Attribute histograms for the frameshift vs. gene overlap classes are plotted in Figure 9; performance of the classifier was compared to the simpler method of checking only that $P(FS@f) >$ total_coding_min (Table 2). We have evaluated several machine learning classifiers, including SVM, decision trees, perceptron networks, and Naive Bayes classifiers (data not shown). The Naive Bayes classifier (John and Langley, 1995) [11] appeared to be the best performing and best generalizing classifier on our data. A Naive Bayes classifier works by making the assumption that the attributes discussed above are independently distributed. Using the Bayes' rule, the probability of a frameshift given the set of attribute values $x_i$ associated with the point $f_{max}$ is

$$P(FS \mid \{x_i\}) = P(FS \mid x_1, x_2, x_3, ...) \propto P(\{x_i\} \mid FS)P(FS).$$

According to this assumption, the last expression can be factored as a product of probabilities,

$$P(\{x_i\} \mid FS) = \prod_i P(x_i \mid FS).$$

The log likelihood ratio for the frameshift vs. non-frameshift events is computed as follows:

$$\ln \frac{P(FS \mid f_{\max})}{P(\neg FS \mid f_{\max})} = \sum_{i \in \text{params}} \ln \frac{P(x_i \mid FS)}{P(x_i \mid \neg FS)}.$$

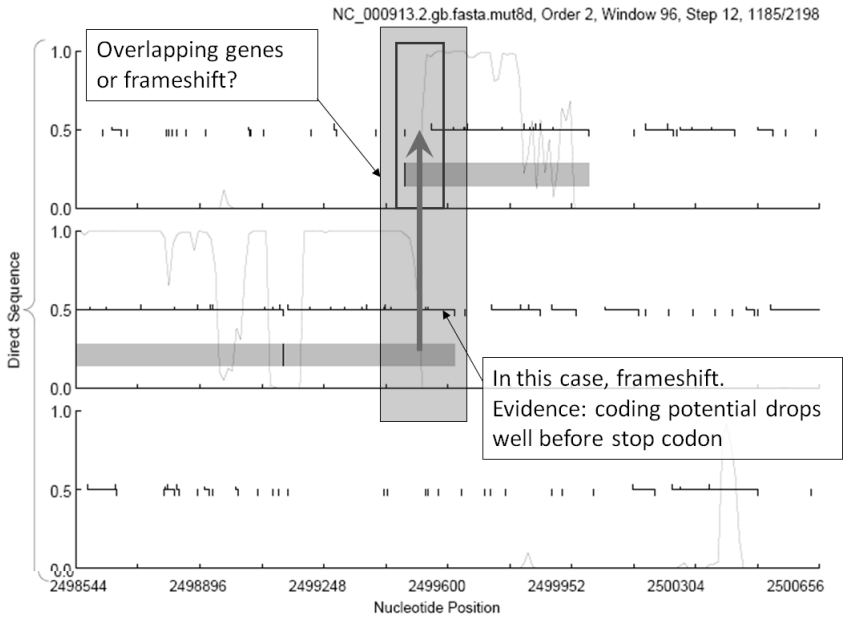



**Figure 4:** Diagram of an open reading frame fragmentation into two overlapping ORFs by a frameshift. A fragment of the Escherichia coli chromosome is shown with an artificial frameshift. Three curves indicate coding potentials in the three coding frames (averaged over 96-nucleotide windows). Open reading frames of significant size are indicated by horizontal lines plotted over the 0.5 line; start and stop codons are shown as upward and downward ticks, respectively. Gene predictions are indicated by grey bars. The frameshift prediction is marked by an arrow and shaded box.

If the ratio exceeds 0 (i.e., the cumulative probability of frameshift exceeds that of non- frameshift), the instance is classified as a frameshift; otherwise, as a non-frameshift (gene overlap). The distributions of attribute values were obtained by assuming Gaussian distributions and estimating the means and variances (given in Table 4).

## *3.7 Acknowledgements*

We are grateful to Wenhan Zhu for providing the database of protein sequences, to Stephan Trong for the help with 454 pyrosequencing error analysis, to James McIninch and William Hayes for the valuable contribution into the "prior design" version of the algorithm.

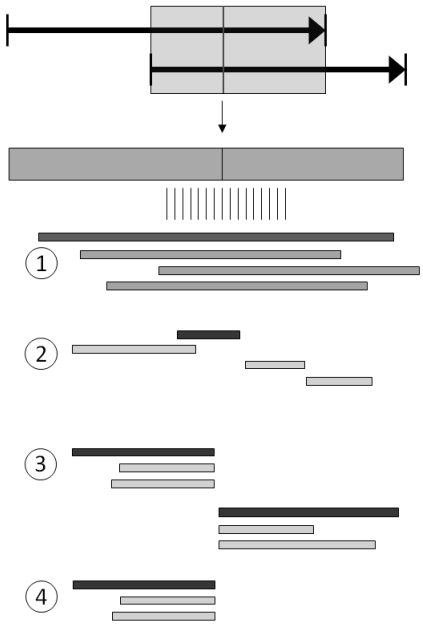



**Figure 5:** Four frameshift verification scenarios. The thick bar represents a conceptual translation of the ORFs with the possible frameshift. The thinner bars below represent similarity search hits in the protein database; the hits providing critical information are highlighted in darker color. Cases 1 and 2 provide positive evidence of a frameshift. Cases 3 and 4 provide negative evidence of a frameshift.

**Table 2:** Accuracy parameters of the different versions of the algorithm as well as the FrameD program determined on genomic sequences with synthetic frameshifts. A/ for the old algorithm, described in the "Prior design" section in Appendix; B/for the new algorithm with coding potential analysis only (the classifier algorithm off); C/full ab initio prediction (includes the coding potential analysis and the classifier algorithm); D/ for the ab initio prediction followed by protein database search and alignment and rejection of ab initio predictions with negative evidence (scenarios 3 and 4, Fig. 2); E/for the ab initio prediction followed by the protein database search and alignment with acceptance of the predictions possessing a positive evidence (scenarios 1 and 2, Fig. 2); F/ for the FrameD program (Schiex et al., 2003). Bold numbers indicate best performance among A-E as measured by the average value (Sn+Sp)/2.

| Species | G+C% | | A Old algorithm | | B New algorithm | | C "Ab initio" B+classifier | | D C+Rejecting negative evidence | | E C+Accepting positive evidence | | F FrameD | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Mycoplasma mycoides* | 24 | Sn | 0.11 | 0.32 | 0.65 | 0.59 | 0.61 | 0.61 | 0.59 | 0.62 | 0.45 | **0.67** | 0.75 | 0.46 |
| | | Sp | 0.54 | | 0.52 | | 0.61 | | 0.64 | | 0.89 | | 0.16 | |
| *Fusobacterium nucleatum* | 27 | Sn | 0.14 | 0.41 | 0.81 | 0.65 | 0.66 | 0.65 | 0.65 | **0.74** | 0.56 | 0.73 | 0.70 | 0.73 |
| | | Sp | 0.69 | | 0.49 | | 0.65 | | 0.83 | | 0.91 | | 0.77 | |
| *Clostridium botulinum* | 28 | Sn | 0.08 | 0.36 | 0.75 | 0.66 | 0.63 | 0.69 | 0.61 | 0.74 | 0.52 | **0.75** | 0.51 | 0.68 |
| | | Sp | 0.64 | | 0.58 | | 0.75 | | 0.87 | | 0.97 | | 0.86 | |
| *Rickettsia prowazekii str. Madrid-E* | 29 | Sn | 0.14 | 0.34 | 0.69 | 0.56 | 0.67 | 0.66 | 0.65 | **0.72** | 0.56 | 0.72 | 0.50 | 0.61 |
| | | Sp | 0.55 | | 0.44 | | 0.65 | | 0.78 | | 0.87 | | 0.72 | |
| *Haemophilus influenzae* | 38 | Sn | 0.28 | 0.39 | 0.74 | 0.62 | 0.62 | 0.62 | 0.60 | **0.67** | 0.53 | 0.66 | 0.71 | 0.67 |
| | | Sp | 0.50 | | 0.49 | | 0.63 | | 0.75 | | 0.79 | | 0.64 | |
| *Lactobacillus reuteri* | 38 | Sn | 0.19 | 0.32 | 0.74 | 0.63 | 0.62 | 0.66 | 0.60 | **0.74** | 0.55 | **0.74** | 0.56 | 0.69 |
| | | Sp | 0.46 | | 0.52 | | 0.71 | | 0.89 | | 0.93 | | 0.82 | |
| *Bacillus subtilis* | 43 | Sn | 0.32 | 0.32 | 0.65 | 0.56 | 0.56 | 0.59 | 0.53 | 0.63 | 0.42 | **0.64** | 0.54 | 0.68 |
| | | Sp | 0.32 | | 0.48 | | 0.62 | | 0.73 | | 0.87 | | 0.82 | |
| *Shewanella putrefaciens* | 44 | Sn | 0.26 | 0.38 | 0.64 | 0.65 | 0.51 | 0.67 | 0.50 | **0.72** | 0.44 | 0.71 | 0.53 | 0.72 |
| | | Sp | 0.50 | | 0.66 | | 0.84 | | 0.94 | | 0.98 | | 0.92 | |
| *Escherichia coli* | 50 | Sn | 0.38 | 0.36 | 0.69 | 0.63 | 0.59 | 0.67 | 0.55 | **0.73** | 0.48 | 0.72 | 0.65 | 0.75 |
| | | Sp | 0.34 | | 0.58 | | 0.75 | | 0.90 | | 0.95 | | 0.85 | |
| *Methanocorpusculum labreanum Z* | 50 | Sn | 0.46 | 0.28 | 0.59 | 0.54 | 0.59 | 0.60 | 0.58 | 0.70 | 0.46 | **0.71** | 0.66 | 0.70 |
| | | Sp | 0.10 | | 0.48 | | 0.62 | | 0.83 | | 0.97 | | 0.75 | |
| *Shewanella loihica PV-4* | 53 | Sn | 0.40 | 0.31 | 0.71 | 0.72 | 0.69 | 0.77 | 0.66 | **0.81** | 0.56 | 0.78 | 0.77 | 0.85 |
| | | Sp | 0.22 | | 0.72 | | 0.85 | | 0.95 | | 1.00 | | 0.93 | |
| *Mycobacterium tuberculosis CDC1551* | 65 | Sn | 0.55 | 0.39 | 0.65 | 0.48 | 0.66 | 0.51 | 0.59 | 0.55 | 0.39 | **0.67** | 0.73 | 0.69 |
| | | Sp | 0.23 | | 0.32 | | 0.37 | | 0.52 | | 0.96 | | 0.65 | |
| *Frankia sp. EAN1pec* | 69 | Sn | 0.54 | 0.37 | 0.61 | 0.51 | 0.64 | 0.54 | 0.58 | 0.54 | 0.37 | **0.66** | 0.69 | 0.67 |
| | | Sp | 0.21 | | 0.40 | | 0.45 | | 0.51 | | 0.96 | | 0.65 | |
| *Clavibacter michiganensis* | 72 | Sn | 0.60 | 0.48 | 0.67 | 0.59 | 0.67 | 0.66 | 0.61 | **0.68** | 0.32 | 0.66 | 0.41 | 0.34 |
| | | Sp | 0.37 | | 0.52 | | 0.66 | | 0.75 | | 1.00 | | 0.28 | |
| *Anaeromyxobacter dehalogenans 2CP-C* | 75 | Sn | 0.56 | 0.54 | 0.61 | 0.56 | 0.61 | 0.64 | 0.56 | **0.67** | 0.27 | 0.63 | 0.31 | 0.24 |
| | | Sp | 0.52 | | 0.51 | | 0.67 | | 0.79 | | 0.99 | | 0.16 | |

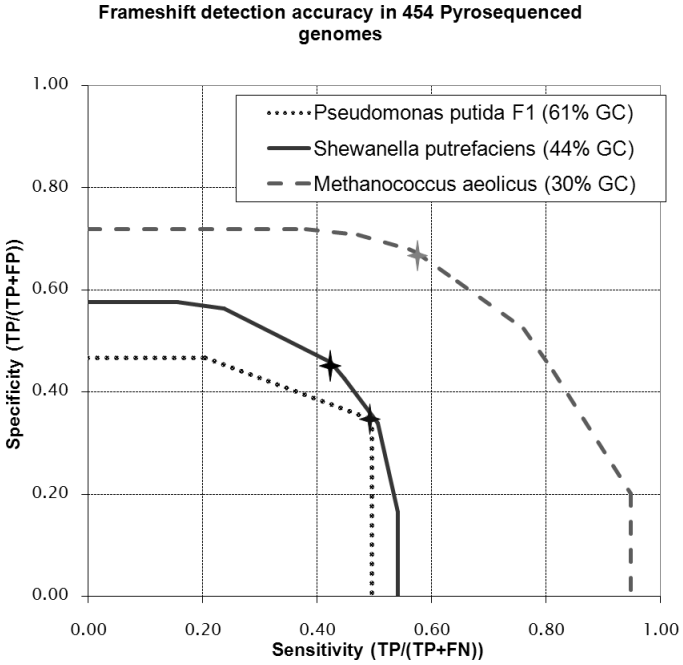



**Figure 6:** Sensitivity/specificity analysis of performance of the ab initio algorithm with homopolymer correction on 454 pyrosequenced genomes. Stars indicate trade-off points selected as optimal on the basis of a maximum (Sn+Sp)/2.

**Table 3:** Characteristics of the algorithm performance on genomes sequenced by 454 pyrosequencing method. Designations of the methods B and C are the same as in Table 2; C*, D*, E* are analogous to Table 2; * indicates that the algorithm was using the homopolymer correction (see text). Bold numbers indicate best performance among B-E* as measured by the average, (Sn+Sp)/2.

| Species | G+C % | | B No classifier | | C Ab initio | | C* Ab initio with HMP detection | | D* C*+Rejecting negative evidence | | E* C*+Accepting positive evidence | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Methanococcus aeolicus* | 30 | Sn | 0.66 | 0.52 | 0.56 | 0.55 | 0.54 | 0.54 | 0.52 | 0.63 | 0.44 | **0.65** |
| | | Sp | 0.38 | | 0.53 | | 0.54 | | 0.74 | | 0.85 | |
| *Shewanella putrefaciens* | 44 | Sn | 0.41 | 0.40 | 0.30 | 0.47 | 0.24 | 0.49 | 0.23 | 0.58 | 0.21 | **0.60** |
| | | Sp | 0.39 | | 0.63 | | 0.74 | | 0.93 | | 0.98 | |
| *Pseudomonas putida F1* | 61 | Sn | 0.39 | 0.30 | 0.33 | 0.28 | 0.33 | 0.31 | 0.33 | 0.31 | 0.11 | **0.50** |
| | | Sp | 0.25 | | 0.23 | | 0.28 | | 0.28 | | 0.89 | |

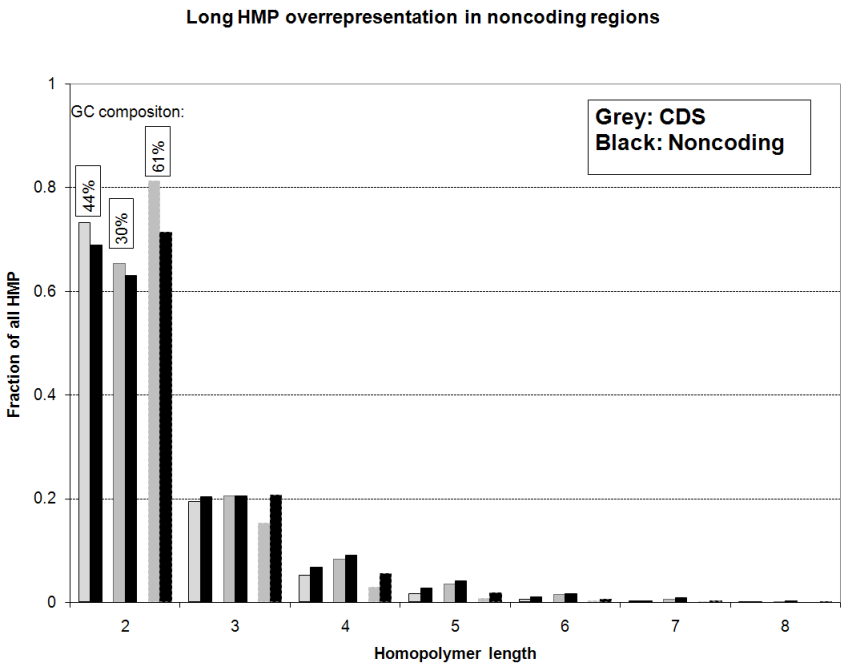



**Figure 7:** Homopolymer (HMP) frequencies were computed for three genomes (M. aeolicus, P. putida F1, S. putrefaciens) and are shown by pairs of bars (for protein coding and non-coding regions) for each homopolymer longer than 1nt. Long homopolymers are relatively more frequent in non-coding sequence, making 454 pyrosequencing errors more likely to occur in non-coding sequence. For each homopolymer length, data for three G+C ranges are presented.

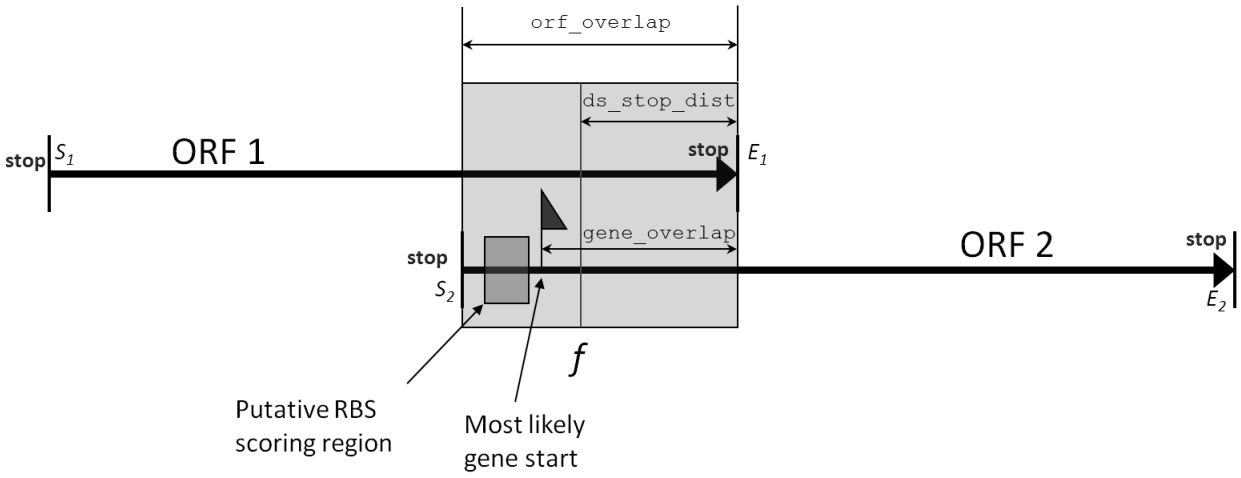



**Figure 8:** Geometry of the ORF overlap and definitions of parameters of the algorithm. Two overlapping ORFs are shown with the overlap region and salient parameters highlighted; f indicates the putative frameshift position.

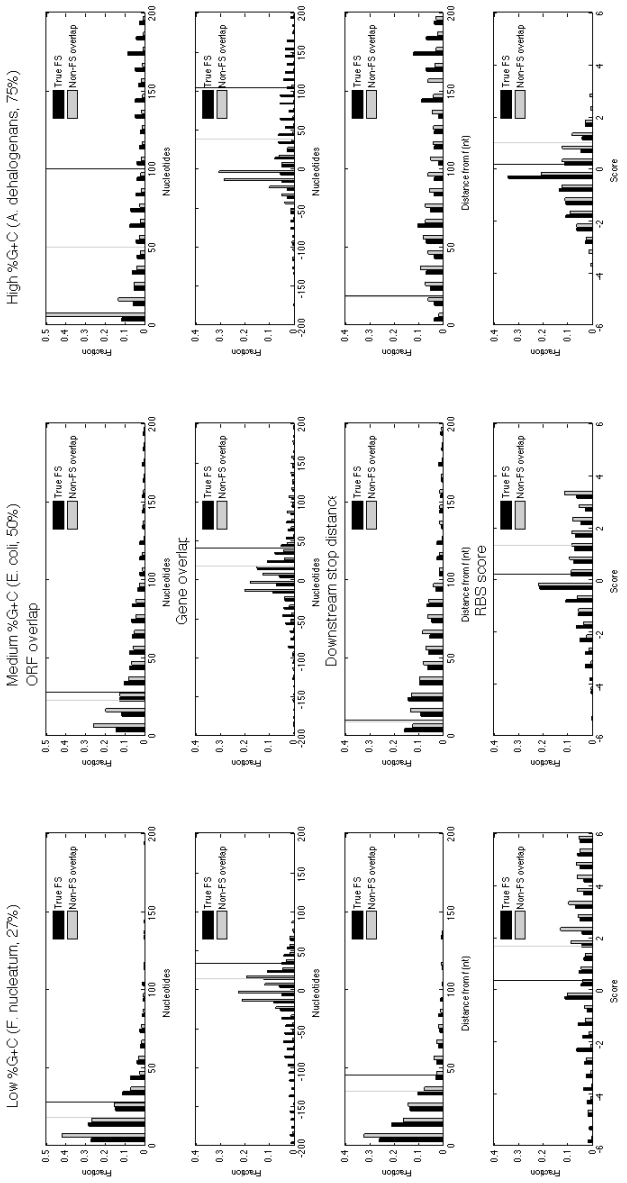



**Figure 9:** Histograms of the values of attributes used by the classifier to distinguish true frameshifts from gene overlap events. Data for genomes representative of the three G+C ranges are presented in the three columns. In a given genome the attribute values were tabulated from all ORF overlaps which satisfied the conditions for candidate pairs (see Methods). A putative frameshift location, f, was assigned for each ORF overlap; position-dependent attribute values were computed with regard to that location. Vertical bars indicate the means of the normal distributions fitted for attribute values characteristic for true frameshifts and gene overlaps, respectively. Negative values of the "gene overlap" parameter correspond to cases where no gene overlap is present, but instead a gap of the corresponding number of nucleotides exists between the upstream stop codon and the putative gene start. ORF overlaps are longer on average for higher GC due to lower frequencies of the three stop codons than in high AT genomes. Short "gene overlap" values are more frequent in non-frameshift events than in the cases of true frameshifts due to the fact that short gene overlaps and short intergenic distances are typical for prokaryotic genomes, while frameshift errors are more likely to produce longer apparent overlaps. RBS scores for frameshifts are lower on average than for gene overlaps due to the low probability of finding by chance a strong RBS motif outside a gene start region.

**Table 4:** Parameters of the fitted normal distributions for the values of orf_overlap, gene_overlap, ds_stop_dist, rbs_score as observed in the sets of sequences with artificial frameshifts and sequences with gene overlaps. These parameters describing the "true frameshift" vs. "gene overlap" class distributions were used in the classifier algorithm.

| | G+C% <40 | | | | 40<G+C% <60 | | | | G+C% >60 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | True FS | | Non-FS | | True FS | | Non-FS | | True FS | | Non-FS | |
| **Parameter** | μ | σ | μ | σ | μ | σ | μ | σ | μ | σ | μ | σ |
| orf_overlap | 28 | 44 | 18 | 33.5 | 28 | 66 | 23 | 48 | 100 | 239 | 49 | 147 |
| gene_overlap | 33 | 67 | 14 | 18 | 40 | 56 | 17.5 | 16 | 104 | 493 | 38 | 124 |
| ds_stop_dist | 45 | 24 | 35 | 22 | 10 | 34 | 9.5 | 32 | 19 | 99 | 18 | 91 |
| rbs_score | 0.35 | 2.6 | 1.6 | 2.5 | 0.3 | 2.02 | 1.32 | 1.89 | 0.24 | 1.55 | 1.0 | 1.38 |

**Table 5:** Quantitative analysis of frameshift predictions. Designations for columns C, D, and E are identical to Table 2. FS/Kbp, artificial frameshifts per 1000 base pairs. Pred, predicted frameshifts. TP, true positives.

| Species | G+C% | Length, Mbp | Total artificial frameshifts | FS/Kbp | ***C*** | | ***D*** | | ***E*** | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Pred. | TP | Pred. | TP | Pred. | TP |
| *Mycoplasma mycoides* | 24 | 1.21 | 242 | 0.2 | 242 | 148 | 223 | 143 | 122 | 109 |
| *Fusobacterium nucleatum* | 27 | 2.17 | 434 | 0.2 | 441 | 286 | 340 | 282 | 267 | 243 |
| *Clostridium botulinum* | 28 | 4.00 | 799 | 0.2 | 671 | 503 | 560 | 487 | 428 | 415 |
| *Rickettsia prowazekii str. Madrid-E* | 29 | 1.11 | 222 | 0.2 | 229 | 149 | 185 | 144 | 143 | 124 |
| *Haemophilus influenzae* | 38 | 1.83 | 366 | 0.2 | 360 | 227 | 293 | 220 | 246 | 194 |
| *Lactobacillus reuteri* | 38 | 2.00 | 399 | 0.2 | 348 | 247 | 269 | 239 | 236 | 219 |
| *Bacillus subtilis* | 43 | 4.21 | 842 | 0.2 | 761 | 472 | 611 | 446 | 406 | 354 |
| *Shewanella putrefaciens* | 44 | 4.66 | 931 | 0.2 | 565 | 475 | 495 | 466 | 418 | 410 |
| *Escherichia coli* | 50 | 4.64 | 927 | 0.2 | 729 | 547 | 567 | 510 | 468 | 445 |
| *Methanocorpusculum labreanum Z* | 50 | 1.80 | 360 | 0.2 | 343 | 212 | 252 | 209 | 171 | 166 |
| *Shewanella loihica PV-4* | 53 | 4.60 | 920 | 0.2 | 747 | 635 | 639 | 607 | 515 | 515 |
| *Mycobacterium tuberculosis CDC1551* | 65 | 4.40 | 880 | 0.2 | 1570 | 581 | 998 | 519 | 358 | 343 |
| *Frankia sp. EAN1pec* | 69 | 5.43 | 1086 | 0.2 | 1545 | 695 | 1235 | 630 | 419 | 402 |
| *Clavibacter michiganensis* | 72 | 3.30 | 659 | 0.2 | 669 | 442 | 536 | 402 | 211 | 211 |
| *Anaeromyxobacter dehalogenans 2CP-C* | 75 | 5.01 | 1002 | 0.2 | 912 | 611 | 710 | 561 | 273 | 271 |

**Table 6:** Parameters of the coding potential analysis algorithm

| Parameter | G+C% <40 | 40<G+C%<60 | G+C% >60 |
|---|---|---|---|
| min_orflen | 90 | 90 | 72 |
| min_orf_overlap | 5 | 10 | 15 |
| coding_min | 0.9 | 0.9 | 0.7 |
| total_coding_min | 0.85 | 0.85 | 0.6 |

## *3.8 Funding*

The work of AK, AL and MB presented in this chapter was supported in part by the NIH grant HG00783 to MB.

# CHAPTER IV

# GENOME ASSEMBLY AND ANNOTATION PIPELINE

New sequencing technologies have accelerated research on prokaryotic genomes and have made genome sequencing operations outside major genome sequencing centers routine. However, no off-the-shelf solution exists for the combined assembly, gene prediction, genome annotation, and data presentation necessary to interpret sequencing data. The resulting requirement to invest significant resources into custom informatics support for genome sequencing projects remains a major impediment to the accessibility of high-throughput sequence data.

We present a self-contained, automated high-throughput open source genome sequencing and computational genomics pipeline suitable for prokaryotic sequencing projects. The pipeline has been used at the Georgia Institute of Technology and the Centers for Disease Control and Prevention for the analysis of *Neisseria meningitidis* and *Bordetella bronchiseptica* genomes. The pipeline is capable of enhanced or manually assisted reference-based assembly using multiple assemblers and modes; gene predictor combining; and functional annotation of genes and gene products. Because every component of the pipeline is executed on a local machine with no need to access resources over the Internet, the pipeline is suitable for projects of a sensitive nature. Annotation of virulence-related features makes the pipeline particularly useful for projects working with pathogenic prokaryotes. Although developed and tested on whole-genome sequencing projects, all stages of the pipeline are also applicable to metagenome data, with the caveat that the output can no longer be assumed to come from a single genome but must be analyzed on a contig-by-contig basis.

The pipeline is licensed under the open-source GNU General Public License and

available at the Georgia Tech Neisseria Base (http://nbase.biology.gatech.edu). The pipeline is implemented with a combination of Perl, Bourne Shell, and MySQL and is compatible with Linux and other Unix systems.

The rest of this chapter is based on published work which first appeared in the following article:

A. O. Kislyuk, L. S. Katz, S. Agrawal, M. S. Hagen, A. B. Conley, P. Jayaraman, V. Nelakuditi, J. C. Humphrey, S. A. Sammons, D. Govil, R. D. Mair, K. M. Tatti, M. L. Tondella, B. H. Harcourt, L. W. Mayer, and I. K. Jordan, "A computational genomics pipeline for prokaryotic sequencing projects," Bioinformatics, vol. 26, no. 15, pp. 1819-1826, August 2010.

## *4.1 Introduction*

Genome sequencing projects, pioneered in the 1990s [58], require large-scale computational support in order to make their data accessible for use and interpretation by biologists. Large sequencing centers have traditionally employed or collaborated with teams of software engineers and computational biologists to develop the software and algorithms for sequencing hardware interfaces, enterprise data storage, sequence assembly and finishing, genome feature prediction and annotation, database mining, comparative analysis, and database user interface development. While many of the components developed by these teams are now available online under open-access terms, the development of new, high-throughput sequencing technologies has necessitated updates to these tools and development of even more sophisticated algorithms to address the challenges raised by the new data. These new technologies – 454 pyrosequencing [110], ABI SOLiD [152], and Illumina [20] – are now collectively referred to as second generation sequencing technologies. Similar updates will be needed as the third generation of sequencing technologies, such as Pacific Biosciences' SMRT sequencing [55], enter production use. New and improved tools released for these

technologies on a monthly basis include assemblers, mapping algorithms, base calling and error correction tools, and a multitude of other programs. Because of this fast pace of development, few experts are able to keep up with the state of the art in the field of computational genomics. Accordingly, the rate limiting step in genome sequencing projects is no longer the experimental characterization of the data but rather the availability of experts and resources for computational analysis.

At the same time, the increased affordability of these new sequencing machines has spawned a new generation of users who were previously unable to perform their own genome sequencing, and thus collaborated with large sequencing centers for genome sequencing and subsequent computational analysis. While these users are now able to experimentally characterize genomes in house, they often find themselves struggling to take full advantage of the resulting data and to make it useful to the scientific community since the informatics support for their genome projects is not sufficient.

Several large sequencing consortia [15, 112, 149] have produced comprehensive, centralized web-based portals for the analysis of genomic and metagenomic data. While extremely useful for many types of projects and collaborations, these solutions inherently result in a loss of data processing flexibility compared to locally installed resources and may be unsuitable for projects dealing with sensitive data. Recently, another group [163] has published DIYA, a software package for gene prediction and annotation in bacterial genomes with a modularized, open source microbial genome processing pipeline. However, DIYA does not include a genome assembly component, and does not provide for the combination of complementary algorithms for genome analysis.

To address the outstanding challenges for local computational genomics support, we have developed a state of the art, self-contained, automated high-throughput open source software pipeline for computational genomics in support of prokaryotic sequencing projects. To ensure the relevance of our pipeline, we checked the latest

developments in computational genomics software for all stages of the pipeline, such as new versions of assembly and gene prediction programs and comparative surveys, and selected what we deemed to be the most suitable software packages. The pipeline is self-contained; that is, we used locally installable versions of all third-party tools instead of web-based services provided by many groups. We chose to do so for three reasons: first, because some of the applications we envision for this pipeline are of sensitive nature; second, to enhance robustness to external changes (e.g., online API changes or website address changes); and third, to improve the ability of developers to customize and derive from our pipeline. The pipeline is also automated and high-throughput: all components are organized in a hierarchical set of readily modifiable scripts, and the use of safe programming practices ensures that multiple copies of the pipeline can be run in parallel, taking advantage of multiple processors where possible.

Importantly, by using and combining the outputs of competitive, complementary algorithms for multiple stages of genome analysis, our pipeline allows for substantial improvement upon single-program solutions. The use of multiple algorithms also provides a way to improve robustness and conduct more comprehensive quality control when the output of one program is significantly different from that of another.

Computational support provided to prokaryotic genome projects by our pipeline can be subdivided into three stages: first, sequencing and assembly; second, feature prediction; and third, functional annotation. For the assembly stage, we developed a custom protocol specific to 454 pyrosequenced data, which resulted in a significant improvement to assembly quality of our test data compared to the baseline assembler bundled by the manufacturer. Other assemblers can be plugged in if necessary, and data from other sequencing technologies such as ABI SOLiD, Illumina and Sanger capillary-based machines can be used. For the prediction stage, we again included a custom combination of feature prediction methods for protein-coding genes, RNA

genes, operon and promoter regions, which improves upon the individual constituent methods. The annotation stage includes several types of protein functional prediction algorithms. We also developed components for comparative analysis, interpretation and presentation (a web-based genome browser), which can be used downstream of our pipeline.

We have tested the pipeline on the bacterium *Neisseria meningitidis*, which is a human commensal of the nasopharanx and which can sometimes cause meningitis or septicemia [143]. When *N. meningitidis* does cause disease, it can be devastating with an approximately 10% fatality rate and 15% sequelae rate. *N. meningitidis* is a highly competent organism with a high recombination rate, and large chromosomal changes are common [86, 146]. This complicates computational genome analysis and makes N. meningitidis an appropriately challenging test for our pipeline. To demonstrate the general applicability of the pipeline, we have also tested it on a different pathogen, *Bordetella bronchiseptica*. *B. bronchiseptica* is a Gram-negative bacterium that can cause bronchitis in humans, although it is more commonly found in smaller mammals [128]. Much like Neisseria, Bordetella has extensive plasticity, likely due to the large number of repeat elements [63]. Here, we analyze the first two complete genome sequences of *B. bronchiseptica* strains isolated from human hosts.

The rest of this chapter is organized as follows. The System and Methods section describes the genomes which we used to test our pipeline, overall organization of the pipeline, and details of the algorithms used to perform tasks in the pipeline. In the Discussion section, we discuss the objectives of our work on the pipeline and how these relate to larger developments in computational biology for next-generation sequencing.

## *4.2 System and Methods*

### 4.2.1 Genome test data

*N. meningitidis* genomes were characterized via 454 pyrosequencing [110] using either a half or one quarter plate runs on the Roche 454 GS-20 or GS Titanium instrument (Table 7). For each genome, a random shotgun library was produced using Roche protocols for nebulization, end-polishing, adaptor ligation, nick repair and single-stranded library formation. Following emulsion PCR, DNA bound beads were isolated and sequenced using long read (LR) sequencing kits. The number of reads produced in the experiments ranged from 200,000 to 600,000, and the average read lengths were between 100 and 330 bases. These data yielded 47.6-94.3 million bases per genome amounting to 20-40x coverage for the approx. 2.1 megabase *N. meningitidis* genomes. After read trimming and re-filtering to recover short quality reads, the data were passed to the first stage of the pipeline - genome assembly.

### 4.2.2 Pipeline organization

The analytical pipeline consists of three integrated subsystems: genome assembly, feature prediction and functional annotation. Each subsystem consists of a top-level execution script managing the input, output, format conversion, and combination of results for a number of distinct software components. A hierarchy of scripts and external programs then performs the tasks required to complete each stage of analysis (Figure 4.2.2).

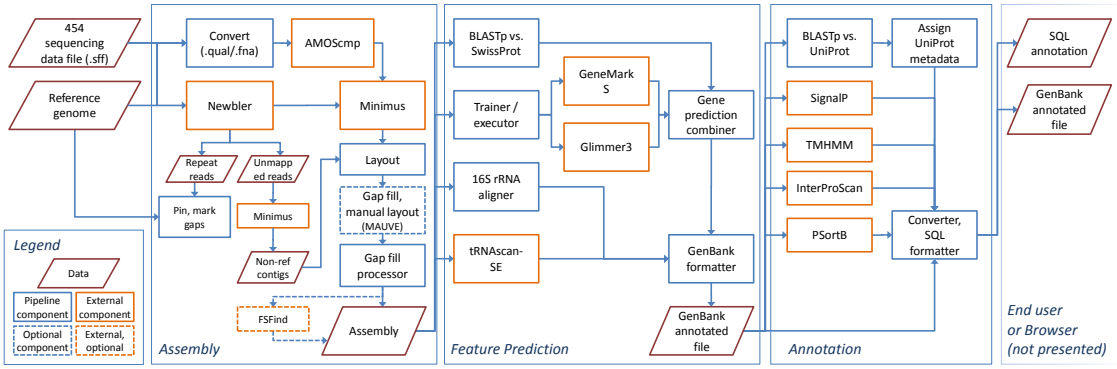
### 4.2.3 Assembly

Genome assembly was performed by evaluating multiple configurations of assemblers including the standard 454 assembler, Newbler (version 2.3), as well the Celera Assembler [118], the Phrap assembler (http://www.phrap.org/) and the AMOScmp mapped assembler [132]. Several other assemblers were evaluated but ultimately excluded from the pipeline due to use limitations: for instance, the ALLPATHS 2

**Table 7:** Summary of sequencing projects used in the pipeline development. Data for each strain are presented in rows.

| Strain ID | Sequence type[a] | Serogroup[b] | Geographic origin[c] | Date collected | Genome size | Closest reference[d] | Substitutions per position vs. ref.[e] | Total reads | Total bases sequenced | Average read length | Coverage[f] | Instrument standard[g] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Neisseria meningitidis* | | | | | | | | | | | | |
| NM13220 | ST-7 | A | Philippines | 2005 | 2.2M | Z2491 | 0.076 | 197067 | 47569493 | 241 | 21× | GS-20 |
| NM10699 | ST-32 | B | Oregon, USA | 2003 | 2.2M | MC58 | 0.053 | 418751 | 81775264 | 195 | 37× | GS-20 |
| NM15141 | ST-11 | C | New York, USA | 2006 | 2.2M | FAM18 | 0.028 | 378773 | 94288660 | 249 | 42× | GS-20 |
| NM9261 | ST-11 | W135 | Burkina Faso | 2002 | 2.2M | FAM18 | 0.030 | 206634 | 69957473 | 338 | 31× | GS Ti |
| NM18575 | ST-2859 | A | Burkina Faso | 2003 | 2.2M | Z2491 | 0.033 | 283888 | 84013571 | 296 | 38× | GS Ti |
| NM5178 | ST-32 | B | Oregon, USA | 1998 | 2.2M | MC58 | 0.050 | 270332 | 88664981 | 328 | 40× | GS Ti |
| NM15293 | ST-32 | B | Georgia, USA | 2006 | 2.2M | MC58 | 0.054 | 276733 | 90951566 | 329 | 41× | GS Ti |
| *Bordetella bronchiseptica* | | | | | | | | | | | | |
| BBE001 | N/A[g] | N/A | Georgia, USA | 1956 | 5.3M | RB50 | 0.056 | 566834 | 229098141 | 404 | 43× | GS Ti |
| BBF579 | N/A | N/A | Mississippi, USA | 2007 | 5.3M | RB50 | 0.104 | 533099 | 228467710 | 429 | 43× | GS Ti |

**Figure 10:** Chart of data flow, major components and subsystems in the pipeline. Three subsystems are presented: genome assembly, feature prediction and functional annotation. Each subsystem consists of a top-level execution script managing the input, output, format conversion, and combination of results for a number of components. A hierarchy of scripts and external programs then performs the tasks required to complete each stage.



assembler [106] required paired-end reads to operate; our evaluation data contained no paired-end reads, and such a requirement unnecessarily constrains the user's options. The widely used Velvet assembler [189] was originally developed as a de novo assembler for Illumina sequencing technology, but its capability has been extended to accommodate 454 data as well. However, we were unable to configure the Velvet assembler to produce a usable assembly or take advantage of reference genomes using 454 data alone.

Evaluation of the results indicated that mapped assemblies of *N. meningitidis* genomes using previously finished strains were of superior quality to de novo assemblies. Using the most appropriate reference strains, it was found that Newbler and AMOScmp complement each other's performance in the assembly stage, with Newbler being able to join some contigs AMOScmp left gapped and vice versa. As a result, we decided to use a combination of these two assemblers' outputs for the final assembly. Then, the Minimus assembler [159] from the AMOS package, a simple assembler for short genomes, was used to combine the constituent assemblies.

We also evaluated alternative base calling algorithms for 454 pyrosequencing data [135] but detected no improvement. Over the course of our project, accuracy of base calling in the Newbler assembler was reported to be significantly improved. We used the latest version of the assembler available at publication time (2.3).

An optional component of the pipeline was created for frameshift detection using FSFind (see Chapter 3). Frameshifts in protein-coding sequences are a known result of pyrosequencing errors caused by undercalls and overcalls in homopolymer runs [97]. The error-correcting algorithm predicts sites of frameshifts caused by sequencing errors, which can then be verified experimentally or corrected speculatively. The user can inspect the dataset to decide whether locations predicted to contain frameshifts break gene models, and patch the sequences to fix up these positions. The prediction stage can then be re-run to correct the gene predictions. While further experimental

analysis to address such errors is desirable (e.g., targeted PCR of predicted error locations or a recently popular choice of combining sequencing technologies such as 454 and Illumina), it incurs extra costs which we aim to avoid.

Unfinished assemblies produced in this stage contained 90-300 contigs each. No paired-end libraries or runs were available for the strains analyzed, and therefore scaffolding of the contigs was a challenge. Manual examination of the assemblies using the MAUVE [44] multiple whole-genome alignment and visualization package revealed numerous locations where contigs could be scaffolded with a small gap or minimal overlap (Figure 11). As an optional step, we produced a table of such positions and a script which would scaffold contigs joined by the gap. Although there is a possibility that rearrangements exist in those gaps as mapped to the closest reference genome, joining was only done after manual examination on a case-by-case basis in positions of high homology and full consensus between four of the reference strains, to minimize this possibility. While we provide the scripts and data format definitions necessary to complete this stage of the pipeline, it involves manual processing of the assembly and is therefore optional. This component is similar in function to Mauve Contig Mover [140] but expands upon it in several ways. An option is provided in the pipeline to use Mauve Contig Mover.

The manually assisted genome assembly procedure resulted in an order-of-magnitude decrease in the number of gaps in comparison to the Newbler assembler (which in turn performed the best out of all standalone assemblers evaluated). In addition, the fully automated assembly metrics (N50 and contig count at equal minimal size) are an approximately 20-50% improvement upon baseline Newbler performance (Table 8).

The contigs in the assembly stage output were named according to the following format: prefix_contig#, where the prefix represents a unique strain identifier and # represents the zero-padded sequential number indicating the contig's predicted order

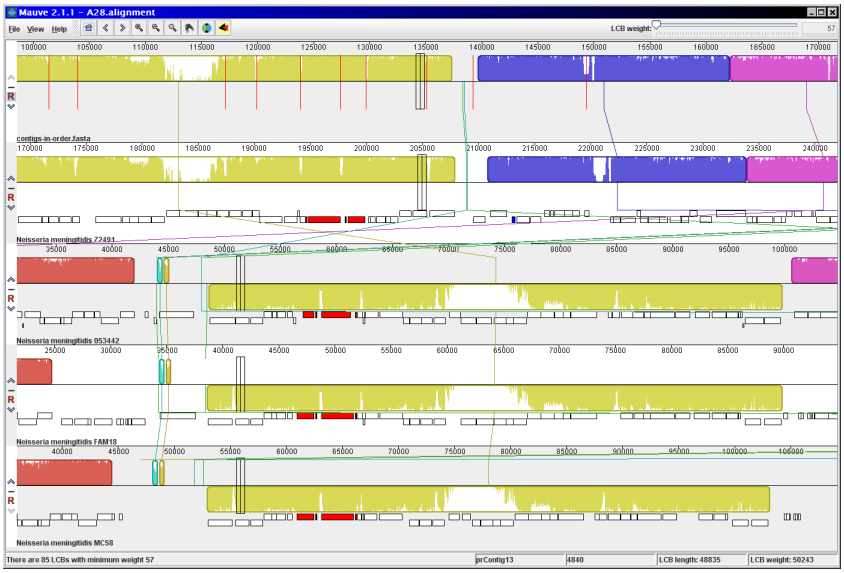



**Figure 11:** Comparative analysis of draft assembly with MAUVE. The top pane represents the active assembly; vertical lines indicate contig boundaries (gaps). The reference genomes are arranged in subsequent panes in order of phylogenetic distance. Blocks of synteny (LCBs) are displayed in different colors (an inversion of a large block is visible between panes 1-2 and 3-5). Most gaps within LCBs were joined in the manually assisted assembly, while considering factors such as sequence conservation on contig flanks and presence of protein-coding regions.

**Table 8:** Summary of assembler performance. Data for each strain are presented in rows. Statistics from standalone assemblers (Newbler and AMOScmp) are presented together with results of the combining protocol (default output of the pipeline) and an optional, manually assisted predictive gap closure protocol. (a) N50 is a standard quality metric for genome assemblies that summarizes the length distribution of contigs. It represents the size N such that 50% of the genome is contained in contigs of size N or greater. Greater N50 values indicate higher quality assemblies. (b) No improvement was detected from the combined assembly in strain BBF579, and the original Newbler assembly was automatically selected. (c) The manual combined assembly protocol was not performed for these projects.

| Strain ID | Newbler statistics | | AMOScmp statistics | | Automatic combined assembly | | Manual combined assembly | |
|---|---|---|---|---|---|---|---|---|
| | Contigs > 500 nt, total size | N50[a], Longest contig | Contigs > 500 nt, total size | N50, Longest contig | Contigs > 500 nt, total size | N50, Longest contig | Contigs > 500 nt, total size | % gapfill, Longest contig |
| NM13220 | 175 | 22K | 202 | 21K | 195 | 31K | 57 | 1.8% |
| | 2.07M | 106K | 2.06M | 77K | 2.25M | 107K | 2.30M | 398K |
| NM10699 | 102 | 52K | 116 | 43K | 83 | 59K | 40 | 1.1% |
| | 2.10M | 143K | 2.10M | 113K | 2.17M | 143K | 2.18M | 435K |
| NM15141 | 147 | 33K | 190 | 22K | 139 | 36K | 50 | 2.0% |
| | 2.06M | 171K | 2.05M | 115K | 2.21M | 171K | 2.28M | 759K |
| NM9261 | 99 | 51K | 133 | 37K | 128 | 64K | 27 | 1.6% |
| | 2.09M | 184K | 2.07M | 170K | 2.16M | 231K | 2.21M | 866K |
| NM18575 | 133 | 30K | 147 | 29K | 220 | 53K | N/A[c] | N/A |
| | 2.09M | 172K | 2.09M | 88K | 2.40M | 231K | | |
| NM5178 | 89 | 56K | 107 | 42K | 104 | 59K | N/A | N/A |
| | 2.13M | 136K | 2.12M | 131K | 2.17M | 136K | | |
| NM15293 | 92 | 52K | 110 | 42K | 107 | 59K | N/A | N/A |
| | 2.08M | 144K | 2.06M | 132K | 2.10M | 144K | | |
| BBE001 | 146 | 70K | 178 | 61K | 214 | 80K | N/A | N/A |
| | 5.05M | 212K | 5.04M | 173K | 5.03M | 252K | | |
| BBF579 | 272 | 57K | 321 | 46K | 272[b] | 57K | N/A | N/A |
| | 4.84M | 88K | 4.84M | 94K | 4.84M | 88K | | |

on the chromosome. For example, the 25th contig for the *N. meningitidis* strain M13220 assembly would be named as CDC_NME_M13320_025. The prefix used in the pipeline is configurable by the user with a command line option.

### 4.2.4 Feature prediction

Feature prediction was performed in the genome using a suite of several programs. To predict genes, we used a combination of de novo and comparative methods. The Glimmer [46] and GeneMark [24] microbial gene predictors were used for de novo prediction, and BLASTp alignment [12] of putative proteins was used for comparative prediction. Self-training procedures were followed for both de novo predictors, and the results, while highly concordant, were different enough (Table 9) to justify the inclusion of both algorithms. BLASTp alignment of all open reading frames (ORFs) at least 90 nt long was performed using the Swiss-Prot protein database [27].

The results of these three methods were combined together using a combiner strategy outlined in Figure 12. In this strategy, we first check that at least half of the predictors report a gene in a given ORF – in our configuration, 2 of the 3 predictors. Then the Met (putative translation start) codon closest to the beginning of the BLAST alignment is found and declared to be the gene start predicted by BLAST. We then find the gene start coordinate reported by the majority of the three predictors and report the resulting gene prediction. If no majority exists, we select the most upstream gene start predicted.

In addition to protein-coding gene prediction, ribosomal genes were predicted using alignment to a reference database of ribosomal operons, and tRNA genes were predicted using the tRNAScan-SE package [104]. The results are summarized in Table 9.

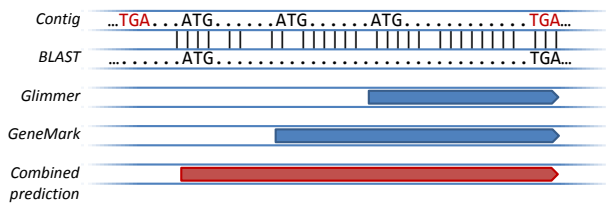



**Figure 12:** Schematics of combining strategy for prediction stage. BLAST alignment start, which may not coincide exactly with a start codon, is pinned to the closest start codon. Then, a consensus or most upstream start is selected.

**Table 9:** Prediction algorithm performance comparison and statistics. Data for each strain are presented in rows. Prediction counts from the 3 standalone gene prediction methods are presented. Counts of protein-coding gene predictions reported by our algorithm and tRNA genes are also shown. Data presented are based on the automatic combined assemblies from Table 8. (a) Number of ORFs with protein-coding gene predictions where all 3 predictors agreed exactly or with a slight difference in the predicted start site. (b) ORFs where only 2 of the 3 predictors made a prediction. (c) Total protein-coding gene predictions reported by the pipeline.

| Strain ID | Gene predictions by GeneMark | Gene predictions by Glimmer3 | Gene predictions by BLAST | ORFs with full consensus[a] | ORFs with partial consensus[b] | Total gene predictions reported[c] | tRNAs predicted by tRNAScan-SE |
|---|---|---|---|---|---|---|---|
| NM13220 | 2530 | 2725 | 1353 | 1325 | 974 | 2299 | 52 |
| NM10699 | 2366 | 2494 | 1317 | 1284 | 826 | 2110 | 51 |
| NM15141 | 2411 | 2578 | 1369 | 1343 | 841 | 2184 | 57 |
| NM9261 | 2370 | 2553 | 1341 | 1308 | 802 | 2110 | 51 |
| NM18575 | 2751 | 2927 | 1495 | 1448 | 1023 | 2471 | 63 |
| NM5178 | 2377 | 2510 | 1315 | 1281 | 816 | 2097 | 52 |
| NM15293 | 2062 | 2040 | 1285 | 1261 | 802 | 2063 | 51 |
| BBE001 | 4793 | 4793 | 2744 | 2732 | 2067 | 4799 | 48 |
| BBF579 | 4649 | 4646 | 2652 | 2635 | 2021 | 4656 | 48 |

Results of the feature prediction stage are saved in a multi-extent GenBank formatted file. Features were named according to the following convention: contig-name_feature-id, where contig-name is as described earlier, and feature-id is a sequential zero-padded number unique to the feature across all contigs. For example, a gene with feature ID 1293 on contig 25 might have the name CDC_NME_M13320_025_1293.

To validate the overall accuracy of the gene prediction stage of the pipeline, we ran our gene prediction tools on the genome of Escherichia coli K12, one of the best-annotated bacterial genomes. Our pipeline was able to detect 97.6% of the annotated E. coli K12 protein-coding genes (analysis described in Section 4.4).

### 4.2.5 Functional annotation

Functional annotation of genome features was also performed using a combination of tools. Annotation of protein coding genes was based on an integrated platform that makes use of six distinct annotation tools, four of which employ intrinsic sequence characteristics for annotation and two that use extrinsic homology-based approaches to compare sequences against databases of sequences and structures with known functions. Information on Gene Ontology (GO) terms, domain architecture and identity, subcellular localization, signal peptides, transmembrane helices and lipoprotein motifs is provided for each protein coding gene (Figure 13).

BLASTp alignment of predicted proteins was performed against the UniProt database (Uniprot, 2009). Homology-based searches were also made across thirteen sequence and protein domain databases with the InterProScan suite [122]. Parsing of the results was carried out against the corresponding InterPro database. The pipeline also stores the top five hits for each gene against the NCBI non-redundant protein database, to provide potentially useful information. All homology searches were run locally. Signal peptides were annotated using the SignalP package [18] and transmembrane domains were annotated with the TMHMM package [93]. State of the art

in subcellular localization algorithms was examined to ensure the best performance given our operational requirements. Insertion sequences (transposases) and proteins reported as virulence factors by VFDB [38, 187] were also annotated. These annotations of virulence-related features make the pipeline particularly useful for projects working with pathogenic prokaryotes. Results of this analysis are summarized in Table 10.

After the functional annotations were determined, a naming scheme was employed for each locus to conform to standard annotation terminology. Specific gene names were assigned according to homology-based results. For genes that had a Uniprot result with a best hit at greater than 91% amino acid sequence identity and an e-value less than 1e-9, the gene assumed the best hit's name. If the best hit had the keyword "hypothetical," then we used a domain name from InterPro to name the gene. For example, if a gene was given the name "hypothetical" from Uniprot and a domain name of "transferase" from InterPro, then the final name was "hypothetical transferase protein."

Therefore most genes that were given "hypothetical" or "putative" prefixes could then be given a more comprehensive name based on further information such as domain names or protein functions. Genes with unknown functions found across many genomes were given the name "conserved hypothetical protein," and all other putative genes with unknown functions were given the name "putative uncharacterized protein."

### 4.2.6 Availability

The pipeline software package is available at our website (http://nbase.biology.gatech.edu). The package contains detailed instructions and scripts for installation of the pipeline and all external programs, documentation on usage of the pipeline and its organization. Components which require large biological databases automatically download

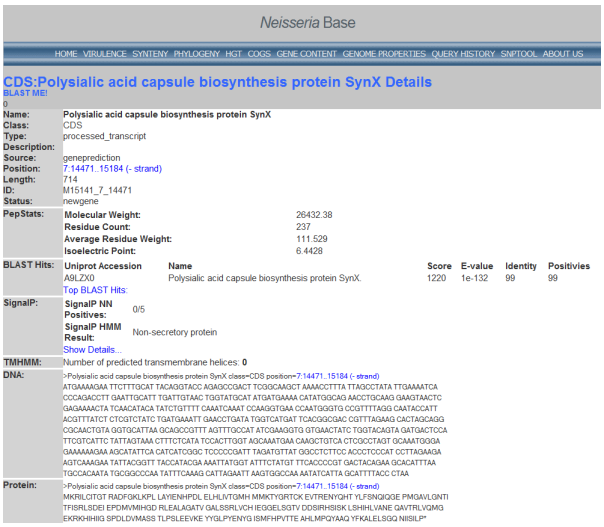



**Figure 13:** Example functional annotation listing of a *N. meningitidis* gene in the Neisseria Base. Draft genome data are shown including gene location, prediction and annotation status, peptide statistics, BLAST hits, signal peptide properties, transmembrane helix presence, DNA and protein sequence. All names, locations, functional annotations, and other fields are searchable, and gene data are accessible from GBrowse genome browser tracks.

**Table 10:** Feature annotation statistics. Data for each strain are presented in rows. Data presented are based on the automatic combined assemblies from Table 8 and the gene predictions from Table 9. (a) Total putative protein-coding sequences analyzed. (b) As predicted by SignalP (Bendtsen, et al., 2004); percentage of total CDS indicated in parentheses. (c) As predicted by TMHMM [93]. (d) As predicted by BLASTp alignment against VFDB [38, 187]; http://www.mgc.ac.cn/VFs/

| Strain ID | Total number of CDS[a] | Signal peptides[b] | Transmembrane helices[c] | Conserved hypothetical proteins | Putative uncharacterized proteins | Functional assignment inferred from homology | Virulence factors[d] |
|---|---|---|---|---|---|---|---|
| NM13220 | 2299 | 326 (14.2%) | 184 (8.0%) | 10 (0.4%) | 708 (30.8%) | 603 (26.2%) | 36 (1.6%) |
| NM10699 | 2110 | 310 (14.7%) | 180 (8.5%) | 5 (0.2%) | 652 (30.9%) | 577 (27.3%) | 45 (2.1%) |
| NM15141 | 2184 | 317 (14.5%) | 173 (7.9%) | 16 (0.7%) | 590 (27.0%) | 583 (26.7%) | 50 (2.3%) |
| NM9261 | 2110 | 303 (14.4%) | 166 (7.9%) | 13 (0.6%) | 591 (28.0%) | 558 (26.4%) | 37 (1.8%) |
| NM18575 | 2471 | 349 (14.1%) | 193 (7.8%) | 13 (0.5%) | 725 (29.3%) | 668 (27.0%) | 48 (1.9%) |
| NM5178 | 2097 | 298 (14.2%) | 177 (8.4%) | 3 (0.1%) | 646 (30.8%) | 572 (27.3%) | 45 (2.1%) |
| NM15293 | 2063 | 304 (14.7%) | 168 (8.1%) | 6 (0.3%) | 613 (29.7%) | 567 (27.5%) | 47 (2.3%) |
| BBE001 | 4799 | 977 (20.4%) | 368 (7.7%) | 9 (0.2%) | 807 (16.8%) | 1184 (24.7%) | 54 (1.1%) |
| BBF579 | 4656 | 934 (20.1%) | 339 (7.3%) | 9 (0.2%) | 739 (15.9%) | 1171 (25.2%) | 45 (1.0%) |

local copies of those databases upon installation. All of the *N. meningitidis* genomes reported here, along with custom annotations and tools for searching and comparative sequence analysis, are available for researchers online at our genome browser database (http://nbase.biology.gatech.edu).

## *4.3 Discussion*

### 4.3.1 Genome biology of *N. meningitidis* and *B. bronchiseptica*

We have used the pathogen *N. meningitidis* for the majority of developmental and production testing of our pipeline. Although *N. meningitidis* gains no fitness advantage from virulence, it occasionally leaves its commensal state and causes devastating disease [117]. Several recent studies have used whole-genome analysis to determine the basis of virulence in this species but none have been conclusive [80, 130, 146]. With the recent advent of next-generation sequencing and the application of an analytical pipeline, such as presented here, this problem and other problems like it can be addressed in individual laboratories on a genome-wide scale. Here, we briefly speculate on a few of the implications of our findings for the genome biology of *N. meningitidis* to underscore the potential utility of our pipeline.

Whole genome analysis of microbes has led to the development of the "pan-genome" concept [170]. A pan-genome refers to the collection of all genes found within different strains of the same species. An open pan-genome means that the genome of any given strain will contain unique genes not found within the genomes of other known strains of the same species. The extent to which microbial pan-genomes are open is a matter of debate [99]. Recent studies have suggested that the *N. meningitidis* pan-genome is essentially open [146], consistent with the fact that it is known to be a highly com-petent species [37, 94]. We evaluated this hypothesis by finding the number of unique genes in each of the seven strains reported here along with seven previously published strains, using the results of our analytical pipeline.

Our findings are consistent with [146] in the sense that every genome sequence was found to contain at least 43 unique genes not found in any other strain. Thus, the *N. meningitidis* pan-genome does appear to be open.

*N. meningitidis* is a human commensal that most often does not cause disease, and avirulent strains of the species are referred to as carriage strains. Results of previous comparative genomic analyses have been taken to suggest that carriage strains represent a distinct evolutionary group that is basal to a group of related virulent strains of *N. meningitidis* [146]. We tested this hypothesis using the results of our analytical pipeline applied to three carriage strains and eight virulent strains of *N. meningitidis*. Whole genome sequences were aligned and pairwise distances between genomes, based on nucleotide diversity levels, were compared within and between groups of carriage and virulent strains. We found that average of the pairwise genome sequence distances within ($w$) the carriage and virulent groups of strains was not significantly different from the average pairwise distances between (b) groups ($w = 0.074 \pm 0.027$ $b = 0.090 \pm 0.014$, $t = 0.693$, $P = 0.491$). This result is inconsistent with the previously held notion that carriage and virulent strains represent distinct evolutionary groups based on whole genome analysis. However, our findings are consistent with earlier work that found little genetic differentiation between carriage and virulent strains of *N. meningitidis* [86].

Currently, there is no unambiguous molecular assay to distinguish *B. bronchiseptica* from other *Bordetella* species. One reason the two *B. bronchiseptica* genomes reported here were characterized was to discover genes unique to the species (i.e. not present in any other *Bordetella* species) to facilitate the development of a *B. bronchiseptica*-specific PCR assay. To identify such genes, we performed BLASTn with *B. bronchiseptica* query genes uncovered by our pipeline against other *B. bronchiseptica* strain genomes along with four genomes of closely related *Bordetella* species. We uncovered a total of 223 genes that are present in all *B. bronchiseptica* strains

and absent in all other *Bordetella* species. To narrow down this set of potential PCR assay targets, we searched for the most conserved *B. bronchiseptica*-specific genes. As a point of reference, we determined the sodC gene used in the *N. meningitidis*-specific PCR assay [94] to be 99.6% identical among all six completely sequenced strains of *N. meningitidis*. There are 7 *B. bronchiseptica*-specific genes with $\geq$99.6% sequence identity; these genes represent a prioritized list of potential PCR assay targets.

### 4.3.2 Computational genomics pipeline

We have presented our computational genomics pipeline, a local solution for automated, high-throughput computational support of prokaryotic genome sequencing projects. While the revolution in sequencing technology makes possible the execution of genome projects within individual laboratories, the computational infrastructure to fully realize this possibility does not yet exist. We made a comprehensive effort to put the tools required for this infrastructure into the hands of biologists working with next-generation sequencing data. Our aim in the course of this project was to facilitate decentralized biological discoveries based on affordable whole-genome prokaryotic sequencing, a mode of science termed "investigator-initiated genomics". For example, one project enabled by the pipeline in our laboratory is a platform for SNP detection and analysis in groups of bacterial genomes.

One of our major goals was to provide full automation of our pipeline's entire workflow, and this has been achieved. On the other hand, to allow computationally savvy users to realize the power of customizability, a semi-automated process is desirable. We have made an effort to strike a balance between these objectives, and provide a modular, hierarchically organized structure to permit maximum customization when so desired.

The state of the art in prokaryotic computational genomics moves at a formidable

pace. The modular organization of our pipeline, along with the emphasis on integration of complementary software tools, allows us to continually update our platform to keep pace with developments in computational genomics. For instance, if a new, better assembler becomes available, we can include its results in the assembly stage with a simple change to the pipeline code.

## *4.4 Validation on known data*

Optional parts of the assembly stage included manual gap joining curation for scaffolding in the absence of paired-end reads, and frameshift detection for homopolymer-induced frameshifts.

The manual gap joining stage involved the layout of contigs according to their aligned position on the reference using the AMOS package and manual examination of each gap, adjacent contig alignments and reference annotation in the MAUVE visualization tool. We then recorded all gaps considered safe to join on the basis of this information into a gap fill specification file, which is a tabulated file in the format "contig 1 name, contig 1 end position, reference start position, gap length, reference end position, contig 2 name, contig 2 start position", with one gap fill description per line. A script was then used to produce the final FASTA formatted output, with gaps filled with N (unknown nucleotides) by default, or optionally with sequence from the reference strain.

The homopolymer-induced frameshift stage used the FSFind package from (Kislyuk et al., 2009). Briefly, this package creates a GeneMark model of the genome, makes gene predictions, and then scans the genome for possible frameshift positions on the basis of ORF configuration and coding potential. Once the possible frameshift sites are identified, a putative translation of the protein possibly encoded by the broken gene is compared against a protein database (SwissProt by default). The predicted

frameshift site is also scanned for adjacent homopolymers. A heuristic set of confidence score cutoffs is then used to provide a set of frameshift predictions while minimizing the false positive rate. The resulting homopolymer error predictions can be used for either targeted re-sequencing or predictive correction using a supplied script. The output can be manually run through the gene prediction and annotation stages of the pipeline again.

To demonstrate the overall accuracy of the prediction stage, we ran it on the genome of E. coli K12, one of the best-annotated bacterial genomes. Our stage was able to detect 97.6% of intact ORFs annotated as protein-coding, and exactly predict starts in 74% of those.

The complete genome of Escherichia coli K12, accession number NC_000913.2, was downloaded from GenBank and its DNA sequence extracted into a FASTA file. The file was then given as input to the prediction component of the pipeline, which utilized the combination of GenMark, Glimmer3 and BLAST vs. SwissProt. To remove bias caused by the presence of most E. coli protein-coding sequences in SwissProt, we also ran the same component configured to run without BLAST based prediction, using only the de novo predictors. The component used the input data to self-train the predictors. See main text for details of the combination algorithm.

GenBank-formatted output of the component was tabulated to include only CDS sequence annotation boundaries. The same procedure was done for the reference E. coli annotation from the original file. Sequences with frameshifted and interrupted CDS (i.e. non-intact ORFs) were omitted from the comparison due to lack of capability in our prediction component to detect such structures at this time.

## *4.5 Acknowledgements*

We are grateful to all participants of the Georgia Tech Computational Genomics class; to Leonardo Mariño-Ramírez for valuable guidance and input; and to Joshua S. Weitz

for his support.

## *4.6 Funding*

The work presented in this chapter was supported by Defense Advanced Research Projects Agency [A.O.K. by HR0011-05-1-0057]; The Alfred P. Sloan Foundation [I.K.J. by BR-4839]; Georgia Research Alliance [I.K.J., P.J., S.A. by GRA.VAC09.O]; Centers for Disease Control and Prevention [L.S.K. by 1 R36 GD 000075-1]; and Bioinformatics program, Georgia Institute of Technology [J.H., P.J, V.N., S.A.].

# CHAPTER V

# METAGENOMIC BINNING

The development of effective environmental shotgun sequence binning methods remains an ongoing challenge in algorithmic analysis of metagenomic data. While previous methods have focused primarily on supervised learning involving extrinsic data, a first-principles statistical model combined with a self-training fitting method has not yet been developed.

We derive an unsupervised, maximum-likelihood formalism for clustering short sequences by their taxonomic origin on the basis of their $k$-mer distributions. The formalism is implemented using a Markov Chain Monte Carlo approach in a $k$-mer feature space. We introduce a space transformation that reduces the dimensionality of the feature space and a genomic fragment divergence measure that strongly correlates with the method's performance. Pairwise analysis of over 1000 completely sequenced genomes reveals that the vast majority of genomes have sufficient genomic fragment divergence to be amenable for binning using the present formalism. Using a high-performance implementation, the binner is able to classify fragments as short as 400 nt with accuracy over 90% in simulations of low-complexity communities of 2 to 10 species, given sufficient genomic fragment divergence. The method is available as an open source package called LikelyBin.

An unsupervised binning method based on statistical signatures of short environmental sequences is a viable stand-alone binning method for low complexity samples. For medium and high complexity samples, we discuss the possibility of combining the current method with other methods as part of an iterative process to enhance the resolving power of sorting reads into taxonomic and/or functional bins.

The rest of this chapter is based on published work which first appeared in the following article:

A. Kislyuk, S. Bhatnagar, J. Dushoff, and J. Weitz, "Unsupervised statistical clustering of environmental shotgun sequences," BMC Bioinformatics, vol. 10, no. 1, pp. 316+, 2009.

## *5.1 Background*

Metagenomics, the study of the combined genomes of communities of organisms, is a rapidly expanding area of genome research. The field is driven by environmental shotgun sequencing (ESS), a technique of applying high-throughput genome sequencing to non-clonal DNA purified directly from an environmental sample. This removes the requirement to isolate and cultivate clonal cultures of each species, allowing an unprecedented broad view of microbial communities.

Thus far, environments such as acid mine drainage [174], Scottish soil [173], open ocean [144], termite gut [179], human gut [67], and neanderthal [126] have been sequenced, to name a few. Attention has been directed to bacterial and viral fractions of these communities, with eukaryotic metagenomics pioneered by projects such as the marine protist census [127]. Complexity of these communities varies greatly from 5 to several thousand identifiable bacterial species. These projects have uncovered vast amounts of previously unobserved genetic diversity [13, 82]. For example, "deep sequencing" using 454 pyrosequencing suggests that possibly tens of thousands of species coexist in a single ml of seawater [157].

Given this wealth of genomic data it is becoming possible to make increasingly precise biological inferences regarding the structure and functioning of microbial communities [74, 188, 51]. As but one example, the discovery of a novel proteorhodopsin gene was the first step in uncovering a previously unknown, yet apparently dominant,

mechanism for phototrophy in the oceans [17]. Characterization of functional diversity is limited by our ability to classify sequences into distinct groups that reflect a desired taxonomic or functional resolution.

Shotgun metagenomic DNA is sequenced in fragments of 50 to 1000 nucleotides, then possibly assembled into longer sequences (contigs). Phylogenetic binning, the task of classifying these sequences into bins by taxonomic origin, then becomes critical to separate metagenomic data into coherent subsets plausibly belonging to separate organisms. This task is challenging due to the short length of available fragments. Bacterial communities of very high complexity, with thousands of species present, further complicate the task.

While methods such as 16S bacterial community censuses [123] and functional- or sequence-based screening surveys are the forerunners of modern metagenomics, indiscriminate whole-genome shotgun sequencing may be the defining approach of the discipline today. This approach has recently generated vast amounts of data, facilitated by continual capacity increases and quality improvements at major sequencing centers and the emergence of cost effective very high throughput Next Generation sequencing (NGS) (454 pyrosequencing [111], Illumina [21] and SOLiD [153]). At the highest diversity levels, the reads may not be assembled at all due to the sparseness of even the highest throughput sequencing methods and the danger of chimeric assemblies, arising from sampling so many organisms at once, leaving the binner with raw reads. Binning methods therefore aim to be able to operate on very short read lengths provided by next-generation sequencing, although most, including the present approach, are only able to go down to 454 pyrosequencing read length (about 400 nt) and not to microread length (30 to 100 nt).

Classic approaches to phylogenetic determination of species identities from environmental sequences rely on identifying variants of highly conserved genes, like 16S rRNA or recA [98]. This approach is not applicable on a full metagenomic scale for

two reasons: first, ribosomal or marker gene sequences comprise a small fraction of the bacterial genome, so most shotgun sequences do not contain them and cannot be classified this way; and second, organisms with identical or closely related 16S genes have been shown to exhibit variations in essential physiological functions [177]. Other approaches are broadly divided into sequence similarity based classifiers such as MEGAN [83], which rely on BLAST or other alignments, and sequence composition based classifiers, which rely on statistical patterns of oligonucleotide distributions.

Many solutions integrate the task of phylogenetic assignment (labeling) together with that of binning per se (clustering) of genomic fragments. However, with unsupervised methods, like the one presented here, labeling is not possible as part of the algorithm and has to be performed by other means, like analyzing the correspondence of generated clusters to known phylogenies.

Sequence classification based on oligonucleotide distributions has been the basis for gene finding applications since the early 1990s. In 1995, Karlin and Burge [88] noted that dinucleotide distribution is relatively constant within genomes but varies between genomes. Since then, this property has been extensively studied and generalized to other oligonucleotide lengths [49]. With the advent of ESS, several binning methods have used oligonucleotide distributions of various orders to build supervised and semi-supervised classifiers. These include PhyloPythia [115], CompostBin [36], and self-organizing map (SOM) based methods [7, 34, 35].

Machine learning-based classification algorithms like those used for binning are categorized into supervised, semi-supervised, and unsupervised classes. Supervised algorithms accept a training set of labeled data used to build their models, which are then applied to the query data. In case of binning, this training set consists of genomic sequences labeled according to the species they originate from. Semi-supervised algorithms use both training set data and query data to build their models. Unsupervised algorithms use no training data and derive their models directly from

the query input. While methods described above have achieved considerable success in classifying short anonymous genomic fragments, their supervised nature makes them reliant on previously sequenced data. For example, BLAST-based methods are completely dependent on the presence of sequences related to the query in the database. While semi-supervised clustering methods can have significant generalizing power, their accuracy still depends on similarity of input data to their training set.

To our knowledge, two approaches to unsupervised metagenomic binning have been published. TETRA [168, 169] explores the applications of $k$-mer frequency statistics to metagenomic data. The authors state that their method is suitable as a "fingerprinting technique" for longer DNA fragments, though not as a general-purpose binning method for single-read 454 pyrosequenced or Sanger fragments, and an application of methods including TETRA to binning of fosmid-sized DNA is used in [183]. Abe *et al.* [7] used self-organizing maps (SOM) in combination with principal component analysis (PCA) on 1- and 10-Kb fragments, and this method was evaluated and enhanced in [34] using growing self-organizing maps (GSOM), an extension of SOM, on 8- and 10-kb fragments.

Given the apparent diversity of metagenomic samples and the significant fraction of the full bacterial phylogeny with no sequenced representatives [3, 177], as well as possible undiscovered diversity of the tree of life, binning methods must perform well on previously unseen data. Semi-supervised methods may be able to extrapolate on this data, but if not, unsupervised clustering will be a necessary part of a combined-method binning approach. We present LikelyBin, a new statistical approach to unsupervised classification of metagenomic reads based on an explicit likelihood model of short genomic fragments [5].

The rest of this chapter is organized as follows. The Methods section introduces a formal definition of the binning problem, the application of the Markov Chain Monte Carlo (MCMC) formalism, and the feature space and likelihood model used. We

discuss numerical methods used in the implementation, including a novel coordinate transformation which achieves dimension reduction for the feature space of $k$-mer frequencies, and the genomic fragment divergence measure $D_n$, a novel statistical measure we developed for performance evaluation of our algorithm. The Results section presents performance evaluations of our method on mixtures of 2 to 10 species compiled from completed genomes available in GenBank, with fragment lengths starting at 400 nt, as well as accuracy trends over different fragment lengths and mixing ratios. We also present results on the FAMeS [114] dataset and compare the current method to a semi-supervised binning method based on $k$-mer distributions [36]. The Conclusion section explains the applicability of our method, its speed and availability, as well as important future directions for improvement.

## *5.2 Methods*

### 5.2.1 The binning problem

We state the problem as follows: given a collection of $N$ short sequence reads from $M$ complete genomes, how can we predict which sequences derive from the same genome? In our model, we represent a genome as a string of characters deriving from a stochastic model with parameters $\Theta$, referred to here as a master distribution. We make the simplifying assumption that the oligonucleotide distribution is uniform across the bacterial chromosome. This assumption is not satisfied biologically; gene-coding, RNA-coding, and noncoding regions, leading and lagging strands of replication, and genomic islands resulting from horizontal gene transfer can all exhibit distinct oligonucleotide distributions. Accurate classification of these regions in metagenomic fragments is an open problem which requires complex statistical models that we have yet to incorporate into our framework, and which are targets for subsequent model development. Nonetheless we have found that clustering of short reads using the above assumption is sufficiently accurate for use in low complexity

metagenome samples.

Given this assumption of statistical homogeneity, we model a collection of sequences from a single genome as realizations of a single stochastic process. Similarly, we model a collection of sequences from multiple genomes as realizations of multiple stochastic processes, one per genome, each with its own master distribution. We are interested in determining which sequences in a metagenomic survey are likely to have been drawn from the same genome and, consequently, the statistical distributions of oligonucleotides within each of the master distributions. If the number of master distributions is unknown, then we must include some prior estimate to close the model. Thus, even in cases where due to insufficient coverage it is impossible to assemble disparate segments of a consensus genome together, a binning algorithm should still be able to group reads together based on their statistical distribution of oligonucleotides.

The simplest model of a genome would be a random collection of letters, A, T, C, and G. The master distribution of a single genome can then be represented as a single probability, $p_A$, denoting the fraction of A-s in the genome. Base complementarity requires $p_A = p_T$ and $p_C = 1/2 - p_A = p_G$. A more complex representation would be to assume that genomes are random collections of $k$-mers. When $k = 1$, each nucleotide is independent of the previous. When $k = 2$, the genomes are random collections of dimers and so on. However, when $k \geq 2$, inherent symmetries are present in this representation since all but the first letters of the current $k$-mer are also contained in the next $k$-mer. In a metagenomic dataset, each short fragment derives from a single master distribution, $\theta_i$, which is represented a fraction $f_i$ of times.

How then can we infer the most likely $\Theta \equiv (\theta_1, \theta_2, \ldots, \theta_M)$ and $F \equiv (f_1, f_2, \ldots, f_M)$ given a set of $N$ sequences $S \equiv (s_1, s_2, \ldots, s_N)$? To do so, we must calculate the likelihood $\mathcal{L}(S|\Theta, F)$ of observing the sequences $S$ given the parameters $\Theta$ and $F$. Then, we must estimate the values of $\Theta$ and $F$ that maximize the likelihood $\mathcal{L}$. Below, we demonstrate the use of a MCMC algorithm to perform this task.

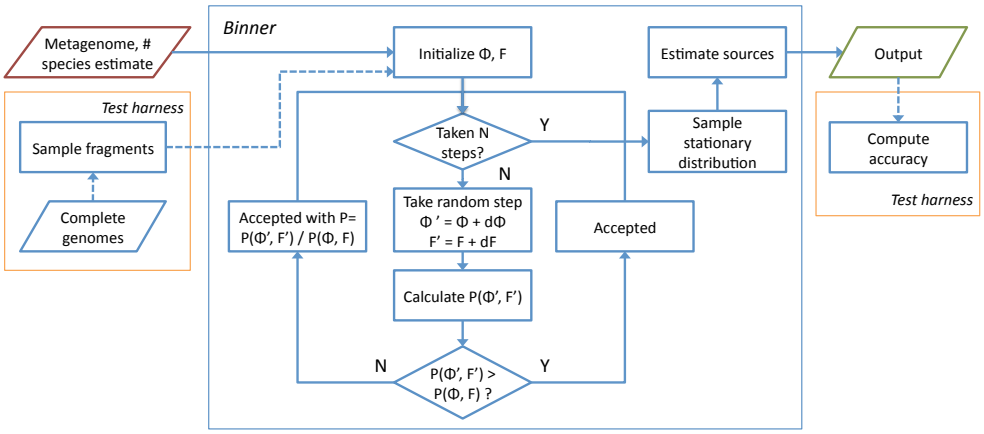



**Figure 14:** Diagram of binning data pathways and main MCMC iteration loop.

### 5.2.2 MCMC framework

We are interested in finding the values of $\Theta$ and $F$ that maximize the likelihood, $\mathcal{L}$. The MCMC approach has been described in detail elsewhere [160]. Given an initial parameter setting and a metagenomic data set, we implement the following Metropolis-Hastings algorithm to MCMC maximum likelihood estimation: (i) Determine the likelihood of the dataset $\mathcal{L}(\Theta, F|S)$; (ii) Choose some $\Phi = \Theta + \mathrm{d}\Theta$, and $G = F + \mathrm{dF}$ and determine its likelihood, $\mathcal{L}'(\Phi, G)$, such that both $\Phi$ and $G$ exist in the same high-dimensional simplex as $\Theta$ and $F$ respectively; (iii) Accept the new value given a probability 1 if $\mathcal{L}'(\Phi, G) > \mathcal{L}(\Theta, F)$ and with probability $\mathcal{L}'(\Phi, G)/\mathcal{L}(\Theta, F)$ otherwise; (iv) Repeat, and after a burn-in period determine the values $\hat{\Theta}$ and $\hat{F}$ which maximize $\mathcal{L}(S|\Theta, F)$. We can then utilize the resulting model of sequence parameters to classify sequences and estimate the most likely oligonucleotide distribution of each of the originating master distributions. The iterative process, together

with key stages of the entire binning algorithm, is illustrated in Figure 14. Some technical details necessary for the implementation follow.

#### *5.2.2.1 Likelihood model.*

Consider a nucleotide sequence $s = c_1c_2c_3 \dots c_\ell$. We would like to know the probability of observing such a sequence given some underlying model. We assume that our sequence is selected from broken pieces of double-stranded DNA, and thus that complementary nucleotide sequences have the same probability: i.e., $L(s) = L(s')$, where $s' = c'_\ell \dots c'_1$, and $c'_i$ is the nucleotide complementary to the nucleotide $c_i$.

We assume that the probability of our sequence is determined by a set of $2^k$ $k$-mer probabilities $p_{c_1 \dots c_k}$. That is, we write:

$$P(s) = p_{c_1 \dots c_k} \prod_{j=k+1}^{\ell} P(c_j | c_{j+1-k} \dots c_{j-1}) \tag{4}$$

Assuming we know probabilities for all of our $k$-mers, we have probabilities for $k-1$-mers as marginals. Thus we can write:

$$P(s) = p_{c_1 \dots c_k} \prod_{j=k+1}^{\ell} \frac{p_{(c_{j+1-k} \dots c_j)}}{p_{(c_{j+1-k} \dots c_{j-1})}} \tag{5}$$

As an example, the probability of a sequence given a set of known dimer frequencies is:

$$P(s) = p_{c_1 c_2} \prod_{j=3}^{\ell} \frac{p_{(c_{j-1} c_j)}}{p_{(c_{j-1})}} \tag{6}$$

Note that we assume the marginal probabilities are well defined: i.e., that we get the same marginal probability if we collapse a $k$-mer to a $k-1$-mer by summing over the first, or the last, nucleotide.

The likelihood of observing $N$ sequences given $M$ master distributions is

$$\mathcal{L} = \prod_{i=1}^{N} \left( \sum_{m=1}^{M} f_m P_m(s_i) \right), \tag{7}$$

where $P_m(s_i)$ is the probability of generating the $i$-th sequence given the $m$-th master distribution.

A simple example of likelihood computation according to the described model is given in the Appendix.

*5.2.2.2 The space of k-mer frequencies.*

Given the assumption of uniformity of the $k$-mer (oligonucleotide) distribution across each genome, we can impose three kinds of constraints on the $k$-mer frequency space. This space is a subspace of $\mathbf{R}^{4^k}$, subject to three kinds of constraints: all $k$-mer frequencies sum to 1, e.g.

$$p_{AAA} + p_{AAT} + \ldots + p_{CCC} = 1;$$

each $k$-mer has the same frequency as its complement; and all marginal probabilities are consistent over all margins, e.g.

$$p_{AAA} + p_{AAT} + p_{AAG} + p_{AAC} = p_{AA}.$$

We then derive a transformation of the original $k$-mer frequency vector,

$$x = [p_A, p_T, p_G, p_C, p_{AA}, p_{AT}, p_{AG}, p_{AC}, p_{TA}, \ldots],$$

into the independent coordinate space. To generalize and automate the process, we perform it for each case from 1-mers (4 dimensions before removing redundancies) to 5-mers (1364 dimensions before removing redundancies) by generating all equations governing the constraints above. We use the notation $[A|b]$ to denote the matrices of the constraint equation $Ax = b$ by generating rows for each constraint type. For example, for $k = 2$, we write the summation, complementarity and marginality constraints as follows:

$$\text{Summation: } \left[\begin{array}{cccccccccccccccccccc|c} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right], \tag{8}$$

**Table 11:** Redundancies in oligonucleotide dimension space

| $k$ | Total dimensions | Independent dimensions |
|---|---|---|
| 1 | 4 | 1 |
| 2 | 20 | 7 |
| 3 | 84 | 25 |
| 4 | 340 | 103 |
| 5 | 1364 | 391 |

$$\text{Complementarity:} \left[\begin{array}{cccccccccccccccccccc|c} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & & & \vdots & & & & & & & & & & & \end{array}\right], \quad (9)$$

$$\text{Marginality:} \left[\begin{array}{cccccccccccccccccccc|c} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & & & \vdots & & & & & & & & & & & \end{array}\right]. \quad (10)$$

We find the nullspace of the resulting matrix $A$ and use it to perform the transformation. The resulting number of independent dimensions is shown in Table 11. The MCMC simulation then performs the search in the independent coordinate space. For $k > 6$, the matrix $A$ becomes too big to compute its nullspace using a non-parallelized algorithm. Even for $k = 6$, the number of independent dimensions is so large that the MCMC simulation takes an intractable amount of time. Therefore, we only generalize our algorithm up to $k = 5$.

*5.2.2.3 Initial conditions.*

The choice of initial conditions can dramatically alter the speed of convergence of a MCMC solver. We used the same initial conditions for comparison of model results, specified by the frequencies of $k$-mers in the entire dataset provided as input (i.e., the weighted average of all sources' contributions to the dataset). Other possibilities,

implemented but not chosen as the default, include taking uniformly distributed frequencies, randomizing the starting condition, or using principal components analysis with $K$-means clustering to obtain initial cluster centroids. We verified that convergence, when it did occur, did not depend sensitively on initial conditions (Additional files 20 and 21).

#### *5.2.2.4 Finding the maximum likelihood model.*

Once the predefined number of timesteps has elapsed, the model with the largest log likelihood is selected.

Note that the MCMC framework is amenable to a Bayesian approach, which we implemented as an alternative. Once the equilibrium state has been reached we calculate the autocorrelation of frequencies and estimate a window over which frequencies show no significant autocorrelations. Given a specified prior distribution $p(\Theta, F)$ for the master distribution and frequencies, the Metropolis-Hastings approach will converge to the true posterior distribution of $\pi(\Theta, F|S) \propto \mathcal{L}(S|\Theta, F)p(\Theta, F)$. In our case we used an uninformed prior distribution so long as positivity and all other specified constraints among $k$-mer probabilities were preserved. We then sample from the equilibrium state to find $\pi(\Theta, F)$. Averages of master distributions in the posterior distribution also preserve the constraint conditions because of the linearity of the averaging operator. Accuracy of the model was similar whether using the maximum likelihood model or the average of the posterior distribution (Additional file 22). Full posterior distributions of $k$-mer models could be used to estimate posterior distributions of binning accuracy.

### 5.2.3 Numerical details

#### *5.2.3.1 Precision.*

Due to precision limitations of the machine double precision floating point format, the model likelihood calculation is performed in log space. Denote the old model

under consideration as $\mathbf{M} = \{M_1, M_2, \ldots M_m\}$, and the new (perturbed) model as $\tilde{\mathbf{M}} = \{\tilde{M}_1, \tilde{M}_2, \ldots \tilde{M}_m\}$. The log likelihood of a single model is

$$\begin{aligned}
\log \mathcal{L} &= \log \prod_{i=1}^{N} \left( \sum_{m=1}^{M} f_m P_m(s_i) \right), \\
&= \sum_{i=1}^{N} \log \sum_{m=1}^{M} f_m P_m(s_i), \\
&= \sum_{i=1}^{N} \log \sum_{m=1}^{M} f_m \left( p^m_{c_1 c_2} \prod_{j=3}^{l} \frac{p^m_{c_{j-1} c_j}}{p^m_{c_{j-1}}}, \right)
\end{aligned}$$

and note that the innermost fraction contains higher-order terms when working with Markov chain orders higher than 2. The innermost product term is a product of on the order of 1000 terms of magnitude $\approx 1/4$. However, $1/4^n$ exceeds double floating point precision at $n \approx 540$. To prevent underflow, we find the $P_m(s_i)$ of highest magnitude and divide the inner sum by it. This allows log space evaluation of the highest magnitude term and ensures that any terms whose precision is lost are at least $\approx$ 1e300 times smaller.

The model log likelihood ratio is then $\log \frac{\mathcal{L}(\tilde{\mathbf{M}}|\mathbf{S})}{\mathcal{L}(\mathbf{M}|\mathbf{S})} = \log \mathcal{L}(\tilde{\mathbf{M}}|\mathbf{S}) - \log \mathcal{L}(\mathbf{M}|\mathbf{S})$. If this term exceeds 0, the new model is more likely to be observed than the old.

The MCMC iteration loop was implemented with the Metropolis-Hastings criterion. From an initial model, a perturbed model $M_N$ is generated. The new model's probability is evaluated as above and compared to that of the currently selected model $M_C$. If higher, the new model is selected; otherwise, the new model is selected with probability $p = \exp\left(\log \mathcal{L}(M_N|S) - \log \mathcal{L}(M_C|S)\right)$. The step is repeated $N$ times ($N$ is fixed at 40000 for the experiments described). Each selected model is stored in a model record for later sampling.

#### *5.2.3.2 Computing the perturbation.*

The statistical model consists of sub-models for each source. The perturbation step is performed for every sub-model independently. Every sub-model consists of a complete

$k$-mer frequency vector, $\{p_A, p_T, p_G, p_C, p_{AA} \ldots\}$. It is perturbed by scaling each vector of the basis matrix $A$ by a random number $r_i$ drawn from a Gaussian distribution with mean 0 and constant variance (computed as described below), then adding each scaled vector in succession to the frequency vector. The basis matrix $A$ is precomputed for each $k$-mer model order from 2 to 5 and supplied with the program. The computation is performed by generating a system of equations representing the base complementarity, marginal, and summation constraints and using the standard nullspace algorithm supplied with GNU Octave.

The perturbation step variance must be calibrated independently for each dataset. An excessive variance will result in too many suboptimal perturbations as well as perturbations placing the frequency vector outside the unit hypercube (those perturbations are rejected). A variance that is too small can result in an inability to escape local maxima in the model search space and an inability to reach the stationary phase before the pre-determined number of steps is taken. To calibrate the variance, the MCMC iteration is started independently for a reduced number of steps, and different variances ranging from $1e-3$ down to $1e-8$ are tried. With each trial, the number of new model acceptances is recorded. We consider the fraction $f = \frac{\#\texttt{acceptances}}{\#\texttt{timesteps}}$. Once the variance yielding $f$ closest to 0.234 is found (a heuristic level of acceptances that has become standard[160], p. 504), we use this variance for the main run. Convergence to the stationary phase occurred after 40,000 iterations in all cases of interest.

#### *5.2.3.3 Computing the prediction.*

To derive the final model prediction, the model with the overall maximum log likelihood is selected.

The full MCMC simulation is repeated a selected number of times (to increase performance, the classifier was run in parallel on an 8-core machine; each core was

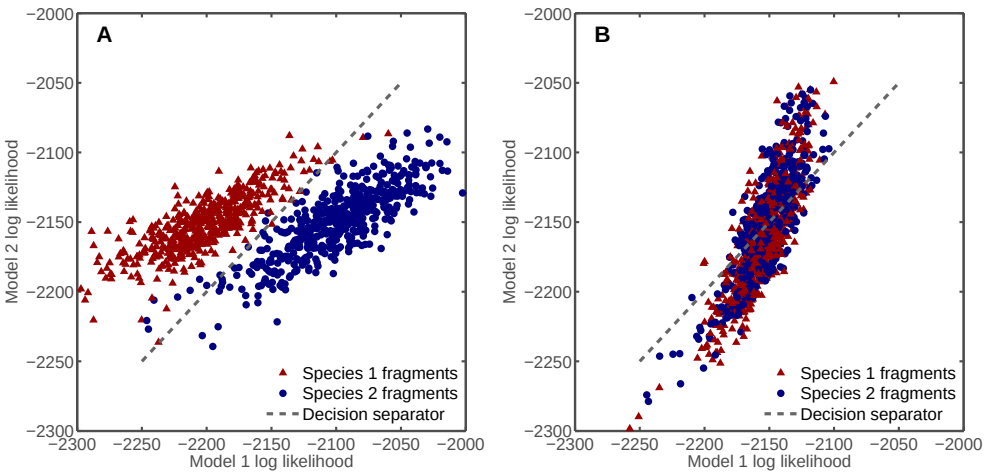



**Figure 15:** Log likelihood values of fragments from pairs of species according to models fitted by the classifier. Points' positions on the two axes represent log likelihoods of each fragment according to the first and second model, respectively. A, *Helicobacter acinonychis* vs. *Vibrio fischeri*, good separation (98% accuracy, D=1.31); B, *Streptococcus pneumoniae* vs. *Streptococcus pyogenes*, poor separation (57% accuracy, D=0.22). Fragment length was 800 in both cases. 500 fragments per species were supplied.

assigned to run one MCMC simulation for a total of 8 restarts). Final model predictions are compared between different runs, and the best overall prediction is selected according to its model likelihood (described above).

The classifier then assigns a putative source to each sequence fragment it was initially queried with. For every fragment, its likelihood according to each sub-model in the final predicted model is computed, and the sub-model supplying the highest likelihood is selected. Since the sources are anonymous, they are referred to simply by indices from 1 to $n$ corresponding to each sub-model's index in the final predicted model. Figure 15 illustrates the log likelihood comparison process for all fragments in a given dataset, according to the best model selected as a result of this process.

### 5.2.4 Testing methodology

Simulated metagenomic datasets were created by selecting two or more genomic sequences as source DNA. Sequence fragments were selected at random positions within

source sequences; overlaps were allowed to occur. Fragment size was fixed for all fragments for each experiment. The total number of fragments per source was selected either according to overall source length or at specified frequency ratios (e.g., 2:1, 10:1:1). The number of sources in each testing dataset was supplied to the classifier.

Accuracy of the classifier is calculated as follows. Every possible matching of source genomic sequence names to classifier output indices is considered, e.g. $\{seq1 \rightarrow 1, seq2 \rightarrow 2\}, \{seq1 \rightarrow 2, seq2 \rightarrow 1\}$. The number of correct assignments made by the classifier is then counted for each matching and the matching with the highest number of correct assignments is selected. Accuracy is then given as $\frac{\#\texttt{correct assignments}}{\#\texttt{fragments}}$.

To evaluate separability of the randomly generated datasets according to the classifier's model, we also define and compute the *genomic fragment divergence* between two sources' $k$-mer distributions. First, we compute the mean, $\mu$, and standard deviation, $\sigma$, of each $k$-mer frequency for each source across fragments originating from that source. The genomic fragment divergence of $k$-mer order $n$ is then given by

$$D_n(S_1, S_2) = \sum_{k=1}^{n} \frac{1}{4^k} \sum_{\substack{i \in \{\text{k-mers} \\ \text{of order } k\}}} \frac{(\mu_i^{S_1} - \mu_i^{S_2})^2}{(\sigma_i^{S_1})^2 + (\sigma_i^{S_2})^2}. \tag{11}$$

Generalizing to $M$ species, let $\{S\} = \{S_1, S_2, \ldots S_M\}$. Then we define

$$D_n(\{S\}) = \min_{\substack{\forall i,j \in [1,M] \\ i \neq j}} \left(D_n\left(S_i, S_j\right)\right). \tag{12}$$

Figure 17 illustrates the distribution of genomic fragment divergences between completed bacterial genomes.

A different formula for intergenomic difference, called the *average absolute dinucleotide relative abundance difference* is [33]: $\delta^\star\left(f, g\right) = \frac{1}{16}\sum_{X,Y} |\rho^\star_{XY}(f) - \rho^\star_{XY}(g)|$, where $\rho^\star_{XY} = \frac{f^\star_{XY}}{f^\star_X f^\star_Y}$. This formula encompasses dinucleotides and pairwise comparisons of entire sequences only, and uses dimer frequency biases instead of absolute frequencies and their deviations in a hierarchical fashion. The advantage of the proposed genomic fragment divergence is in its consideration of fragment length induced

variation in $k$-mer frequency distributions and integration of information content from multiple $k$-mer lengths into one measure.

## *5.3 Results and Discussion*

The accuracy and applicability of the present method in binning short sequence fragments from low complexity communities (2-10 species) was systematically analyzed using a variety of species, varying fragment lengths, and varying ratios of fragment representation.

First, a set of 1055 completed bacterial chromosomes was retrieved from GenBank. This set was randomly sampled for sets of 2, 3, 5, 10 genomes at a time, representative of various genomic fragment $k$-mer distribution divergences. Binning results for nearly 1800 simulated communities comprised of 2 or 3 genomes at a time are summarized in the top panels of Figure 16. There is a strong positive correlation between genomic fragment divergence and average performance. Classification accuracy was consistently above 85% for fragment divergences when $D_3 > 2$. Results for Bayesian posterior distribution sampling were not substantially different (Additional file 22).

Accuracy of binning simulated communities of 5-10 species was consistent with the results from 2-3 species communities. The accuracy of binning was strongly positively correlated with genomic fragment divergence with accuracies consistently above 85% for $D_3 > 2$. Note that accurate binning was possible when fragment length was either $L = 400$ nt or $L = 800$ nt (middle and bottom panels of Figure 16 respectively). For 5 and 10 species, a total of 1815 simulated communities were tested in the $L = 400$ nt case and a total of 425 simulated communities were tested in the $L = 800$ nt case.

Next, we evaluated the robustness of our binning method to changes in fragment length and to changes in fragment ratios using five distinct genome pairs from the

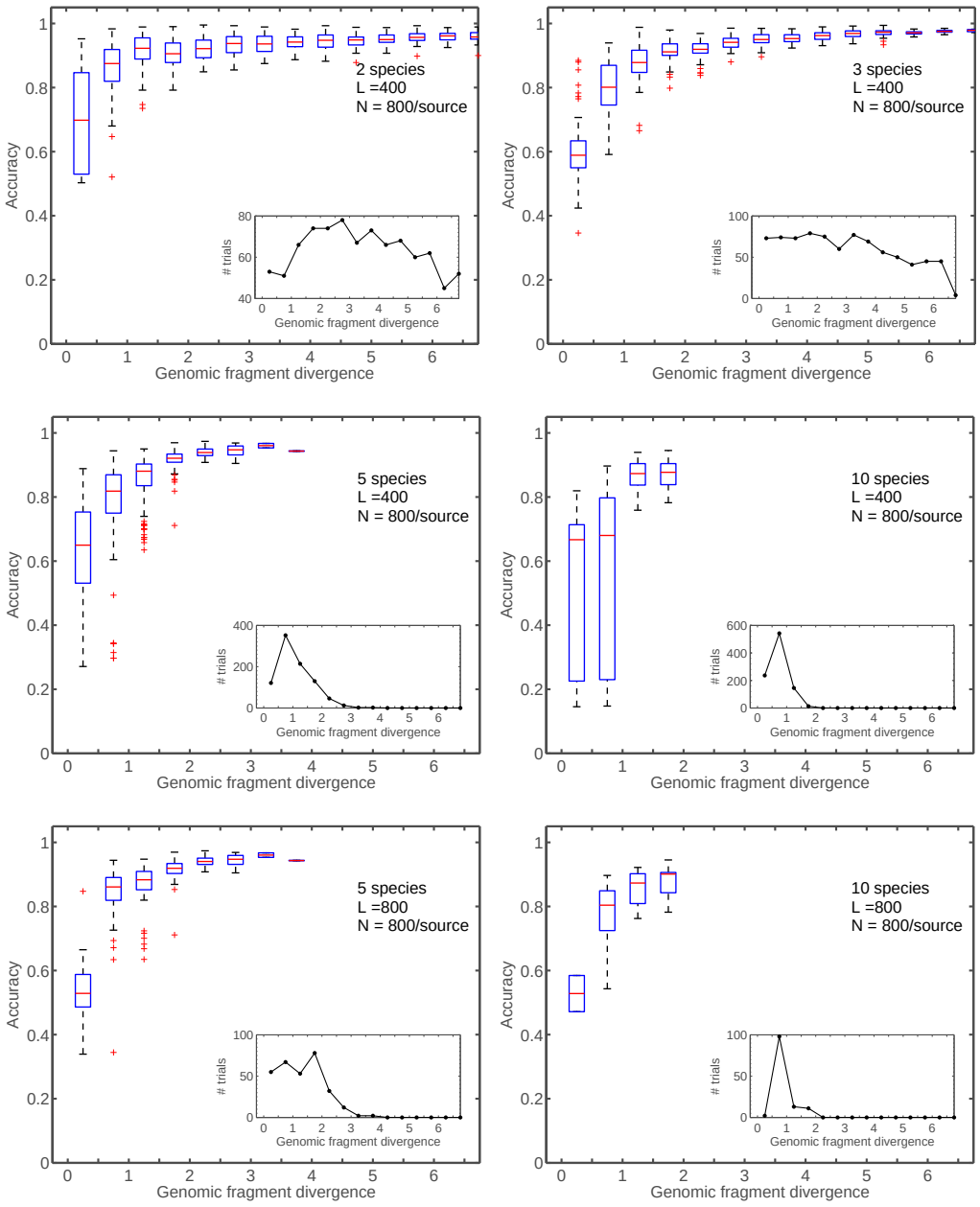



**Figure 16:** Sets of 2, 3, 5, 10 genomes were sampled randomly from a set of 1055 completed bacterial chromosomes, and experiments were conducted as described in Materials and Methods. Trials were conducted with 400- and 800-nt long fragments. Classification accuracy for the majority of genome pairs above overall divergence 1 is in the high performance range (accuracy $> 0.9$), while above divergence 3 accuracy is above 0.9 for over 95% of the trials. Results for Bayesian posterior distribution sampling were not significantly different (Additional file 22).

**Table 12:** Summary of species' characteristics, including all independent monomer and dimer frequencies, in the subset of trials on 5 pairs of genomes performed in Figures 18 and 19.

| Species composition | GC content | $p_A$ | $p_{AA}$ | $p_{AC}$ | $p_{AT}$ | $p_{CA}$ | $p_{CG}$ | $p_{GC}$ |
|---|---|---|---|---|---|---|---|---|
| *Arthrobacter aurescens TC1* | 63% | 0.186 | 0.041 | 0.044 | 0.048 | 0.054 | 0.127 | 0.114 |
| *Sinorhizobium meliloti 1021* | 62% | 0.189 | 0.040 | 0.057 | 0.037 | 0.068 | 0.097 | 0.098 |
| *Lactococcus lactis subsp. cremoris MG1363* | 36% | 0.322 | 0.128 | 0.046 | 0.092 | 0.063 | 0.025 | 0.037 |
| *Francisella tularensis subsp. holarctica FTA* | 32% | 0.337 | 0.118 | 0.047 | 0.109 | 0.059 | 0.015 | 0.038 |
| *Helicobacter pylori HPAG1* | 40% | 0.301 | 0.105 | 0.050 | 0.082 | 0.066 | 0.027 | 0.042 |
| *Streptococcus pneumoniae R6* | 39% | 0.303 | 0.126 | 0.040 | 0.079 | 0.058 | 0.037 | 0.060 |
| *Staphylococcus aureus RF122* | 35% | 0.324 | 0.122 | 0.042 | 0.097 | 0.060 | 0.017 | 0.037 |
| *Prochlorococcus marinus str. NATL2A* | 33% | 0.333 | 0.121 | 0.053 | 0.110 | 0.066 | 0.026 | 0.035 |
| *Staphylococcus aureus subsp. aureus COL* | 31% | 0.343 | 0.134 | 0.038 | 0.110 | 0.055 | 0.008 | 0.027 |
| *Methanocaldococcus jannaschii DSM 2661* | 33% | 0.335 | 0.122 | 0.053 | 0.112 | 0.065 | 0.026 | 0.033 |

preceding experiment (see Table 12). The pairs were selected based on their relatively low genomic fragment divergence, $D_3 \approx 1$, given a fragment length of $L = 400$ nt. Binning results on these 2-species tests were evaluated using sequence fragments whose lengths ranged from 40 to 1000 nt. The results are shown in Figure 18. Performance stabilizes close to its optimal value at fragment length 400. Again, results for Bayesian posterior distribution sampling were not substantially different than the maximum likelihood approach (Table 15).

For the same five pairs as in Figure 18, we performed a test of fragment ratio-dependent contributions to accuracy (Figure 19). The binner successfully classifies mixtures with species' fractional content of 20% and above. Although robust to moderate variation in fragment ratios, these results indicate that binning relatively rare species may require modifications to the present likelihood formalism.

We also tested our method using subsets of the JGI FAMeS [1, 114] simulated low-complexity dataset (simLC). We took 5 genomic sources at a time, using 500 fragments, each of length $L = 400$ nt. The accuracy results for binning these simulated low complexity communities are summarized in Table 13. The binning method has approximately 80% accuracy for a five-species community despite the genomic divergence, $D_3$, being approximately 1.5 (an indicator of a community with similar $k$-mer distributions).

**Table 13:** Summary of algorithm performance on JGI FAMeS data. Random subsets of 5 sources each were selected from the FAMeS simLC dataset, with a genomic fragment divergence, $D_3$, as shown. Fragments were truncated to the indicated length where appropriate. Reads from the dataset were used raw with no trimming.

| FAMeS identifiers | min $D_3$ | Fragment count | Fragment length | Accuracy |
|---|---|---|---|---|
| APOW1005, PPD1199, AIBF1022, AHZI1134, AHXO1014 | 2.3451 | 500 | 400 | 0.87 |
| BCSB1222, ABFI1048, AHYP1295, AKNK1296, AAZH3626 | 1.9598 | 500 | 400 | 0.69 |
| AHYT1136, AHYI1010, PIT10099, AINZ1029, AHZF1044 | 1.9314 | 500 | 400 | 0.85 |
| PPD1199, AUNI1013, ABSU1031, AABS2846, AHXO1014 | 1.8881 | 500 | 400 | 0.89 |
| AOTU1003, BCSB1222, AIOH1083, AIFS1040, AHXX1063 | 1.8032 | 500 | 400 | 0.86 |
| BCSB1222, VNY1182, AHXF1121, AKNK1296, AHZI1134 | 1.3563 | 500 | 400 | 0.81 |
| KPY1561, AOTY1222, BAHF1005, POG1025, AAOP1172 | 1.2429 | 500 | 400 | 0.79 |
| BCSB1222, AADD1003, AUNI1013, KPR1102, AHXO1014 | 1.1571 | 500 | 400 | 0.87 |
| AICI1287, AAOO1711, AKNK1296, AHXX1063, KPR1102 | 1.0279 | 500 | 400 | 0.72 |
| AHYT1136, AAWX1070, WBJ1361, AIAI1092, AXBY1147 | 0.9987 | 500 | 400 | 0.65 |
| AICI1287, AHYT1136, AAWX1070, AADE1259, AINZ1029 | 0.9856 | 500 | 400 | 0.72 |
| AUSC1572, AHYF1232, AAON1449, AIAX1019, ACBK1133 | 0.8884 | 500 | 400 | 0.78 |
| Average (12 trials, 5 sources, $L = 400$) | 1.46 | 500 | 400 | 0.79 |

We also compared our method to CompostBin [36], a semi-supervised algorithm that utilizes a PCA method to bin fragments based on their $k$-mer distributions (Table 5.3). We performed comparisons on pairs of genomes with fragment divergence $D_3 \approx 1$ using the same dataset analyzed in Figures 18, 19 and Table 12. The results indicated that our method performs on par with or better than CompostBin, even though CompostBin required a fraction of input fragments to be labeled to initialize its clustering algorithm. Run time and memory performance was comparable between the two methods.

The algorithm is implemented in portable Perl and C code that can be compiled and run on any platform supporting a Perl interpreter. Both memory use and run time scale linearly with the number of fragments and species, and sub-linearly with fragment length. Memory complexity scales quadratically with the number of dimensions in the search space, or exponentially with $k$ (as shown in Table 11). We selected

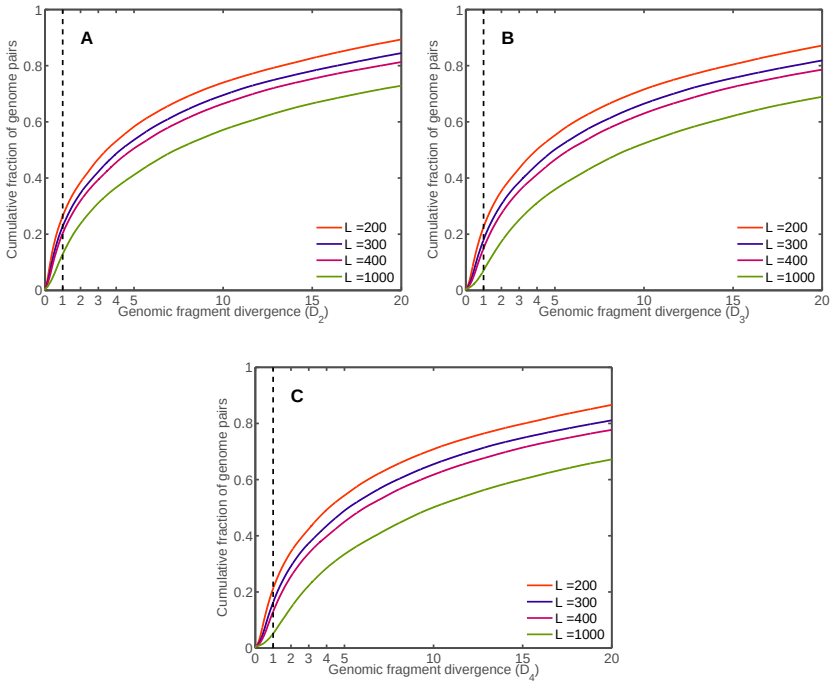



**Figure 17:** Cumulative distributions of pairwise divergences ($D_n$) between all completed bacterial genomes retrieved from GenBank. Fragment lengths of 400 to 1000 were used to compute $D_n$. Divergences based on $k$-mer order 2, 3, and 4 are represented in panels A, B, and C, respectively. The vertical cut-off line at $D = 1$ indicates an empirical boundary above which the binning algorithm works with high accuracy. For fragment length 400, over 80% of all randomly selected pairs are observed to have divergences above this line.

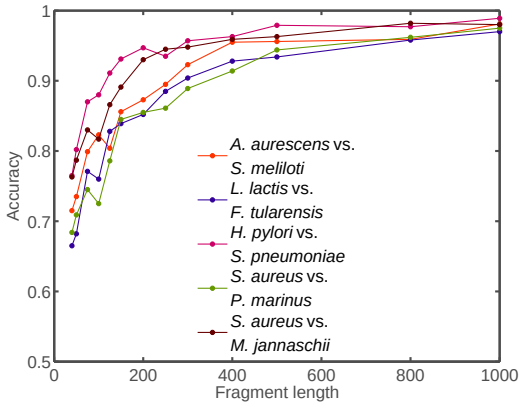



**Figure 18:** Fragment length-dependent performance on 2-species datasets. Same trials as in Figure 16 were performed on a subset of pairs of genomes while varying simulated fragment size from 40 to 1000. The species' characteristics are given in Table 12.

**Table 14:** Performance comparison of LikelyBin and CompostBin on pairs of genomes analyzed in Figures 18, 19, Table 12. *Frag L*, Fragment length; *Frag N*, Number of fragments per source; *CB seeds*, labeled fragments supplied to CompostBin for training. LikelyBin consistently performed equally to or above CompostBin performance despite being completely unsupervised, while CompostBin required a fraction of input fragments to be labeled to seed its clustering alorithm. We supplied training fragments to CompostBin without regard to their origin (protein or RNA-coding). In a likely practical scenario, only 16S RNA-coding fragments would be labeled, but would have different $k$-mer distributions from protein-coding regions, possibly confounding classification. $(*)$ Convergence toward a good clustering was not observed in CompostBin for these datasets; accuracy can be less than 50% due to labeled input.

| Org 1 | Org 2 | Frag L | Frag N | $D_3$ | LikelyBin accuracy | CB seeds | CompostBin accuracy |
|---|---|---|---|---|---|---|---|
| *S. meliloti* | *A. aurescens* | 400 | 500 | 1.02 | 0.94 | 10 | 0.93 |
| | | | | | | 25 | 0.93 |
| *L. lactis* | *F. tularensis* | 400 | 500 | 1.15 | 0.92 | 10 | 0.76 |
| | | | | | | 25 | 0.12$*$ |
| *S. pneumoniae* | *H. pylori* | 400 | 500 | 0.97 | 0.96 | 10 | 0.12$*$ |
| | | | | | | 25 | 0.96 |
| *P. marinus* | *S. aureus* | 400 | 500 | 0.99 | 0.93 | 10 | 0.73 |
| | | | | | | 25 | 0.83 |
| *M. jannaschii* | *S. aureus* | 400 | 500 | 0.92 | 0.94 | 10 | 0.17$*$ |
| | | | | | | 25 | 0.91 |

$k = 3$ as the default $k$-mer length, with user-defined options for 2, 4, or 5 available. We have not yet formalized convergence time performance as a function of $k$. In practice, a 3-species dataset of 1000 fragments per species, with $k$-mer order set to 3, takes approximately 2 minutes to run on an Intel Core 2 Duo-class processor.

## *5.4 Conclusions*

We developed an unsupervised, maximum likelihood approach to the binning problem - called LikelyBin. LikelyBin uses a MCMC framework to estimate the set of master distributions and relative frequencies most likely to give rise to an observed collection of short reads. The likelihood approach is based on $k$-mer distributions, for which we

developed an index of separability of any pair of genomes, which we termed the genomic fragment divergence measure, $D_n$. We found that the vast majority of genomes have sufficient divergence to be distinguished using the present method (Figure 17).

Using a high-performance implementation, LikelyBin can be used to cluster sequences with high accuracy (in some cases, $> 95\%$) even when the mononucleotide content of the original genomes is essentially identical (Figure 16). The method does as well or better than a comparable semi-supervised method (CompostBin [36]) that also uses $k$-mer distributions as the statistical basis for binning (Table 5.3). Performance of LikelyBin is consistently good for synthesized low-complexity datasets (2-10 species) with fragments of length as low as 400 nt, which corresponds to the characteristic single-read length of a 454 pyrosequencing FLX machine. Microread sequencing technologies such as Solexa and SOLiD are currently out of reach of any non-alignment-based binning method when applied to single reads, which range from 30 to 50 base pairs with these technologies.

The unsupervised nature of our approach makes it potentially useful for classifying mixtures of novel sequences for which supervised learning-based methods may have difficulties. A future direction for our work is to combine our statistical formalism with alignment and supervised composition-based models. For example, we could develop a feature selection framework that would transform the input fragments' features such as $k$-mer statistics, coding frame information, and variable-length motifs into a lower-dimensional space. We could then feed these features to an unsupervised MCMC-based classifier in tandem with an alignment-based classifier that can partially label fragments based on known taxonomic information, then compare and combine their results.

A number of challenges remain to broaden the scope and applicability of the current method. At present, our method is scalable for $k$-mer length from $k = 2$ to $k = 5$. We intend to expand the method's ability to capture longer motif

frequencies by using dimension transformation or feature selection in a future work. Intra-genomic heterogeneity of oligonucleotide distributions is another topic that is yet to be addressed. A confidence measure that serves as a performance self-check is already available as part of our method but we have not incorporated it into the program's output yet.

Further, applying the current method in an environmental context requires an estimation of the number of bins. The problem of identifying the necessary number of distinct models, or groups thereof, to represent all components of a given genome, is related to the problem of identifying the number of distinct genomes in the mixture. A combination of jump diffusion and grouped models is our currently planned solution. In this respect, the use of phylogenetic markers to estimate the number of bins will provide important prior information.

In summary, the unsupervised method we proposed is based on a maximum likelihood formalism and can bin short fragments ($L = 400$ nt) of low complexity communities (2-10 species) with high accuracy (in some cases, $> 95\%$) given sufficient genomic divergence. The maximum likelihood formalism and its MCMC implementation make the current approach amenable to extension and incorporation into other packages.

The MCMC binner application is provided as an open-source downloadable package, LikelyBin [5], that can be installed on any platform that supports Perl and C and is fully automated to facilitiate use in genome processing pipelines. The latest version of the source code is available on our website [5].

### 5.4.1 Example application of likelihood model

Suppose we have two source genomes, $G_1$ and $G_2$, with two fragments from each: $G_1 \rightarrow \{\texttt{ATGTTA}, \texttt{TGTAAT}\}$, $G_2 \rightarrow \{\texttt{CCTGTC}, \texttt{AGGCCTC}\}$. We wish to evaluate the likelihood of observing these sequences according to a dimer model of 2 sources, $M =$

$\{S_1, S_2\}$, which we have generated. Assume the model's source frequency vector is $F = [0.6, 0.4]$, its monomer frequencies are $\{S_1 : \{p_A = 0.3, p_T = 0.3, p_G = 0.2, p_C = 0.2\}, S_2 : \{p_A = 0.2, p_T = 0.2, p_G = 0.3, p_C = 0.3\}\}$ and its dimer frequencies are $\{S_1 : \{p_{AA} = 0.09, p_{AT} = 0.09, p_{AG} = 0.06, p_{AC} = 0.06, p_{TA} = 0.07, p_{TT} = 0.09, p_{TG} = 0.06, p_{TC} = 0.08 p_{GA} = 0.08, p_{GT} = 0.06, p_{GG} = 0.04, p_{GC} = 0.02 p_{CA} = 0.06, p_{CT} = 0.06, p_{CG} = 0.04, p_{CC} = 0.04\}$,

$S_2 : \{p_{AA} = 0.02, p_{AT} = 0.04, p_{AG} = 0.08, p_{AC} = 0.06, p_{TA} = 0.04, p_{TT} = 0.02, p_{TG} = 0.06, p_{TC} = 0.08 p_{GA} = 0.08, p_{GT} = 0.06, p_{GG} = 0.07, p_{GC} = 0.09 p_{CA} = 0.06, p_{CT} = 0.08, p_{CG} = 0.09, p_{CC} = 0.07\}\}$

Then the likelihoods of observing the first fragment, `ATGTTA`, given master distributions $S_1$ and $S_2$, respectively, are

$$\begin{aligned} P(\texttt{ATGTTA}|S_1) &= p^{S_1}_{c_1c_2} \prod_{j=3}^{\ell} \frac{p^{S_1}_{(c_{j-1}c_j)}}{p^{S_1}_{(c_{j-1})}} = \frac{p^{S_1}_{\texttt{AT}} p^{S_1}_{\texttt{TG}} p^{S_1}_{\texttt{GT}} p^{S_1}_{\texttt{TT}} p^{S_1}_{\texttt{TA}}}{p^{S_1}_{\texttt{T}} p^{S_1}_{\texttt{G}} p^{S_1}_{\texttt{T}} p^{S_1}_{\texttt{T}}} \\ &= \frac{0.09 \cdot 0.06 \cdot 0.06 \cdot 0.09 \cdot 0.07}{0.3 \cdot 0.2 \cdot 0.3 \cdot 0.3} = 0.000378 \\ P(\texttt{ATGTTA}|S_2) &= p^{S_2}_{c_1c_2} \prod_{j=3}^{\ell} \frac{p^{S_2}_{(c_{j-1}c_j)}}{p^{S_2}_{(c_{j-1})}} = \frac{p^{S_2}_{\texttt{AT}} p^{S_2}_{\texttt{TG}} p^{S_2}_{\texttt{GT}} p^{S_2}_{\texttt{TT}} p^{S_2}_{\texttt{TA}}}{p^{S_2}_{\texttt{T}} p^{S_2}_{\texttt{G}} p^{S_2}_{\texttt{T}} p^{S_2}_{\texttt{T}}} \\ &= \frac{0.04 \cdot 0.06 \cdot 0.06 \cdot 0.02 \cdot 0.04}{0.2 \cdot 0.3 \cdot 0.2 \cdot 0.2} = 0.000048 \end{aligned}$$

where superscripts $S_1$ and $S_2$ denote the master distribution. Similarly,

$$P(\texttt{TGTAAT}|S_1) = 0.000378; P(\texttt{TGTAAT}|S_2) = 0.000048;$$

$$P(\texttt{CCTGTC}|S_1) = 0.000192; P(\texttt{CCTGTC}|S_2) = 0.000448;$$

$$P(\texttt{AGGCCTC}|S_1) = 0.0000056\bar{8}; P(\texttt{AGGCCTC}|S_2) = 0.0004704$$

The overall posterior likelihood of the model is then

$$\begin{aligned}
\mathcal{L} &= \prod_{i=1}^{N}\left(\sum_{m=1}^{M} f_m P_m(s_i)\right) = \\
&= (f_{S_1}P(\texttt{ATGTTA}|S_1) + f_{S_2}P(\texttt{ATGTTA}|S_2)) \cdot (f_{S_1}P(\texttt{TGTAAT}|S_1) + f_{S_2}P(\texttt{TGTAAT}|S_2)) \\
&\quad \cdot (f_{S_1}P(\texttt{CCTGTC}|S_1) + f_{S_2}P(\texttt{CCTGTC}|S_2)) \cdot (f_{S_1}P(\texttt{AGGCCTC}|S_1) + f_{S_2}P(\texttt{AGGCCTC}|S_2)) \\
&= (0.6 \cdot 0.000378 + 0.4 \cdot 0.000048) \cdot (0.6 \cdot 0.000378 + 0.4 \cdot 0.000048) \\
&\quad \cdot (0.6 \cdot 0.000192 + 0.4 \cdot 0.000448) \cdot (0.6 \cdot 0.0000056\bar{8} + 0.4 \cdot 0.0004704) \\
&= 3.4131\texttt{E} - 15
\end{aligned}$$

## 5.5 Acknowledgements

We would like to thank Jonathan Eisen for many inspiring discussions. We would also like to thank Amol Shetty, Michael Raghib-Moreno, Sourav Chatterji, Luca Giuggoli, and Simon Levin for their suggestions on a preliminary version of the present model, and thank three anonymous reviewers for their helpful suggestions. We are grateful to Sourav Chatterji, Jonathan Eisen, and Ichitaro Yamazaki for their help in the utilization of CompostBin.

## 5.6 Funding

The work presented in this chapter was supported by the Defense Advanced Research Projects Agency under grant HR0011-05-1-0057. Joshua S. Weitz, Ph.D., holds a Career Award at the Scientific Interface from the Burroughs Wellcome Fund.

**Table 15:** The method of sampling the posterior distribution of the MCMC chain by averaging random accepted models from the steady state was compared to the method of selecting the model with the overall maximum log likelihood. The resulting accuracy differences were negligible. Accuracy was also compared in 3-mer models vs. 4-mer models. While 4-mer models slightly outperformed 3-mer models on average, a significant run time increase was observed (not shown). NC_ identifiers refer to GenBank accession numbers for genomes listed in each trial.

| Org 1 | Org 2 | Frag L | Sampling type | Order 3 model | | | Order 4 model | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | $D_3$ | Accuracy | LL | $D_4$ | Accuracy | LL |
| *Arthrobacter aurescens TC1* vs. *Sinorhizobium meliloti 1021* | | | | | | | | | |
| NC_003047 | NC_008711 | 400 | Steady state sampled | 1.08 | 0.95 | -1054490.36 | | | |
| | | 400 | Maximum log likelihood | 1.02 | 0.94 | -1055584.16 | 1.09 | 0.94 | -1040007.41 |
| | | 1000 | Steady state sampled | 1.95 | 0.97 | -2648159.80 | | | |
| | | 1000 | Maximum log likelihood | 2.12 | 0.98 | -2645204.57 | 2.52 | 0.99 | -2637429.69 |
| *Lactococcus lactis subsp. cremoris MG1363* vs. *Francisella tularensis subsp. holarctica FTA* | | | | | | | | | |
| NC_009004 | NC_009749 | 400 | Steady state sampled | 1.08 | 0.90 | -1045063.72 | | | |
| | | 400 | Maximum log likelihood | 1.15 | 0.92 | -1047966.99 | 1.33 | 0.95 | -1040811.10 |
| | | 1000 | Steady state sampled | 2.02 | 0.96 | -2624742.76 | | | |
| | | 1000 | Maximum log likelihood | 2.19 | 0.96 | -2626080.18 | 2.22 | 0.97 | -2615376.71 |
| *Helicobacter pylori HPAG1* vs. *Streptococcus pneumoniae R6* | | | | | | | | | |
| NC_003098 | NC_008086 | 400 | Steady state sampled | 0.93 | 0.96 | -1059955.55 | | | |
| | | 400 | Maximum log likelihood | 0.97 | 0.96 | -1061298.85 | 1.18 | 0.93 | -1045561.25 |
| | | 1000 | Steady state sampled | 1.71 | 0.99 | -2656860.50 | | | |
| | | 1000 | Maximum log likelihood | 1.69 | 0.98 | -2658488.27 | 2.28 | 0.99 | -2634722.55 |
| *Staphylococcus aureus RF122* vs. *Prochlorococcus marinus str. NATL2A* | | | | | | | | | |
| NC_007335 | NC_007622 | 400 | Steady state sampled | 0.99 | 0.90 | -1049716.33 | | | |
| | | 400 | Maximum log likelihood | 0.99 | 0.93 | -1050316.80 | 1.00 | 0.95 | -1045188.54 |
| | | 1000 | Steady state sampled | 1.92 | 0.97 | -2636903.64 | | | |
| | | 1000 | Maximum log likelihood | 1.75 | 0.97 | -2636046.52 | 2.21 | 0.97 | -2624299.41 |
| *Staphylococcus aureus subsp. aureus COL* vs. *Methanocaldococcus jannaschii DSM 2661* | | | | | | | | | |
| NC_000909 | NC_002951 | 400 | Steady state sampled | 0.96 | 0.95 | -1037936.55 | | | |
| | | 400 | Maximum log likelihood | 0.92 | 0.94 | -1037505.67 | 1.05 | 0.89 | -1033285.36 |
| | | 1000 | Steady state sampled | 1.84 | 0.98 | -2598584.81 | | | |
| | | 1000 | Maximum log likelihood | 1.94 | 0.98 | -2601394.32 | 2.36 | 0.99 | -2581181.80 |

*Frag L, Fragment length; LL, Output model log likelihood*

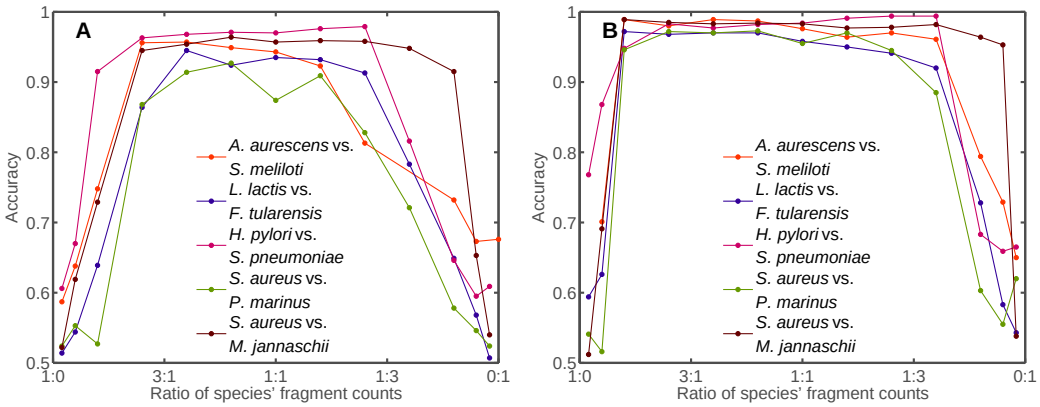



**Figure 19:** Fragment ratio-dependent performance on 2-species datasets. Same trials as in Figure 16 were performed on a subset of pairs of genomes while varying species' contributions to the dataset from 2% to 98%. Fragment sizes were fixed at 400 nt (A) and 1000 nt (B). The species' characteristics are given in Table 12.

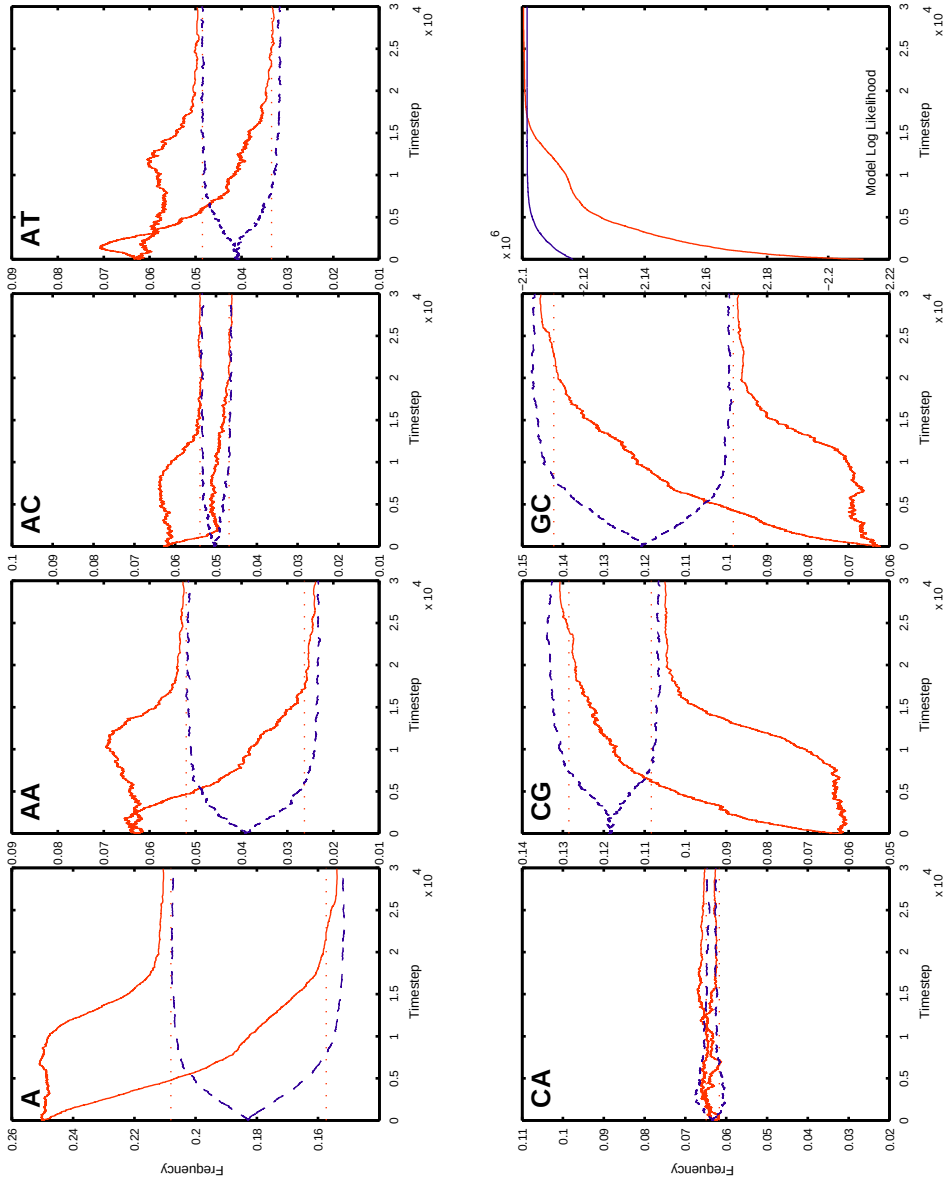



**Figure 20:** Convergence dynamics for good accuracy, *Mycoplasma capricolum subsp. capricolum ATCC 27343* vs. *Campylobacter jejuni subsp. jejuni 81-176* ($D_3 = 2.8$). A single MCMC simulation was completed for this pair of genomes as described in Methods. $k$-mer order 3 model was used with 30000 steps, and expected nucleotide frequencies in accepted models were plotted over time for all independent mono- and dinucleotides in the model. Two starting conditions were compared: uniform initial frequencies (solid line) and frequencies at dataset mean (dashed line). Dotted lines indicate true average frequencies in the constituent species' fragment datasets. Convergence was observed to be substantially the same, demonstrating robustness of the algorithm to initial starting conditions. Final model accuracy was $\approx 95\%$ in both cases.

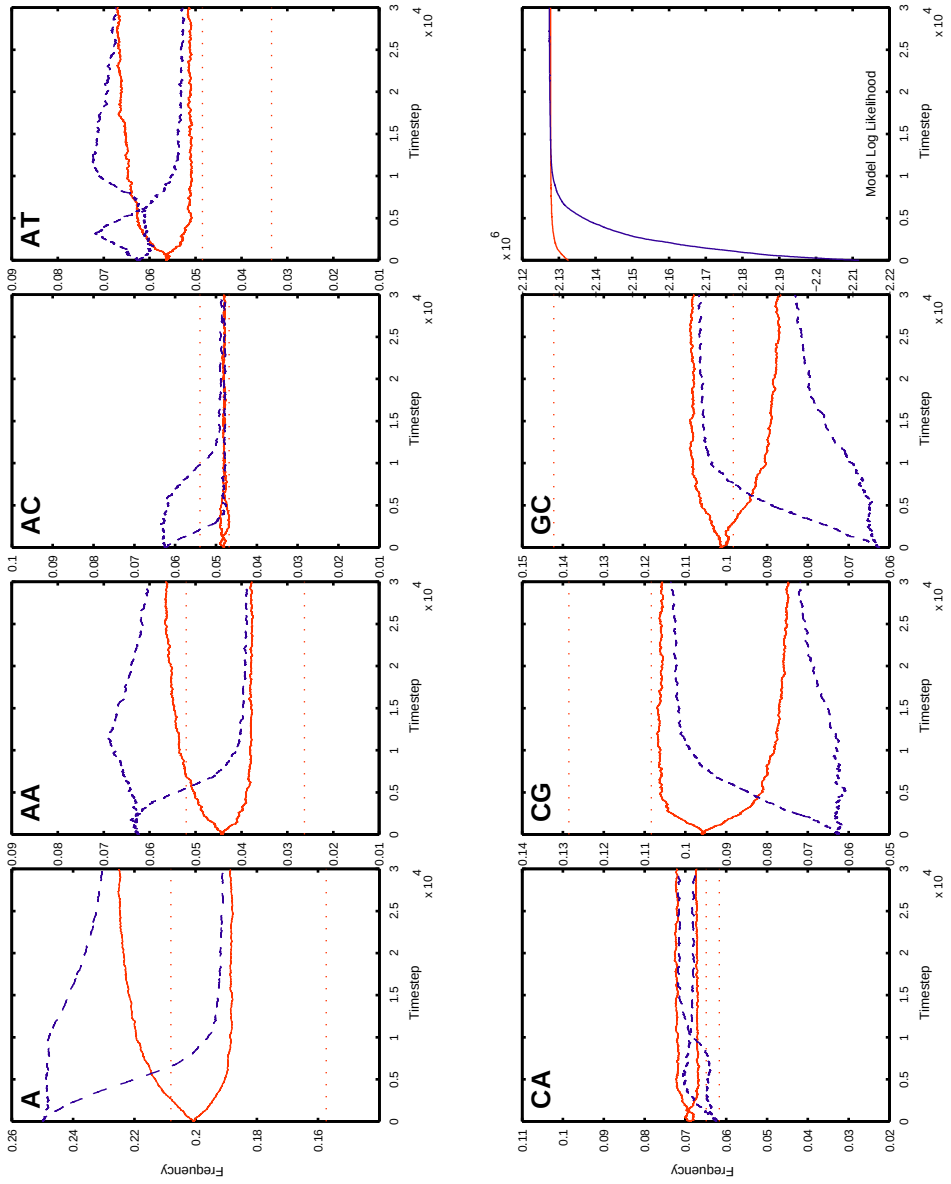



**Figure 21:** Convergence dynamics for poor accuracy, *Granulibacter bethesdensis CGDNIH1* vs. *Gluconobacter oxydans 621H* ($D_3 = 0.45$). Details are identical to Additional file 20, but final model accuracy was $\approx 60\%$ in both cases.

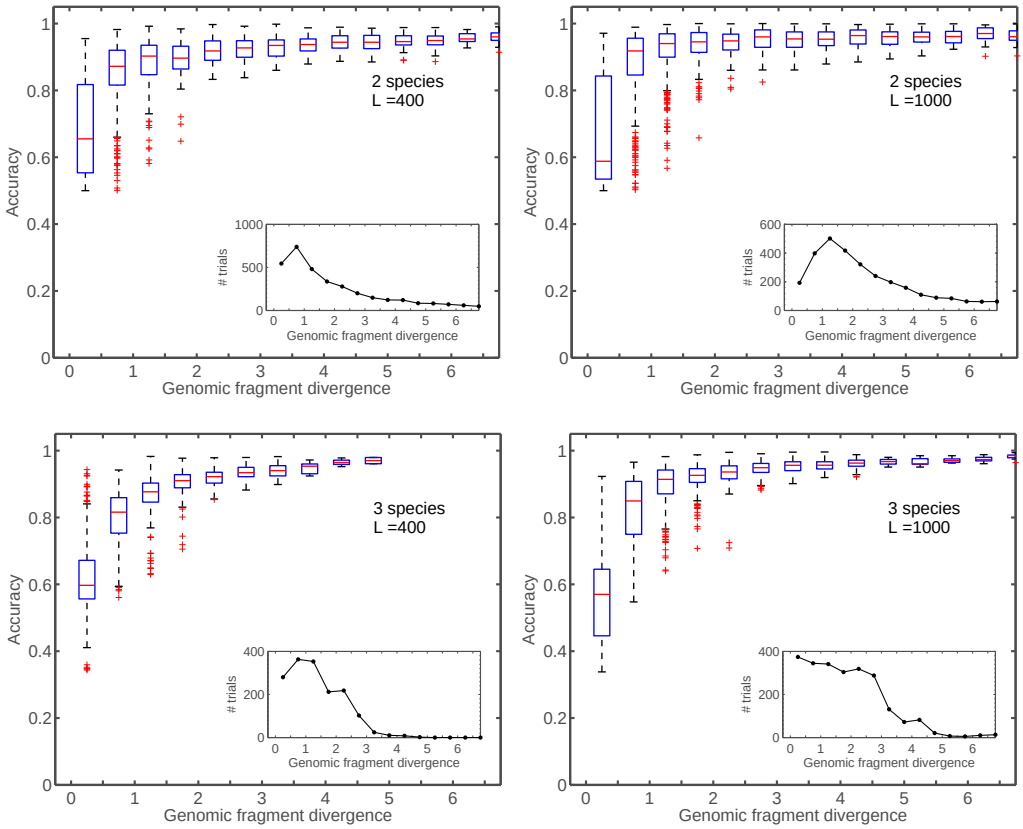



**Figure 22:** Pairs and triples of genomes were sampled randomly from a set of 1055 completed bacterial chromosomes, and experiments were conducted using Bayesian posterior distribution sampling on the stationary distribution of the MCMC simulation. The results were found to not be significantly different from those for maximum likelihood sampling (Figure 16).

# CHAPTER VI

# CORE- AND PAN-GENOMES

The dual concepts of pan and core genomes have recently been utilized to assess the distribution of gene families within genomes of closely related organisms, e.g., a bacterial species or genus. The estimation of both pan and core genome sizes has led to incongruous results such as claims that the pan genomes of a given microbial species range from infinite to finite, but strongly dependent on the number of genomes sampled. Here, we demonstrate mathematically that pan and core genome sizes cannot be estimated accurately except in highly contrived settings. Instead, we introduce an alternative metric, genomic fluidity, which represents an integrative measure of the relative dissimilarity of gene families within a group of closely related organisms. Fluidity can be accurately estimated from a small number of sequenced genomes and comparisons of fluidity between groups is robust to variation arising in gene alignment parameters used in tabulating gene families. We estimate genomic fluidity in 7 multiply-sequenced species containing 109 sequenced genomes using an automated bioinformatics pipeline. Using fluidity estimates, we are able to reliably rank order the cumulative effect of gene acquisition and loss within bacterial species. In so doing, we demonstrate the limits to what can be known about the gene diversity of an entire group of organisms when analyzing the properties of just a few.

The rest of this chapter is based on unpublished work currently submitted for peer review and publication:

A. O. Kislyuk, B. Haegeman, N. H. Bergman, and J. S. Weitz, "Genomic fluidity: an integrative view of gene diversity within microbial populations." Submitted (2010).

## 6.1 Introduction

The advent of technologies to rapidly sequence entire genomes provides a resource of sequenced genomes spanning the entire tree of life [85, 186, 109, 151]. Indeed, as the cost and time to sequence genomes have decreased, it has become possible to sequence multiple individuals from within a species. Re-sequencing efforts have led to the following discovery: the representation of gene families in isolates from the same bacterial species is highly variable [170, 80, 78, 76, 136]. This variability poses conceptual as well as applied problems. Conceptually, the variability suggests the need to further re-visit species definitions that rely upon comparisons of highly conserved components of the genome, such as 16S rRNA sequences [64, 89, 9, 53, 60]. In addition, horizontal gene transfer and other genome rearrangements such as gene deletions and duplications can radically change the phenotype of a bacterium, even within individuals of the same species [68]. For example, the introduction of toxin genes can render a bacterium pathogenic. Hence, from an applied perspective, there is an increasing need to quantify the gene diversity of a species or genus with pathogenic potential [80, 78, 147, 10, 39, 45]. The core and pan genome concepts have been proposed as a way to characterize the distribution of gene families within a group of organisms, e.g., within a species or genus [170, 80, 171, 32, 138, 147, 39]. The core genome is the set of genes found in every organism within a group (whether sequenced or not). The pan genome is the set of all genes found within organisms of a group (whether sequenced or not), including core genes and genes which appear in a fraction of genomes. Intuitively, the core genome preserves the notion that genomes of closely related organisms have something in common, while the pan genome is in accord with the finding that gene composition differs even among genomes of closely related organisms. In that sense, the core and pan genome concepts begin to address both conceptual problems (e.g., what is a bacterial species?) and applied problems (e.g., how likely is it that an individual of a given bacterial species is a pathogen?).

Multiple attempts have been made to estimate the size of pan and core genomes in hopes of quantifying how open or closed a particular set of genomes is to gene exchange [170, 78, 76, 156, 99]. However, estimating the actual list of genes in the pan and core genomes remains intractable.

Thus far, attempts to quantify the size of the core and pan genomes have been based on extrapolations from a limited number of sequenced strains (usually on the order of a dozen or few dozen genomes) to the entire group (generally unknown, but easily upwards of $10^{12}$ genomes). Results of such extrapolations have been widely divergent. In the most well-studied case, the pathogen *Streptococcus agalactiae*, estimates of the pan genome size vary from tens of thousands [156] to infinite [170]. Extreme variation in estimates of core and pan genome sizes makes it difficult to utilize these measures to quantify or compare the degree of acquisition and loss of gene families within a particular group or to make meaningful biological interpretations of the core and pan genome concepts. One might suspect that robust quantification of core and pan genomes sizes could be achieved with improved statistical estimation methods, combined with increased sequencing coverage. This is not the case. The problem of estimating pan and core genome sizes will not be resolved by gradual improvements in sequencing.

In this chapter we demonstrate that current methods to estimate pan and core genome sizes are statistically ill-posed. We do so by demonstrating that sample gene distributions drawn from artificially generated groups of genomes with radically different pan and core genomes sizes are statistically indistinguishable. In contrast, we present an alternative diversity metric, genomic fluidity, whose expected value is equivalent whether estimated from the sample or from the true gene distribution. We then apply a bioinformatics pipeline so as to estimate genomic fluidity within 7 multiply-sequenced bacterial species containing 109 sequenced genomes. We test the robustness of genomic fluidity to changes in the number of sequenced genomes

as well as to changes in alignment parameters. In so doing we demonstrate when it is possible to reliably rank order species in terms of genomic fluidity and discuss the implications of our work for inferring information about gene distributions based on subsamples.

## *6.2 Results*

### 6.2.1 Pan and core genome sizes cannot be reliably estimated

We claim that current methods to estimate pan and core genome sizes are statistically ill-posed [170, 32, 156, 99]. To demonstrate this in a case where the pan and core genome sizes are known, we artificially generated gene distributions for three "species" such that their pan genome sizes were $10^5$ (A), $10^7$ (B), and $10^5$ (C) and their core genome sizes were $10^3$ (A), 10 (B), and $10^3$ (C) (See Figure 23A and 24A). Note that Species A and C had distinct gene frequency distributions despite having the same pan and core genome sizes. Next, we computationally generated ensembles of genomes for each species, each of which had 2000 genes. Each gene in a genome was chosen at random from a frequency distribution specific to a given species, i.e., some genes occurred in all, or nearly all, genomes and some genes occurred very rarely. Importantly, a gene that only appears in 0.00001% of genomes (1 in $10^7$ occurrence) contributes as much to the pan genome as does a core gene (Figure 23A), however, the rare gene will almost certainly not be detected in a sample set of tens or hundreds of sequenced genomes (Figure 23B). Furthermore, none of the genes that are detected in the sample set of genomes provide any indication that this rare gene exists while performing standard rarefaction analysis (Figure 23C). In essence, the problem of estimating the pan genome is equivalent to estimating the level of rare genes, which, because they are rare, are recalcitrant to quantification. Similar difficulties are faced when trying to quantify the size of the core genome. For example, a gene that appears in 99.999% of genomes is technically not a core gene (Figure 23A). Yet the rare genome

without this core gene will not be detected in a sample set of genomes (Figure 23B), nor will the sample provide any indication that an apparent core gene is absent from some small number of organisms in the group (see Figure 23D). Intuitively, both pan and core genome size estimates depend on accurate estimation of the frequency of rare events that any small sub-sample of sequenced genomes will not enable. This is not to say that the pan and core genome concepts should be discarded, however in practice, estimates of pan and core genome sizes may have no correspondence to true values. Instead, some alternative metric is needed that (i) is robust to small sample size (can be reliably estimated from few genomes); (ii) quantifies the relative degree of gene acquisition and loss within a group of genomes; and (iii) validates prior expectations that gene diversity increases within groups of increasingly unrelated organisms.

### 6.2.2 Genomic fluidity is a robust and reliable estimator of gene diversity

We propose the use of genomic fluidity, $\varphi$, as a robust diversity metric which can be applied to small numbers of sequenced genomes whether at the species level or amongst groups of increasingly unrelated organisms. Genomic fluidity is defined as the ratio of unique gene families to the sum of gene families in pairs of genomes averaged over randomly chosen genome pairs from within a group:

$$\varphi = \left\langle \frac{U_k + U_l}{M_k + M_l} \right\rangle_{\forall k \neq l}, \tag{13}$$

where $U_k$ and $U_l$ are the number of gene families found only in genomes $k$ and $l$ respectively and $M_k$ and $M_l$ are the total number of gene families found in $k$ and $l$ respectively. In other words, genomic fluidity is an estimate of gene-ic dissimilarity, akin to similarity measures used in the study of ecological communities [69]. Genomic fluidity also provides information on novelty in sequencing projects. To see how, note that the best estimate for the probability that a random gene from a newly sequenced genome is not found in a randomly selected prior sequenced genome is simply $\varphi$. Importantly, genomic fluidity is robust to small sample size: it can be

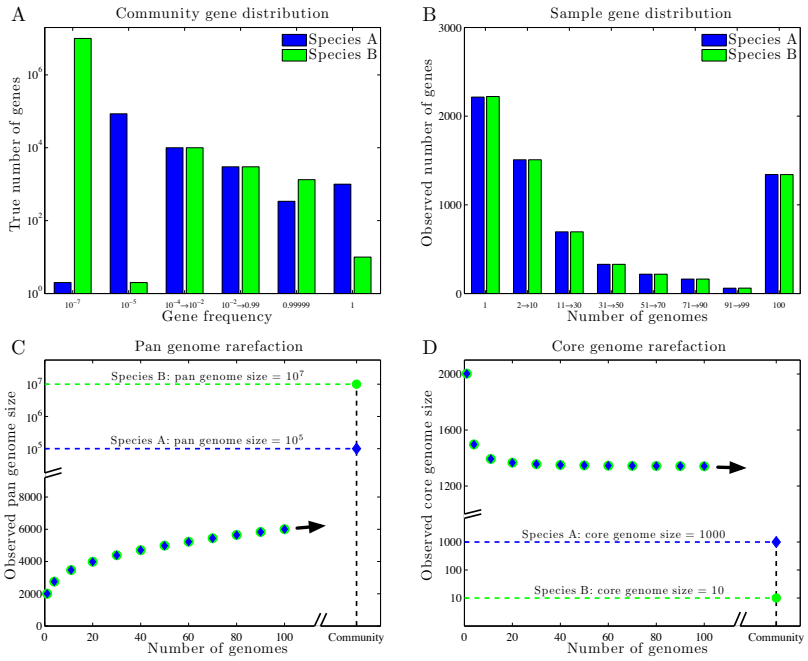



**Figure 23:** Radically different pan and core genome sizes cannot be estimated from sampled genomes. (A) Two species with vastly different true gene distributions: (i) Species A (blue) w/pan genome of $10^5$ genes and core genome of $10^3$ genes; (ii) Species B (green) w/pan genome of $10^7$ genes and core genome of 10 genes. Each genome has 2000 genes randomly chosen from the true gene distribution according to its frequency. (B) The number of genes (y-axis) observed as a function of the number of sampled genomes (x-axis). Note that despite differences in the true distribution, the observed gene distributions are statistically indistinguishable given 100 sampled genomes. For example, there were approximately 2200 genes found in just 1 of 100 genomes for both Species A and Species B. (C) Observed pan genome size as a function of the number of sampled genomes. There is no possibility to extrapolate the true pan genome size from the observed pan genome curves. (D) Observed core genome size as a function of the number of sampled genomes. There is no possibility to extrapolate the true core genome size from the observed core genome curves.

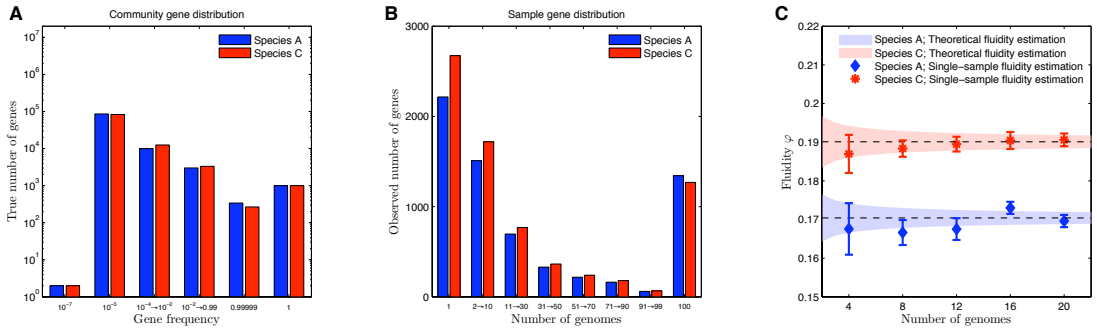



**Figure 24:** True differences in genomic fluidity $\varphi$ can be detected from a small number of sampled genomes. (A) Two species with subtle differences in true gene distributions: (i) Species A (blue) as in Figure 1, w/pan genome of $10^5$ genes and core genome of $10^3$ genes; (ii) Species C (red) w/pan genome of $10^5$ genes and core genome of $10^3$ genes. Each genome has 2000 genes randomly chosen from the true gene distribution according to its frequency. (B) The number of genes (y-axis) observed as a function of the number of sampled genomes (x-axis). The observed gene distributions are statistically distinguishable. (C) Fluidity as a function of the number of sampled genomes is an unbiased estimator of the true value (dashed lines within red and blue shaded regions). The shaded regions denote the theoretical prediction for mean and standard deviations as inferred from the jacknife estimate.

reliably estimated from a few sampled genomes. For example, in Figure 24 we show how the genomic fluidities for synthetically generated gene distributions are equivalent whether estimated from the true distribution or from a few dozen sampled genomes. In addition, subtle differences in the genomic fluidity between two species can be detected from a small number of sampled genomes. The estimated variance of fluidity was calculated using the jackknife estimate [54], which is based on leave-one-out statistics (see Materials and Methods for more details). In contrast, rarefaction curves used to estimate pan and core genome sizes are statistically indistinguishable for synthetically generated gene distributions, even when the underlying pan and core genome sizes are radically different (see Figure 23c,d).

### 6.2.3 Fluidity and its variance can be estimated from a group of sequenced genomes

We developed a bioinformatics pipeline to estimate genomic fluidity at the species level among sequenced genomes (see Figure 26 and Methods Summary), but later, as in Figure 29, we apply it to more diverse groups. Using this pipeline we calculated genomic fluidity for 7 species including 109 sequenced genomes from: *Bacillus anthracis*, *Escherichia coli*, *Neisseria meningitidis*, *Staphylococcus aureus*, *Streptococcus agalactiae*, *Streptococcus pneumoniae*, and *Streptococcus pyogenes* (see Table 22 for a list of all genomes analyzed in this study). We find that estimates of fluidity converge rapidly even when evaluated on a small number of sequenced genomes, as has been the case for all published studies of gene diversity within a species or genus. These results are consistent with the rapid convergence of fluidity when estimated from synthetically generated genomes (see Figure 23). When applied to genomes from multiply resequenced bacterial species we find the mean value of fluidity is consistent when evaluated on a small subsample or on the entire sample (Figure 25). We find convergence of fluidity estimates to approximately 10% relative standard deviation after a dozen or so genomes (see Figure 25). The variation in fluidity estimates found in small subsamples of sequenced genomes suggests caution should be applied in attempting to establish when we can reliably say that the fluidity of a particular species is greater than that of another. Importantly, the use of the jack-knife estimate of variance permits us to evaluate how both the mean and the variance of fluidity converge as more genomes are added and provides a metric to indicate when sufficient sequencing has been accomplished for use in comparing relative values of fluidity between species or between groups.

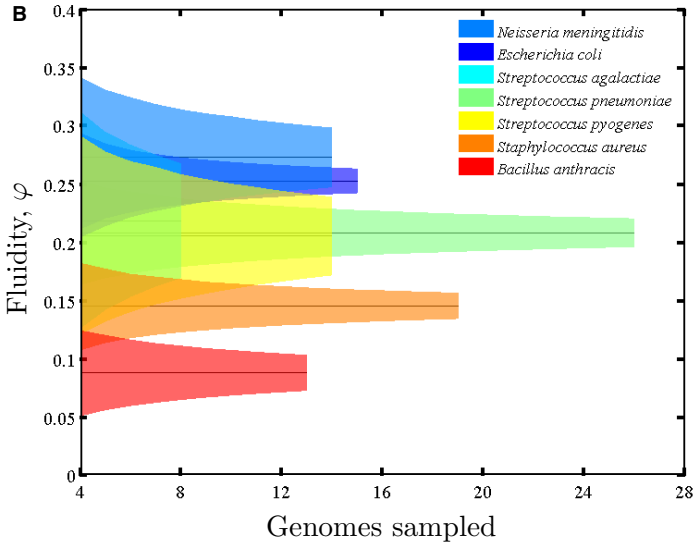



**Figure 25:** Estimates of mean fluidity converge with increases in the number of sampled genomes. Fluidity was calculated as described in the text given alignment parameters $i = 0.74$ and $c = 0.74$. The variance of fluidity is estimated as a total variance, containing both the variance due to subsampling within the sample of genomes, and the variance due to the limited number of sampled genomes. For dependence of fluidity on genomes sampled for the two other sets of alignment parameters in Figure 28, see Figure 27.

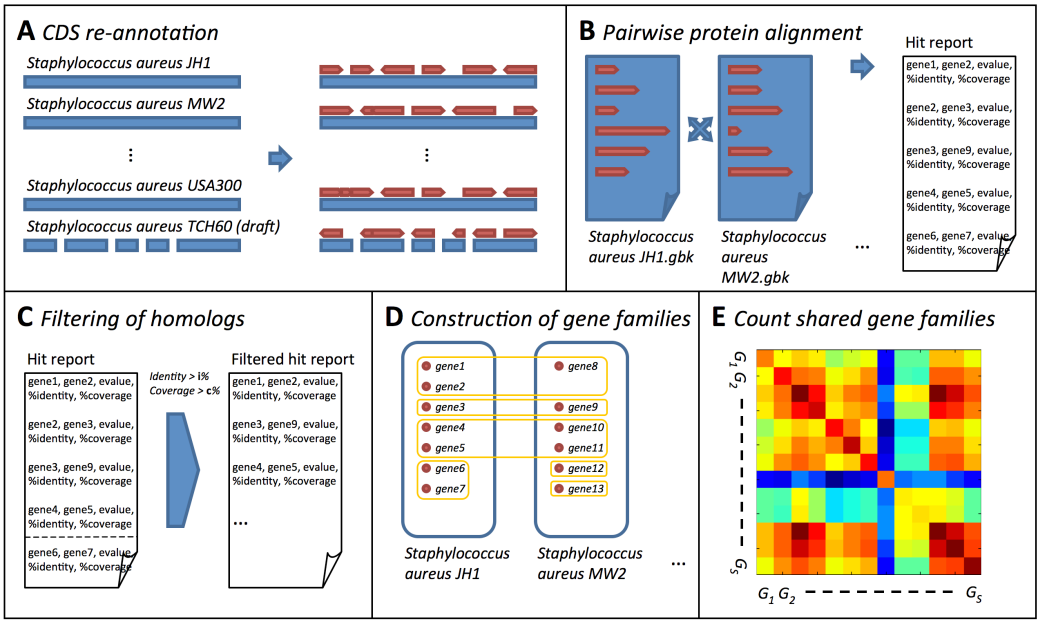



**Figure 26:** Computation of shared genes among genomes (see Materials and Methods for complete details of the pipeline and Table 22 for a complete list of genomes analyzed). (A) Genomes are annotated automatically to minimize curation bias (see Chapter 4); (B) For a given pair of genomes, all genes are compared using an all vs. all protein alignment; (C) Shared genes are identified based on whether alignment identity and coverage exceed i and c respectively; (D) Gene families are calculated based on a maximal clustering rule; (E) The number of shared genes is found for each pair of genomes, $G_i$ and $G_j$, from which the number of unique genes can be calculated.

### 6.2.4 Rank-ordering of genomic fluidity is robust to variation in alignment parameters

The estimate of genomic fluidity varied with alignment parameters as expected. When either minimum alignment identity or coverage is increased, more gene families are formed and fluidity increases (see Figure 27). Nonetheless, the relative values of fluidity between species remained nearly invariant even as the magnitude of fluidity changed. We applied the fluidity pipeline detailed in Figure 26 and restricted our analysis to gene family assembly values of alignment identity ($i$) and coverage ($c$) from 0.5 to 0.8 in increments of 0.02 (see Materials and Methods). In 225 trials, we found 4 distinct orderings of genomic fluidity, three of which accounted for 224/225 orderings (see Figure 28a for the three dominant rank orderings). The robust rank-ordering suggests that it is possible to make comparative statements classifying one group as being more or less "open" to net gene acquisition. Specifically, we used the mean and variances estimates of fluidity to determine whether the $\varphi$ of one species is significantly greater or less than another (see Materials and Methods). We find there is a statistically significant and unambiguous rank order of genomic fluidity for 11/21 comparisons of relative rank order among the 7 species examined in all 3 alignment parameter conditions corresponding to the dominant rank orderings ($p < 0.05$; see Tables 16-21). In all conditions tested, *B. anthracis* had the lowest value of $\varphi$ and either *N. meningitidis* or *E. coli* had the highest value of $\varphi$. Further, *Strep. agalactiae* always had an intermediate value of fluidity. However, *Strep. agalactiae* had a particularly high variance and we were unable to rank-order it relative to any other genome with the exception of *B. anthracis*. These results are generally consistent with previous suggestions that *B. anthracis* has a closed genome, that *N. meningitidis* may have an open genome due to its natural competence, and that *Strep. agalactiae* has an open genome [170]. However, now we can describe a group of organisms as being *relatively* open or closed, instead of being strictly open or strictly closed. In addition,

we can utilize variance estimates to suggest when greater sequencing is needed. The comparison of the rank order of $\varphi$ between species is consistent with recent calls [150] to utilize the rank, not the absolute magnitude, when comparing the relative diversity of complex ecological communities. This issue is particularly important in the case of gene diversity studies when identification of gene families is strongly depend on thresholds utilized in bioinformatics pipelines.

### 6.2.5 Genomic fluidity is a natural metric spanning phylogenetic scales from species to kingdom

Thus far we have estimated genomic fluidity within a bacterial species, though the metric can be applied, in principle, to any group of genomes. Therefore, we estimated values of $\varphi$ at the species level and at higher taxonomic groupings and found that $\varphi$ varies from close to 0 (at the species level) to nearly 1 (at the phylum level) (see Figure 29). A phylogenetic tree of 29 bacterial species was assembled using AMPHORA [186]. Species in this calculation were chosen to include those whose strain-level variation we had analyzed, as well as a hand-curated selection of genomes from different parts of the tree. Each leaf with a corresponding strain group therefore represents a collapsed subtree that clusters closely around the representative strain with respect to the overall tree. The phylogenetic tree selected here is not meant to represent the entire diversity of life, but rather to illustrate how fluidity changes when closely and distantly related organisms are grouped together. Note the transition from relatively "solid" genomes at the level of isolates from within a bacterial species to a nearly totally "fluid" bacterial kingdom. Further, estimates of genomic fluidity are consistent with expectations that $\varphi$ should increase as we move up the phylogenetic tree from species to genus to family, etc. Hence, we find that genomic fluidity is a natural metric for describing gene level similarity between groups of closely and distantly related organisms. These results suggest the suitability of genomic fluidity at coarse-grained scales, e.g. bacterial kingdom [99] and microbial

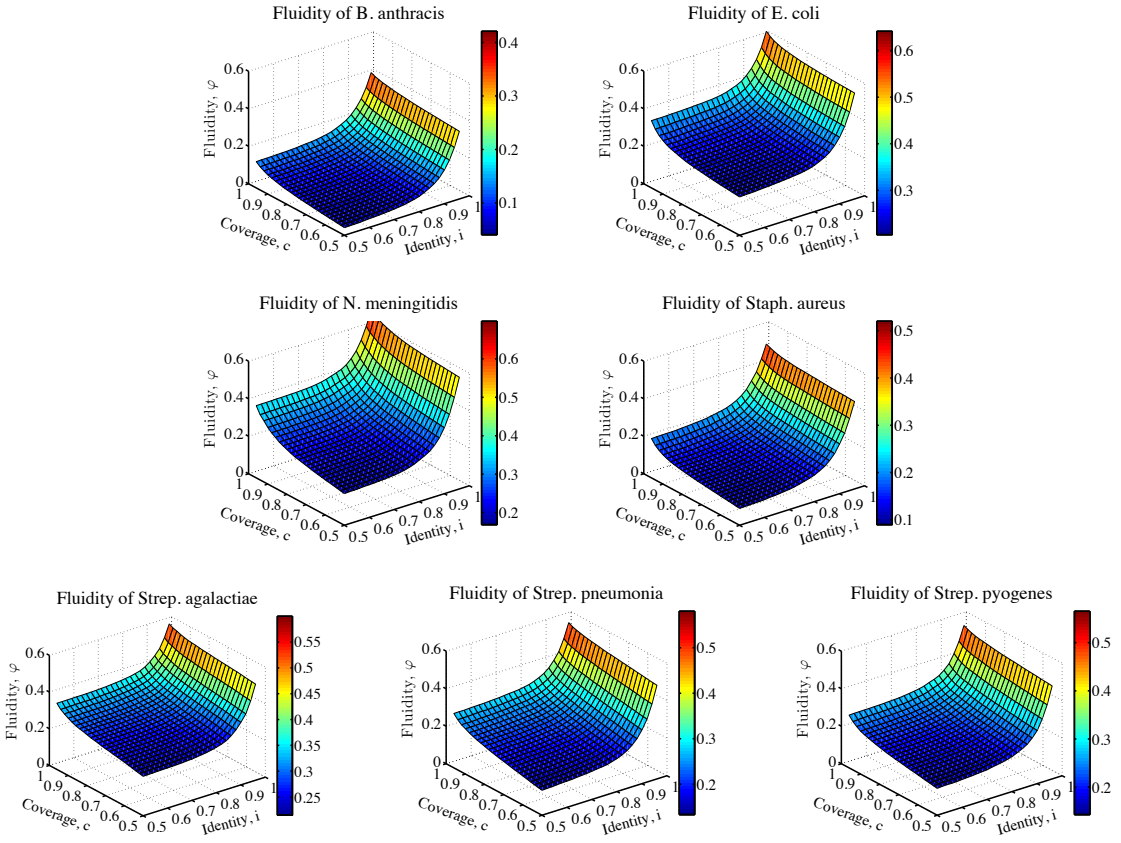



**Figure 27:** Estimates of fluidity depend on gene alignment parameters that determine the grouping of genes into gene families. We calculated fluidity for each of the 7 species examined in the main text with varying alignment parameter levels of identity ($i$) and coverage ($c$). We chose levels such that $0.5 \leq i \leq 0.96$ and $0.5 \leq c \leq 0.96$. Computations of $\varphi$ are based on estimating the fraction of unique genes between any two random genomes. Unsurprisingly, fluidity increases with increases in either $i$ or $c$. This increase arises because greater stringency of alignment causes the bioinformatics pipeline algorithm to infer that there are more unique genes. For each of the 7 species examined, genomic fluidity is more sensitive to changes in identity than to changes in coverage. This result suggests the importance of considering the robustness of results derived from bioinformatics pipelines to changes in parameters. Despite the change in fluidity values, the actual value of fluidity is relatively insensitive to changes in alignment parameters so long as neither parameter is greater than approximately 0.8. Hence, in the main text we restrict sensitivity analyses to $0.5 \leq i < 0.8$ and $0.5 \leq c < 0.8$.

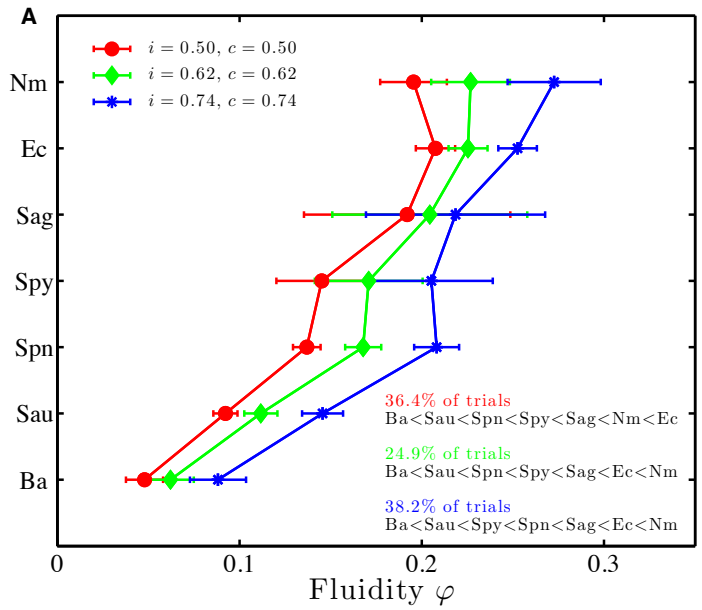



**Figure 28:** Estimates of mean and standard deviation of fluidity for *B. anthracis* (Ba), *E. coli* (Ec), and *N. meningitidis* (Nm). *Staph. aureus* (Sa), *Strep. agalactiae* (Sag). *Strep. pneumoniae* (Spn), and *Strep. pyogenes* (Spy) as a function of alignment parameters. Although fluidity increases with higher values of identity ($i$) and coverage ($c$) (see Figure 27), only three rank-orderings of fluidity (of 5040 possible orderings) are found in 224/225 combinations of alignment parameters.

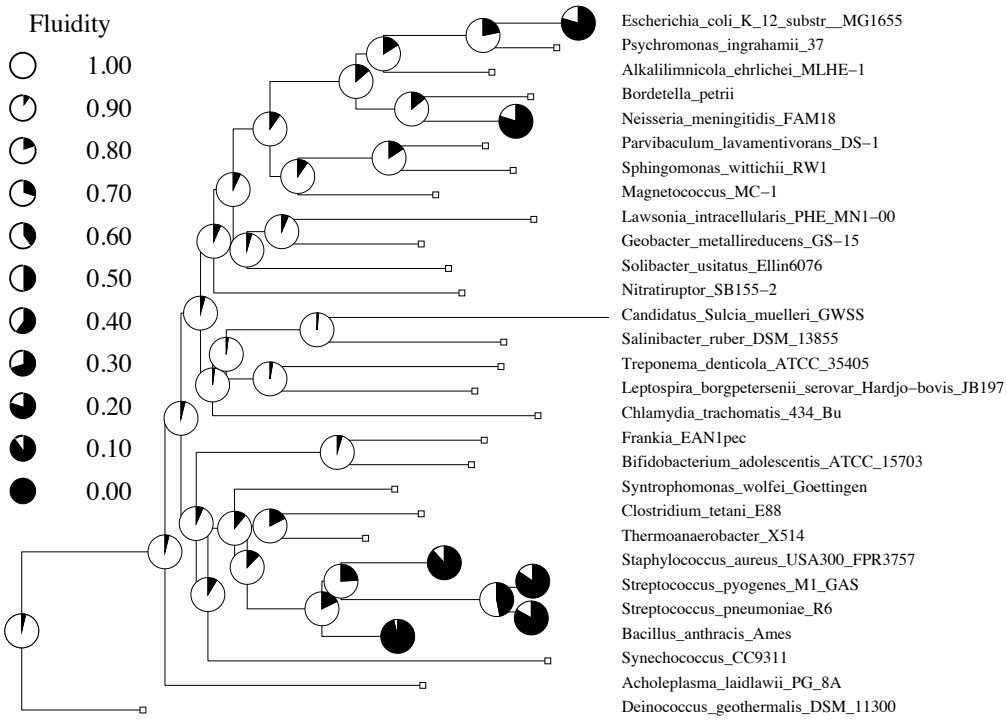



**Figure 29:** Fluidity increases with phylogenetic scale such that the fluidity of multiply-resequenced species is in the range of $0.1-0.3$ and the fluidity of all genomes included in the analysis approaches 1. Each circle represents the relative fluidity at a species (with multiple sequenced genomes) or internal node (the fluidity of all the genomes in the tree below it). Open circles are $\varphi = 1$ and black circles are $\varphi = 0$. The phylogenetic tree of 29 bacterial species was assembled using AMPHORA [186]. Branch lengths correspond to the average number of amino acid substitutions per position in well-conserved marker genes.

community levels [134]. In contrast, estimates of pan-genome sizes at such scales will be problematic for the same reasons as outlined here when applied to closely related organisms. As a general rule, similarity based approaches to quantifying other forms of genome diversity are likely to be robust whereas estimates of the total diversity will be less so.

## *6.3 Discussion*

The proposal that there exists a core and pan genome for bacterial species represents a significant advance in the conceptualization of gene variability within microorganisms [170]. The basic premise of these two concepts have been borne out by the finding that the gene content of bacteria can vary significantly when comparing the sequence of two isolates from a species or genus [170, 80, 78, 76, 136, 138, 39, 45]. For example, it is now well established that some genes are found in most, if not all, sequenced genomes of isolates from within a sample. In addition, it is also well established that some genes are found in very few, if only one, sequenced isolate within a sample. However, as we have demonstrated here, efforts to infer the size of the pan and core genomes of an entire species or genus from the frequency distribution of genes within a small sample of sequenced genomes will almost certainly fail. Similarly, efforts to compare the core or pan genomes sizes of bacterial species or genera will be uninformative. The reason is that pan and core genome sizes depend sensitively on the frequency of rare events (such as a rare gene occurring in a genome) whose frequency cannot be accurately estimated from a small sample of sequenced genomes. Instead, we have proposed the use of an alternative diversity metric – genomic fluidity – which is a reliable and robust estimator of the gene dissimilarity amongst a group of sequenced genomes.

This study has a number of key implications for future sequencing efforts. First, it suggests that efforts to understand a single species by sequencing as many isolates as possible may be limited in their ability to comprehensively define the diversity within that species [79]. Clearly, such studies will remain important in their ability to describe expected genomic differences (in contrast to rare genomic differences). Next, our findings also suggest that the expected gene dissimilarity within a given species can be well characterized by sequencing a relatively small number of well-chosen representatives. Sequencing a few dozen genomes is a fairly straightforward

task given recent advances in sequencing technology. Finally, perhaps the most far-reaching implication of the work presented here is that we have shown it is possible to compare the relative genomic fluidity of different groups of bacteria (e.g. species, genera, or higher). We have shown that genomic fluidity can reliably distinguish between subtle differences in true gene distributions (in a computational study) as well as determine when it is possible to rank-order a set of 7 species based on the analysis of 109 whole genomes (in a bioinformatics analysis).

Despite its merits, genomic fluidity is not meant to describe all forms of genome variation. Genomic fluidity can provide a reliable estimate for how many new genes additional sequencing is likely to reveal, with respect to a previously sequenced genome. It cannot, however, provide an estimate of the amount of sequencing necessary to cover the gene novelty in the entire group (for reasons similar to why estimates of the pan genome size are impossible). In addition, genomic fluidity restricts itself to one component of genomic difference. There are a variety of forms of genomic differences beyond gene compositional differences or the more classic finding of single-nucleotide polymorphisms. Genomes may differ in terms of gene synteny [22], copy number variation [137, 164], plasmid and/or prophage presence [10], codon biases [185, 95], and methylation state [59]. It would be prudent to consider other diversity metrics, in addition to the metric of genomic fluidity studied here, that account for forms of variation in genome state amongst closely related organisms.

In summary, genomic fluidity is an integrated measure of gene diversity within a group of organisms. Genomic fluidity is both estimable given a small number of sequenced genomes and robust to variation in alignment parameters. As such, we recommend that genomic fluidity be used in place of pan and core genome size estimates when assessing gene diversity within a species or a group of closely related organisms. However, the precise relationship between variation in gene composition and genomic fluidity with underlying mechanisms of gene family diversification are yet

to be resolved [68]. Recent calls for comparing and contrasting the average overlap of gene content with respect to average nucleotide divergence provide one possible route to disentangling the effects of ecological and genomic structure [90], but much work remains at the interface of bioinformatics and ecological analysis. For example, the detailed comparison of complete bacterial genomes from closely related biofilm-forming bacteria revealed how and why different organisms have adapted to and shaped their environment [47]. Similarly, genomic analysis of cyanoviruses sampled in the oceans helped uncover photosynthetic pathways which enable the exploitation of a niche distinct from previously cultured *E. coli* based phages despite sharing many common genes and genome architecture [165]. Genomic fluidity complements the detailed functional comparison of genomes by robustly estimating dissimilarity of genes within groups of genomes and providing insight into their potential evolvability. In so doing, our results highlight the need for continued focus on developing new toolsets for assessing what can be inferred about the genome composition and diversity of prokaryotic species and communities based on analysis of a sub-sample of genomes.

## *6.4 Materials and Methods*

### 6.4.1 Fluidity estimator pipeline

Complete annotated genomes and draft annotated genomes were retrieved from NCBI GenBank in the GenBank format. Genomes were automatically re-annotated without hand-curation using a recently developed infrastructure resulting in new GenBank-formatted files (see Chapter 4). Automatic re-annotation removes annotation bias arising from variability in annotation methods, depth of curation, and the resulting impact on the list of candidate genes – a similar approach was recently used in the analysis of genomes within a bacterial genus [39]. Following this process, putative protein sequences were extracted from annotated CDS regions and aligned using BLASTP [12] in all vs. all pairwise amino acid alignment. A pair of genes were

considered homologous if the protein alignment covered more than $c$ fraction of each gene's length and identity in the alignment exceeded $i$. To improve performance, alignments were parallelized between nodes on a compute cluster using the Torque PBS job scheduler. Next, genes were clustered into gene families using a strict clique requirement, i.e. each new gene considered for inclusion into a family must have an alignment with every member of the family satisfying the minimum criteria described above. Alignments were processed in order of increasing E-value, to prevent lower quality alignments from disrupting formation of families using higher quality alignments. Each gene was allowed to participate in only one family; if the gene could not be joined into any gene family, it formed its own singleton family. Gene family assignments were used to calculate fluidity using Eq. 13. We used the jackknife estimator [54] to estimate the variance of the fluidity estimator $\mathrm{Var}[\widehat{\varphi}]$. Explicitly, for a group of $N$ genomes, the variance is

$$\hat{\sigma}^2 = \widehat{\mathrm{Var}}[\widehat{\varphi}] = \frac{N-1}{N} \sum_{\kappa} \left( \widehat{\varphi}_\kappa - \widehat{\varphi} \right)^2, \tag{14}$$

where $\widehat{\varphi}_\kappa$ are the leave-one-out statistics,

$$\widehat{\varphi}_\kappa = \frac{1}{\binom{N-1}{2}} \sum_{\substack{k<l \\ k \neq \kappa \neq l}} \frac{U_k + U_l}{M_k + M_l}. \tag{15}$$

### 6.4.2 Significance test for fluidity differences

Consider two sets of genomes, the first set consisting of $n_1$ genomes, the second set consisting of $n_2$ genomes. For each pair of genomes, we determine the fraction of the total number of unique genes and the total number of genes. Averaging over all pairs in the first set gives the fluidity $\hat{\varphi}_1$; in the second set $\hat{\varphi}_2$. Suppose $\hat{\varphi}_1 > \hat{\varphi}_2$. We want to determine whether this inequality is significant.

From the theory of $U$-statistics it is known that the estimated fluidity has approximately a normal distribution [100]. The mean of this distribution is estimated to be $\widehat{\varphi}_1$ in the first set and $\widehat{\varphi}_2$ in the second set. The variance is estimated (by jackknifing)

to be $\widehat{\sigma}_1^2$ in the first set and $\widehat{\sigma}_2^2$ in the second set. We use the parameters of the approximate normal distributions to compute the significance of the observed fluidity differences. Formally, this corresponds to a two-sample two-sided $z$-test with one degree of freedom (the effective number of degrees of freedom are taken into account by the jackknife estimation).

## 6.5 Acknowledgements

We thank King Jordan, Jessa Lee, Tim Read and Matt Sullivan for their feedback and suggestions on the work presented in this chapter. We thank Anju Varadarajan for her assistance in the implementation of the bioinformatics pipeline. We thank Lee Katz, Scott Sammons, Dhwani Govil, Brian Harcourt, King Jordan and Leonard Mayer for providing access to *N. meningitidis* genomes sequenced at the Centers for Disease Control and Prevention and for help with their analysis.

## 6.6 Funding

The work presented in this chapter was supported by the Defense Advanced Research Projects Agency under grants HR0011-05-1-0057 and HR0011-09-1-0055. Joshua S. Weitz, Ph.D., holds a Career Award at the Scientific Interface from the Burroughs Wellcome Fund. Andrey Kislyuk was supported, in part, by a Georgia Tech Research and Innovation travel grant.

**Table 16:** Significant fluidity differences for $i = 0.5$ and $c = 0.5$ (see Materials and Methods). Species are ordered such that in the upper part of the table fluidity differences are positive, i.e., in all three cases Ba has the lowest fluidity. The null hypothesis that the fluidity difference is not significant can be rejected with a $p$-value of 0.05 is noted with a $\star$, whereas comparisons for which the null hypothesis cannot be rejected are noted with a $\circ$.

| | Ec | Nm | Sag | Spy | Spn | Sau | Ba |
|---|---|---|---|---|---|---|---|
| Ec | $\times$ | $\circ$ | $\circ$ | $\star$ | $\star$ | $\star$ | $\star$ |
| Nm | | $\times$ | $\circ$ | $\circ$ | $\star$ | $\star$ | $\star$ |
| Sag | | | $\times$ | $\circ$ | $\circ$ | $\circ$ | $\star$ |
| Spy | | | | $\times$ | $\circ$ | $\star$ | $\star$ |
| Spn | | | | | $\times$ | $\star$ | $\star$ |
| Sau | | | | | | $\times$ | $\star$ |
| Ba | | | | | | | $\times$ |

**Table 17:** $p$-values for fluidity differences for $i = 0.5$ and $c = 0.5$. Details of the significance test are provided in the Materials and Methods.

| | Ec | Nm | Sag | Spy | Spn | Sau | Ba |
|---|---|---|---|---|---|---|---|
| Ec | $\times$ | $5.70\ 10^{-1}$ | $7.87\ 10^{-1}$ | $2.16\ 10^{-2}$ | $8.57\ 10^{-8}$ | $8.93\ 10^{-20}$ | $5.42\ 10^{-27}$ |
| Nm | | $\times$ | $9.53\ 10^{-1}$ | $1.03\ 10^{-1}$ | $3.11\ 10^{-3}$ | $1.14\ 10^{-7}$ | $1.81\ 10^{-12}$ |
| Sag | | | $\times$ | $4.49\ 10^{-1}$ | $3.35\ 10^{-1}$ | $8.03\ 10^{-2}$ | $1.22\ 10^{-2}$ |
| Spy | | | | $\times$ | $7.52\ 10^{-1}$ | $4.04\ 10^{-2}$ | $3.04\ 10^{-4}$ |
| Spn | | | | | $\times$ | $9.53\ 10^{-6}$ | $2.44\ 10^{-12}$ |
| Sau | | | | | | $\times$ | $2.60\ 10^{-4}$ |
| Ba | | | | | | | $\times$ |

**Table 18:** Significant fluidity differences for $i = 0.62$ and $c = 0.62$. Species are ordered such that in the upper part of the table fluidity differences are positive, i.e., in all three cases Ba has the lowest fluidity. The null hypothesis that the fluidity difference is not significant can be rejected with a $p$-value of 0.05 is noted with a $\star$, whereas comparisons for which the null hypothesis cannot be rejected are noted with a $\circ$.

| | Nm | Ec | Sag | Spy | Spn | Sau | Ba |
|---|---|---|---|---|---|---|---|
| Nm | × | ∘ | ∘ | ∘ | ⋆ | ⋆ | ⋆ |
| Ec | | × | ∘ | ∘ | ⋆ | ⋆ | ⋆ |
| Sag | | | × | ∘ | ∘ | ∘ | ⋆ |
| Spy | | | | × | ∘ | ∘ | ⋆ |
| Spn | | | | | × | ⋆ | ⋆ |
| Sau | | | | | | × | ⋆ |
| Ba | | | | | | | × |

**Table 19:** $p$-values for fluidity differences for $i = 0.62$ and $c = 0.62$. Details of the significance test are provided in the Materials and Methods.

| | Nm | Ec | Sag | Spy | Spn | Sau | Ba |
|---|---|---|---|---|---|---|---|
| Nm | × | $9.53\ 10^{-1}$ | $6.98\ 10^{-1}$ | $1.27\ 10^{-1}$ | $1.30\ 10^{-2}$ | $8.98\ 10^{-7}$ | $5.23\ 10^{-11}$ |
| Ec | | × | $7.00\ 10^{-1}$ | $8.33\ 10^{-2}$ | $7.71\ 10^{-5}$ | $5.66\ 10^{-16}$ | $1.13\ 10^{-22}$ |
| Sag | | | × | $5.83\ 10^{-1}$ | $5.01\ 10^{-1}$ | $8.74\ 10^{-2}$ | $9.66\ 10^{-3}$ |
| Spy | | | | × | $9.22\ 10^{-1}$ | $5.58\ 10^{-2}$ | $7.41\ 10^{-4}$ |
| Spn | | | | | × | $2.92\ 10^{-5}$ | $5.95\ 10^{-11}$ |
| Sau | | | | | | × | $1.58\ 10^{-3}$ |
| Ba | | | | | | | × |

**Table 20:** Significant fluidity differences for $i = 0.74$ and $c = 0.74$. Species are ordered such that in the upper part of the table fluidity differences are positive, i.e., in all three cases Ba has the lowest fluidity. The null hypothesis that the fluidity difference is not significant can be rejected with a $p$-value of 0.05 is noted with a $\star$, whereas comparisons for which the null hypothesis cannot be rejected are noted with a $\circ$.

| | Nm | Ec | Sag | Spn | Spy | Sau | Ba |
|---|---|---|---|---|---|---|---|
| Nm | × | ∘ | ∘ | ⋆ | ∘ | ⋆ | ⋆ |
| Ec | | × | ∘ | ⋆ | ∘ | ⋆ | ⋆ |
| Sag | | | × | ∘ | ∘ | ∘ | ⋆ |
| Spn | | | | × | ∘ | ⋆ | ⋆ |
| Spy | | | | | × | ∘ | ⋆ |
| Sau | | | | | | × | ⋆ |
| Ba | | | | | | | × |

**Table 21:** $p$-values for fluidity differences for $i = 0.74$ and $c = 0.74$. Details of the significance test are provided in the Materials and Methods.

| | Nm | Ec | Sag | Spn | Spy | Sau | Ba |
|---|---|---|---|---|---|---|---|
| Nm | × | $4.68\ 10^{-1}$ | $3.28\ 10^{-1}$ | $2.30\ 10^{-2}$ | $1.10\ 10^{-1}$ | $5.34\ 10^{-6}$ | $6.37\ 10^{-10}$ |
| Ec | | × | $4.98\ 10^{-1}$ | $6.28\ 10^{-3}$ | $1.80\ 10^{-1}$ | $4.19\ 10^{-12}$ | $1.41\ 10^{-18}$ |
| Sag | | | × | $8.37\ 10^{-1}$ | $8.24\ 10^{-1}$ | $1.47\ 10^{-1}$ | $1.13\ 10^{-2}$ |
| Spn | | | | × | $9.36\ 10^{-1}$ | $1.76\ 10^{-4}$ | $1.19\ 10^{-9}$ |
| Spy | | | | | × | $9.21\ 10^{-2}$ | $1.54\ 10^{-3}$ |
| Sau | | | | | | × | $2.60\ 10^{-3}$ |
| Ba | | | | | | | × |

Table 22: Accession information for all bacterial genomes used in this project. Strain lists the strain name. Accession is the NCBI accession identifier that is hyper-linked to the NCBI website. The final 5 columns denote the number of coding sequences (CDS) identified in the genome using various schemes: first, the number of CDS in the annotated genome (if available), then the number of CDS identified using the re-annotation scheme described in Materials and Methods (CDS Re-annot), and finally the number of CDS identified using Glimmer [46], GeneMarkS [24] and BLAST [12].

| Strain | Accession | CDS Original | CDS Re-annot | CDS Glimmer | CDS GeneMark | CDS BLAST |
|---|---|---|---|---|---|---|
| *Bacillus anthracis* – 13 genomes | | | | | | |
| A0174 (Draft) | NZ_ABLT01000001 | 5198 | 5512 | 5641 | 5782 | 3302 |
| A0193 (Draft) | NZ_ABKF01000001 | 5309 | 5601 | 5740 | 5889 | 3360 |
| A0389 (Draft) | NZ_ABLB01000001 | 5296 | 5644 | 5783 | 5940 | 3398 |
| A0442 (Draft) | NZ_ABKG01000001 | 5256 | 5598 | 5742 | 5866 | 3353 |
| A0465 (Draft) | NZ_ABLH01000001 | 5300 | 5649 | 5782 | 5925 | 3386 |
| A0488 (Draft) | NZ_ABJC01000001 | 5288 | 5599 | 5733 | 5900 | 3358 |
| A2012 (Draft) | NZ_AAAC02000001 | 5352 | 5474 | 5892 | 5939 | 3341 |
| Tsiankovskii-I (Draft) | NZ_ABDN01000001 | 6051 | 5704 | 5838 | 6025 | 3399 |
| A0248 (Finished) | NC_012659 | 5291 | 5711 | 5855 | 6005 | 3477 |
| AE017225 (Finished) | AE017225 | 5287 | 5427 | 5563 | 5694 | 3360 |
| Ames (Finished) | NC_003997 | 5311 | 5432 | 5568 | 5695 | 3362 |
| Ames_ancestor (Finished) | NC_007530 | 5617 | 5713 | 5856 | 6007 | 3477 |
| CDC684 (Finished) | NC_012581 | 5902 | 5715 | 5859 | 6016 | 3477 |
| *Escherichia coli* – 15 genomes | | | | | | |
| 536 (Finished) | NC_008253 | 4629 | 4553 | 4787 | 4699 | 4190 |
| APEC01 (Finished) | NC_008563 | 4879 | 5208 | 5508 | 5385 | 4524 |
| CFT073 (Finished) | NC_004431 | 5378 | 4894 | 5150 | 5055 | 4400 |
| E24377A (Finished) | NC_009801 | 4997 | 4947 | 5174 | 5138 | 4474 |
| EDL933 (Finished) | NC_002655 | 5419 | 5366 | 5744 | 5564 | 4566 |
| HS (Finished) | NC_009800 | 4384 | 4316 | 4430 | 4449 | 4117 |
| K12 (Finished) | NC_000913 | 4244 | 4320 | 4443 | 4442 | 4294 |
| Sakai (Finished) | NC_002695 | 5341 | 5345 | 5672 | 5534 | 4563 |
| UT189 (Finished) | NC_007946 | 5211 | 4822 | 5054 | 4979 | 4342 |
| 101_1 (Draft) | NZ_AAMK01000001 | 4234 | 4700 | 4852 | 4876 | 4334 |
| B171 (Draft) | NZ_AAJX01000001 | 4705 | 5229 | 5463 | 5416 | 4574 |
| B7A (Draft) | NZ_AAJT01000001 | 4628 | 5070 | 5257 | 5287 | 4494 |
| E110019 (Draft) | NZ_AAJW01000001 | 4742 | 5239 | 5507 | 5448 | 4550 |

| | | | | | | |
|---|---|---|---|---|---|---|
| E22 (Draft) | NZ_AAJV01000001 | 4781 | 5328 | 5607 | 5544 | 4550 |
| F11 (Draft) | NZ_AAJU01000001 | 4461 | 4884 | 5151 | 5046 | 4353 |
| *Neisseria meningitidis* – 14 genomes | | | | | | |
| 053442 (Finished) | NC_010120 | N/A | 2020 | | Not performed | |
| FAM18 (Finished) | NC_008767 | N/A | 1918 | | Not performed | |
| MC58 (Finished) | NC_003112 | N/A | 2063 | | Not performed | |
| Z2491 (Finished) | NC_003116 | N/A | 2049 | | Not performed | |
| alpha14 (Finished) | NC_013016 | N/A | 2059 | | Not performed | |
| alpha153 (Draft) | N/A | N/A | 2354 | | Not performed | |
| alpha275 (Draft) | N/A | N/A | 2565 | | Not performed | |
| NM10699 (Draft) | N/A | N/A | 2110 | 2494 | 2366 | 1317 |
| NM13220 (Draft) | N/A | N/A | 2299 | 2725 | 2530 | 1353 |
| NM15141 (Draft) | N/A | N/A | 2184 | 2578 | 2411 | 1369 |
| NM15293 (Draft) | N/A | N/A | 2063 | 2040 | 2062 | 1285 |
| NM18575 (Draft) | N/A | N/A | 2471 | 2927 | 2751 | 1495 |
| NM5178 (Draft) | N/A | N/A | 2097 | 2510 | 2377 | 1315 |
| NM9261 (Draft) | N/A | N/A | 2110 | 2553 | 2370 | 1341 |
| *Staphylococcus aureus* – 19 genomes | | | | | | |
| JKD6008 (Draft) | NZ_ABRZ01000084 | 2662 | 2681 | 2733 | 2791 | 1854 |
| JKD6009 (Draft) | NZ_ABSA01000082 | 2684 | 2666 | 2720 | 2776 | 1843 |
| MN8 (Draft) | NZ_ACJA01000014 | 2901 | 2714 | 2768 | 2845 | 1841 |
| TCH60 (Draft) | NZ_ACHC01000045 | 2738 | 2551 | 2613 | 2666 | 1816 |
| USA300_TCH959 (Draft) | NZ_AASB01000107 | 2853 | 2784 | 2826 | 2936 | 1899 |
| COL (Finished) | NC_002951 | 2618 | 2568 | 2612 | 2680 | 1843 |
| JH1 (Finished) | NC_009632 | 2780 | 2726 | 2775 | 2835 | 1890 |
| JH9 (Finished) | NC_009487 | 2726 | 2726 | 2773 | 2836 | 1890 |
| MRSA252 (Finished) | NC_002952 | 2656 | 2669 | 2728 | 2792 | 1888 |
| MRSA_USA300_TCH1516 (Finished) | NC_010079 | 2689 | 2696 | 2744 | 2805 | 1890 |
| MSSA476 (Finished) | NC_002953 | 2598 | 2555 | 2599 | 2671 | 1834 |
| MW2 (Finished) | NC_003923 | 2632 | 2541 | 2580 | 2668 | 1832 |
| Mu3 (Finished) | NC_009782 | 2698 | 2647 | 2701 | 2748 | 1876 |
| Mu50 (Finished) | NC_002758 | 2731 | 2677 | 2730 | 2778 | 1885 |
| N315 (Finished) | NC_002745 | 2619 | 2578 | 2624 | 2677 | 1880 |
| NCTC8325 (Finished) | NC_007795 | 2892 | 2608 | 2660 | 2729 | 1830 |
| Newman (Finished) | NC_009641 | 2614 | 2677 | 2722 | 2805 | 1841 |
| RF122 (Finished) | NC_007622 | 2515 | 2589 | 2630 | 2707 | 1841 |
| USA300 (Finished) | NC_007793 | 2604 | 2701 | 2756 | 2806 | 1884 |

| | | | | | | |
|---|---|---|---|---|---|---|
| *Streptococcus agalactiae* – 8 genomes | | | | | | |
| 18RS21 (Draft) | NZ_AAJO01000553 | 2146 | 2179 | 2326 | 2448 | 1316 |
| 515 (Draft) | NZ_AAJQ01000155 | 2275 | 2150 | 2248 | 2203 | 1356 |
| COH1 (Draft) | NZ_AAJR01000393 | 2376 | 2295 | 2437 | 2341 | 1414 |
| H36B (Draft) | NZ_AAJS01000345 | 2376 | 2305 | 2466 | 2354 | 1430 |
| CJB111 (Draft) | NZ_AAJP01000255 | 2197 | 2099 | 2209 | 2137 | 1363 |
| NEM316 (Finished) | NC_004368 | 2094 | 2127 | 2191 | 2161 | 1358 |
| 2603V/R (Finished) | NC_004116 | 2124 | 2108 | 2164 | 2146 | 1385 |
| A909 (Finished) | NC_007432 | 1996 | 2060 | 2127 | 2094 | 1387 |
| *Streptococcus pneumoniae* – 26 genomes | | | | | | |
| CDC0288-04 (Draft) | NZ_ABGF01000001 | 1825 | 2015 | 2105 | 2131 | 1311 |
| CDC1087-00 (Draft) | NZ_ABFT01000001 | 1763 | 2153 | 2230 | 2329 | 1369 |
| CDC1873-00 (Draft) | NZ_ABFS01000001 | 2026 | 2297 | 2390 | 2464 | 1372 |
| CDC3059-06 (Draft) | NZ_ABGG01000001 | 2088 | 2293 | 2373 | 2456 | 1327 |
| MLV016 (Draft) | NZ_ABGH01000001 | 1851 | 2163 | 2253 | 2340 | 1393 |
| SP11-BS70 (Draft) | NZ_ABAC01000001 | 2365 | 2095 | 2154 | 2221 | 1343 |
| SP14-BS69 (Draft) | NZ_ABAD01000001 | 2807 | 2524 | 2625 | 2675 | 1461 |
| SP18-BS74 (Draft) | NZ_ABAE01000001 | 2415 | 2144 | 2200 | 2282 | 1377 |
| SP19-BS75 (Draft) | NZ_ABAF01000001 | 2480 | 2220 | 2300 | 2339 | 1371 |
| SP195 (Draft) | NZ_ABGE01000001 | 1945 | 2204 | 2297 | 2353 | 1331 |
| SP23-BS72 (Draft) | NZ_ABAG01000001 | 2416 | 2154 | 2227 | 2294 | 1337 |
| SP3-BS71 (Draft) | NZ_AAZZ01000001 | 2378 | 2110 | 2191 | 2250 | 1334 |
| SP6-BS73 (Draft) | NZ_ABAA01000001 | 2507 | 2240 | 2298 | 2373 | 1380 |
| SP9-BS68 (Draft) | NZ_ABAB01000001 | 2429 | 2159 | 2236 | 2298 | 1336 |
| TIGR4-454 (Draft) | NZ_AAGY02000001 | 1878 | 1994 | 2036 | 2117 | 1294 |
| 70585 (Finished) | NC_012468 | 2202 | 2214 | 2289 | 2340 | 1364 |
| ATCC700669 (Finished) | NC_011900 | 1990 | 2195 | 2300 | 2319 | 1357 |
| CGSP14 (Finished) | NC_010582 | 2206 | 2164 | 2231 | 2293 | 1353 |
| D39 (Finished) | NC_008533 | 1914 | 2031 | 2100 | 2149 | 1306 |
| G54_MLSTST63 (Finished) | NC_011072 | 2115 | 2085 | 2163 | 2199 | 1326 |
| Hungary19A-6 (Finished) | NC_010380 | 2155 | 2249 | 2338 | 2365 | 1358 |
| JJA (Finished) | NC_012466 | 2123 | 2118 | 2203 | 2247 | 1328 |
| P1031 (Finished) | NC_012467 | 2073 | 2135 | 2221 | 2252 | 1331 |
| R6 (Finished) | NC_003098 | 2043 | 2021 | 2087 | 2144 | 1301 |
| TIGR4 (Finished) | NC_003028 | 2094 | 2139 | 2209 | 2268 | 1354 |
| Taiwan19F-14 (Finished) | NC_012469 | 2044 | 2092 | 2158 | 2224 | 1309 |

*Streptococcus pyogenes* – 14 genomes

| | | | | | | |
|---|---|---|---|---|---|---|
| M49591 (Draft) | NZ_AAFV01000001 | 1365 | 1426 | 1457 | 1501 | 846 |
| M1GAS (Finished) | NC_002737 | 1697 | 1791 | 1839 | 1863 | 1177 |
| MGAS10270 (Finished) | NC_008022 | 1987 | 1894 | 1946 | 1976 | 1183 |
| MGAS10394 (Finished) | NC_006086 | 1886 | 1826 | 1874 | 1911 | 1197 |
| MGAS10750 (Finished) | NC_008024 | 1979 | 1893 | 1950 | 1972 | 1199 |
| MGAS2096 (Finished) | NC_008023 | 1898 | 1813 | 1853 | 1898 | 1202 |
| MGAS315 (Finished) | NC_004070 | 1865 | 1864 | 1920 | 1964 | 1167 |
| MGAS5005 (Finished) | NC_007297 | 1865 | 1788 | 1840 | 1871 | 1187 |
| MGAS6180 (Finished) | NC_007296 | 1894 | 1813 | 1871 | 1897 | 1176 |
| MGAS8232 (Finished) | NC_003485 | 1845 | 1881 | 1924 | 1966 | 1191 |
| MGAS9429 (Finished) | NC_008021 | 1877 | 1755 | 1800 | 1826 | 1163 |
| Mabfredo (Finished) | NC_009332 | 1745 | 1802 | 1851 | 1893 | 1175 |
| NZ131 (Finished) | NC_011375 | 1699 | 1734 | 1811 | 1817 | 1162 |
| SSI-1 (Finished) | NC_004606 | 1861 | 1862 | 1912 | 1961 | 1167 |

# CHAPTER VII

# CONCLUSION

We have presented a number of methods and innovations in DNA sequence analysis applicable to genomic and metagenomic data (except in Chapter 6, for which applications to metagenomic data are in development). We hope that in the course of our presentation, a better understanding of the state of the art in genome and metagenome analysis could be gained.

The complexity of biological systems which remains hidden from our understanding is matched by the intellectual reward of discovery when another aspect of these systems is characterized. Direct sequencing of environmental DNA is a key tool for this process. In the next few decades, we can expect the coalescence of accumulated knowledge and experimental methods to produce a qualitative improvement of our knowledge of life. Correspondingly more sophisticated algorithms will be needed for parallelized analysis of massive samples of biological sequence, biological network data, genetic features, bioengineering applications and other biological problems. This is an exciting time for biology, and our work is cut out for us.

# REFERENCES


[1] "Fames: Fidelity of analysis of metagenomic samples. http://fames.jgi-psf.org/."

[2] "Fsfind website. http://topaz.gatech.edu/ kislyuk/fsfind/."

[3] "A genomic encyclopedia of bacteria and archaea (geba). http://www.jgi.doe.gov/programs/GEBA/index.html."

[4] "Intel threading building blocks. http://www.threadingbuildingblocks.org/."

[5] "Likelybin webpage. http://ecotheory.biology.gatech.edu/likelybin."

[6] "Nvidia cuda. http://www.nvidia.com/object/cuda_home_new.html."

[7] Abe, T., Kanaya, S., Kinouchi, M., Ichiba, Y., Kozuki, T., and Ikemura, T., "Informatics for unveiling hidden genome signatures.," *Genome research*, vol. 13, pp. 693–702, April 2003.

[8] Abelson, H. and Sussman, G. J., *Structure and Interpretation of Computer Programs*. MIT Electrical Engineering and Computer Science, The MIT Press, second ed., July 1996.

[9] Achtman, M. and Wagner, M., "Microbial diversity and the genetic nature of microbial species.," *Nature reviews. Microbiology*, vol. 6, pp. 431–440, June 2008.

[10] Ahmed, N., Dobrindt, U., Hacker, J., and Hasnain, S. E. E., "Genomic fluidity and pathogenic bacteria: applications in diagnostics, epidemiology and intervention.," *Nature reviews. Microbiology*, April 2008.

[11] Aldinucci, M., Torquati, M., and Meneghin, M., "Fastflow: Efficient parallel streaming applications on multi-core," Sep 2009.

[12] Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J., "Gapped blast and psi-blast: a new generation of protein database search programs.," *Nucleic Acids Res*, vol. 25, pp. 3389–3402, September 1997.

[13] Angly, F. E., Felts, B., Breitbart, M., Salamon, P., Edwards, R. A., Carlson, C., Chan, A. M., Haynes, M., Kelley, S., Liu, H., Mahaffy, J. M., Mueller, J. E., Nulton, J., Olson, R., Parsons, R., Rayhawk, S., Suttle, C. A., and Rohwer, F., "The marine viromes of four oceanic regions.," *PLoS Biol*, vol. 4, November 2006.

[14] Antonov, I. and Borodovsky, M., "Genetack: frameshift identification in protein-coding sequences by the viterbi algorithm.," *Journal of bioinformatics and computational biology*, vol. 8, pp. 535–551, June 2010.

[15] Aziz, R., Bartels, D., Best, A., DeJongh, M., Disz, T., Edwards, R., Formsma, K., Gerdes, S., Glass, E., Kubal, M., Meyer, F., Olsen, G., Olson, R., Osterman, A., Overbeek, R., McNeil, L., Paarmann, D., Paczian, T., Parrello, B., Pusch, G., Reich, C., Stevens, R., Vassieva, O., Vonstein, V., Wilke, A., and Zagnitko, O., "The rast server: rapid annotations using subsystems technology," *BMC Genomics*, vol. 9, no. 1, p. 75, 2008.

[16] Bailly, J., Fraissinet-Tachet, L., Verner, M.-C. C., Debaud, J.-C. C., Lemaire, M., Wésolowski-Louvel, M., and Marmeisse, R., "Soil eukaryotic functional diversity, a metatranscriptomic approach.," *The ISME journal*, vol. 1, pp. 632–642, November 2007.

[17] Béjà, O., Spudich, E. N., Spudich, J. L., Leclerc, M., and DeLong, E. F., "Proteorhodopsin phototrophy in the ocean.," *Nature*, vol. 411, pp. 786–789, June 2001.

[18] Bendtsen, J. D. A., Nielsen, H., von Heijne, G., and Brunak, S. A., "Improved prediction of signal peptides: Signalp 3.0," *Journal of molecular biology*, vol. 340, no. 4, pp. 783–795, 2004.

[19] Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., and Sayers, E. W., "Genbank.," *Nucleic acids research*, vol. 37, pp. D26–31, January 2009.

[20] Bentley, D., Balasubramanian, S., Swerdlow, H., Smith, G., Milton, J., Brown, C., Hall, K., Evers, D., Barnes, C., Bignell, H., Boutell, J., Bryant, J., Carter, R., Cheetham, K., Cox, A., Ellis, D., Flatbush, M., Gormley, N., Humphray, S., Irving, L., Karbelashvili, M., Kirk, S., Li, H., Liu, X., Maisinger, K., Murray, L., Obradovic, B., Ost, T., Parkinson, M., Pratt, M., Rasolonjatovo, I., Reed, M., Rigatti, R., Rodighiero, C., Ross, M., Sabot, A., Sankar, S., Scally, A., Schroth, G., Smith, M., Smith, V., Spiridou, A., Torrance, P., Tzonev, S., Vermaas, E., Walter, K., Wu, X., Zhang, L., Alam, M., Anastasi, C., Aniebo, I., Bailey, D., Bancarz, I., Banerjee, S., Barbour, S., Baybayan, P., Benoit, V., Benson, K., Bevis, C., Black, P., Boodhun, A., Brennan, J., Bridgham, J., Brown, R., Brown, A., Buermann, D., Bundu, A., Burrows, J., Carter, N., Castillo, N., Catenazzi, M., Chang, S., Cooley, N., Crake, N., Dada, O., Diakoumakos, K., Dominguez-Fernandez, B., Earnshaw, D., Egbujor, U., Elmore, D., Etchin, S., Ewan, M., Fedurco, M., Fraser, L., Fuentes, Furey, S., George, D., Gietzen, K., Goddard, C., Golda, G., Granieri, P., Green, D., Gustafson, D., Hansen, N., Harnish, K., Haudenschild, C., Heyer, N., Hims, M., Ho, J., Horgan, A., Hoschler, K., Hurwitz, S., Ivanov, D., Johnson, M., James, T., Jones, H., Kang, G.-D., Kerelska, T., Kersey, A., Khrebtukova, I., Kindwall, A., Kingsbury, Z., Kokko-Gonzales, P., Kumar, A., Laurent, M., Lawley, C., Lee, S., Lee, X., Liao, A., Loch, J., Lok, M., Luo, S., Mammen, R., Martin, J., McCauley, P., McNitt, P., Mehta, P., Moon, K., Mullens, J., Newington, T., Ning, Z., Ng, B., Novo,

S., O'Neill, M., Osborne, M., Osnowski, A., Ostadan, O., Paraschos, L., Pickering, L., Pike, A., Pike, A., Pinkard, C., Pliskin, D., Podhasky, J., Quijano, V., Raczy, C., Rae, V., Rawlings, S., Rodriguez, A., Roe, P., Rogers, J., Bacigalupo, M., Romanov, N., Romieu, A., Roth, R., Rourke, N., Ruediger, S., Rusman, E., Sanches-Kuiper, R., Schenker, M., Seoane, J., Shaw, R., Shiver, M., Short, S., Sizto, N., Sluis, J., Smith, M., Sohna, Spence, E., Stevens, K., Sutton, N., Szajkowski, L., Tregidgo, C., Turcatti, G., Vandevondele, S., Verhovsky, Y., Virk, S., Wakelin, S., Walcott, G., Wang, J., Worsley, G., Yan, J., Yau, L., Zuerlein, M., Rogers, J., Mullikin, J., Hurles, M., McCooke, N., West, J., Oaks, F., Lundberg, P., Klenerman, D., Durbin, R., and Smith, A., "Accurate whole human genome sequencing using reversible terminator chemistry," *Nature*, vol. 456, no. 7218, pp. 53–59, 2008.

[21] Bentley, D. R., "Whole-genome re-sequencing," *Current Opinion in Genetics & Development*, vol. 16, pp. 545–552, December 2006.

[22] Bentley, S. D., Chater, K. F., Cerdeño Tárraga, A.-M. M., Challis, G. L., Thomson, N. R., James, K. D., Harris, D. E., Quail, M. A., Kieser, H., Harper, D., Bateman, A., Brown, S., Chandra, G., Chen, C. W., Collins, M., Cronin, A., Fraser, A., Goble, A., Hidalgo, J., Hornsby, T., Howarth, S., Huang, C.-H. H., Kieser, T., Larke, L., Murphy, L., Oliver, K., O'Neil, S., Rabbinowitsch, E., Rajandream, M.-A. A., Rutherford, K., Rutter, S., Seeger, K., Saunders, D., Sharp, S., Squares, R., Squares, S., Taylor, K., Warren, T., Wietzorrek, A., Woodward, J., Barrell, B. G., Parkhill, J., and Hopwood, D. A., "Complete genome sequence of the model actinomycete streptomyces coelicolor a3(2).," *Nature*, vol. 417, pp. 141–147, May 2002.

[23] Besemer, J. and Borodovsky, M., "Heuristic approach to deriving models for gene finding," *Nucl. Acids Res.*, vol. 27, pp. 3911–3920, October 1999.

[24] Besemer, J., Lomsadze, A., and Borodovsky, M., "Genemarks: a self-training method for prediction of gene starts in microbial genomes. implications for finding sequence motifs in regulatory regions.," *Nucleic Acids Res*, vol. 29, pp. 2607–2618, June 2001.

[25] Birney, E., Thompson, J., and Gibson, T., "Pairwise and searchwise: finding the optimal alignment in a simultaneous comparison of a protein profile against all dna translation frames," *Nucl. Acids Res.*, vol. 24, pp. 2730–2739, July 1996.

[26] Birney, E., Clamp, M., and Durbin, R., "Genewise and genomewise," *Genome Res.*, vol. 14, pp. 988–995, May 2004.

[27] Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M. C., Estreicher, A., Gasteiger, E., Martin, M. J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S., and Schneider, M., "The swiss-prot protein knowledgebase and its supplement trembl in 2003," *Nucleic Acids Res*, vol. 31, no. 1, pp. 365–370, 2003.

[28] Borodovsky, M. and McIninch, J., "Recognition of genes in dna sequence with ambiguities.," *Biosystems*, vol. 30, no. 1-3, pp. 161–171, 1993.

[29] Brady, A. and Salzberg, S. L., "Phymm and phymmbl: metagenomic phylogenetic classification with interpolated markov models," *Nature Methods*, vol. 6, pp. 673–676, August 2009.

[30] Brudno, M., Chapman, M., Gottgens, B., Batzoglou, S., and Morgenstern, B., "Fast and sensitive multiple alignment of large genomic sequences," *BMC Bioinformatics*, vol. 4, pp. 66+, December 2003.

[31] Burrows, M. and Wheeler, D. J., "A block-sorting lossless data compression algorithm.," Tech. Rep. 124, 1994.

[32] Callister, S. J., McCue, L. A. A., Turse, J. E., Monroe, M. E., Auberry, K. J., Smith, R. D., Adkins, J. N., and Lipton, M. S., "Comparative bacterial

proteomics: analysis of the core genome concept.," *PloS one*, vol. 3, pp. e1542+, February 2008.

[33] Campbell, A., Mrázek, J., and Karlin, S., "Genome signature comparisons among prokaryote, plasmid, and mitochondrial dna," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 96, no. 16, pp. 9184–9189, 1999.

[34] Chan, C. K., Hsu, A. L., Tang, S. L., and Halgamuge, S. K., "Using growing self-organising maps to improve the binning process in environmental whole-genome shotgun sequencing.," *Journal of biomedicine & biotechnology*, vol. 2008, no. 1, 2008.

[35] Chan, C.-K. K., Hsu, A. L., Halgamuge, S. K., and Tang, S.-L., "Binning sequences using very sparse labels within a metagenome," *BMC Bioinformatics*, vol. 9, pp. 215+, April 2008.

[36] Chatterji, S., Yamazaki, I., Bai, Z., and Eisen, J., "Compostbin: A dna composition-based algorithm for binning environmental shotgun reads," *ArXiv e-prints*, vol. 708, August 2007.

[37] Chen, I. and Dubnau, D., "Dna uptake during bacterial transformation," *Nat Rev Microbiol*, vol. 2, pp. 241–9, Mar 2004.

[38] Chen, L., Yang, J., Yu, J., Yao, Z., Sun, L., Shen, Y., and Jin, Q., "Vfdb: a reference database for bacterial virulence factors," *Nucleic Acids Res*, vol. 33, pp. D325–8, Jan 1 2005.

[39] Chen, P., Cook, C., Stewart, A., Nagarajan, N., Sommer, D., Pop, M., Thomason, B., Thomason, M., Lentz, S., Nolan, N., Sozhamannan, S., Sulakvelidze, A., Mateczun, A., Du, L., Zwick, M., and Read, T., "Genomic characterization of the yersinia genus," *Genome Biology*, vol. 11, no. 1, pp. R1+, 2010.

[40] Claverie, J. M., "Detecting frame shifts by amino acid sequence comparison.," *J Mol Biol*, vol. 234, pp. 1140–1157, December 1993.

[41] Clemente, J. C., Satou, K., and Valiente, G., "Phylogenetic reconstruction from non-genomic data," *Bioinformatics*, vol. 23, pp. e110–115, January 2007.

[42] Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., and de Hoon, M. J. L., "Biopython: freely available python tools for computational molecular biology and bioinformatics," *Bioinformatics*, vol. 25, pp. 1422–1423, June 2009.

[43] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., *Introduction to Algorithms*. McGraw-Hill Science / Engineering / Math, 2nd ed., December 2003.

[44] Darling, A. C., Mau, B., Blattner, F. R., and Perna, N. T., "Mauve: multiple alignment of conserved genomic sequence with rearrangements.," *Genome Res*, vol. 14, pp. 1394–1403, July 2004.

[45] D'Auria, G., Jimenez-Hernandez, N., Peris-Bondia, F., Moya, A., and Latorre, A., "Legionella pneumophila pangenome reveals strain-specific virulence factors," *BMC Genomics*, vol. 11, pp. 181+, March 2010.

[46] Delcher, A. L., Harmon, D., Kasif, S., White, O., and Salzberg, S. L., "Improved microbial gene identification with glimmer," *Nucleic acids research*, vol. 27, no. 23, pp. 4636–4641, 1999.

[47] Denef, V. J., Kalnejais, L. H., Mueller, R. S., Wilmes, P., Baker, B. J., Thomas, B. C., VerBerkmoes, N. C., Hettich, R. L., and Banfield, J. F., "Proteogenomic basis for ecological divergence of closely related bacteria in natural acidophilic microbial communities," *Proceedings of the National Academy of Sciences*, vol. 107, pp. 2383–2390, February 2010.

[48] DeSantis, T. Z., Hugenholtz, P., Larsen, N., Rojas, M., Brodie, E. L., Keller, K., Huber, T., Dalevi, D., Hu, P., and Andersen, G. L., "Greengenes,

a chimera-checked 16s rrna gene database and workbench compatible with arb," *Appl. Environ. Microbiol.*, vol. 72, no. 7, pp. 5069–5072, 2006.

[49] Deschavanne, P. J., Giron, A., Vilain, J., Fagot, G., and Fertil, B., "Genomic signature: characterization and classification of species assessed by chaos game representation of sequences," *Mol Biol Evol*, vol. 16, pp. 1391–1399, October 1999.

[50] Diaz, N., Krause, L., Goesmann, A., Niehaus, K., and Nattkemper, T., "Tacoa - taxonomic classification of environmental genomic fragments using a kernelized nearest neighbor approach," *BMC Bioinformatics*, vol. 10, no. 1, pp. 56+, 2009.

[51] Dinsdale, E. A., Edwards, R. A., Hall, D., Angly, F., Breitbart, M., Brulc, J. M., Furlan, M., Desnues, C., Haynes, M., Li, L., McDaniel, L., Moran, M. A. A., Nelson, K. E., Nilsson, C., Olson, R., Paul, J., Brito, B. R. R., Ruan, Y., Swan, B. K., Stevens, R., Valentine, D. L., Thurber, R. V. V., Wegley, L., White, B. A., and Rohwer, F., "Functional metagenomic profiling of nine biomes.," *Nature*, vol. 452, pp. 629–632, April 2008.

[52] Dittrich, M. T., Klau, G. W., Rosenwald, A., Dandekar, T., and Müller, T., "Identifying functional modules in protein-protein interaction networks: an integrated exact approach.," *Bioinformatics (Oxford, England)*, vol. 24, July 2008.

[53] Doolittle, W. F. and Zhaxybayeva, O., "On the origin of prokaryotic species.," *Genome research*, vol. 19, pp. 744–756, May 2009.

[54] Efron, B., "Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods," *Biometrika*, vol. 68, pp. 589–599, December 1981.

[55] Eid, J., Fehr, A., Gray, J., Luong, K., Lyle, J., Otto, G., Peluso, P., Rank, D., Baybayan, P., Bettman, B., Bibillo, A., Bjornson, K., Chaudhuri, B., Christians, F., Cicero, R., Clark, S., Dalal, R., deWinter, A., Dixon, J., Foquet, M., Gaertner, A., Hardenbol, P., Heiner, C., Hester, K., Holden,

D., Kearns, G., Kong, X., Kuse, R., Lacroix, Y., Lin, S., Lundquist, P., Ma, C., Marks, P., Maxham, M., Murphy, D., Park, I., Pham, T., Phillips, M., Roy, J., Sebra, R., Shen, G., Sorenson, J., Tomaney, A., Travers, K., Trulson, M., Vieceli, J., Wegener, J., Wu, D., Yang, A., Zaccarin, D., Zhao, P., Zhong, F., Korlach, J., and Turner, S., "Real-time dna sequencing from single polymerase molecules," *Science*, vol. 323, no. 5910, pp. 133–138, 2009.

[56] Felsenstein, J., *Inferring Phylogenies*. Sinauer Associates, 2 ed., September 2003.

[57] Fichant, G. A. and Quentin, Y., "A frameshift error detection algorithm for dna sequencing projects.," *Nucleic Acids Res*, vol. 23, pp. 2900–2908, August 1995.

[58] Fleischmann, R. D., Adams, M. D., White, O., Clayton, R. A., Kirkness, E. F., Kerlavage, A. R., Bult, C. J., Tomb, J. F., Dougherty, B. A., and Merrick, J. M., "Whole-genome random sequencing and assembly of haemophilus influenzae rd," *Science (New York, N.Y.)*, vol. 269, no. 5223, pp. 496–512, 1995.

[59] Flusberg, B. A., Webster, D. R., Lee, J. H., Travers, K. J., Olivares, E. C., Clark, T. A., Korlach, J., and Turner, S. W., "Direct detection of dna methylation during single-molecule, real-time sequencing," *Nature Methods*, vol. 7, pp. 461–465, May 2010.

[60] Fraser, C., Alm, E. J., Polz, M. F., Spratt, B. G., and Hanage, W. P., "The bacterial species challenge: making sense of genetic and ecological diversity.," *Science (New York, N.Y.)*, vol. 323, pp. 741–746, February 2009.

[61] Frias-Lopez, J., Shi, Y., Tyson, G. W., Coleman, M. L., Schuster, S. C., Chisholm, S. W., and Delong, E. F., "Microbial community gene expression in ocean surface waters.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, pp. 3805–3810, March 2008.

[62] García Martín, H., Ivanova, N., Kunin, V., Warnecke, F., Barry, K. W., McHardy, A. C., Yeates, C., He, S., Salamov, A. A., Szeto, E., Dalin,

E., Putnam, N. H., Shapiro, H. J., Pangilinan, J. L., Rigoutsos, I., Kyrpides, N. C., Blackall, L. L. L., McMahon, K. D., and Hugenholtz, P., "Metagenomic analysis of two enhanced biological phosphorus removal (ebpr) sludge communities.," *Nature biotechnology*, vol. 24, pp. 1263–1269, October 2006.

[63] Gerlach, G., von Wintzingerode, F., Middendorf, B., and Gross, R., "Evolutionary trends in the genus bordetella," *Microbes and infection / Institut Pasteur*, vol. 3, no. 1, pp. 61–72, 2001.

[64] Gevers, D., Cohan, F. M., Lawrence, J. G., Spratt, B. G., Coenye, T., Feil, E. J., Stackebrandt, E., de Peer, Y. V., Vandamme, P., Thompson, F. L., and Swings, J., "Re-evaluating prokaryotic species," *Nature Reviews Microbiology*, vol. 3, pp. 733–739, August 2005.

[65] Gilbert, J. A., Field, D., Huang, Y., Edwards, R., Li, W., Gilna, P., and Joint, I., "Detection of large numbers of novel sequences in the metatranscriptomes of complex marine microbial communities.," *PloS one*, vol. 3, pp. e3042+, August 2008.

[66] Gilbert, J. A., Thomas, S., Cooley, N. A., Kulakova, A., Field, D., Booth, T., McGrath, J. W., Quinn, J. P., and Joint, I., "Potential for phosphonoacetate utilization by marine bacteria in temperate coastal waters.," *Environmental microbiology*, vol. 11, pp. 111–125, January 2009.

[67] Gill, S. R., Pop, M., Deboy, R. T., Eckburg, P. B., Turnbaugh, P. J., Samuel, B. S., Gordon, J. I., Relman, D. A., Fraser-Liggett, C. M., and Nelson, K. E., "Metagenomic analysis of the human distal gut microbiome," *Science*, vol. 312, pp. 1355–1359, June 2006.

[68] Gogarten, J. P., Doolittle, W. F., and Lawrence, J. G., "Prokaryotic evolution in light of gene transfer.," *Mol Biol Evol*, vol. 19, pp. 2226–2238, December 2002.

[69] Gotelli, N. J. and Ellison, A. M., *A Primer Of Ecological Statistics*. Sinauer Associates, 1 ed.

[70] Grant, S., Grant, W. D., Cowan, D. A., Jones, B. E., Ma, Y., Ventosa, A., and Heaphy, S., "Identification of eukaryotic open reading frames in metagenomic cdna libraries made from environmental samples.," *Applied and environmental microbiology*, vol. 72, pp. 135–143, January 2006.

[71] Guan, X. and Uberbacher, E. C., "Alignments of dna and protein sequences containing frameshift errors.," *Comput Appl Biosci*, vol. 12, pp. 31–40, February 1996.

[72] Guo, Z., Li, Y., Gong, X., Yao, C., Ma, W., Wang, D., Li, Y., Zhu, J., Zhang, M., Yang, D., and Wang, J., "Edge-based scoring and searching method for identifying condition-responsive protein protein interaction sub-network," *Bioinformatics*, vol. 23, pp. 2121–2128, August 2007.

[73] Gupta, R. S. and Griffiths, E., "Critical issues in bacterial phylogeny.," *Theoretical population biology*, vol. 61, pp. 423–434, June 2002.

[74] Handelsman, J., "Metagenomics: application of genomics to uncultured microorganisms.," *Microbiol Mol Biol Rev*, vol. 68, pp. 669–685, December 2004.

[75] Heymans, M. and Singh, A. K., "Deriving phylogenetic trees from the similarity analysis of metabolic pathways," *Bioinformatics*, vol. 19, pp. i138–146, July 2003.

[76] Hiller, N. L., Janto, B., Hogg, J. S., Boissy, R., Yu, S., Powell, E., Keefe, R., Ehrlich, N. E., Shen, K., Hayes, J., Barbadora, K., Klimke, W., Dernovoy, D., Tatusova, T., Parkhill, J., Bentley, S. D., Post, J. C., Ehrlich, G. D., and Hu, F. Z., "Comparative genomic analyses of seventeen streptococcus pneumoniae strains: insights into the pneumococcal supragenome.," *Journal of bacteriology*, vol. 189, pp. 8186–8195, November 2007.

[77] Hoff, K., Tech, M., Lingner, T., Daniel, R., Morgenstern, B., and Meinicke, P., "Gene prediction in metagenomic fragments: A large scale machine learning approach," *BMC Bioinformatics*, vol. 9, pp. 217+, April 2008.

[78] Hogg, J., Hu, F., Janto, B., Boissy, R., Hayes, J., Keefe, R., Post, J. C., and Ehrlich, G., "Characterization and modeling of the haemophilus influenzae core and supragenomes based on the complete genomic sequences of rd and 12 clinical nontypeable strains," *Genome Biology*, vol. 8, pp. R103+, June 2007.

[79] Holt, K. E., Parkhill, J., Mazzoni, C. J., Roumagnac, P., Weill, F.-X., Goodhead, I., Rance, R., Baker, S., Maskell, D. J., Wain, J., Dolecek, C., Achtman, M., and Dougan, G., "High-throughput sequencing provides insights into genome variation and evolution in salmonella typhi," *Nature Genetics*, vol. 40, pp. 987–993, July 2008.

[80] Hotopp, J. D., Grifantini, R., Kumar, N., Tzeng, Y. L., Fouts, D., Frigimelica, E., Draghi, M., Giuliani, M. M., Rappuoli, R., Stephens, D., Grandi, G., and Tettelin, H. A., "Comparative genomics of neisseria meningitidis: core genome, islands of horizontal transfer and pathogen-specific genes," *Microbiology (Reading, England)*, vol. 152, no. Pt 12, pp. 3733–3749, 2006.

[81] Huse, S., Huber, J., Morrison, H., Sogin, M., and Welch, D., "Accuracy and quality of massively parallel dna pyrosequencing," *Genome Biology*, vol. 8, no. 7, p. R143, 2007.

[82] Huse, S. M., Dethlefsen, L., Huber, J. A., Welch, D. M., Relman, D. A., and Sogin, M. L., "Exploring microbial diversity and taxonomy using ssu rrna hypervariable tag sequencing," *PLoS Genet*, vol. 4, pp. e1000255+, November 2008.

[83] Huson, D. H., Auch, A. F., Qi, J., and Schuster, S. C., "Megan analysis of metagenomic data.," *Genome Res*, vol. 17, pp. 377–386, March 2007.

[84] Ideker, T., Ozier, O., Schwikowski, B., and Siegel, A. F., "Discovering regulatory and signalling circuits in molecular interaction networks," *Bioinformatics*, vol. 18, pp. S233–240, July 2002.

[85] International, "Initial sequencing and analysis of the human genome.," *Nature*, vol. 409, pp. 860–921, February 2001.

[86] Jolley, K. A., Wilson, D. J., Kriz, P., McVean, G., and Maiden, M. C. J., "The influence of mutation, recombination, population history, and selection on patterns of genetic diversity in neisseria meningitidis," *Mol Biol Evol*, vol. 22, no. 3, pp. 562–569, 2005.

[87] Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T., and Yamanishi, Y., "Kegg for linking genomes to life and the environment.," *Nucleic acids research*, vol. 36, pp. D480–484, January 2008.

[88] Kariin, S. and Burge, C., "Dinucleotide relative abundance extremes: a genomic signature," *Trends in Genetics*, vol. 11, pp. 283–290, July 1995.

[89] Konstantinidis, K. T., Ramette, A., and Tiedje, J. M., "The bacterial species definition in the genomic era.," *Philos Trans R Soc Lond B Biol Sci*, vol. 361, pp. 1929–1940, November 2006.

[90] Konstantinidis, K. T. and Tiedje, J. M., "Genomic insights that advance the species definition for prokaryotes.," *Proc Natl Acad Sci U S A*, vol. 102, pp. 2567–2572, February 2005.

[91] Krause, L., Diaz, N. N., Goesmann, A., Kelley, S., Nattkemper, T. W., Rohwer, F., Edwards, R. A., and Stoye, J., "Phylogenetic classification of short environmental dna fragments," *Nucl. Acids Res.*, vol. 36, pp. 2230–2239, April 2008.

[92] Krogh, A., "Chapter 4 an introduction to hidden markov models for," in *Biological Sequences, Computational Methods in Molecular Biology, Elsevier*, 1998.

[93] Krogh, A., Larsson, B. A., von Heijne, G., and Sonnhammer, E., "Predicting transmembrane protein topology with a hidden markov model: application to complete genomes1," *Journal of molecular biology*, vol. 305, no. 3, pp. 567–580, 2001.

[94] Kroll, J. S., Wilks, K. E., Farrant, J. L., and Langford, P. R., "Natural genetic exchange between haemophilus and neisseria: intergeneric transfer of chromosomal genes between major human pathogens," *Proc Natl Acad Sci U S A*, vol. 95, pp. 12381–5, Oct 13 1998.

[95] Kudla, G., Murray, A. W., Tollervey, D., and Plotkin, J. B., "Coding-sequence determinants of gene expression in escherichia coli," *Science*, vol. 324, pp. 255–258, April 2009.

[96] Kunin, V., Copeland, A., Lapidus, A., Mavromatis, K., and Hugenholtz, P., "A bioinformatician's guide to metagenomics.," *Microbiology and molecular biology reviews : MMBR*, vol. 72, pp. 557–578, December 2008.

[97] Kuo, A. and Grigoriev, I., "Challenges in whole-genome annotation of pyrosequenced fungal genomes," 2009.

[98] Lane, D. J., Pace, B., Olsen, G. J., Stahl, D. A., Sogin, M. L., and Pace, N. R., "Rapid determination of 16s ribosomal rna sequences for phylogenetic analyses.," *Proceedings of the National Academy of Sciences*, vol. 82, pp. 6955–6959, October 1985.

[99] Lapierre, P. and Gogarten, J. P., "Estimating the size of the bacterial pan-genome," *Trends in Genetics*, vol. 25, pp. 107–110, March 2009.

[100] Lee, A., *U-statistics: theory and practice*. CRC Press, 1990.

[101] Lee, E., Chuang, H.-Y. Y., Kim, J.-W. W., Ideker, T., and Lee, D., "Inferring pathway activity toward precise disease classification.," *PLoS computational biology*, vol. 4, pp. e1000217+, November 2008.

[102] Li, H. and Durbin, R., "Fast and accurate long-read alignment with burrows-wheeler transform," *Bioinformatics*, vol. 26, pp. 589–595, March 2010.

[103] Liu, B., Bazinet, A., and Adelfio, M., "Comparative analysis of metabolic pathways in metagenomics."

[104] Lowe, T. M. and Eddy, S. R., "trnascan-se: a program for improved detection of transfer rna genes in genomic sequence," *Nucleic acids research*, vol. 25, no. 5, pp. 955–964, 1997.

[105] Lukashin, A. V. and Borodovsky, M., "Genemark.hmm: new solutions for gene finding," *Nucl. Acids Res.*, vol. 26, pp. 1107–1115, February 1998.

[106] Maccallum, I., Przybylski, D., Gnerre, S., Burton, J., Shlyakhter, I., Gnirke, A., Malek, J., McKernan, K., Ranade, S., Shea, T. P., Williams, L., Young, S., Nusbaum, C., and Jaffe, D. B., "Allpaths 2: small genomes assembled accurately and with high continuity from short paired reads.," *Genome biology*, vol. 10, pp. R103+, October 2009.

[107] Maiden, M., Bygraves, J., Feil, E., Morelli, G., Russell, J., Urwin, R., Zhang, Q., Zhou, J., Zurth, K., Caugant, D., Feavers, I., Achtman, M., and Spratt, B., "Multilocus sequence typing: A portable approach to the identification of clones within populations of pathogenicâmicroorganisms," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, no. 6, pp. 3140–3145, 1998.

[108] Majoros, W. H., *Methods for Computational Gene Prediction*. Cambridge University Press, 1 ed., September 2007.

[109] Mardis, E. R., "The impact of next-generation sequencing technology on genetics.," *Trends in genetics : TIG*, vol. 24, pp. 133–141, March 2008.

[110] Margulies, M., Egholm, M., Altman, W., Attiya, S., Bader, J., Bemben, L., Berka, J., Braverman, M., Chen, Y.-J., Chen, Z., Dewell, S., Du, L.,

Fierro, J., Gomes, X., Godwin, B., He, W., Helgesen, S., Ho, C., Irzyk, G., Jando, S., Alenquer, M., Jarvie, T., Jirage, K., Kim, J.-B., Knight, J., Lanza, J., Leamon, J., Lefkowitz, S., Lei, M., Li, J., Lohman, K., Lu, H., Makhijani, V., McDade, K., McKenna, M., Myers, E., Nickerson, E., Nobile, J., Plant, R., Puc, B., Ronan, M., Roth, G., Sarkis, G., Simons, J., Simpson, J., Srinivasan, M., Tartaro, K., Tomasz, A., Vogt, K., Volkmer, G., Wang, S., Wang, Y., Weiner, M., Yu, P., Begley, R., and Rothberg, J., "Genome sequencing in microfabricated high-density picolitre reactors," *Nature*, vol. 437, no. 7057, pp. 376–380, 2005.

[111] Margulies, M., Egholm, M., Altman, W. E., Attiya, S., Bader, J. S., Bemben, L. A., Berka, J., Braverman, M. S., Chen, Y.-J., Chen, Z., Dewell, S. B., Du, L., Fierro, J. M., Gomes, X. V., Godwin, B. C., He, W., Helgesen, S., Ho, C. H., Irzyk, G. P., Jando, S. C., Alenquer, M. L. I., Jarvie, T. P., Jirage, K. B., Kim, J.-B., Knight, J. R., Lanza, J. R., Leamon, J. H., Lefkowitz, S. M., Lei, M., Li, J., Lohman, K. L., Lu, H., Makhijani, V. B., Mcdade, K. E., Mckenna, M. P., Myers, E. W., Nickerson, E., Nobile, J. R., Plant, R., Puc, B. P., Ronan, M. T., Roth, G. T., Sarkis, G. J., Simons, J. F., Simpson, J. W., Srinivasan, M., Tartaro, K. R., Tomasz, A., Vogt, K. A., Volkmer, G. A., Wang, S. H., Wang, Y., Weiner, M. P., Yu, P., Begley, R. F., and Rothberg, J. M., "Genome sequencing in microfabricated high-density picolitre reactors," *Nature*, July 2005.

[112] Markowitz, V., Chen, I. M., Palaniappan, K., Chu, K., Szeto, E., Grechkin, Y., Ratner, A., Anderson, I., Lykidis, A., Mavromatis, K., Ivanova, N., and Kyrpides, N., "The integrated microbial genomes system: an expanding comparative analysis resource," *Nucl. Acids Res.*, p. gkp887, 2009.

[113] Markowitz, V. M., Ivanova, N. N., Szeto, E., Palaniappan, K., Chu, K., Dalevi, D., Chen, I.-M. M., Grechkin, Y., Dubchak, I., Anderson, I., Lykidis, A., Mavromatis, K., Hugenholtz, P., and Kyrpides, N. C., "Img/m:

a data management and analysis system for metagenomes.," *Nucleic acids research*, vol. 36, pp. D534–538, January 2008.

[114] Mavromatis, K., Ivanova, N., Barry, K., Shapiro, H., Goltsman, E., McHardy, A. C. C., Rigoutsos, I., Salamov, A., Korzeniewski, F., Land, M., Lapidus, A., Grigoriev, I., Richardson, P., Hugenholtz, P., and Kyrpides, N. C. C., "Use of simulated data sets to evaluate the fidelity of metagenomic processing methods.," *Nat Methods*, April 2007.

[115] Mchardy, A. C., Martín, H. G., Tsirigos, A., Hugenholtz, P., and Rigoutsos, I., "Accurate phylogenetic classification of variable-length dna fragments," *Nature Methods*, vol. 4, pp. 63–72, December 2006.

[116] Medvedev, P. and Brudno, M., "Maximum likelihood genome assembly.," *Journal of computational biology : a journal of computational molecular cell biology*, vol. 16, pp. 1101–1116, August 2009.

[117] Meyers, L. A., Levin, B., Richardson, A., and Stojiljkovic, I., "Epidemiology, hypermutation, within-host evolution and the virulence of neisseria meningitidis," *Proceedings. Biological sciences / The Royal Society*, vol. 270, no. 1525, pp. 1667–1677, 2003.

[118] Miller, J. R., Delcher, A. L., Koren, S., Venter, E., Walenz, B. P., Brownley, A., Johnson, J., Li, K., Mobarry, C., and Sutton, G., "Aggressive assembly of pyrosequencing reads with mates," *Bioinformatics*, vol. 24, pp. 2818–2824, December 2008.

[119] Miller, J. R., Koren, S., and Sutton, G., "Assembly algorithms for next-generation sequencing data.," *Genomics*, vol. 95, pp. 315–327, June 2010.

[120] Moon, S., Byun, Y., Kim, H.-J., Jeong, S., and Han, K., "Predicting genes expressed via -1 and +1 frameshifts," *Nucl. Acids Res.*, vol. 32, pp. 4884–4892, September 2004.

[121] Morgan, J. L., Darling, A. E., and Eisen, J. A., "Metagenomic sequencing of an in vitro-simulated microbial community.," *PloS one*, vol. 5, pp. e10209+, April 2010.

[122] Mulder, N. and Apweiler, R., "Interpro and interproscan: Tools for protein sequence classification and comparison," *Methods Mol Biol*, vol. 396, pp. 59–70, 2007.

[123] Muyzer, G., de Waal, E. C., and Uitterlinden, A. G., "Profiling of complex microbial populations by denaturing gradient gel electrophoresis analysis of polymerase chain reaction-amplified genes coding for 16s rrna.," *Appl. Environ. Microbiol.*, vol. 59, pp. 695–700, March 1993.

[124] Mdigue, C., Rose, M., Viari, A., and Danchin, A., "Detecting and analyzing dna sequencing errors: toward a higher quality of the bacillus subtilis genome sequence.," *Genome Res*, vol. 9, pp. 1116–1127, November 1999.

[125] Nacu, S., Critchley-Thorne, R., Lee, P., and Holmes, S., "Gene expression network analysis and applications to immunology," *Bioinformatics*, vol. 23, pp. 850–858, April 2007.

[126] Noonan, J. P., Coop, G., Kudaravalli, S., Smith, D., Krause, J., Alessi, J., Chen, F., Platt, D., Pääbo, S., Pritchard, J. K., and Rubin, E. M., "Sequencing and analysis of neanderthal genomic dna.," *Science*, vol. 314, pp. 1113–1118, November 2006.

[127] Not, F., Gausling, R., Azam, F., Heidelberg, J. F., and Worden, A. Z., "Vertical distribution of picoeukaryotic diversity in the sargasso sea," *Environmental Microbiology*, vol. 9, no. 5, pp. 1233–1252, 2007.

[128] Parkhill, J., Sebaihia, M., Preston, A., Murphy, L., Thomson, N., Harris, D., Holden, M., Churcher, C., Bentley, S., Mungall, K., A¡rraga, A. C. A.-T., Temple, L., James, K., Harris, B., Quail, M., Achtman, M., Atkin, R., Baker, S., Basham, D., Bason, N., Cherevach, I., Chillingworth, T., Collins, M., Cronin, A., Davis, P., Doggett, J., Feltwell, T., Goble, A.,

Hamlin, N., Hauser, H., Holroyd, S., Jagels, K., Leather, S., Moule, S., Norberczak, H., O'Neil, S., Ormond, D., Price, C., Rabbinowitsch, E., Rutter, S., Sanders, M., Saunders, D., Seeger, K., Sharp, S., Simmonds, M., Skelton, J., Squares, R., Squares, S., Stevens, K., Unwin, L., Whitehead, S., Barrell, B., and Maskell, D., "Comparative analysis of the genome sequences of bordetella pertussis, bordetella parapertussis and bordetella bronchiseptica," *Nature Genetics*, vol. 35, no. 1, pp. 32–40, 2003.

[129] Pearson, W. R., Wood, T., Zhang, Z., and Miller, W., "Comparison of dna sequences with protein sequences.," *Genomics*, vol. 46, pp. 24–36, November 1997.

[130] Perrin, A. A., Bonacorsi, S. A., Carbonnelle, E., Talibi, D., Dessen, P., Nassif, X., and Tinsley, C., "Comparative genomics identifies the genetic islands that distinguish neisseria meningitidis, the agent of cerebrospinal meningitis, from other neisseria species," *Infection and immunity*, vol. 70, no. 12, pp. 7063–7072, 2002.

[131] Pinter, R. Y., Rokhlenko, O., Yeger-Lotem, E., and Ziv-Ukelson, M., "Alignment of metabolic pathways.," *Bioinformatics*, vol. 21, pp. 3401–3408, August 2005.

[132] Pop, M., Phillippy, A., Delcher, A. L., and Salzberg, S. L., "Comparative genome assembly.," *Brief Bioinform*, vol. 5, pp. 237–248, September 2004.

[133] Posfai, J. and Roberts, R. J., "Finding errors in dna sequences.," *Proc Natl Acad Sci U S A*, vol. 89, pp. 4698–4702, May 1992.

[134] Qin, J., Li, R., Raes, J., Arumugam, M., Burgdorf, K. S., Manichanh, C., Nielsen, T., Pons, N., Levenez, F., Yamada, T., Mende, D. R., Li, J., Xu, J., Li, S., Li, D., Cao, J., Wang, B., Liang, H., Zheng, H., Xie, Y., Tap, J., Lepage, P., Bertalan, M., Batto, J.-M., Hansen, T., Le Paslier, D., Linneberg, A., Nielsen, H. B., Pelletier, E., Renault, P., Sicheritz-Ponten, T., Turner, K., Zhu, H., Yu, C., Li, S., Jian, M., Zhou, Y., Li, Y., Zhang, X., Li, S., Qin, N., Yang, H., Wang, J., Brunak, S., Dore, J., Guarner, F.,

Kristiansen, K., Pedersen, O., Parkhill, J., Weissenbach, J., Bork, P., Ehrlich, S. D., and Wang, J., "A human gut microbial gene catalogue established by metagenomic sequencing," *Nature*, vol. 464, pp. 59–65, March 2010.

[135] Quinlan, A., Stewart, D., Stromberg, M., and Marth, G., "Pyrobayes: an improved base caller for snp discovery in pyrosequences," *Nat Meth*, vol. 5, no. 2, pp. 179–181, 2008.

[136] Rasko, D. A., Rosovitz, M. J., Myers, G. S., Mongodin, E. F., Fricke, W. F., Gajer, P., Crabtree, J., Sebaihia, M., Thomson, N. R., Chaudhuri, R., Henderson, I. R., Sperandio, V., and Ravel, J., "The pangenome structure of escherichia coli: comparative genomic analysis of e. coli commensal and pathogenic isolates.," *Journal of bacteriology*, vol. 190, pp. 6881–6893, October 2008.

[137] Redon, R., Ishikawa, S., Fitch, K. R., Feuk, L., Perry, G. H., Andrews, T. D., Fiegler, H., Shapero, M. H., Carson, A. R., Chen, W., Cho, E. K., Dallaire, S., Freeman, J. L., Gonzalez, J. R., Gratacos, M., Huang, J., Kalaitzopoulos, D., Komura, D., MacDonald, J. R., Marshall, C. R., Mei, R., Montgomery, L., Nishimura, K., Okamura, K., Shen, F., Somerville, M. J., Tchinda, J., Valsesia, A., Woodwark, C., Yang, F., Zhang, J., Zerjal, T., Zhang, J., Armengol, L., Conrad, D. F., Estivill, X., Tyler-Smith, C., Carter, N. P., Aburatani, H., Lee, C., Jones, K. W., Scherer, S. W., and Hurles, M. E., "Global variation in copy number in the human genome," *Nature*, vol. 444, pp. 444–454, November 2006.

[138] Reno, M. L., Held, N. L., Fields, C. J., Burke, P. V., and Whitaker, R. J., "Biogeography of the sulfolobus islandicus pan-genome.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, pp. 8605–8610, May 2009.

[139] Richter, D. C., Ott, F., Auch, A. F., Schmid, R., and Huson, D. H., "Metasima sequencing simulator for genomics and metagenomics," *PLoS ONE*, vol. 3,

pp. e3373+, October 2008.

[140] Rissman, A., Mau, B., Biehl, B., Darling, A., Glasner, J., and Perna, N., "Reordering contigs of draft genomes using the mauve aligner," *Bioinformatics*, vol. 25, no. 16, pp. 2071–2073, 2009.

[141] Ritz, A., Bashir, A., and Raphael, B. J., "Structural variation analysis with strobe reads," *Bioinformatics*, vol. 26, pp. 1291–1298, May 2010.

[142] Rosen, G. L., Sokhansanj, B. A., Polikar, R., Bruns, M. A. A., Russell, J., Garbarine, E., Essinger, S., and Yok, N., "Signal processing for metagenomics: extracting information from the soup.," *Current genomics*, vol. 10, pp. 493–510, November 2009.

[143] Rosenstein, N. E., Perkins, B. A., Stephens, D. S., Popovic, T., and Hughes, J. M., "Meningococcal disease," *The New England journal of medicine*, vol. 344, no. 18, pp. 1378–1388, 2001.

[144] Rusch, D. B., Halpern, A. L., Sutton, G., Heidelberg, K. B., Williamson, S., Yooseph, S., Wu, D., Eisen, J. A., Hoffman, J. M., Remington, K., Beeson, K., Tran, B., Smith, H., Baden-Tillson, H., Stewart, C., Thorpe, J., Freeman, J., Andrews-Pfannkoch, C., Venter, J. E., Li, K., Kravitz, S., Heidelberg, J. F., Utterback, T., Rogers, Y.-H., Falcón, L. I., Souza, V., Bonilla-Rosso, G., Eguiarte, L. E., Karl, D. M., Sathyendranath, S., Platt, T., Bermingham, E., Gallardo, V., Tamayo-Castillo, G., Ferrari, M. R., Strausberg, R. L., Nealson, K., Friedman, R., Frazier, M., and Venter, C. J., "The sorcerer ii global ocean sampling expedition: Northwest atlantic through eastern tropical pacific," *PLoS Biology*, vol. 5, pp. e77+, March 2007.

[145] Schiex, T., Gouzy, J., Moisan, A., and de Oliveira, Y., "Framed: A flexible program for quality check and gene prediction in prokaryotic genomes and noisy matured eukaryotic sequences.," *Nucleic Acids Res*, vol. 31, pp. 3738–3741, July 2003.

[146] Schoen, C., Blom, J., Claus, H., Schramm-Glück, A., Brandt, P., Müller, T., Goesmann, A., Joseph, B., Konietzny, S., Kurzai, O., Schmitt, C., Friedrich, T., Linke, B., Vogel, U., and Frosch, M., "Whole-genome comparison of disease and carriage strains provides insights into virulence evolution in neisseria meningitidis.," *Proc Natl Acad Sci U S A*, vol. 105, pp. 3473–3478, March 2008.

[147] Schoen, C., Tettelin, H., Parkhill, J., and Frosch, M., "Genome flexibility in neisseria meningitidis.," *Vaccine*, vol. 27 Suppl 2, June 2009.

[148] Schreiber, F., Gumrich, P., Daniel, R., and Meinicke, P., "Treephyler: fast taxonomic profiling of metagenomes," *Bioinformatics*, vol. 26, pp. btq070–961, February 2010.

[149] Seshadri, R., Kravitz, S., Smarr, L., Gilna, P., and Frazier, M., "Camera: A community resource for metagenomics," *PLoS Biol*, vol. 5, no. 3, p. e75, 2007.

[150] Shaw, A. K. K., Halpern, A. L. L., Beeson, K., Tran, B., Venter, J. C. C., and Martiny, J. B. H. B., "It's all relative: ranking the diversity of aquatic bacterial communities.," *Environmental microbiology*, July 2008.

[151] Shendure, J. and Ji, H., "Next-generation dna sequencing," *Nature Biotechnology*, vol. 26, pp. 1135–1145, October 2008.

[152] Shendure, J., Porreca, G., Reppas, N., Lin, X., McCutcheon, J., Rosenbaum, A., Wang, M., Zhang, K., Mitra, R., and Church, G., "Accurate multiplex polony sequencing of an evolved bacterial genome," *Science (New York, N.Y.)*, vol. 309, no. 5741, pp. 1728–1732, 2005.

[153] Shendure, J., Porreca, G. J., Reppas, N. B., Lin, X., Mccutcheon, J. P., Rosenbaum, A. M., Wang, M. D., Zhang, K., Mitra, R. D., and Church, G. M., "Accurate multiplex polony sequencing of an evolved bacterial genome," *Science*, vol. 309, pp. 1728–1732, September 2005.

[154] Simpson, J. T. and Durbin, R., "Efficient construction of an assembly string graph using the fm-index.," *Bioinformatics (Oxford, England)*, vol. 26, pp. i367–373, June 2010.

[155] Sinha, R., Puglisi, S., Moffat, A., and Turpin, A., "Improving suffix array locality for fast pattern matching on disk," in *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, (New York, NY, USA), pp. 661–672, ACM, 2008.

[156] Snipen, L., Almoy, T., and Ussery, D., "Microbial comparative pan-genomics using binomial mixture models," *BMC Genomics*, vol. 10, pp. 385+, August 2009.

[157] Sogin, M. L., Morrison, H. G., Huber, J. A., Welch, D. M., Huse, S. M., Neal, P. R., Arrieta, J. M., and Herndl, G. J., "Microbial diversity in the deep sea and the underexplored rare biosphere," *Proceedings of the National Academy of Sciences*, vol. 103, pp. 12115–12120, August 2006.

[158] Sohler, F., Hanisch, D., and Zimmer, R., "New methods for joint analysis of biological networks and expression data," *Bioinformatics*, vol. 20, pp. 1517–1521, July 2004.

[159] Sommer, D., Delcher, A., Salzberg, S., and Pop, M., "Minimus: a fast, lightweight genome assembler," *BMC bioinformatics*, vol. 8, p. 64, 2007.

[160] Sorensen, D. and Gianola, D., *Likelihood, Bayesian and MCMC Methods in Quantitative Genetics*. Springer, March 2007.

[161] Stajich, J. E., Block, D., Boulez, K., Brenner, S. E., Chervitz, S. A., Dagdigian, C., Fuellen, G., Gilbert, J. G. R., Korf, I., Lapp, H., Lehväslaiho, H., Matsalla, C., Mungall, C. J., Osborne, B. I., Pocock, M. R., Schattner, P., Senger, M., Stein, L. D., Stupka, E., Wilkinson, M. D., and Birney, E., "The bioperl toolkit: Perl modules for the life sciences," *Genome Research*, vol. 12, pp. 1611–1618, October 2002.

[162] States, D. J. and Botstein, D., "Molecular sequence accuracy and the analysis of protein coding regions.," *Proc Natl Acad Sci U S A*, vol. 88, pp. 5518–5522, July 1991.

[163] Stewart, A., Osborne, B., and Read, T., "Diya: a bacterial annotation pipeline for any genomics lab," *Bioinformatics (Oxford, England)*, vol. 25, no. 7, pp. 962–963, 2009.

[164] Stranger, B. E., Forrest, M. S., Dunning, M., Ingle, C. E., Beazley, C., Thorne, N., Redon, R., Bird, C. P., de Grassi, A., Lee, C., Tyler-Smith, C., Carter, N., Scherer, S. W., Tavaré, S., Deloukas, P., Hurles, M. E., and Dermitzakis, E. T., "Relative impact of nucleotide and copy number variation on gene expression phenotypes.," *Science (New York, N.Y.)*, vol. 315, pp. 848–853, February 2007.

[165] Sullivan, M. B., Coleman, M. L., Weigele, P., Rohwer, F., and Chisholm, S. W., "Three prochlorococcus cyanophage genomes: signature features and ecological interpretations.," *PLoS Biol*, vol. 3, May 2005.

[166] Sun, Z., Luo, J., Zhou, Y., Luo, J., Liu, K., and Li, W., "Exploring phenotype-associated modules in an oral cavity tumor using an integrated framework," *Bioinformatics*, vol. 25, pp. 795–800, March 2009.

[167] Taylor, R. J., Siegel, A., and Galitski, T., "Network motif analysis of a multimode genetic-interaction network," *Genome Biology*, vol. 8, pp. R160+, August 2007.

[168] Teeling, H., Meyerdierks, A., Bauer, M., Amann, R., and Glöckner, F. O., "Application of tetranucleotide frequencies for the assignment of genomic fragments.," *Environ Microbiol*, vol. 6, pp. 938–947, September 2004.

[169] Teeling, H., Waldmann, J., Lombardot, T., Bauer, M., and Glöckner, F. O., "Tetra: a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in dna sequences.," *BMC Bioinformatics*, vol. 5, October 2004.

[170] Tettelin, H., Masignani, V., Cieslewicz, M., Donati, C., Medini, D., Ward, N., Angiuoli, S., Crabtree, J., Jones, A., Durkin, S., Deboy, R., Davidsen, T., Mora, M., Scarselli, M., Margarit y Ros, I., Peterson, J., Hauser, C., Sundaram, J., Nelson, W., Madupu, R., Brinkac, L., Dodson, R., Rosovitz, M., Sullivan, S., Daugherty, S., Haft, D., Selengut, J., Gwinn, M., Zhou, L., Zafar, N., Khouri, H., Radune, D., Dimitrov, G., Watkins, K., O'Connor, K., Smith, S., Utterback, T., White, O., Rubens, C., Grandi, G., Madoff, L., Kasper, D., Telford, J., Wessels, M., Rappuoli, R., and Fraser, C., "Genome analysis of multiple pathogenic isolates of streptococcus agalactiae: implications for the microbial "pan-genome"," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 39, pp. 13950–13955, 2005.

[171] Tettelin, H., Riley, D., Cattuto, C., and Medini, D., "Comparative genomics: the bacterial pan-genome.," *Current opinion in microbiology*, vol. 11, pp. 472–477, October 2008.

[172] Trapnell, C. and Schatz, M. C., "Optimizing data intensive gpgpu computations for dna sequence alignment," *Parallel Computing*, vol. 35, pp. 429–440, August 2009.

[173] Tringe, S. G., von Mering, C., Kobayashi, A., Salamov, A. A., Chen, K., Chang, H. W., Podar, M., Short, J. M., Mathur, E. J., Detter, J. C., Bork, P., Hugenholtz, P., and Rubin, E. M., "Comparative metagenomics of microbial communities," *Science*, vol. 308, pp. 554–557, April 2005.

[174] Tyson, G. W., Chapman, J., Hugenholtz, P., Allen, E. E., Ram, R. J., Richardson, P. M., Solovyev, V. V., Rubin, E. M., Rokhsar, D. S., and Banfield, J. F., "Community structure and metabolism through reconstruction of microbial genomes from the environment.," *Nature*, vol. 428, pp. 37–43, March 2004.

[175] Tyson, G. W., Chapman, J., Hugenholtz, P., Allen, E. E., Ram, R. J., Richardson, P. M., Solovyev, V. V., Rubin, E. M., Rokhsar, D. S., and

Banfield, J. F., "Community structure and metabolism through reconstruction of microbial genomes from the environment.," *Nature*, vol. 428, pp. 37–43, March 2004.

[176] Wang, Z., Chen, Y., and Li, Y., "A brief review of computational gene prediction methods.," *Genomics, proteomics & bioinformatics / Beijing Genomics Institute*, vol. 2, pp. 216–221, November 2004.

[177] Ward, B. B., "How many species of prokaryotes are there?," *Proc Natl Acad Sci U S A*, vol. 99, pp. 10234–10236, August 2002.

[178] Ward, D. M., Weller, R., and Bateson, M. M., "16s rrna sequences reveal numerous uncultured microorganisms in a natural community.," *Nature.*, vol. 345, pp. 63–65, May 1990.

[179] Warnecke, F., Luginbühl, P., Ivanova, N., Ghassemian, M., Richardson, T. H., Stege, J. T., Cayouette, M., Mchardy, A. C., Djordjevic, G., Aboushadi, N., Sorek, R., Tringe, S. G., Podar, M., Martin, H. G., Kunin, V., Dalevi, D., Madejska, J., Kirton, E., Platt, D., Szeto, E., Salamov, A., Barry, K., Mikhailova, N., Kyrpides, N. C., Matson, E. G., Ottesen, E. A., Zhang, X., Hernández, M., Murillo, C., Acosta, L. G., Rigoutsos, I., Tamayo, G., Green, B. D., Chang, C., Rubin, E. M., Mathur, E. J., Robertson, D. E., Hugenholtz, P., and Leadbetter, J. R., "Metagenomic and functional analysis of hindgut microbiota of a wood-feeding higher termite," *Nature*, vol. 450, no. 7169, pp. 560–565, 2007.

[180] Wernicke, S. and Rasche, F., "Simple and fast alignment of metabolic pathways by exploiting local diversity.," *Bioinformatics*, May 2007.

[181] Woese, C. R. and Fox, G. E., "Phylogenetic structure of the prokaryotic domain: the primary kingdoms," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 74, pp. 5088–5090, November 1977.

[182] Wooley, J. C., Godzik, A., and Friedberg, I., "A primer on metagenomics," *PLoS Comput Biol*, vol. 6, pp. e1000667+, February 2010.

[183] Woyke, T., Teeling, H., Ivanova, N. N., Huntemann, M., Richter, M., Gloeckner, F. O. O., Boffelli, D., Anderson, I. J., Barry, K. W., Shapiro, H. J., Szeto, E., Kyrpides, N. C., Mussmann, M., Amann, R., Bergin, C., Ruehland, C., Rubin, E. M., and Dubilier, N., "Symbiosis insights through metagenomic analysis of a microbial consortium.," *Nature*, vol. 443, pp. 950–955, October 2006.

[184] Woyke, T., Xie, G., Copeland, A., González, J. M., Han, C., Kiss, H., Saw, J. H., Senin, P., Yang, C., Chatterji, S., Cheng, J.-F., Eisen, J. A., Sieracki, M. E., and Stepanauskas, R., "Assembling the marine metagenome, one cell at a time," *PLoS ONE*, vol. 4, pp. e5299+, April 2009.

[185] Wright, F. and Bibb, M. J., "Codon usage in the g+c-rich streptomyces genome.," *Gene*, vol. 113, pp. 55–65, April 1992.

[186] Wu, M. and Eisen, J., "A simple, fast, and accurate method of phylogenomic inference," *Genome Biology*, vol. 9, pp. R151+, October 2008.

[187] Yang, J., Chen, L., Sun, L., Yu, J., and Jin, Q., "Vfdb 2008 release: an enhanced web-based resource for comparative pathogenomics," *Nucleic Acids Res*, vol. 36, pp. D539–42, Jan 2008.

[188] Yooseph, S., Sutton, G., Rusch, D. B. B., Halpern, A. L. L., Williamson, S. J. J., Remington, K., Eisen, J. A. A., Heidelberg, K. B. B., Manning, G., Li, W., Jaroszewski, L., Cieplak, P., Miller, C. S. S., Li, H., Mashiyama, S. T. T., Joachimiak, M. P. P., van Belle, C., Chandonia, J.-M. M., Soergel, D. A. A., Zhai, Y., Natarajan, K., Lee, S., Raphael, B. J. J., Bafna, V., Friedman, R., Brenner, S. E. E., Godzik, A., Eisenberg, D., Dixon, J. E. E., Taylor, S. S. S., Strausberg, R. L. L., Frazier, M., and Venter, J. C. C., "The sorcerer ii global ocean sampling expedition: Expanding the universe of protein families.," *PLoS Biol*, vol. 5, March 2007.

[189] Zerbino, D. and Birney, E., "Velvet: algorithms for de novo short read assembly using de bruijn graphs," *Genome Research*, vol. 18, no. 5, pp. 821–829, 2008.

[190] Zhang, K., Martiny, A. C., Reppas, N. B., Barry, K. W., Malek, J., Chisholm, S. W., and Church, G. M., "Sequencing genomes from single cells by polymerase cloning," *Nature Biotechnology*, vol. 24, pp. 680–686, May 2006.

[191] Zhenping, L., Zhang, S., Wang, Y., Zhang, X.-S. S., and Chen, L., "Alignment of molecular networks by integer quadratic programming.," *Bioinformatics (Oxford, England)*, vol. 23, pp. 1631–1639, July 2007.

[192] Zhu, W., Lomsadze, A., and Borodovsky, M., "Ab initio gene identification in metagenomic sequences.," *Nucleic acids research*, vol. 38, pp. e132+, July 2010.