Discovery and Applications of Bacterial Noncoding RNAs

Huei-Hun Elizabeth Tseng

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2012

Reading Committee:

Walter L. Ruzzo, Chair

Meredith Hullar

Martin Tompa

Program Authorized to Offer Degree:
Computer Science & Engineering

University of Washington

Abstract

Discovery and Applications of Bacterial Noncoding RNAs

Huei-Hun Elizabeth Tseng

Chair of the Supervisory Committee:
Professor Walter L. Ruzzo
Computer Science & Engineering

Noncoding RNAs (ncRNAs) are functional transcripts that do not code for proteins. Many of them play indispensible roles in the cell. For example, the ribosomal RNAs make up the ribosome that is the factory for making proteins and riboswitches bind to small metabolites in the cell and regulate gene expression. Computational discovery of ncRNAs is challenging, however, because ncRNAs evolve rapidly on the nucleotide level while preserving secondary structure. In the first part of this thesis, we develop two clustering algorithms that are robust to weak sequence homology signals and are applicable on the genomic scale. We show that both algorithms can recover most known ncRNA families and as few as 5 homologous sequences are needed to predict a strong motif.

In the second part of the thesis, we investigate whether secondary structure information improves maximum likelihood tree inference for ncRNAs. An accurate phylogenetic tree has important biological and clinical applications: it can be used to infer the function of novel organisms and understand the evolutionary history of species. We show that using structure information, a more realistic gap model, and a maximum likelihood approach improves phylogenetic tree inference.

In the third part of the thesis, we develop a method for profiling human gut microbial communities using high-throughput sequencing. Our method works on Illumina short reads and does not require assembly or taxonomic identification. We

show that it can differentiate between the gut microbiota of healthy individuals at low sequencing depth, making it a cost-effective screening tool for large population studies.

In the final part of the thesis, we use a standard additions experiment to examine sequencing bias and errors in Illumina HiSeq. We identify features associated with systematic errors and develop an error correction pipeline. We show that our method reduces base errors and produces better species diversity estimates.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank the following people: Larry Ruzzo and Meredith Hullar, who are my main advisors and have provided guidance and mentoring over the years. John Stamatoyannopoulos, who has gracially funded my research and without whom we would not have many of our valuable sequencing data. Martin Tompa, who advised me during my first year and always provided feedback and help when I needed it. Johanna Lampe and all her lab members, who are the folks who do real scientific research on real people. Zizhen Yao and Zasha Weinberg, whose works I heavily depend on in my ncRNA research. Lindsay Michimoto, the goddess of the UW CSE graduate program. All the grads from my year—the "never-grads"—who are all awesomely smart people. Ian Simon, because he provided massages to incentivize thesis writing. Finally, my mom, who I think did not really know what a CSE PhD was good for, but nevertheless told me to come to the US for it.

DEDICATION

to Miska, who died on May 25th, 2012.

INTRODUCTION

In all living organisms, the genetic information is carried in DNA. We can think of DNA as a string over four letters (*nucleotides*): A (*adenine*), G (*guanine*), C(*cytosine*), and T (*thymine*). Usually, DNA exists as a double helix, where two opposing strands of DNA are held together through hydrogen bonds between complementary bases (A pairs with T, C pairs with G). RNA can also be thought of as a string over four letters: the same A, G, C as DNA, plus U (*uracil*). Unlike DNA, RNA is single stranded and can form complex secondary and tertiary structures on its own through the same hydrogen bonding of complementary bases.

To define noncoding RNA (ncRNA), we first describe how proteins, the main workhorse of cell functions, are made: DNA is first transcribed by RNA polymerase into messenger RNAs (mRNA). Then ribosomes, which consist of two subunits of ribosomal RNAs (rRNA) and proteins, are joined by transfer RNA (tRNA) in the cytoplasm to translate mRNA into peptide chains that ultimately fold into proteins. For many years, RNA was considered to have mainly the three flavors described above: mRNA, tRNA, and rRNA. We now know this is not the case. Many RNAs are transcribed from DNA into single strand RNA, remain untranslated, and serve important functions in the cell. These ncRNAs are found in both eukaryotes and prokaryotes, although few families have been found to exist across both domains of life. In humans, ncRNAs regulate transcription, DNA methylation, transposons, and gene silencing [22]. In bacteria, ncRNAs are involved in the regulation of metabolism, signaling, and mobility [9, 18]. As sequencing and computational technologies advance, it is apparent that the discovery and understanding of novel ncRNAs is an important topic. Not only could they become potential therapeutic targets for diseases, their discoveries spark the interesting hypothesis that ncRNAs might be relics from an ancient RNA world [15, 116].

In the first part of my thesis, I focus on noncoding RNAs in bacteria—specifically, ncRNAs that function through conserved secondary structures. I describe computational methods that have been developed to model and predict conserved structures and present my contribution to finding candidate ncRNA sequences. I then focus on the structural aspect of ncRNAs and develop a method for incorporating secondary structure information in phylogenetic tree estimation.

The second part of my thesis focuses on another emerging area of research: analysis of microbial communities in the human gut using high-throughput methods. Up until recently, Sanger sequencing or tRFLP (terminal restriction fragment length polymorphism [49]) were used to characterize bacteria from an environmental sample (e.g., human gut, soil). The former produces long accurate sequences but is expensive and yields at most hundreds of sequences per sample. The latter is cost-efficient, can be massively applied to lots of samples, but produces only a "fingerprint" (fragment length after enzyme digestion) of the community and not species-level information. With high-throughput sequencing that produces short reads at millions of reads per run, we can now answer the following questions with higher coverage: What and how many species are in the human gut? What is their relationship with the host diet and health? Before we can answer them with confidence, however, we need to know what biases and errors exist in high-throughput sequencing. In this thesis, I review the advances in the field, present my findings on our own data, and propose computational methods that can accurately characterize human gut microbial samples.

In Chapter 1, I describe a computational pipeline for discovering ncRNAs in bacteria. I compare two clustering approaches: quasi-clique finding and hierarchical clustering, and show that both resulted in similarly good ncRNA motif predictions. In Chapter 2, I tackle another aspect of ncRNAs: does using secondary structure information improve phylogenetic tree estimation for ncRNAs? I describe a maximum likelihood tree estimation approach (Gapped Base Paired Maximum Liklihood, GBPML) and show that for ncRNAs, using secondary structure information and a more realistic gap model improves tree estimation. In Chapter 3, I develop a pro-

filing method (Microbial Nucleotide Signature, MNS) for characterizing human gut microbiota. I show that our method worked on very short reads at low sequencing depth and differentiated the microbiota of different healthy individuals. Finally, in Chapter 4, I look at sequencing biases and errors present in Illumina short reads and propose an error correction method (ErrCor) that reduced base errors in our dataset.

Chapter 1

DISCOVERING BACTERIAL NCRNAS THROUGH GENOMIC SCALE CLUSTERING

1.1 Introduction

Noncoding RNAs (ncRNAs) are transcripts that have a functional role other than coding for proteins. Recent discoveries have revealed an increasing number of ncRNA families that take part in a wide range of regulatory roles [80, 103]. Of particular interest to us are riboswitches, ncRNAs that regulate gene expression through binding of metabolites, including amino acids (glycine, lysine), nucleobases, biosynthesis intermediates (PreQ1), coenzymes (AdoCbl, TPP, FMN), and metal ions (moco) (Table 1.1). Riboswitches regulate mainly through conformational changes that cause either transcription termination or translation inhibition, but can exert more complex control through use of combined riboswitches. The discoveries of such ncRNAs not only reveal new complexity in the regulatory networks, but also accentuate their potential as disease markers and therapeutic agents [9].

De novo ncRNA discovery is difficult because, unlike protein families which are often highly conserved in primary sequence, homologous ncRNAs can evolve rapidly on the nucleotide level while preserving secondary structure (for example, see the TPP riboswitch alignment in Figure 2.6). A computationally driven approach has been to make use of the conserved secondary structures among homologous ncRNAs: if there is evidence that a group of sequences are homologous, one can look for conservation in secondary structure.

How do we find these groups of homologous sequences? There are two main approaches: a gene-oriented approach and an intergenic region-based approach. In the gene-oriented approach, sequences are grouped based on whether they are upstream of homologous genes or correspond to homologous regions by whole genome align-

ment. Yao et al. clustered upstream regions of conserved bacterial protein genes [111] and discovered 29 novel ncRNA motifs [102]. Torarinsson et al. looked for conserved secondary structures in the UCSC 17-way vertebrate alignment and found thousands of candidate ncRNAs [96]. While both approaches have been successful, functional annotation and whole genome alignment are stringent requirements, and this approach misses ncRNAs that either regulate multiple classes of genes or are not *cis*-regulatory.

The intergenic region-based approach—the approach we use here—is to use BLAST against the entire intergenic regions (IGRs) of bacterial genomes to identify homologous sequences. BLAST is poor at detecting remote RNA homologs, with performance dropping sharply as sequence similarity falls below 60% [27]. Unfortunately, most RNA families of interest have sequence identities of 50% or less (Table 1.1). To deal with the patchy nature of weak BLAST matches, we need a method for creating clusters of homologous sequences.

Hierarchical clustering is a method for forming a hierarchy of clusters. There are generally two approaches: (1) agglomerative ("bottom up"), which starts with every sequence being in a cluster by itself and at each step merges a pair of clusters, moving one step up the hierarchy; and (2) divisive ("top down"), which starts with all sequences in one cluster and at each step divides a cluster into smaller subclusters. Both approaches require the update of similarity scores between clusters at each step, and three common choices are complete (least similar pair between the two clusters), single (most similar), and average (averaging all pairs of similarities between two clusters). In this study, we select average agglomerative clustering.

An alternative clustering method is maximal quasi-clique finding. In graph theory, a clique is a subset of vertices fully connected with each other. A clique is maximal if the clique cannot be extended by adding one more vertex. Finding ncRNA clusters is, in many ways, analogous to finding cliques in a graph: every sequence (vertex) in the cluster (clique) is homologous (connected) to all others, and by including as many sequences as possible (maximal), we increase the likelihood of accurate secondary structure prediction. Since BLAST-inferred homology is patchy, instead

of finding fully connected cliques, we look for densely connected quasi-cliques. A quasi-clique is defined as a subset of q vertices Q such that each vertex is connected to at least $\gamma(q\text{-}1)$ other vertices in Q. By setting γ to a high fraction, e.g., 0.8 or 0.9, we expect the majority but not necessarily all homologous sequences are identified through BLAST. Quasi-clique finding has some theoretical advantages over hierarchical clustering: (1) it still takes advantage of BLAST's efficiency for finding pairs of homologous sequences but does store the scores (the edges are unweighted), reducing memory cost; (2) it is easily parallelizable, whereas hierarchical clustering is not.

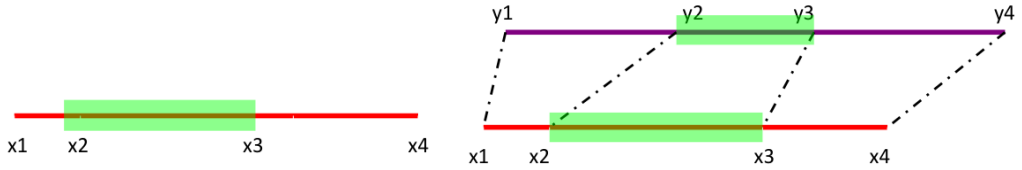
In this chapter, we compare the performance of using average-linkage hierarchical clustering with the maximal quasi-clique finding approach. We show that while both methods predict good motifs for the majority of our ncRNA families, there is still much room for improvement.

1.2 Methods

Our ncRNA discovery pipeline is as follows. First, intergenic regions are extracted from bacterial genomes then BLASTed against one another. We then parse the BLAST output into a homology graph where vertices are intergenic sequences and edges indicate homology (section 1.2.1). Only vertices with BLAST scores above a threshold have a connecting edge. For quasi-clique, the scores are ignored and the edges unweighted; for hierarchical clustering, the scores are kept to determine which nodes to merge at each iteration. We end up with a large undirected graph that is sparse globally because most sequences are unrelated, but dense locally where homologous clusters form. We apply either hierarchical clustering or maximal quasi-clique finding to output clusters of homologous sequences, then predict secondary structure from the clusters using CMFinder [112]. The goodness of the predicted secondary structures are evaluated by using them to scan sequence databases containing both genuine and shuffled ncRNA sequences.

Family	Avg. length	Indel fraction	Seq identity	Description	Ref
23S-methyl	99	0.16	0.59	Hypothetical regulator of 23S rRNA	[102]
6S	198	0.54	0.36	Supression of α^{70}-dependent promoter	[7]
AdoCbl	199	0.78	0.36	Cobalamin (B-12) riboswitch	[69]
Bacillus-plasmid	61	0.23	0.61	Hypothetical plasmid regulator	[104]
coccus-1	109	0.58	0.48	Hypothetical RNA element	[102]
crcB	70	0.69	0.46	Hypothetical DNA repair gene riboswitch	[104]
COG4708	89	0.26	0.60	Hypothetical RNA element	[102]
epsC	119	0.13	0.70	Hypothetical regulator of EAR antitermination	[104]
FMN	143	0.77	0.53	FMN riboswitch	[108]
GEMM	76	0.66	0.51	Cyclic di-GMP riboswitch	[102]
glmS	187	0.66	0.42	Glucosamine-6-phosphate activated riboswitch	[58]
glycine	178	0.79	0.52	Glycine riboswitch	[56]
group-II-D1D4-3	173	0.59	0.48	Group II catalytic intron D1-D4-3	
group-II-D1D4-5	200	0.51	0.50	Group II catalytic intron D1-D4-5	
group-II-intron.fake	81	0.52	0.40	Group II catalytic intron	
L17DE	74	0.46	0.51	Regulation of the L17 ribosomal gene	[104]
lactis-plasmid	107	0.05	0.82	Hypothetical plasmid regulator	[104]
lacto-1	106	0.18	0.47	Unknown function	[102]
lacto-2	75	0.28	0.52	Unknown function	[102]
Lacto-int	108	0.28	0.60	Unknown function	[104]
Lacto-rpoB	54	0.13	0.51	Hypothetical RNA polymerase regulator	[104]
leu-phe-leader	162	0.09	0.68	Ribosomal leader for leucine or phenylalanine.	[104]
lysine	181	0.44	0.43	Lysine riboswitch	[90]
moco	138	0.68	0.41	Moco (and Tuco) riboswitch	[102]
OLE	599	0.26	0.55	Regulation of membrane associated genes	[76]
pan	88	0.39	0.40	Hypothetical pantothenate riboswitch	[104]
pfl	97	0.76	0.44	Hypothetical riboswitch	[104]
preQ1-I	69	0.25	0.47	PreQ1 riboswitch	[82]
purine	101	0.11	0.52	Purine riboswitch	[57]
RNaseP_a	298	0.79	0.50	Ribozyme; maturation of tRNA 5' ends	
RNaseP_b	354	0.69	0.39	Ribozyme; maturation of tRNA 5' ends	
rpsB	90	0.70	0.40	Unknown function	[64]
SAM-I	124	0.57	0.49	S-denosylmethionine (SAM) sensing riboswitch	[67]
SAM-III	181	0.64	0.30	S-denosylmethionine (SAM) sensing riboswitch	[67]
SRP	99	0.30	0.43	SRP	
tbox	231	0.77	0.37	T-box leader	[29]
tmRNA	361	0.61	0.38	transfer-messenger RNA	
TPP	97	0.79	0.46	Thiamine pyrophospate (TPP) riboswitch	
ydaO	168	0.68	0.39	Hypothetical ydaO/yuaA riboswitch	[8]
yjdF	110	0.50	0.55	Unknown function	[104]
ykkC	145	0.41	0.44	Hypothetical ykkC/yxkD riboswitch	[104]
ykoK	179	0.24	0.53	Magnesium ion transport gene riboswitch	[6]
ylbH	158	0.27	0.59	Unknown function	[6]
yybP	128	0.68	0.34	Hypothetical yybP/ykoY riboswitch	[6]

Table 1.1: **Summary of ncRNAs used in this study.** Average sequence lengths, fraction of indels (gaps) and pairwise sequence identities are based on curated alignments provided by the Breaker Lab and may have slight differences with Rfam annotations. References for well-characterized ncRNAs are omitted. We selected these families because they are represented in Firmicutes—the phylum we evaluate our methods on.



(a) Case 1 (b) Case 2

Figure 1.1: **Merging nodes.** (a) Segments $x1$-$x3$ and $x2$-$x4$ overlap significantly, so they are merged into a single node representing $x1$-$x4$. (b) Segment $x1$-$x3$ has a homologous hit with segment $y1$-$y3$, and $x2$-$x4$ with $y2$-$y4$, the result is two nodes, one representing $x1$-$x4$, and one for $y1$-$y4$.

1.2.1 Homology graph construction

We extract intergenic regions (IGRs) from bacterial genomes using RefSeq annotations and create the homology graph using WU-BLAST [30][1]. WU-BLAST is run with parameters chosen to maximize sensitivity:

```
-W 3 -E 2 -M 8 -N -7 -Q 16 -R 2 -noseqs -mformat 2
```

which corresponds to match score +8, mismatch score -7, gap open penalty -16, gap extend penalty -2, and seed length 3. We empirically chose the parameters based on BLAST hits to real and shuffled ncRNA sequences (Appendix A). An E-value cutoff of 2 roughly translates to a (normalized) WU-BLAST score of 25 given the match/mismatch and gap open/extend scoring scheme, well below our actual score cutoff used during the homology graph construction. For quasi-clique finding, the homology graph is constructed using a BLAST score cutoff of 35; for hierarchical clustering, we use 50. With a BLAST score of 35, the sequences are just as likely to be random as they are distant homologs, but with a high γ (quasi-clique connectivity), the quasi-clique algorithm would exclude most of the non-homologous vertices from a homologous cluster.

[1]WU-BLAST was acquired by Advanced Biocomputing, LLC in 2009 and is now available commercially as AB-BLAST.

We process homology hits into nodes and edges. A node represents an IGR segment (a subsequence of the intergenic region) and an edge represents a homology hit. Edge weights are initially defined as homology bit scores. For all homology output, hits with bit scores below threshold (35 for quasi-clique, 50 for hierarchical clustering) are ignored. Homology hits often capture portions of ncRNA motifs, not full motifs. To counteract this undesired fragmentation of the IGRs, we merge nodes in two cases: (1) Two nodes representing segments from the same IGR that overlap by more than 70% of the smaller of two lengths (Figure 1.1a); (2) Two segments of IGR x overlap and two segments of IGR y also overlap, the overlap between neither pair is significant, but the corresponding x-y pairs are joined by homology edges (Figure 1.1b). In practice, we rarely observe case 2. The merged nodes are connected with an edge whose weight is calculated by combining the two original edge weights in proportion to segment lengths. In hierarchical clustering, the edge weights are used; in maximal quasi-clique finding, the edge weights are ignored.

1.2.2 Hierarchical clustering

For hierarchical clustering, we use WPGMA (Weight Pair Group Method using Arithmetic averaging), also known as average-linkage clustering. At each step, the nearest two clusters are combined into a higher-level cluster. Distances between the new merged cluster and other nodes are computed using edge weights weighted by cluster sizes.

1.2.3 Maximal quasi-clique finding

Formally, a graph is denoted by $G = (V, E)$ where V is the set of vertices and E is the set of edges. For the purpose of finding homologous clusters, we do not put weights or directionality on the edges, so $(u,\ v)$ and $(v,\ u)$ both refer to the same undirected edge connecting vertices u and v. Let Q be a subset of V, then $deg_Q(u)$=$|\{(u, v) \mid v \in Q \backslash \{u\}, (u, v) \in E\}|$ is the degree of u in $S(G)$, the subgraph induced by Q. Q is a perfect clique if $S(G)$ is fully connected. Q is a γ-quasi-clique

if $\forall u \in Q, \deg_Q(u) \geq \gamma(|Q| - 1)$. Note that when γ=1, Q is again a perfect clique. Q is a maximal γ-quasi-clique if no vertices outside Q can be added such that it is still a γ-quasi-clique.

The problem of finding maximal cliques is NP-hard, and there have been many proposed heuristic algorithms for finding a near-optimal solution efficiently. Abello et al. [1] developed a heuristic algorithm to find the largest possible clique (maximum clique) in a large directed graph. We modified the algorithm to find as many maximal quasi-cliques as possible in an undirected graph using the same greedy randomized adaptive search (GRASP) technique. For each randomly selected starting seed vertex s, we iteratively construct a randomized perfect clique then try to expand the clique by bringing in nearby vertices. Because GRASP is a greedy randomized heuristic, it isn't guaranteed to always find the optimal solution. To explore the search space and increase the chance of finding the maximal quasi-clique containing s, we repeat our modified GRASP for many iterations and keep the largest quasi-clique found as the final solution.

1.2.3.1 GRASP algorithm for maximal clique finding

Algorithm 1.2.1 shows the modified GRASP procedure `QuasiGrasp` that consists of three procedures: `construct` (Algorithm 1.2.2), `local-pair` (Algorithm 1.2.3), and `local-single` (pseudocode not shown as it is the same as `local-pair` with minor modifications). First `construct` grows a perfect clique Q randomly and greedily given the starting vertex s. At first, the clique Q contains just s, and C is the set of candidate vertices adjacent to s. A restricted candidate list (RCL) contains the candidate vertices that have high degrees in the subgraph induced by C. Because we want to grow to as large a clique as possible, we select vertices from RCL to grow Q since there's a higher chance that these vertices are members of the maximal clique for s. Once a new vertex t is added to Q, C is updated by eliminating all vertices that are not connected to t and RCL is recomputed. The procedure ends when no more vertices can be added to Q. `construct` is random and greedy because when

there is more than one way to grow the clique around s, it randomly selects the next vertex to include in the clique but favors those with higher connectivity. In the next phase (`local-pair`), Q is expanded by continuously swapping a pair of connected vertices (v, u) that is not in Q with a vertex w in Q if $Q \bigcup (v, u) \backslash \{w\}$ still has an edge density $> \gamma$. Each swap increases the clique size by 1, potentially growing Q towards the maximal clique. Note that during `local-pair`, we immediately discard the results and start with another seed vertex if s is swapped out. When no swapping is possible, the final phase (`local-single`) looks for vertices outside Q that could be brought in to maximize the clique size while maintaining γ. `local-single` was not used in [16] because they only implemented it for perfect cliques and it is not possible to bring in more vertices after the local swapping. Since GRASP is heuristic, to escape from poor local solutions, the sub-procedures are run for a number of times (*maxitr*) to explore the search space. In this study, we set *maxitr* = 20.

1.2.3.2 Additional heuristics to reduce memory cost

Because the entire homology graph G is too large and does not fit into memory, we store the graph externally in a MySQL database table (though this could easily be stored in other formats and systems). We assign each vertex a unique integer ID, and store the edges as $(id1, id2)$ where $id1 < id2$. Now consider what vertices we have to consider as potential members of the maximal clique for a given vertex s. If the maximal quasi-clique for s is Q^* and γ is high enough, then every vertex u in Q^* must be either connected to s (denoted 1-adjacent) or is connected to some 1-adjacent vertex (2-adjacent). Thus, to find the maximal quasi-clique of s, we only consider a subgraph of G induced by s and the set of 1-adjacent and 2-adjacent vertices. In reality the set of 2-adjacent vertices is very large, so to reduce memory, we exclude from the subgraph all edges between 2-adjacent vertices. This can result in poorer quasi-cliques, which we compensate by setting a two-tier γ setting. We first set γ to a relatively high threshold (in this study we chose 0.8) and run `QuasiGrasp` on the induced subgraph. After `QuasiGrasp`, we have a quasi-

clique Q that consists of highly interconnected vertices. We run a single iteration of `local-single` on Q with a lower γ (in this study we chose 0.6) to allow more vertices, especially 2-adjacent vertices, to be added without swapping out vertices already in Q. We are interested in finding as many maximal quasi-cliques as we can from G. Since the graph is very large, we can perform parallel local runs of `QuasiGrasp` on different seed vertices. Whenever a local run finds a maximal quasi-clique, it deletes all of the vertices from G. It is possible that by randomly choosing seed vertices, two parallel runs produce overlapping cliques. Extra checking procedures can easily be implemented to detect this, but we did not find the need for it.

Algorithm 1.2.1: QUASIGRASP$(V, E, seed_node, \gamma, maxitr)$

$Q^* = \emptyset$

for $k = 1, 2, \ldots maxitr$

do
- Select α, at random, from interval [0, 1]
- $Q = construct(V, E, seed\_node, \alpha)$
- $Q = local - pair(V, E, Q, \gamma)$
- $Q = local - single(V, E, Q, \gamma)$
- **if** $|Q| > |Q^*|$
  - **then** $Q^* = Q$

return $(Q*)$

Algorithm 1.2.2: CONSTRUCT$(V, E, seed_node, \alpha)$

Set initial clique $Q = \{seed\_node\}$

Set C = neighbors of $seed\_node$

while $|C| > 0$

do
- Let $G(C)$ = subgraph induced by vertices in C
- Let $\deg_{G(C)}$(u) = degree of $u \in C$ with respect to $G(C)$
- $d_{min} = min\{deg_{G(C)}(u) \mid u \in C\}$
- $d_{max} = max\{deg_{G(C)}(u) \mid u \in C\}$
- $RCL = \{u \in C \mid deg_{G(C)}(u) \geq d_{min} + \alpha(d_{max} - d_{min})\}$
- Select u at random from RCL
- $Q = Q \bigcup \{u\}$
- $C = \{v \in C \mid (u, v) \in E\}$

return (Q)

Algorithm 1.2.3: LOCAL-PAIR(V, E, Q, γ)

$H = \{(v, u, w) \mid v, u, w \in V, (v, u) \in E, w \in Q$ and $Q \bigcup \{u, v\} \backslash \{w\}$ is still a γ-quasi-clique$\}$

while $|H| > 0$

do
- Select $(v, u, w) \in H$
- $Q = Q \bigcup \{u, v\} \backslash \{w\}$
- $H = \{(v, u, w) \mid v, u, w \in V, (v, u) \in E, w \in Q$ and $Q \bigcup \{u, v\} \backslash \{w\}$ is still a γ-quasi-clique$\}$

return (Q)

1.2.4 Motif prediction and scanning

To predict motifs from clusters, we run CMfinder [112] and use the same scripts from Weinberg et al. [102] to remove duplicate sequences and combine motifs. Briefly, CMfinder folds each sequence in the input set, and constructs an initial heuristic alignment attempting to match similar sequence and structural features. Next, it builds a covariance model (CM) from the alignment, exploiting both mutual information and single-sequence structure predictions to arrive at a consensus structure prediction. Finally, it performs an EM-like iteration, alternately realigning the sequences to the model and rebuilding the model from the refined alignment. It is robust to non-motif containing sequences and extraneous regions flanking the motifs. For each input dataset, CMfinder predicts zero or more motifs. We take all predicted ncRNA motifs and use their covariance models (CMs) to do a CM scan on our ncRNA dataset to see how well the motifs can recover known ncRNAs. For each ncRNA family, we create a database that has all the sequences from that family and 99 times more di-shuffled (shuffling that preserves di-nucleotide frequencies) sequences. CM scan is done using Infernal 1.0 [71] with parameter `-fil-hmm-T=10` for speedup. Only CM scan hits with scores >= 15 are considered. Any significant CM hit to a real ncRNA sequence is a true positive (TP) and a hit to one of the

di-shuffled sequences is a false positive (FP). To select the motif that had the best CM scan result, we calculate the Matthews Correlation Coefficient (MCC)=$(TP \times TN - FP \times FN)/\sqrt{((TP+FP)(TP+FN)(TN+FP)(TN+FN))}$ for each motif scan result.

1.3 Results

We applied our pipeline to a dataset of 246 Firmicutes genomes. RefSeq version 25 was used for intergenic sequence extraction. The total size of the extracted intergenic sequences was 25.5 Mbp. The resulting homology graph consisted of 119,646 nodes and 24,098,465 edges. We used 10 parallel processes to run the quasi-clique finding algorithm (our implementation of hierarchical clustering is not parallelizable). Because cliques containing too few sequences will not have enough information to sufficiently predict secondary structure, we discarded all cliques of size < 5.

1.3.1 Cluster evaluation

We labeled clusters in the following manner: A sequence is labeled as an F-sequence if it overlaps with a ncRNA sequence from family F. A cluster is an F-cluster if the majority ($> 50\%$) of the sequences are F-sequences. Table 1.2 summarizes the clusters for each ncRNA family. With the exception of the widespread T-box leader (tbox), both methods produced 1-5 clusters per ncRNA family. We were surprised to find that quasi-clique finding did not do better than hierarchical clustering in many cases. A potential explanation for this is that BLAST did a good job at picking up homologous sequences and the simplifications made in quasi-clique to make it fast (bringing in only at most 2-adjacent nodes) plus the random starting node for growing the clique may have harmed its performance.

1.3.2 CM scan results

Since our goal is to detect novel ncRNA families, it matters less how many good ncRNA clusters we get per ncRNA family—all we need is one ncRNA cluster that

Family	Size	Quasi-Clique Clustering		Hierarchical Clustering	
		Clusters	*Avg. Spec.*	Clusters	*Avg. Spec.*
23S-methyl	23	3	0.83	2	0.95
6S	73	5	0.86	5	1.00
AdoCbl	94	1	0.99	2	0.99
Bacillus-plasmid	18	1	0.70	2	0.79
coccus-1	114	8	0.72	5	0.85
COG4708	10	1	0.80	1	0.89
crcB	34	2	1.00	3	1.00
epsC	7	1	0.88	1	0.83
FMN	99	1	0.99	1	0.99
GEMM	76	1	1.00	2	0.96
glmS	37	1	1.00	2	1.00
glycine	44	2	0.73	3	0.77
group-II-D1D4-3	53	4	0.77	1	0.90
group-II-D1D4-5	10	1	0.80	1	0.69
group-II-intron.fake	62	4	0.86	3	1.00
L17DE	21	2	0.80	2	0.90
lactis-plasmid	11	1	0.83	1	0.79
lacto-1	36	1	0.90	2	0.92
lacto-2	91	5	0.74	3	0.74
Lacto-int	15	1	0.75	3	0.67
Lacto-rpoB	22	3	0.62	2	0.72
leu-phe-leader	9	1	0.83	1	1.00
lysine	98	5	0.99	6	0.99
moco	40	2	1.00	2	1.00
OLE	16	1	1.00	1	1.00
pan	22	1	0.89	1	0.88
pfl	8	1	1.00	1	1.00
preQ1-I	43	4	0.85	3	0.81
purine	118	5	0.96	6	0.85
RNaseP_a	13	1	1.00	1	1.00
RNaseP_b	42	3	0.85	1	0.97
rpsB	20	2	0.86	2	0.67
SAM-I	199	6	0.90	3	0.96
SAM-III	27	3	0.73	3	0.94
SRP	64	2	0.89	2	1.00
tbox	1013	63	0.92	55	0.91
tmRNA	65	3	0.78	1	1.00
TPP	173	6	0.97	5	0.98
ydaO	59	1	1.00	2	1.00
yjdF	23	1	1.00	1	0.91
ykkC	30	2	0.90	2	0.93
ykoK	46	1	1.00	2	1.00
ylbH	11	1	0.92	1	0.91
yybP	80	5	0.88	3	0.98

Table 1.2: **Cluster evaluation for quasi-clique versus hierarchical clustering.** For each ncRNA family F and method, we show the number of F-clusters and the average cluster specificity (percentage of cluster sequences that are F-sequences). The method with the higher average specificity is highlighted in each row.

Family	Size	Quasi-Clique			Hierarchical Clustering		
		Sens.	*Spec.*	*MCC*	*Sens.*	*Spec.*	*MCC*
23S-methyl	30	0.77	0.96	0.86	0.77	1.00	0.87
6S	1226	0.50	0.93	0.68	0.40	0.96	0.62
AdoCbl	2654	0.94	1.00	0.97	0.93	0.99	0.96
Bacillus-plasmid	31	0.97	1.00	0.98	1.00	1.00	1.00
coccus-1	206	0.94	0.99	0.97	0.94	1.00	0.97
crcB	360	0.77	0.96	0.86	0.71	0.98	0.83
COG4708	15	0.93	1.00	0.97	0.93	1.00	0.97
epsC	30	1.00	1.00	1.00	1.00	1.00	1.00
FMN	785	1.00	1.00	1.00	1.00	1.00	1.00
GEMM	558	0.97	0.97	0.97	0.99	0.96	0.97
glmS	219	1.00	1.00	1.00	1.00	1.00	1.00
glycine	1531	0.95	0.96	0.96	0.99	1.00	0.99
group-II-D1D4-3	359	0.96	0.99	0.98	0.56	0.92	0.71
group-II-D1D4-5	103	1.00	0.99	1.00	1.00	0.94	0.97
group-II-intron.fake	2132	0.96	0.99	0.98	0.97	0.97	0.97
L17DE	60	0.80	0.70	0.74	0.95	0.86	0.90
lactis-plasmid	14	1.00	1.00	1.00	1.00	1.00	1.00
lacto-1	81	0.96	1.00	0.98	0.86	1.00	0.93
lacto-2	161	0.98	0.96	0.97	0.90	0.99	0.94
Lacto-int	54	0.89	1.00	0.94	1.00	0.98	0.99
Lacto-rpoB	38	0.53	1.00	0.72	0.71	0.96	0.83
leu-phe-leader	9	1.00	1.00	1.00	1.00	1.00	1.00
lysine	438	0.89	0.98	0.93	0.89	0.99	0.94
moco	259	0.93	0.98	0.95	0.90	1.00	0.95
OLE	38	1.00	1.00	1.00	1.00	1.00	1.00
pan	94	0.60	0.89	0.73	0.54	0.94	0.71
pfl	200	0.37	0.94	0.58	0.42	0.84	0.59
preQ1-I	129	0.76	0.82	0.79	0.71	0.86	0.78
purine	304	0.99	0.99	0.99	0.99	0.98	0.99
RNaseP_a	3101	0.98	1.00	0.99	0.98	1.00	0.99
RNaseP_b	1162	0.78	0.97	0.87	0.55	0.98	0.73
rpsB	994	0.23	0.93	0.45	0.12	0.60	0.26
SAM-I	908	0.97	1.00	0.98	0.96	0.98	0.97
SAM-III	44	0.59	0.90	0.73	0.82	0.78	0.80
SRP	2069	0.83	0.92	0.87	0.96	1.00	0.98
tbox	3516	0.96	1.00	0.98	0.96	1.00	0.98
tmRNA	1798	0.97	1.00	0.99	0.92	1.00	0.96
TPP	2813	1.00	0.99	1.00	0.98	0.99	0.99
ydaO	324	1.00	0.96	0.98	0.98	1.00	0.99
yjdF	89	1.00	1.00	1.00	1.00	1.00	1.00
ykkC	226	0.95	0.96	0.96	0.94	0.93	0.94
ykoK	161	1.00	1.00	1.00	1.00	1.00	1.00
ylbH	27	1.00	1.00	1.00	1.00	1.00	1.00
yybP	797	0.53	0.96	0.72	0.20	0.98	0.44

Table 1.3: **CM scan evaluation for quasi-clique versus hierarchical clustering.** For each ncRNA family and method, we list the CM scan result with the highest Matthews Correlation Coefficient (MCC). The method with the higher MCC is colored in each row.

can be used to predict a good ncRNA motif. We ran CMfinder on all ncRNA clusters. For each cluster input, CMfinder predicts zero or more motifs. For each predicted motif (represented by a covariance model, CM), we run CM scan on a synthetic database that consists of all real F-sequences (not limited to just Firimicutes) and di-shuffled them 99 times. In other words, if the predicted motif is not representative, it would result in low sensitivity (TP/(TP+FN)) or specificity (TP/(TP+FP)) in the scans. For each ncRNA family, we output the motif that resulted in the highest Matthews Correlation Coefficient (MCC). We found that both quasi-clique and hierarchical clustering resulted in good motifs (Table 1.3). The best AdoCbl riboswitch motifs came from a 68/69 (TP/FP) cluster for quasi-clique and a 75/77 cluster for hierarchical clustering, respectively. More impressively, the best purine motif came from a 5/6 cluster for quasi-clique (for hierarchical clustering, it was a 55/56 cluster). Both motifs represented the major multi-stem in the real purine riboswitch (Figure 1.2) with slight differences. The hierarchical clustering motif did not contain the unpaired 5' and 3' region and only captured the major multi-stem and as a result, the CM scans consistently recovered positions ∼19-87 of the real purine sequences. The quasi-clique motif also missed the unpaired 5' and 3' regions and also had an extra unpaired loop in the beginning of the 5' region. The extra loop appears relatively unconserved and did not affect the CM scan result—the CM scans consistently recovered positions ∼20-100 of the real purine sequences. Both motifs contained very few false positive hits (quasi-clique: 4, hierarchical clustering: 5).

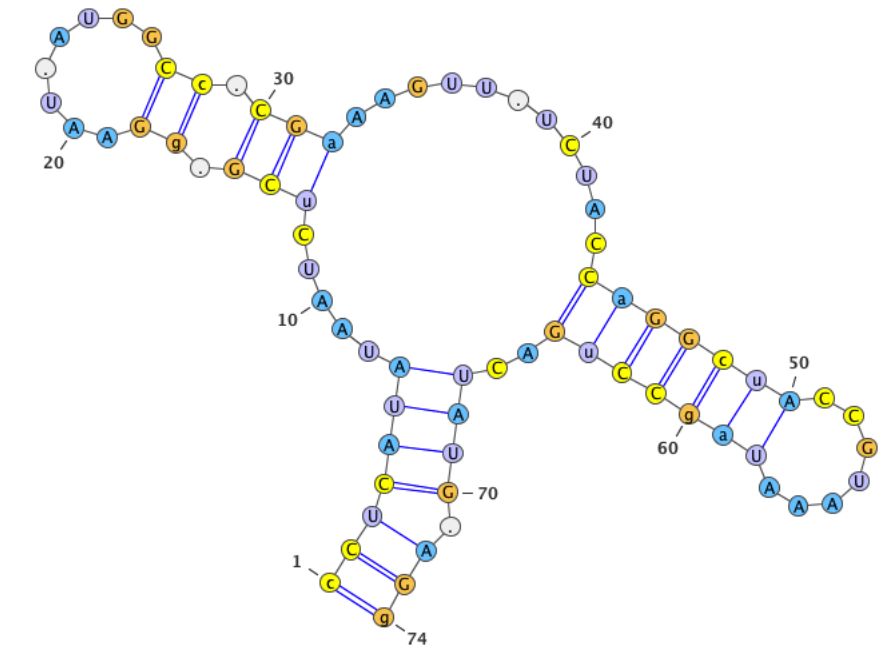
In contrast to the success of AdoCbl and purine riboswitch families, both clustering methods failed to predict a good yybP motif. The consensus structure of the yybP RNA element shows that it contains long stretches of unpaired regions with low conservation (Figure 1.3c). The predicted motifs reflect some of the unpaired loops but do not fully capture the complicated stem loop. The hierarchical clustering motif recovered a longer part of the yybP structure and the CM scan matches also matched longer segments (∼100 nt) of the real yybP sequences. The quasi-clique motif is missing more structure in the 3' region and the CM scan matches matched

∼70 nt of the real yybP sequences. However, the longer motif from hierarchical clustering had worse sensitivity (0.2) compared to quasi-clique's (0.53). The predicted motifs came from good clusters—for hierarchical clustering it was a 18/19 yybP cluster and for quasi-clique it was 12/12—so we can rule out false positives as a factor for bad motif prediction. The sequences in the yybP clusters overlapped the real yybP element completely (so we can rule out incomplete input sequence as a cause), but the predicted motifs were missing ∼20 bp in the 3' region. In other words, CMFinder failed to recognize the last segment of the yybP structure as part of the motif. Furthermore, it's possible that the cluster sequences were Firmicutes-specific, leading to a motif that does not represent the general yybP consensus structure.

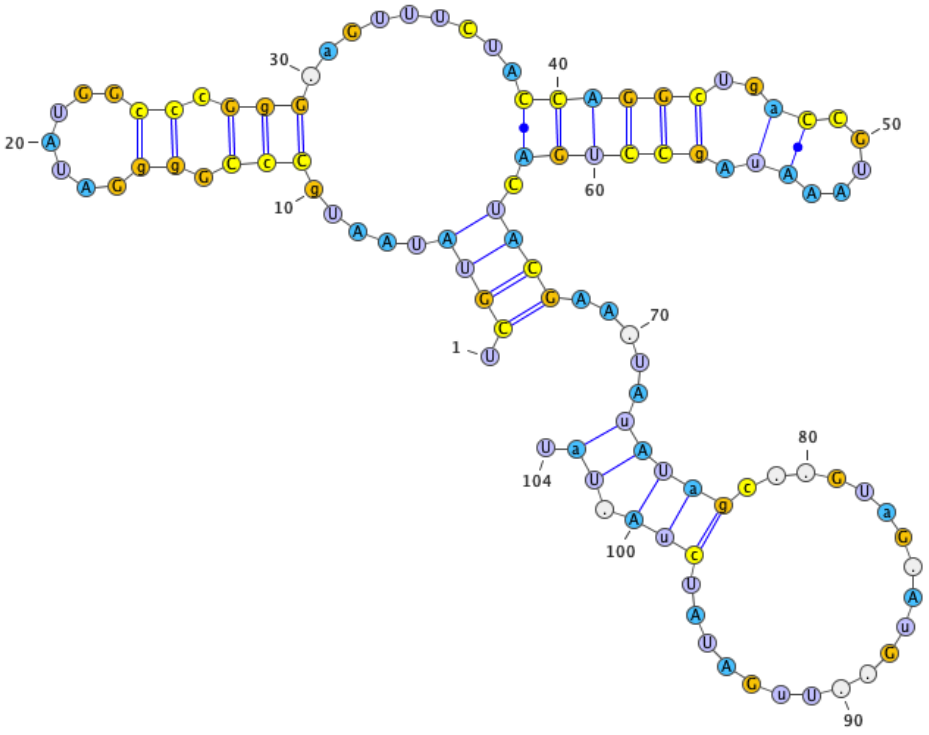
To sum up, for a few families (6S, Lacto-rpoB, pfl, rpsB, and yybP), neither methods produced good motifs. For the majority of the ncRNA families, however, both methods produced at least one good motif that recovered most of the real ncRNA sequences in the CM scan with very few false positive hits.

1.4 Discussion

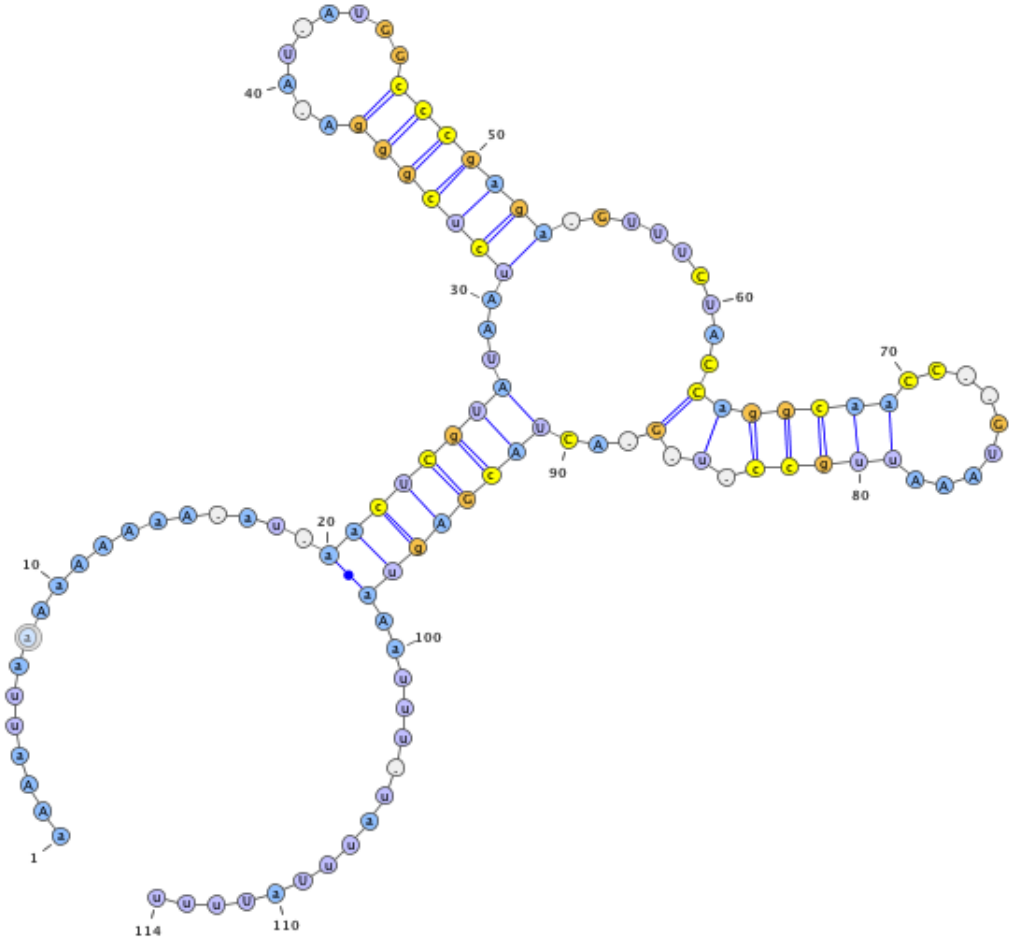
In this chapter, we presented two algorithms for finding homologous ncRNA clusters. Given a homology graph, the maximal quasi-clique finding algorithm greedily and randomly constructs a γ-clique such that the nodes are connected above some threshold γ. Quasi-clique is easy to parameterize (choice of γ can be easily modified) and parallelize. The hierarchical clustering algorithm deterministically merges two closest nodes at a time; it is not easily parallelizable. We applied both methods to the homology graph from 246 Firmicutes genomes and evaluated the clusters using known ncRNA families. Neither method succeeded in clustering all ncRNA families perfectly and not all clusters had high specificity. However, many of these families were well-represented in at least one cluster by both methods and CMFinder was able to predict a good motif that performed well in the CM scans. In the case of the purine riboswitch, only 5 sequences were sufficient for CMFinder to construct the major multi-stem in its secondary structure. On the other hand, neither methods could construct a good motif for the yybP and several other families. Several factors



(a) Best purine motif from hierarchical clustering.

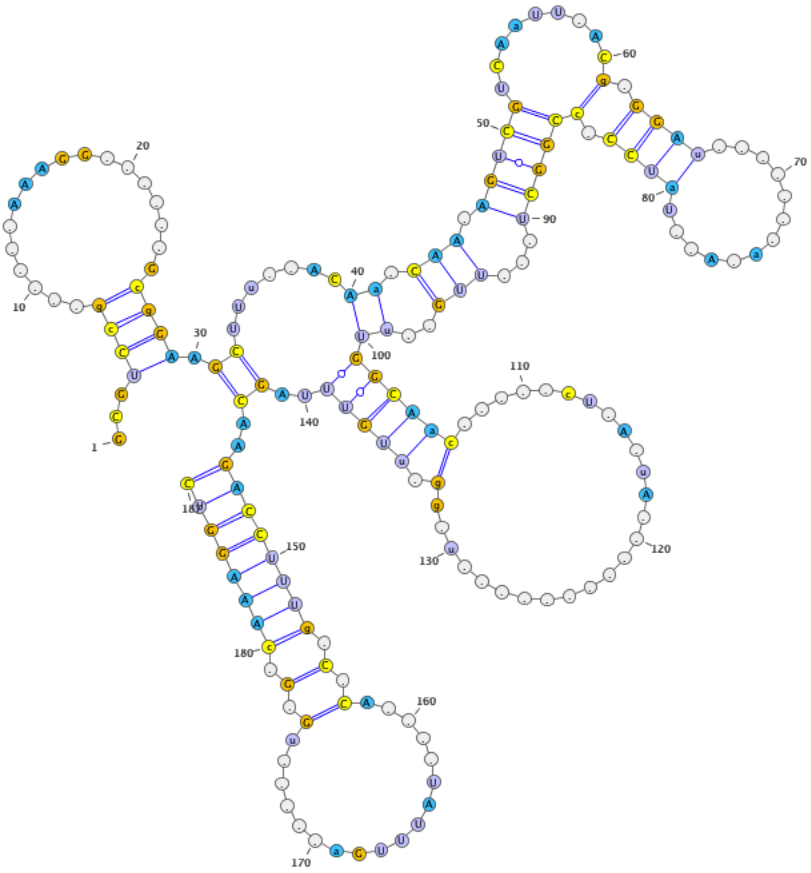


(b) Best purine motif from quasi-clique.

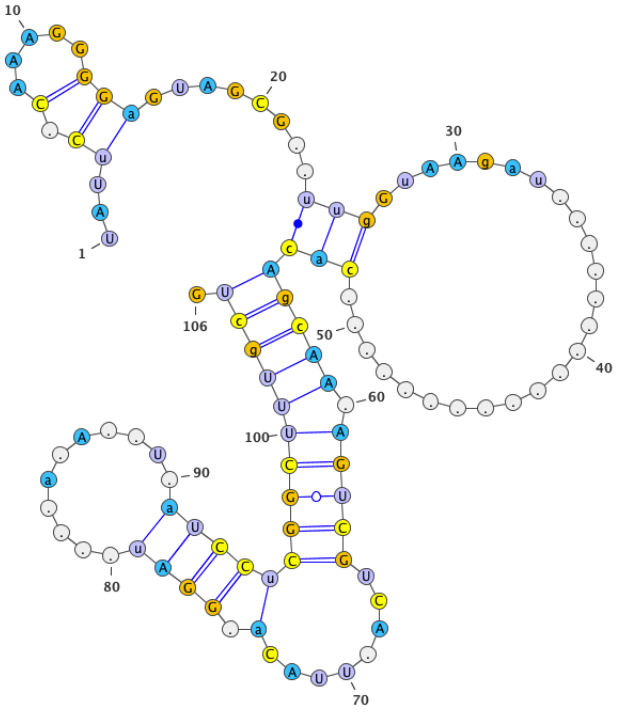


(c) Purine riboswitch consensus structure.

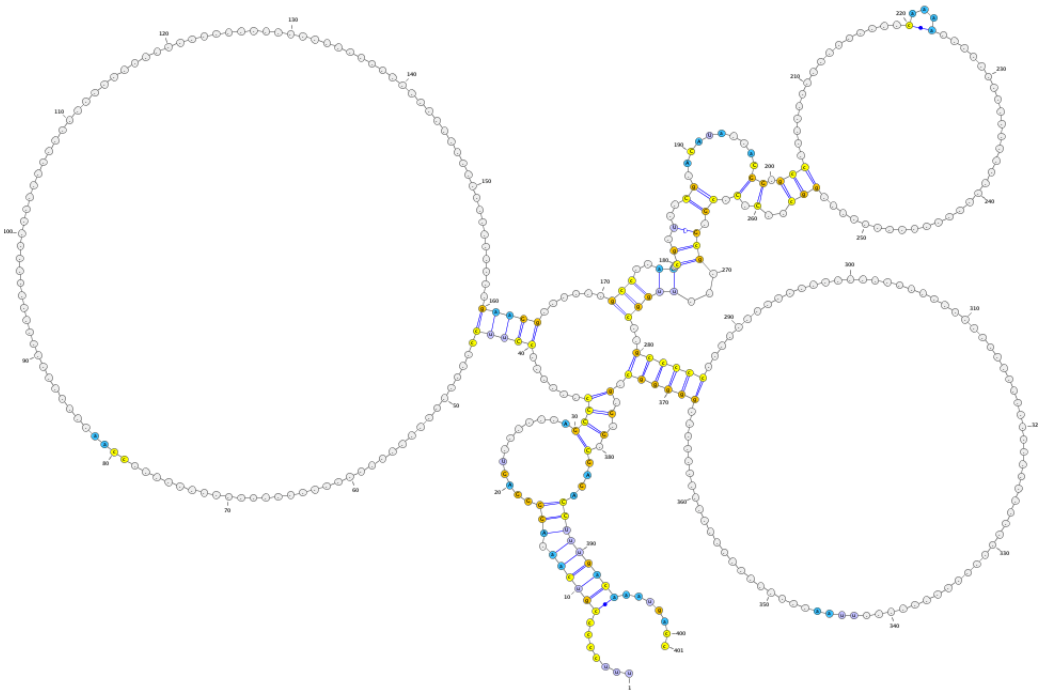
Figure 1.2: **Best purine motif from (a) hierarchical clustering and (b) quasi-clique clustering.** The best motifs were selected based on the sensitivity/specificity of CM scan results. **(c) Consensus structure of the purine riboswitch.** Alignment obtained privately from Breaker lab. Bases are colored by consensus nucleotide (A: blue, T/U: purple, G: orange, C: yellow, gap: gray). Both predicted motifs captured the major multi-stem in the real purine riboswitch and both missed the unpaired 5' and 3' region.



(a) Best yybP motif from hierarchical clustering.



(b) Best yybP motif from quasi-clique.



(c) yybP riboswitch consensus structure.

Figure 1.3: **Best yybP motif from (a) hierarchical clustering and (b) quasi-clique clustering.** The best motifs were selected based on the sensitivity/specificity of CM scan results. **(c) Consensus structure of the yybP riboswitch.** Alignment obtained privately from Breaker lab. The consensus structure contains long unconserved, unpaired regions that the predicted motifs only partially captured.

can contribute to motif finding failure: there wasn't enough sequence conservation to be picked up by BLAST, the parameters used in the clustering methods were not optimal, or lack of sequence covariation.

Beyond cluster and motif prediction, ncRNA discovery presents additional computational and experimental challenges. In this chapter, we focused on clusters that contained known ncRNA sequences, but in practice we are interested in clusters that contain novel ncRNAs. Unfortunately, not all clusters contain ncRNAs. To screen through thousands of clusters manually would be impossible. Ranking methods for predicted motifs have been developed [111, 110] but remain a difficult topic. Finally, experimental validation of ncRNAs is difficult because the binding ligand is unknown. Many of the riboswitches that have been experimentally validated to date were relatively easy guesses: the acting ligand comes from the gene appearing immediately 3' of the riboswitch. Several "orphan riboswitches", like the ykkC, yybP, and pfl RNA elements, are strong candidates for being riboswitches because they have highly conserved structure and appear in a wide range of bacterial phyla, yet so far all screens for binding ligands have returned no results [65].

Next-generation high-throughput sequencing presents a new and perhaps alternative way for ncRNA discovery. Instead of relying on computational predictions, which requires both a sufficient number of diverse yet homologous sequences and a strong consensus structure, whole transcriptome sequencing simply looks at what is transcribed in a single cell, tissue, or environment. If a ncRNA is functional, it must be present in the transcriptome data. In humans, several long ncRNAs have been discovered and shown to be associated with tissue-specific expression and cancer [32]; these ncRNAs span several kb, do not have conserved structures, and some are even spliced. Traditional computational discovery methods such as the one described in this chapter would thus have failed to find these ncRNAs. However, transcriptome data presents its own kinds of challenges: there is a significant amount of noise—it has been estimated that up to 70%-90% of the human genome is transcribed, and it is unlikely that all of them are functional. For an ncRNA transcript to be detectable, it would have to be sufficiently abundant in the sample so that (a) it actually appears

in the sequencing data, and (b) its abundance rises above the background transcription level. Sequencing errors, incomplete reference genomes, antisense transcription (the ncRNA is on the opposite strand on a coding gene), and alternative splicing are other factors that can complicate analysis even further. Nevertheless, there is hope that with more fully sequenced genomes and affordable transcript sequencing, new computational methods could be developed to expedite the process of discovering ncRNAs.

Chapter 2

USING SECONDARY STRUCTURE INFORMATION TO IMPROVE PHYLOGENETIC TREE INFERENCE OF NCRNAS

2.1 Introduction

Phylogenetic trees provide direct insight into the evolutionary history of DNA and protein sequences. In novel bacteria discovery, a reliable species tree built from an universal gene, such as the 16S rRNA gene, provides reference from which information on novel organisms can be inferred. This is the approach taken in the rising field of metagenomics studies, where thousands of uncultured bacteria are being sequenced and categorized by comparing them to their nearest known neighbors. In clinical settings, where identification of novel pathogens is critical and time-sensitive, identifying the bacterial species using the 16S rRNA gene is more accurate than using morphologic and phenotypic traits [16]. On the gene level, knowing the series of mutation and duplication events in proteins help study evolution, a common approach for studying selection pressure of critical proteins in infectious viruses, e.g., H5N1 [101], HIV [35]. Finally, from a global perspective, the Tree of Life is a guide to understanding the relationships between all living species on Earth [55].

Accurate phylogenetic tree inference is not easy, however, because the components to estimating a phylogenetic tree—alignment and evolutionary model—are just our computational guesses. For noncoding RNAs, there is a third component: secondary structure, which can be physically determined via crystallography, but are available only for a handful of sequences due to cost. Yet even when all three components are accurate, inferring the tree itself remains an arduous task. One reason is that the tree search space grows rapidly with the number of species: for 15 species, the number of possible trees is $\sim$8 trillion, a prohibitively big number even for today's computing power. Existing tree algorithms therefore all use some kind of heuristics

to speed up the search.

Another layer of difficulty in the tree inference problem lies in the fact that the alignment, model and tree are not independent issues. Alignment describes the correspondence between bases in homologous sequences; it implicitly states that bases from an alignment column all come from a single base of the last common ancestor, which then evolved according to the evolutionary model and the divergence time (branch lengths in the tree). Some tree algorithms thus don't just assume the alignment and model are given, but try to infer the phylogenetic tree along with one or more of alignment, model, and structure.

Here, we are interested in inferring phylogenetic trees for noncoding RNAs once their alignment and structure have been given. In particular, we want to know if having a separate evolutionary model for structured and unstructured regions would get us closer to the true phylogenetic tree. Intuitively, it seems like it would, because the structured and unstructured regions should be under different modes of selection. In the following sections, we first describe the generalized time reversible model and how it is used to calculate the likelihood of a tree (section 2.2), provide a brief description of existing maximum likelihood tree inference programs (section 2.3), then finally present our tree inference program GBPML (Gapped BasePaired Maximum Likelihood) and evaluate it on simulated and real ncRNA datasets.

2.2 The GTR model and tree likelihood calculation

The GTR (general time reversible) model is the most general neutral, independent, finite-sites, and time-reversible model. Simpler models (e.g., Jukes Cantor model, F84 [24]) all come from adding more restrictions to the GTR parameters. The instantaneous substitution rate matrix R is defined as:

$$R = \begin{pmatrix} R_{A,A} & \pi_C r_{A,C} & \pi_G r_{A,G} & \pi_T r_{A,T} \\ \pi_A r_{C,A} & R_{C,C} & \pi_G r_{C,G} & \pi_T r_{C,T} \\ \pi_A r_{G,A} & \pi_C r_{G,C} & R_{G,G} & \pi_T r_{G,T} \\ \pi_A r_{T,A} & \pi_C r_{T,C} & \pi_G r_{T,G} & R_{T,T} \end{pmatrix} \tag{2.1}$$

where $R_{i,i} = -\sum_{j|j\neq i} R_{i,j}$ and π are the equilibrium base frequencies. To further satisfy the reversibility condition, which is $\pi_i R_{i,j} = \pi_j R_{j,i}$, we set $r_{i,j} = r_{j,i}$, so the matrix simplifies to:

$$R = \begin{pmatrix} R_{A,A} & \pi_C a & \pi_G b & \pi_T c \\ \pi_A a & R_{C,C} & \pi_G d & \pi_T e \\ \pi_A b & \pi_C d & R_{G,G} & \pi_T f \\ \pi_A c & \pi_C e & \pi_G f & R_{T,T} \end{pmatrix} \tag{2.2}$$

which is 6 parameters for substitution rates and 3 for equilibrium frequencies.

Given the rate matrix R, the substitution matrix Q for a given branch length t can be computed as:

$$Q = e^{tR} \tag{2.3}$$

Now that we have $Q_{i,j} = \Pr(\text{base } i \text{ mutates to } j \mid \text{branch length } t)$, the likelihood of a given column in an alignment can be computed using Felsenstein's peeling algorithm [23]. Let $L^{(l)}_{k,s_k}$ be the likelihood of the tree node k at the l-th column where the base is s_k, and the branch lengths to child i and j are t_i and t_j:

$$L^{(l)}_{k,s_k} = \Big(\sum_{s_i=\{A,C,G,T\}} \Pr(s_k \text{ mutates to } s_i \mid t_i) L^{(l)}_{i,s_i}\Big)\Big(\sum_{s_j=\{A,C,G,T\}} \Pr(s_k \text{ mutates to } s_j \mid t_j) L^{(l)}_{j,s_j}\Big) \tag{2.4}$$

At the leaf nodes, where the bases are known, $L^{(l)}_{k,s_k} = 1$ if s_k is the observed base, otherwise it is 0. The likelihood at the root for the l-th column is then:

$$L^{(l)} = \sum_{s_{root}} \pi_{s_{root}} L_{root,s^{(l)}_{root}} \tag{2.5}$$

and the total likelihood of the tree is just the product of $L^{(0)}, L^{(1)}, L^{(2)}, \ldots$ since we assume column independence. The equations described here can be generalized for base paired regions, where we assume pairs of columns to be independent and have 16 instead of 4 possibilities for s_i, s_j, and s_k.

Program	**Input**	**Output**	**Model**	**Gaps**
CMFinder [112]	sequence	structure alignment	EM algorithm for iteratively refining structure & alignment	Inherent in covariance models
stemloc [33]	sequence	structure alignment	Efficient heuristics for Sankoff SCFG model	Inherent in SCFG modeling
SimulFold [63]	sequence	structure alignment tree	Co-estimation of **S**tructure (with pseudoknots), **A**lignment, and **T**ree efficiently using MCMC sampling of $\Pr(\mathbf{S}, \mathbf{A}, \mathbf{T} \mid \mathbf{D}data)$	For $\Pr(\mathbf{D} \mid \mathbf{S}, \mathbf{A}, \mathbf{T})$: treat as unknown letter. For alignment prior: open/extend/close gap penalties
EvoFold [73]	alignment tree	structure	phylo-SCFG with separate models for paired and unpaired (marginalized from paired model)	Treat as unknown letter
Pfold [40, 41]	alignment	structure tree	phylo-SCFG with neighbor joining tree (for topology) and ML-optimized branch lengths	Treat as unknown letter
PETfold [86]	alignment	structure tree	Extends Pfold to use both SCFG and thermodynamic modeling	Treat as unknown letter
DNAML-ϵ [79]	alignment	tree	Extends DNAML for complex indels	Models insertions & deletions
RAxML [89]	alignment (structure)	tree	Efficient heuristics for speeding up DNAML for large datasets	Treat as unknown letter
GBPML	alignment structure	tree	Uses separate models for paired and unpaired	Models insertions but not deletions

Table 2.1: **A brief summary of some of the related works that either focus on predicting secondary structure for ncRNAs or phylogenetic trees from sequence alignments.**

2.3 Existing maximum likelihood tree programs

In this section, I review the three main programs that share major similarities with our tree program. Pfold was our inspiration for incorporating secondary structure information into tree estimation, DNAML-ϵ pushes the possibility of realistic indel modeling while keeping computation tractable, and RAxML is a maximum likelihood tree program that is highly optimized for large datasets. Additional tree programs that either predict ncRNA secondary structure or infer phylogenetic trees are shown in Table 2.1.

2.3.1 Pfold

The main goal of Pfold is secondary structure prediction, though as part of the intermediate step, a phylogenetic tree is estimated. The input to Pfold consists of an alignment A, a stochastic context-free grammar (SCFG) M_g, and a substitution model M_s. The key to Pfold is the ability to calculate the probability of the alignment given a structure σ, a tree t, and the models M_g, M_s:

$$\Pr(A \mid \sigma, t, M_g, M_s)$$

Given M_g, parsing of the alignment can be done using the inside-outside algorithm [20]. The likelihood of seeing a particular column (or pairs of columns) is then the probability of the grammar production rule (from M_g) multiplied by the probability of observing the base (or base pair) transitions given tree t (from M_s). Once this is done for every column, the total likelihood is just the product of the probabilities (column independence is assumed). To find the best structure would then be to pick the structure with the highest posterior probability:

$$\sigma^{MAP} \propto argmax_{\sigma} \Pr(A \mid \sigma, t, M_g, M_s) \Pr(\sigma \mid M_g)$$

This was the approach taken in the original 1999 version of Pfold [40]. The tree was estimated using maximum likelihood; separate models for paired and unpaired

regions were trained from tRNA and rRNA sequences. But the authors considered maximum likelihood too slow (they did not use heuristics for tree searching), so in the newer and currently available version [41], a single unpaired model was made by marginalizing the 1999 rate matrices and used to compute a pairwise distance matrix.

Given the rate matrix R, the distance t between a pair of sequences A and B is found using the objective function:

$$t = \text{argmin} \log \prod_{i=1,2,\ldots N} \Pr(\text{base } A[i] \text{ mutates to base } B[i] \mid \text{branch length } t) \quad (2.6)$$

where $A[i]$ is the base at the i-th position of A. The substitution probabilities $\Pr(\ldots)$ are entries from $Q = e^{tR}$. In other words, the pairwise sequence distance is the branch length t that results in the best log likelihood.

Given the pairwise distance matrix, they then run neighbor joining [83] to get a tree and optimize the branch lengths using maximum likelihood. Instead of simply picking the most probable structure, they pick the best nested structure with the highest expected number of correctly predicted columns (correct here refers to the probabilities from the inside-outside algorithm) as the final output.

To summarize: the Pfold tree estimation subroutine is fast (neighbor joining), does not use structure, and treats gaps as unknown letters (no indel modeling).

2.3.2 DNAML-ϵ

In [79], Rivas & Eddy extend a substitution-only rate model to allow for insertions and deletions. They parameterize deletion with a single parameter μ and insertion with rate λp_k, where p_k is assumed to be the same as the equilibrium base frequencies (π_k). The difficulty with modeling indels is that the ancestral sequences will not have the same lengths — insertions cause descendent sequences to be longer, deletions shorter, and there are bases in the ancestral sequences that have left no trace in the extant sequences. To allow for an arbitrary number of columns in ancestral

sequences, they parameterize ancestral sequence lengths with a geometric distribution; in the actual implementation, this is approximated using the average lengths of the extant sequences. A biologically realistic modeling of insertions and deletions would allow for simultaneous insertions and deletions of several bases (affine gaps), however this makes computation extremely difficult. Instead, the authors assume column independence, which means that every insertion and deletion event is independent and at most one insertion is allowed per column. This simplification allows them to use Felsenstein's peeling algorithm (Eq. 2.5) so that computation remains fast. Despite the assumptions and simplifications made here, this is still by far the most realistic insertion/deletion modeling compared to all the other tree programs in Table 2.1, which all treat gaps as unknown. They modified DNAML [24] to use their model; the substitution part of the model is the F84 model [25], which has the substitution rate:

$$R(i \neq j) = \beta\pi_j + \alpha\Delta_{ij}$$

$$\Delta_{ij} = \pi_j \frac{\epsilon_{ij}}{\sum_k \pi_k \epsilon_{jk}}$$

$$\epsilon_{ij} = \begin{cases} 1 & \text{if i, j are both either purines or pyrimidines} \\ 0 & \text{otherwise} \end{cases}$$

Thus there are 4 $(\alpha, \beta, \mu, \lambda)$ rate parameters that need to be estimated during branch length optimization.

To summarize: DNAML-ϵ is fast, does not use structure, and models insertions and deletions assuming column independence.

2.3.3 RAxML

RAxML (Randomized Accelerated Maximum Likelihood) [89] is a widely used phylogenetic tree program mainly because it is highly optimized for large datasets. Its underlying substitution model and tree search routine is based on DNAML: a GTR model with site rate heterogeneity and branch swapping for tree search. It uses heuristics to speed up estimation of site rates (GTRCAT approximation [88]), tree

branch storage, and tree search exploration. It also takes full advantage of parallelization (MPI or multicore), dramatically reducing runtime. Additionally, it supports secondary structure input, providing several flavors of parameterized PHASE models [78].

To summarize: the RAxML tree estimation subroutine is fast (despite using maximum likelihood), can use structure (both paired and unpaired models are parameterized and estimated during branch length optimization), and treats gaps as unknown (no indel modeling).

2.4 Methods

2.4.1 Evolutionary model

Our evolutionary model is based on the work of Zizhen Yao, who in her thesis proposed separate evolutionary models for unpaired and paired bases [110]. Her model is similar to the ones used by Pfold [40] and later, EvoFold [73], with differences mainly in symmetry assumptions and parameterizations for non-canonical base pair substitutions. Similar to EvoFold, she developed the model for scoring predicted ncRNA structures and assumed the phylogenetic tree was given, whereas here we use the model to infer the tree assuming the structure is given.

Ignoring gaps for now, the substitution model consists of a 4 x 4 substitution rate matrix $R^{unpaired}$ for unpaired bases and a 16 x 16 matrix R^{paired} for paired bases. $R^{unpaired}$ is just the GTR model:

$$R_{i,j}^{unpaired} = \pi_j r_{i,j} \tag{2.7}$$

where $R_{i,j}^{unpaired}$ is the instantaneous substitution rate from base i to j, π_j is the equilibrium base frequency of j. We set $r_{i,j} = r_{j,i}$ for all i, j to reduce the number of parameters we need to estimate while satisfying reversibility (i.e. $\pi_i R_{i,j}^{unpaired} = \pi_j R_{j,i}^{unpaired}$). Thus, for the unpaired gapless model, there are 3 free parameters for the equilibrium base frequencies and 6 for the instantaneous substitution rates.

R^{paired} is designed so that substitutions from one canonical base pair (AU, GC,

or GU) to another follows the GTR model. Additional parameters then categorize substitutions between canonical or non-canonical base pairs to reduce the number of free parameters we have to estimate:

$$R^{paired}_{ij,kl} = \pi_{kl} \left\{ \begin{array}{ll} r_{ij,kl} & \texttt{if both are canonical base pairs} \\ \gamma_1 & \texttt{if only one is canonical and i=k or j=l} \\ \gamma_2 & \texttt{if only one is canonical and i}\neq\texttt{k and j}\neq\texttt{l} \\ \beta_1 & \texttt{if both are not canonical and i=k or j=l} \\ \beta_2 & \texttt{if both are not canonical and i}\neq\texttt{k and j}\neq\texttt{l} \end{array} \right\} \tag{2.8}$$

And again, we set $r_{ij,kl} = r_{kl,ij}$. In addition, we set $r_{ij,kl} = r_{ji,lk}$ and $\pi_{ij} = \pi_{ji}$ (symmetry of base pairs). Thus, for the paired gapless model, there are 9 free parameters for equilibrium base pair frequencies and 13 for the instantaneous substitution rates.

Finally, our gap model describes an evolutionary process that allows deletions ($i \to -$) but not insertions ($- \to i$). Insertions are more difficult to model than deletions, as we would have to model variable ancestral sequence lengths. For simplicity and efficiency, we adopted Yao's choice of modeling only deletions. For unpaired bases, we have one parameter for deletions and a zero probability for insertions:

$$R^{unpaired}_{i,-} = \alpha_1 \tag{2.9}$$

$$R^{unpaired}_{-,i} = 0 \tag{2.10}$$

For paired bases, we allow for one of the bases to be deleted ($ij \to i-$ or $ij \to -i$), for the entire base pair to be deleted ($ij \to --$), or for a base pair to be fully deleted ($i- \to --$ or $-i \to --$). This introduces three more parameters:

$$R^{paired}_{ij,i-} = R^{paired}_{ij,-j} = \alpha_2 \tag{2.11}$$

$$R^{paired}_{ij,--} = \alpha_3 \tag{2.12}$$

$$R^{paired}_{i-,--} = R^{paired}_{-i,--} = \alpha_4 \tag{2.13}$$

For aligned bases that have already lost pairing, substitution follows the unpaired model:

$$R^{paired}_{i-,k-} = R^{paired}_{-i,-k} = R^{unpaired}_{i,k} \tag{2.14}$$

Any other kinds of substitutions that are not covered by the above rules, such as $ij \to k-$ (a substitution and a deletion) or $-- \to ij$ (insertion) have zero probabilities.

To sum up, the gapped unpaired model is a 5 x 5 matrix with 7 instantaneous substitution rate parameters and 3 equilibrium base frequency parameters, and the gapped paired model is a 25 x 25 matrix with 16 and 15 parameters, respectively.

2.4.2 Calculating the tree likelihood

Given a rate matrix R, the substitution matrix Q for a given branch length t can be computed as:

$$Q = e^{tR} \tag{2.15}$$

For a given tree and alignment, we compute the tree's log likelihood using Felsenstein's peeling algorithm [23] separately for unpaired and paired bases, then take the sum of it to form the total log likelihood. Note that, since our gap model is non-reversible, we cannot use the pulley principle[1] to reroot the tree to compute the log likelihood. Instead we use L-BFGS [117] to optimize branch lengths.

2.4.3 Training the evolutionary model

Yao et al. trained the evolutionary model using the 17-vertebrate species tree with a collection of structurally annotated ncRNA alignments from the UCSC browser. MAF blocks from Wang et al. [99] were scanned with Rfam covariance models to get ncRNA matches; the matches were then aligned using Infernal's `cmalign`.

[1] Described in [23], the pulley principle refers to the fact that for reversible models, the likelihood of the tree is the same if it is rerooted at any point. The pulley principle allows for a fast analytical solution to optimize branch lengths.

	A	**C**	**G**	**T**
A	-0.395	0.076	0.229	0.090
C	0.077	-0.397	0.069	0.250
G	0.209	0.062	-0.348	0.077
T	0.081	0.222	0.075	-0.378
α_1: 0.057				
Freq:	0.238	0.235	0.261	0.266

Table 2.2: **Rate matrix for unpaired model (from Yao thesis [110]).** Trained from 264 ncRNA alignments using the 17-vertebrate species tree.

The unpaired and paired models are shown in Table 2.2 and Table 2.3. In Yao's models, the paired model has a scale of 2 (which doubles the branch length) while the unpaired model has a scale of 1. Yao was concerned that the training data is overly conservative and mutation rates for the paired region are underestimated. She found that the 2-to-1 ratio resulted in better false discovery rates for genomic scale ncRNA discovery, so we've adopted the scaling here.

	AA	**AC**	**AG**	**AT**	**CA**	**CC**	**CG**	**CT**	**GA**	**GC**	**GG**	**GT**	**TA**	**TC**	**TG**	**TT**
AA	-2.6469	0.2967	0.1677	0.6306	0.2967	0.0191	0.1380	0.0221	0.1677	0.1380	0.0210	0.0355	0.6306	0.0221	0.0355	0.0258
AC	0.1503	-2.6528	0.1677	0.6306	0.0362	0.1564	0.1380	0.0221	0.0205	0.9395	0.0210	0.0355	0.0926	0.1810	0.0355	0.0258
AG	0.1503	0.2967	-2.8426	0.6306	0.0362	0.0191	0.9395	0.0221	0.0205	0.1380	0.1719	0.0355	0.0926	0.0221	0.2417	0.0258
AT	0.0329	0.0650	0.0367	-0.4806	0.0095	0.0050	0.0075	0.0397	0.0054	0.0351	0.0055	0.1618	0.0208	0.0058	0.0034	0.0464
CA	0.1503	0.0362	0.0205	0.0926	-2.6528	0.1564	0.9395	0.1810	0.1677	0.1380	0.0210	0.0355	0.6306	0.0221	0.0355	0.0258
CC	0.0184	0.2967	0.0205	0.0926	0.2967	-3.1968	0.9395	0.1810	0.0205	0.9395	0.0210	0.0355	0.0926	0.1810	0.0355	0.0258
CG	0.0048	0.0095	0.0367	0.0051	0.0650	0.0343	-0.4204	0.0397	0.0054	0.0064	0.0377	0.0034	0.0235	0.0058	0.1364	0.0068
CT	0.0184	0.0362	0.0205	0.6306	0.2967	0.1564	0.9395	-2.8812	0.0205	0.1380	0.0210	0.2417	0.0926	0.0221	0.0355	0.2116
GA	0.1503	0.0362	0.0205	0.0926	0.2967	0.0191	0.1380	0.0221	-2.8426	0.9395	0.1719	0.2417	0.6306	0.0221	0.0355	0.0258
GC	0.0048	0.0650	0.0054	0.0235	0.0095	0.0343	0.0064	0.0058	0.0367	-0.4171	0.0377	0.1364	0.0051	0.0397	0.0001	0.0068
GG	0.0184	0.0362	0.1677	0.0926	0.0362	0.0191	0.9395	0.0221	0.1677	0.9395	-3.0629	0.2417	0.0926	0.0221	0.2417	0.0258
GT	0.0048	0.0095	0.0054	0.4223	0.0095	0.0050	0.0002	0.0397	0.0367	0.5301	0.0377	-1.1687	0.0088	0.0058	0.0067	0.0464
TA	0.0329	0.0095	0.0054	0.0208	0.0650	0.0050	0.0351	0.0058	0.0367	0.0075	0.0055	0.0034	-0.4806	0.0397	0.1618	0.0464
TC	0.0184	0.2967	0.0205	0.0926	0.0362	0.1564	0.1380	0.0221	0.0205	0.9395	0.0210	0.0355	0.6306	-2.8812	0.2417	0.2116
TG	0.0048	0.0095	0.0367	0.0088	0.0095	0.0050	0.5301	0.0058	0.0054	0.0002	0.0377	0.0067	0.4223	0.0397	-1.1687	0.0464
TT	0.0184	0.0362	0.0205	0.6306	0.0362	0.0191	0.1380	0.1810	0.0205	0.1380	0.0210	0.2417	0.6306	0.1810	0.2417	-2.5544
$alpha_2$:0.0210 $alpha_3$:0.0196 $alpha_4$:0.1319																
Freq:	0.0081	0.0160	0.0091	0.1554	0.0160	0.0084	0.2315	0.0098	0.0091	0.2315	0.0093	0.0596	0.1554	0.0098	0.0596	0.0114

Table 2.3: **Rate matrix for paired model (from Yao thesis [110]).** Trained from 264 ncRNA alignments using the 17-vertebrate species tree. Branch lengths are scaled by 2.

Algorithm 2.4.1: GBPML(alignment A, tree t_0, bool *nobp*)

if t_0 is not a binary tree
  then Convert t_0 into a binary tree
Remove from A all columns with ambiguous bases
Remove from A all unpaired columns with $> 70\%$ gaps
Remove all paired columns from A with $> 50\%$ non-canonical pairs
if *nobp* is $True$
  then Reannoate all columns of A as unpaired
$tree \leftarrow FindMLTree(t_0, 0, 5, 20, 0.1)$
return $(tree)$

Algorithm 2.4.2: FINDMLTREE(tree t_{best}, int rL, int rU, int k, float ϵ)

$rMAX \leftarrow 20$
while $rU \leq rMAX$
do
- $k_{best} \leftarrow SubtreeRearr(t_{best}, rL, rU, k)$
- Optimize branches of all k best trees of k_{best}
- $t \leftarrow$ best tree from k_{best}
- **if** $t.log\_likelihood < t_{best}.log\_likelihood + \epsilon$
  - **then** $t_{best} \leftarrow t$
  - **else**
    - $rL \leftarrow rL + 5$
    - $rU \leftarrow rU + 5$

return (t_{best})

2.4.4 Tree search algorithm using maximum likelihood

Our tree search algorithm (Algorithm 2.4.1) is based on the RAxML algorithm [4]. We begin with an initial tree (in this case, the neighbor joining tree produced by Pfold) and explore the tree space via subtree rearrangements (Algorithm 2.4.3). At

first we limit possible subtree rearrangements to nearby branches and only increase the limit when the likelihood does not improve. The search ends when the log likelihood stops increasing by some small threshold ϵ (default 0.1). To speed up the tree search, we (1) limit tree optimization after rearrangement to only optimizing the local three branches after subtree rearrangement, and (2) parallelize the search for subtree rearrangements (to 5 parallel processes in this study) (Algorithm 2.4.2).

Algorithm 2.4.3: SUBTREEREARR(tree t_{cur}, int rL, int rU, int k)

$k_{best} \leftarrow \emptyset$
for each internal node $n \in t_{cur}$
do {
  $C \leftarrow$ all possible reinsertion points of distance d, $rL \leq d \leq rU$
  $trees \leftarrow$ empty list
  for each $c \in C$ {
    $t' \leftarrow$ new tree with n inserted at point c of t_{cur}
      and local 3 branches optimized
    Insert t' into $trees$
    Order $trees$ by decreasing likelihood
    $k_{best} \leftarrow k_{best}$ combined with $trees$
    $t_1 \leftarrow$ first element of $trees$
    if $t_1.log\_likelihood > t_{cur}.log\_likelihood$
      then $t_{cur} \leftarrow t_1$
  }
}
$k_{best} \leftarrow$ first k elements of k_{best}
return (k_{best})

2.4.5 Pairwise tree distance calculation

To compare how similar two trees are, we calculate the symmetric difference distance (SDD) and the normalized branch score distance (nBSD). The branch score distance (BSD) penalizes heavily branch partitions that appear only in one tree, and less so (or not at all) on partitions shared by the two trees. To account for scaling differences that result from running different evolutionary models, we normalize trees

to an average branch length of 1 before computing BSD. The SDD is similar to BSD except that it ignores branch lengths (or assumes they are all of length 1). A detailed description of SDD and nBSD is provided in the documentation of the `treedist` program in the PHYLIP package [24]. In this study we use the Python library dendropy [92] implementation of SDD and nBSD. We also computed the Robinson-Foulds distances (RBD) using dendropy and found that it gave similar results, so in the results section we mostly show the nBSDs.

2.4.6 Statistical significance of pairwise program comparisons

Given the goodness of inferred trees, as measured by SDD and nBSD, we compare the performance of different programs using a paired Wilcoxon test. To test if program A infers betters trees than B on a dataset, where X_A, X_B are the SDDs (or nBSDs) and $X_A[i]$ and $X_B[i]$ are the values for the i-th dataset, we run a paired Wilcoxon test with alternative hypothesis of $A < B$. The smaller the p-value, the more significant the performance of A over B.

2.5 Results

2.5.1 Evaluation on simulated data

Using simulated data, we first evaluated whether the use of structure information improves tree inference. With simulated data, we know the ground truth and can compare the inferred trees with the true trees. We generated 500 8-taxon trees using DNAML-ϵ's *erate-tree* program (average branch length set to 0.2), which implements the ultrametric tree simulation from [43]. For each tree, we generated alignments of fixed total lengths using our evolutionary model (section 2.4.3) with varying proportions of paired columns; we generated alignment lengths of 200 bp, 350 bp, 500 bp, 1000 bp, and paired column proportions of 0 (no structure), 0.2, 0.4, and 0.6. The average pairwise sequence identity was 71% and the fraction of indels was 4%. We ran GBPML on the alignments once with structure information and once without (GBPML_nobp). We then compared the inferred trees to the true trees and

calculated distance metrics (Figure 2.1). Average runtime is shown in Table 2.4. Considering only topology (SDD), GBPML was slightly better than GBPML_nobp; p-values were significant only for higher paired proportions (Figure 2.1a). Considering both topology and branch length (nBSD and RBD), the paired Wilcoxon tests and the average tree distances both showed that GBPML performed better than GBPML_nobp (Figure 2.1b, Figure 2.1c). What was also worthy of note was: the longer the total alignment length, the easier the tree inference was because there was more information. Yet within a fixed total alignment length, as the proportion of paired columns increased, the programs performed less well. This is not surprising because while one paired column is more informative than one single column (Figure 2.2a), it is not twice as informative. As a result, for a fixed alignment length, the total amount of information (entropy) decreased with increasing paired column proportion (Figure 2.2b).

Total alignment length (bp)	200	350	500	1,000
Average Runtime (sec)	91±34	118±52	176±108	737±647

Table 2.4: **Runtime statistics for GBPML on 8-taxon simulated alignments.**

2.5.2 Evaluation using rRNA concordance test

To evaluate GBPML on real ncRNA datasets, we used the concordance test concept from Rivas & Eddy [79]. We randomly subsampled 16 taxa from a curated rRNA alignment, split the columns randomly in half and ran tree inferences. Instead of having a true tree to compare to, we compared the two trees from the half alignments and calculated tree metrics. The idea was that if the alignments were sufficiently long and contained enough information, inferred trees should concur. While there is no guarantee that any concurring half trees imply the underlying true tree, this may be the best we can do for real datasets. The 16S and 23S rRNA alignments were curated by the Comparative RNA Website [12] and available in the supplement data from Rivas & Eddy [79]. Because we use structure information, we created splits such

Name	Processing	Avg. single (nt)	Avg. paired (bp)	Avg. seq id (%)	Avg. indel (%)
Archaea	preprocessing	854	451	71	18
	postprocessing	554	373	73	3
Bacteria	preprocessing	2,156	469	68	53
	postprocessing	539	336	72	5
Chloroplast	preprocessing	1,568	444	73	39
	postprocessing	573	373	77	3
Eukaryote	preprocessing	6,182	489	57	75
	postprocessing	828	302	65	11
Mitochondria	preprocessing	3,822	447	49	78
	postprocessing	467	225	56	9

Table 2.5: **Alignment statistics for 16S rRNA concordance test pre- and post-processing.** For each 16S rRNA category, we subsampled 100 16-taxon alignments. Post-processing removed (1) all single or paired columns containing ambiguous bases, (2) single columns with $\geq$ 70% gaps, (3) paired columns with $\geq$ 50% non-canonical base pairs. Input to tree programs were post-processed alignments.

that both halves have the same number of single and paired columns. In addition, we pre-processed the halved alignments by (1) removing all single and paired columns with at least one ambiguous base; (2) removing all single columns where more than 70% of the bases were gaps; (3) removing all paired columns where more than 50% of the base pairs were non-canonical pairs (i.e. not A·U, G·C, or G·U pairs; pairs with gaps are also non-canonical). We did (3) because if most base pairs from the subsampled alignment column pair were not canonical pairs, it was likely that they were no longer paired in the subsampled alignment. The postprocessing removed a significant number of gapped single columns (60-80%) and a small number of paired columns (15%-50%), reducing the indel fraction to 0.03-0.11 (Table 2.5, Table 2.6).

For each rRNA gene (16S rRNA or 23S rRNA) and kingdom (Archaea, Bacteria, Chloroplast, Eukaryote, Mitochondria), we subsampled 100 sixteen-taxon alignments and ran the concordance test with GBPML, GBPML_nobp, Pfold, DNAML-ϵ, and RAxML; RAxML was run with secondary structure input and the S16A paired model. We plotted the tree nBSDs (Figure 2.3, Figure 2.4) and for each

Name	Processing	Avg. single (nt)	Avg. paired (bp)	Avg. seq id (%)	Avg. indel (%)
Archaea	preprocessing	1,636	855	66	10
	postprocessing	1,296	705	68	3
Bacteria	preprocessing	2,817	869	66	36
	postprocessing	1,172	656	70	4
Chloroplast	preprocessing	2,543	884	70	32
	postprocessing	1,244	674	74	4
Eukaryote	preprocessing	6,923	1,066	50	59
	postprocessing	1,892	768	59	15

Table 2.6: **Alignment statistics for 23S rRNA concordance test pre- and post-processing.** Same post-processing procedure as Table 2.6.

Program	DNAML-ϵ	GBPML	GBPML_nobp	Pfold	RAxML
DNAML-ϵ		0.95	0.13	4.5e-04	4.2e-31
GBPML	0.49		3e-07	2.4e-05	1.3e-39
GBPML_nobp	0.87	1		0.069	5.3e-31
Pfold	1	1	0.93		7.7e-46
RAxML	1	1	1	1	

Table 2.7: **Significance statistics on the concordance tests for 16S rRNA (Archaea, Bacteria, Eukaryote, Chloroplast, Mitochondria), 16-taxon alignments.** The p-values for $<i, j>$ were calculated using Wilcoxon test (paired, alternative hypothesis $i < j$) using the nBSDs. Statistically significant (p-value $<$ 0.05) comparisons are highlighted in yellow.

pair of programs, the paired Wilcoxon test p-values (Table 2.7, Table 2.8). Except for 16S eukaryotes, GBPML always performed better or at least equivalently when secondary structure information was used. GBPML was not statistically different from DNAML-ϵ for 16S rRNA, but had better concordance for 23S rRNA (p-value: $9.2\text{x}10^{-6}$). In particular, GBPML performed better than DNAML-ϵ on the 16S rRNA dataset for archaea (Figure 2.3a), bacteria (Figure 2.3b), but was not better or worse for eukaryote (Figure 2.3d), chloroplast (Figure 2.3c), and mitochondria (Figure 2.3e). For the 23S rRNA dataset, GBPML performed better than DNAML-ϵ for archaea (Figure 2.4a), bacteria (Figure 2.4b), eukaryote (Figure 2.4d), but was

not better or worse for chloroplast (Figure 2.4c).

It is not clear why GBPML did worse on some of the datasets, but a possible explanation might be that our trained evolutionary model was less similar to the true evolutionary model: GBPML and Pfold both use fixed evolutionary models whereas DNAML-ϵ and RAxML estimate the parameters during tree optimization.

2.5.3 Concordance test on the lysine riboswitch family

We ran 20 iterations of the same concordance test (random 16-taxon alignment) on the lysine riboswitch seed alignment (Rfam ID: RF00168). The lysine riboswitches are significantly shorter than the rRNAs (avg. sequence length: 163 nt). GBPML performed better than all other programs (Figure 2.5).

2.5.4 Case study: a TPP riboswitch phylogenetic tree

We evaluated GBPML on a curated TPP riboswitch alignment. The TPP riboswitch is a structured ncRNA that directly binds to thiamine pyrophosphate (TPP) to regulate gene expression [107]. We chose this ncRNA as a test dataset for several reasons: (1) it is at present the only riboswitch to be found in all three domains of life [91], and (2) the X-ray crystal structure of the TPP riboswitch aptamer has been solved. We randomly selected 7 species from the Rfam alignment (RF00059) whose species accession number appeared multiple times (Figure 2.6). The species were: *Oryza sativa*

Program	**DNAML-ϵ**	**GBPML**	**GBPML_nobp**	**Pfold**	**RAxML**
DNAML-ϵ		1	0.69	0.27	9.7e-22
GBPML	9.2e-06		2.9e-10	2.6e-08	6.2e-35
GBPML_nobp	0.31	1		0.047	1.9e-25
Pfold	0.73	1	0.95		1.9e-33
RAxML	1	1	1	1	

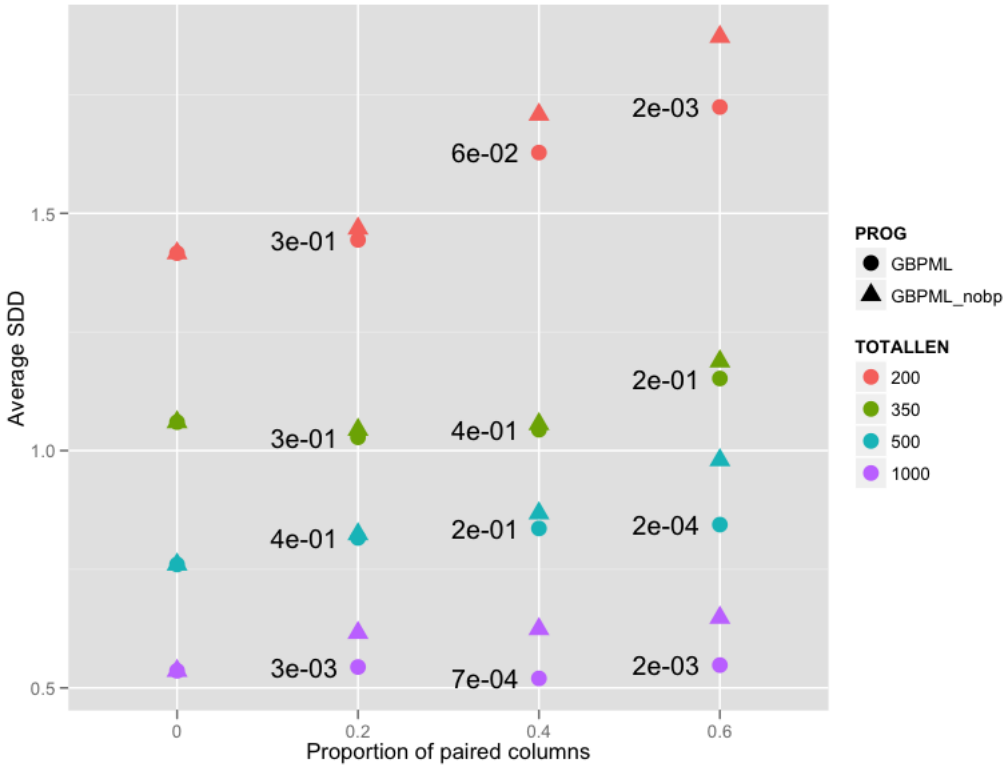
Table 2.8: **Significance statistics on the concordance tests for 23S rRNA (Archaea, Bacteria, Eukaryote, Chloroplast), 16-taxon alignments.** The p-values for $<i, j>$ were calculated using Wilcoxon test (paired, alternative hypothesis $i < j$) using the nBSDs. Statistically significant (p-value < 0.05) comparisons are highlighted in yellow.

Japonica Group (Eukaryota, rice), *Aspergillus oryzae* (Eukaryota, fungi), *Thermoplasma acidophilum* (Archaea), *Mycobacterium tuberculosis* (Actinobacteria), *Streptomyces coelicolor* (Actinobacteria), *Bacillus subtilis* (Firmicutes), *Bacillus lincheniformis (Firmicutes).* The Rfam alignment did not provide genomic locations of the sequences, so we BLASTed them and found the following to be from different genomic locations of the same species (gene paralogs): four *B. subtilis* (avg. pairwise sequence id: 56%), two *T. acidophilum* (75%), three *S. coelicolor* (57%). The two *A. oryzae sequences* were possibly sequencing duplicates as BLAST mapped them back to two independent sequencing attempts of the thiC gene (although the sequence identity is only 68%). *O. sativa* 1, 2, and 4 mapped back to the same chr3 region with slight differences in the first 1-15bp, *O. sativa.3* returned no BLAST results[2]; average pairwise sequence identity between *O. sativa* sequences was 63% with most differences between *O. sativa* 3 and others. The curated alignment of 17 sequences consisted of 58 single columns and 28 paired columns with an average pairwise sequence identity of 49% (Figure 2.6).

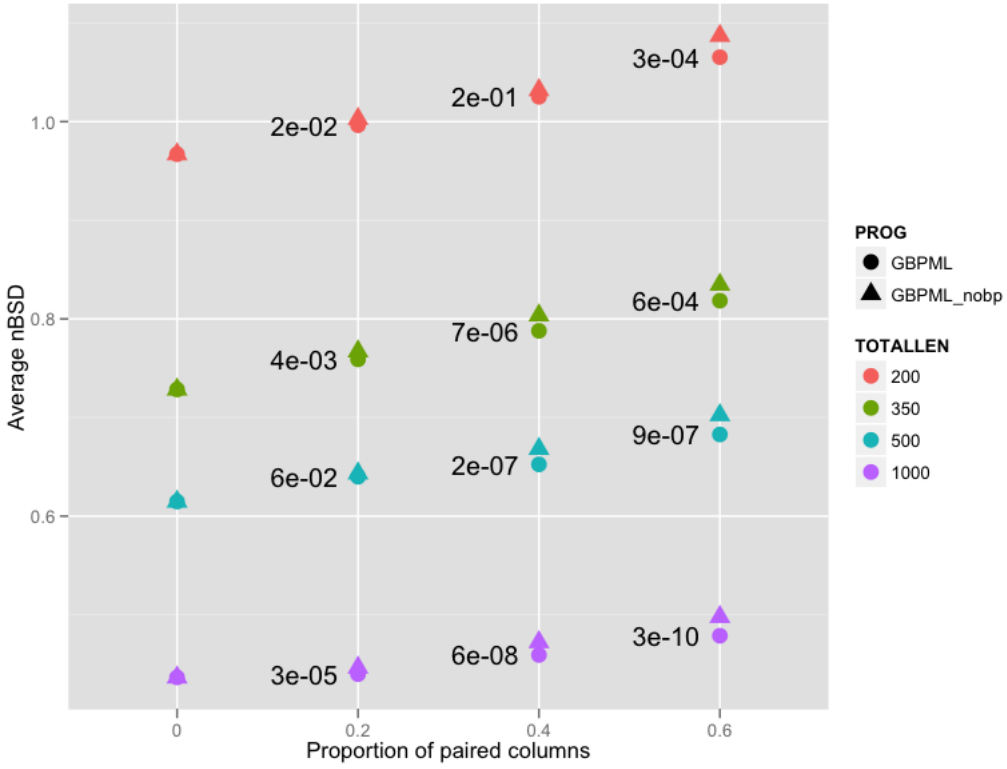
GBPML and GBPML_nobp produced very similar trees with slight disagreement in the placement of *M. tuberculosis* and an *S. coelicolor* subtree, but successfully grouped all other paralogs and duplicates together (Figure 2.7a, Figure 2.7b). Specifically, the eukaryotes (*O. sativa*, *A. oryzae*) formed an outgroup, and *O. sativa* 3, the sequence which BLAST could not map, was an outlier in the *O. sativa* subtree; the alignment shows that the eukaryotes are missing the S4 stem loop as supporting evidence of it being an outgroup. The single *B. lincheniform* sequence was within the *B. subtilis* subtree, which may be biologically valid as both species belong to the Bacillus genus; the alignment shows that the Bacillus sequences are similar in the S1, S2, S4, and S5 regions. The *T. acidophilum* sequences, the only archaean in the alignment, are mixed in with the *S. coelicolor* sequences; the alignment shows that they share similarities in the S2, S3, and S5 regions, and that *S. coelicolor*, an unusually GC-rich species (TPP riboswitch sequence GC content: 65-72%), is not

[2]The only match to the non-redundant database found was to itself (AK119882.1), a 3548 bp cDNA clone. BLAST against the *O. sativa* reference genome showed no results.

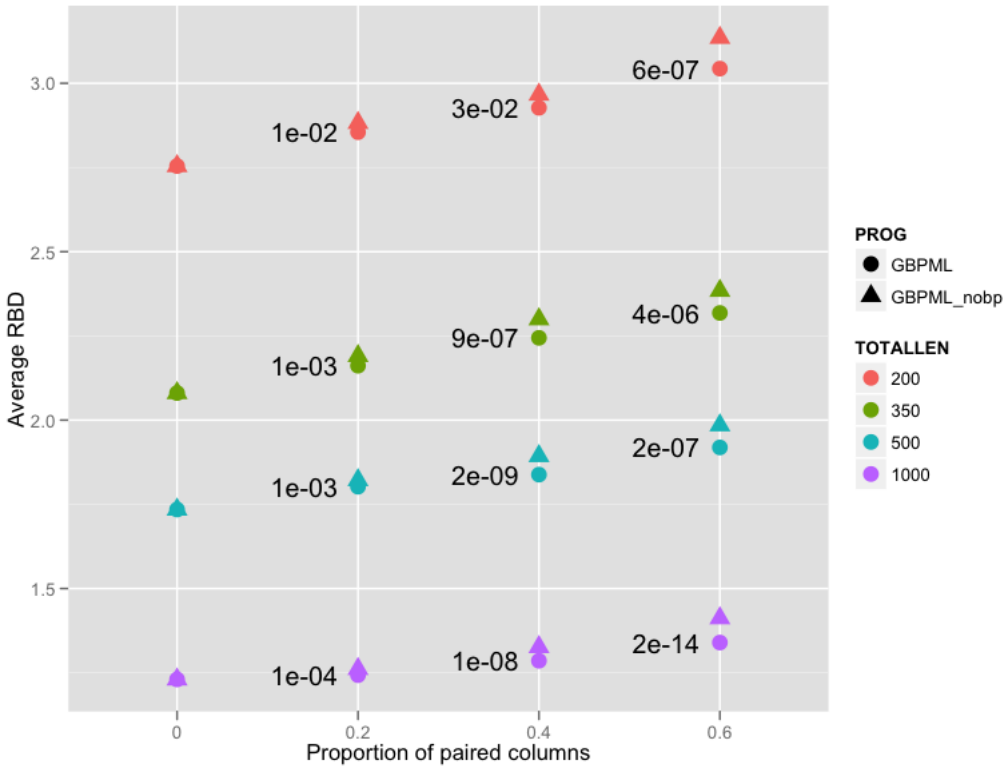
particularly more similar to the other bacterial sequences, which may explain why it was on the outer edge of the bacteria/archaea subtree. Running GBPML without the gap model (treat as missing data) resulted in misplacement of several Bacillus sequences (Figure 2.7c). DNAML-ϵ , RAxML, and Pfold produced trees with similar bacterial groupings but did not group the eukaryotic sequences together (Figure 2.7d, Figure 2.7e, Figure 2.7f).



(a) average SDD.

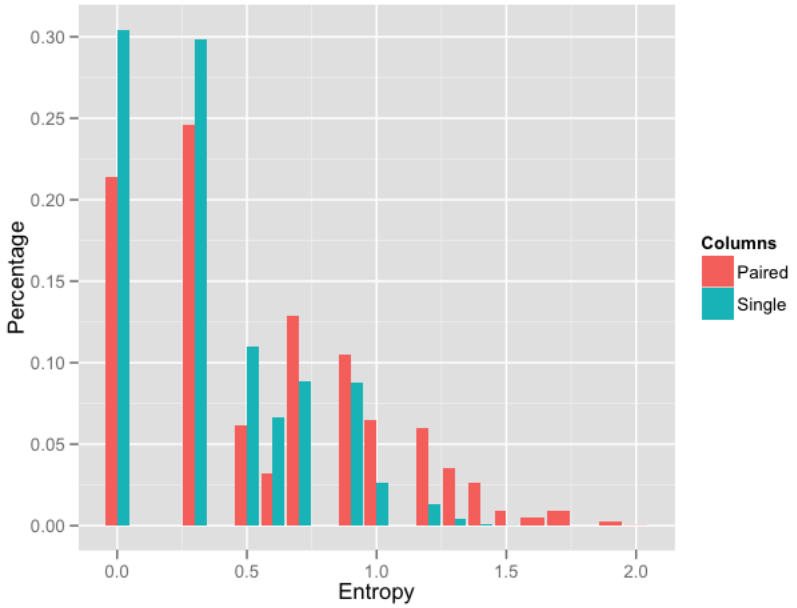


(b) average nBSD.

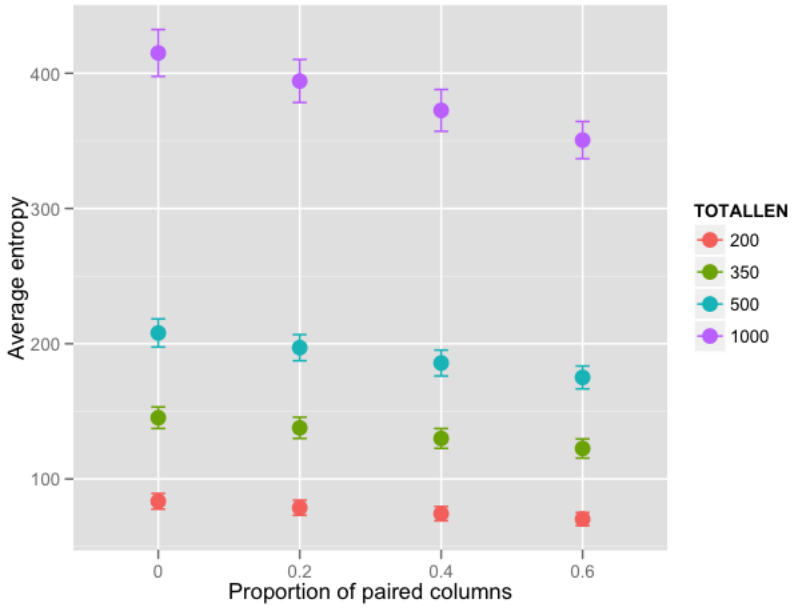


(c) average RBD.

Figure 2.1: **Comparison of GPBML with and without structure on simulated alignments.** Alignments were generated from 8-taxon trees using our evolutionary model with varying total alignment lengths and paired column proportions. We calculated the average distances of the inferred trees to the true trees using (a) SDD. (b) nBSD. (c) RBD. Lower SDD, nBSD and RBD means better tree inference. Numbers shown are Wilcoxon test p-values of GBPML versus GBPML_nobp for a given (*total alignment length, paired proportion*); test is paired with alternative hypothesis: GBPML < GPBML_nobp.

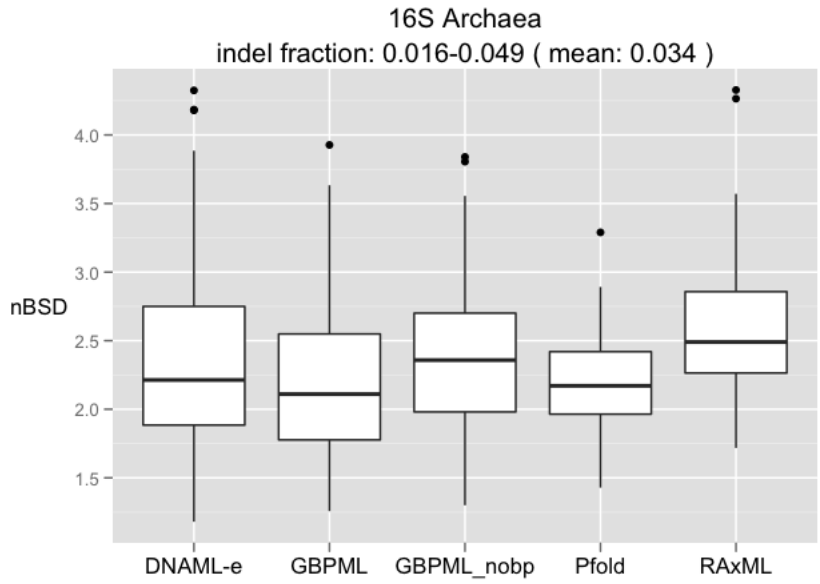


(a) Histogram of paired versus single columns entropies. The average entropy from paired columns (0.57) is higher than unpaired columns (0.37).

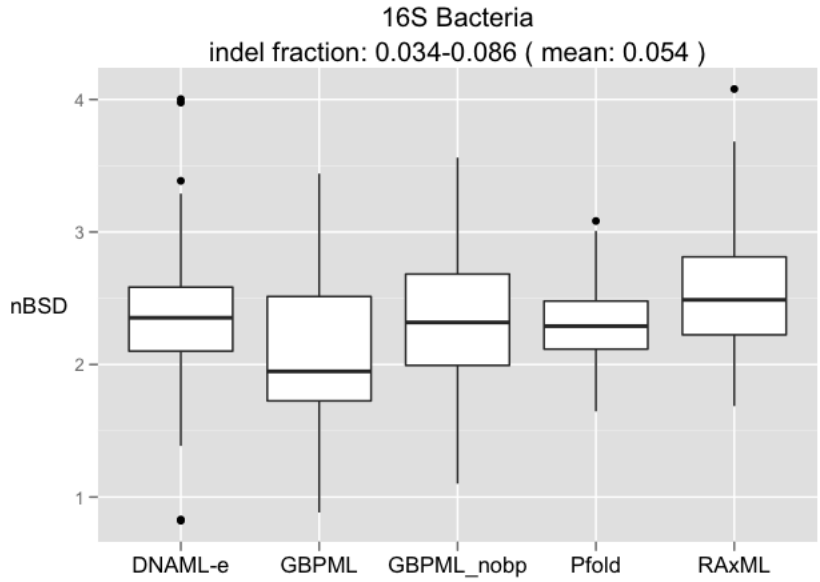


(b) Average total entropy for different total alignment lengths and paired column proportions.

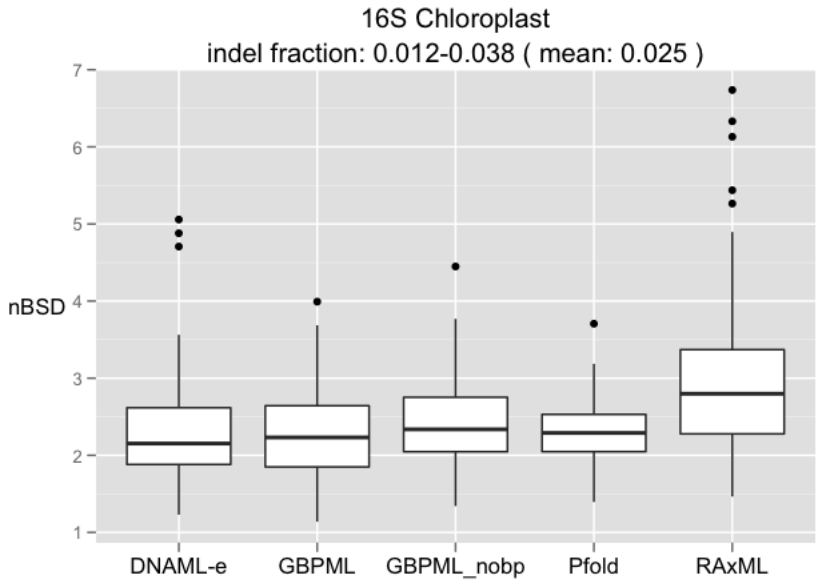
Figure 2.2: **(a) Paired columns are more informative than unpaired columns.** We tallied the entropies for each single column and each pair of paired columns (treating a base pair as a single entity). An entropy of 0 means a 100% identity across the column(s) that provides no information. **(b) For a fixed total alignment length, the average total entropy decreased with increasing paired column proportion.**



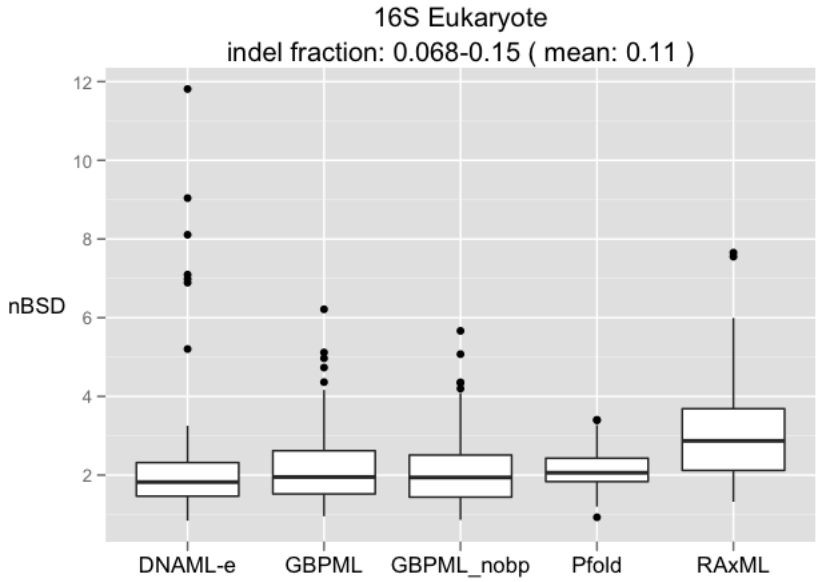
(a) 16S rRNA, 16-taxon alignments, archaea. GBPML performed better than GBPML_nobp (p-value: $8\text{x}10^{-8}$), DNAML-ϵ (p-value: 0.017), and RAxML (p-value: $3.5\text{x}10^{-7}$); GBPML was not statistically different from Pfold.



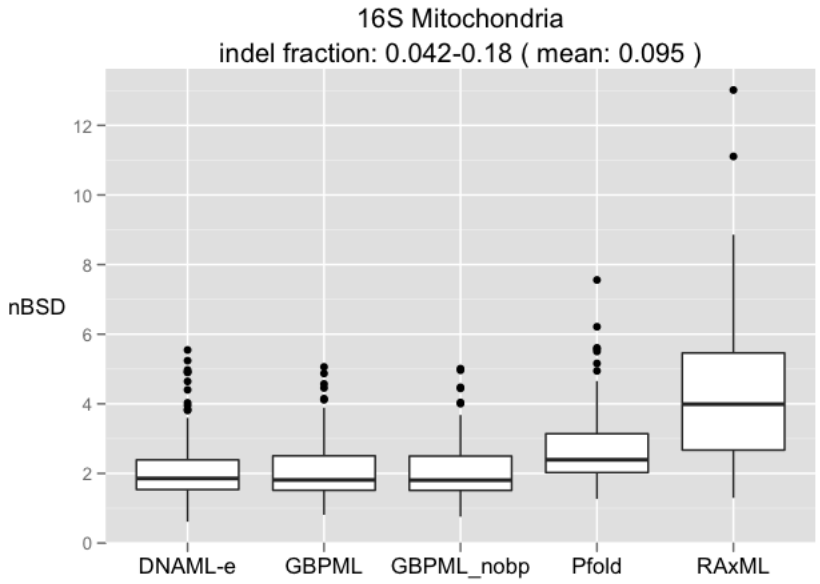
(b) 16S rRNA, 16-taxon alignments, bacteria. GBPML performed better than GBPML_nobp (p-value: $1.8\text{x}10^{-8}$), DNAML-ϵ (p-value: $4.3\text{x}10^{-5}$), Pfold (p-value: $3.6\text{x}10^{-4}$), and RAxML (p-value: $4.5\text{x}10^{-8}$).



(c) 16S rRNA, 16-taxon alignments, chloroplast. GBPML performed better than GBPML_nobp (p-value: 0.011) and RAxML (p-value: $9.6\text{x}10^{-8}$); GBPML is not statistically different from Pfold and DNAML-ϵ (p-value > 0.05).

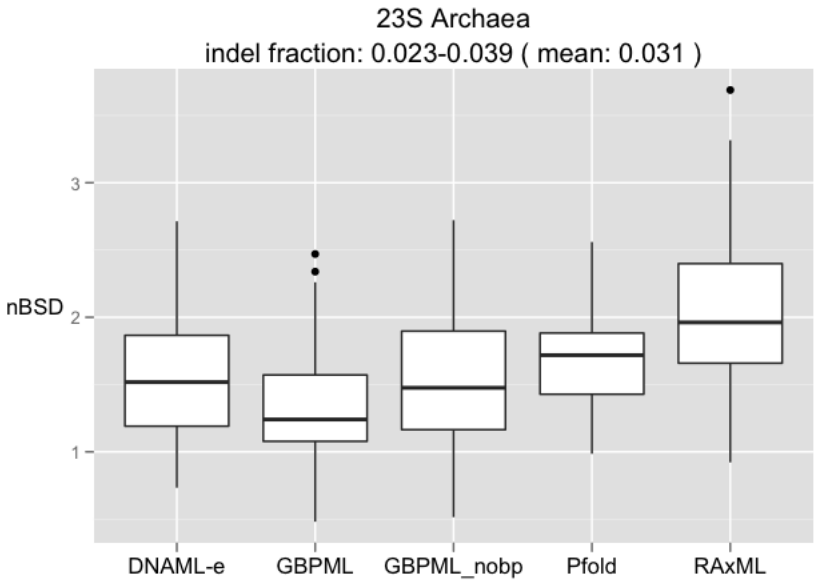


(d) 16S rRNA, 16-taxon alignments, eukaryote. GBPML_nobp performed better than GBPML (p-value: $4.6\text{x}10^{-3}$ and was not statistically different from Pfold or DNAML-ϵ . GBPML and GBPML_nobp both performed better than RAxML.

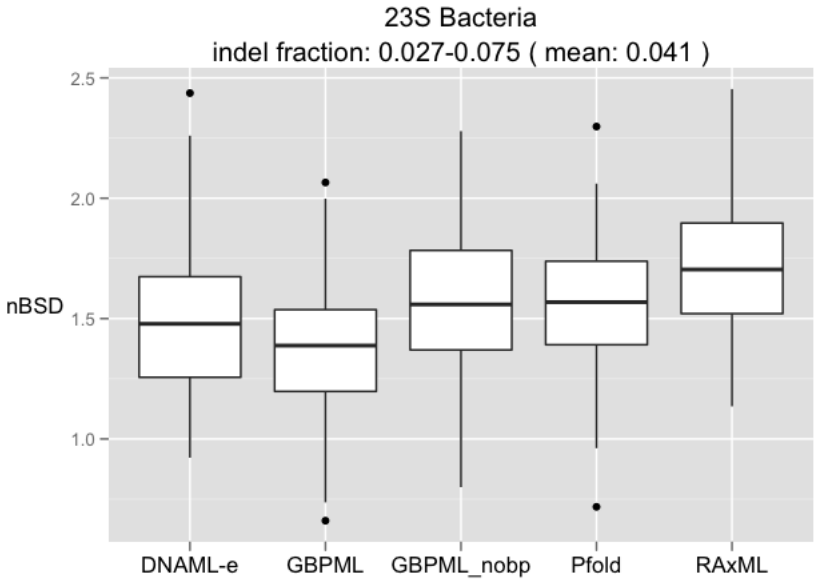


(e) 16S rRNA, 16-taxon alignments, mitochondria. GBPML, GBPML_nobp, and DNAML-ϵ were not statistically different (p-value > 0.05); all three performed better than Pfold and RAxML.

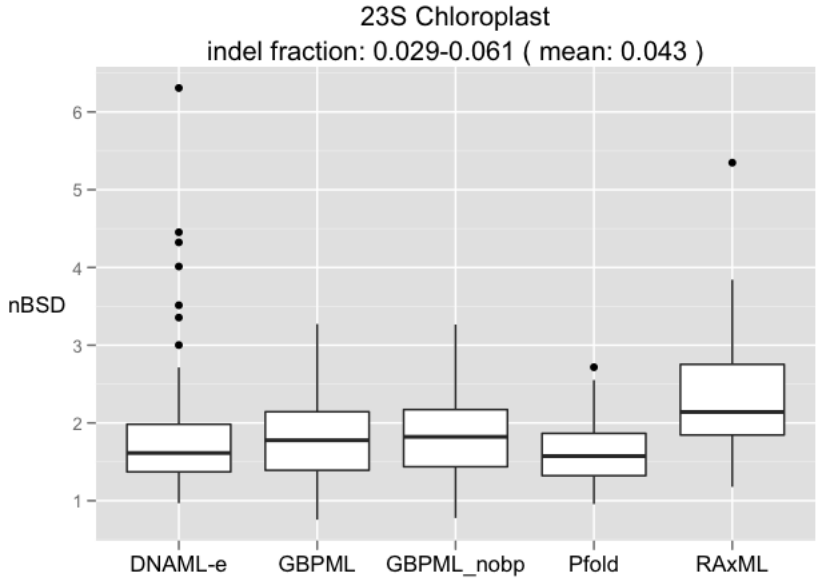
Figure 2.3: **Concordance test using 16S rRNA.** We randomly subsampled 100 sixteen-taxon alignments from a curated 16S rRNA alignment, randomly split the alignment in halves and ran tree inference programs. For each input/program, we computed the normalized branch score distance (nBSD) between each pair of trees.



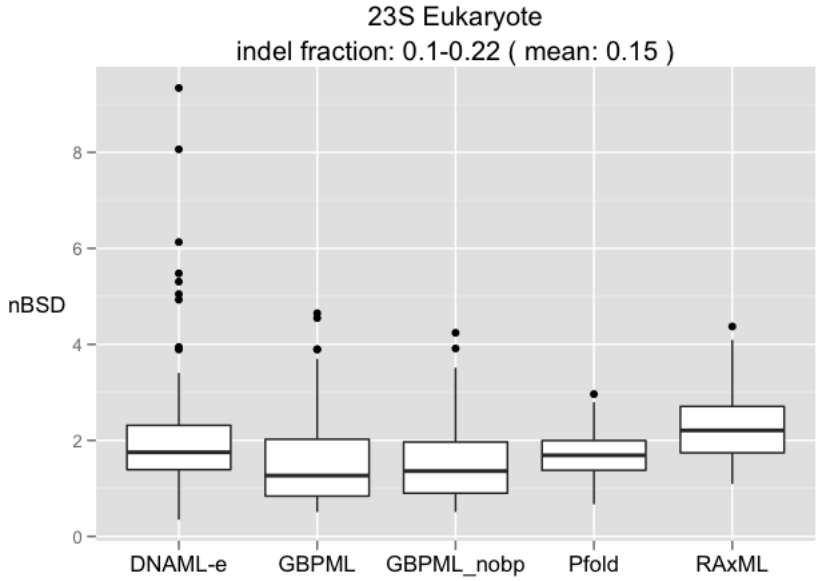
(a) 23S rRNA, 16-taxon alignments, archaea. GBPML performed better than GBPML_nobp (p-value: $4.4\text{x}10^{-8}$), DNAML-ϵ (p-value: $3.3\text{x}10^{-4}$), Pfold (p-value: $2\text{x}10^{-9}$), and RAxML (p-value: $6.1\text{x}10^{-16}$).



(b) 23S rRNA, 16-taxon alignments, bacteria. GBPML performed better than GBPML_nobp (p-value: $1.1\text{x}10^{-12}$), DNAML-ϵ (p-value: $1.7\text{x}10^{-3}$), Pfold (p-value: $6.2\text{x}10^{-7}$), and RAxML (p-value: $1.6\text{x}10^{-12}$).



(c) 23S rRNA, 16-taxon alignments, chloroplast. GBPML, GBPML_nobp and DNAML-ϵ were not statistically different (p-value > 0.05). Pfold was statistically better than GBPML (p-value: 0.028) and GBPML_nobp (p-value: 0.007). GBPML, GBPML_nobp, DNAML-ϵ , Pfold were statistically more significant than RAxML (p-value < 0.05).



(d) 23S rRNA, 16-taxon alignments, eukaryote. GBPML and GBPML_nobp were not statistically different. Both GBPML and GBPML_nobp performed better than DNAML-ϵ , Pfold, and RAxML.

Figure 2.4: **Concordance test using 23S rRNA.** We randomly subsampled 100 sixteen-taxon alignments from a curated 23S rRNA alignment, randomly split the alignment in halves and ran tree inference programs. For each input/program, we computed the normalized branch score distance (nBSD) between each pair of trees.

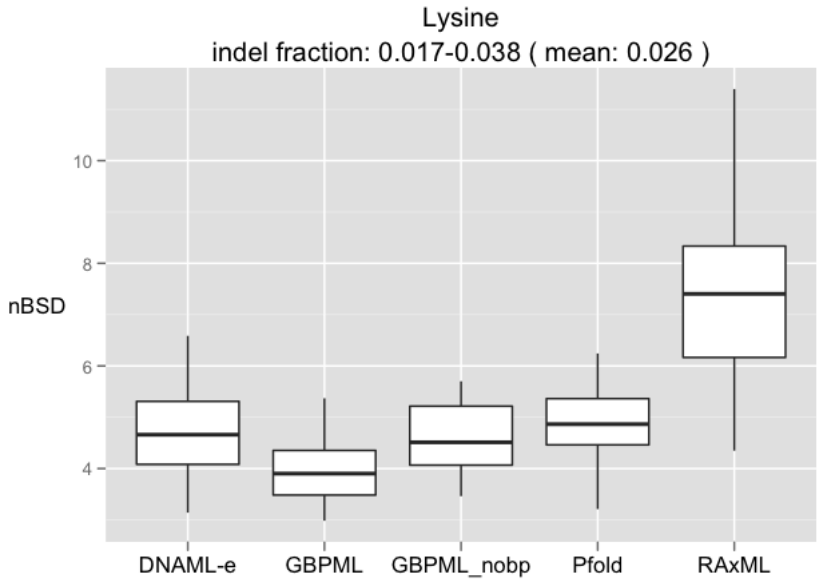


Figure 2.5: **Concordance test using the seed alignment for the lysine riboswitch from Rfam.** We randomly subsampled 20 sixteen-taxon alignments and ran the concordance test in the same manner as the rRNA concordance tests. GBPML performed better than GBPML_nobp (p-value: 9.5x10^{-6}), DNAML-ϵ (p-value: 2.9x10^{-6}), Pfold (p-value: 1.8x10^{-5}), and RAxML (p-value: 9.5x10^{-7}).

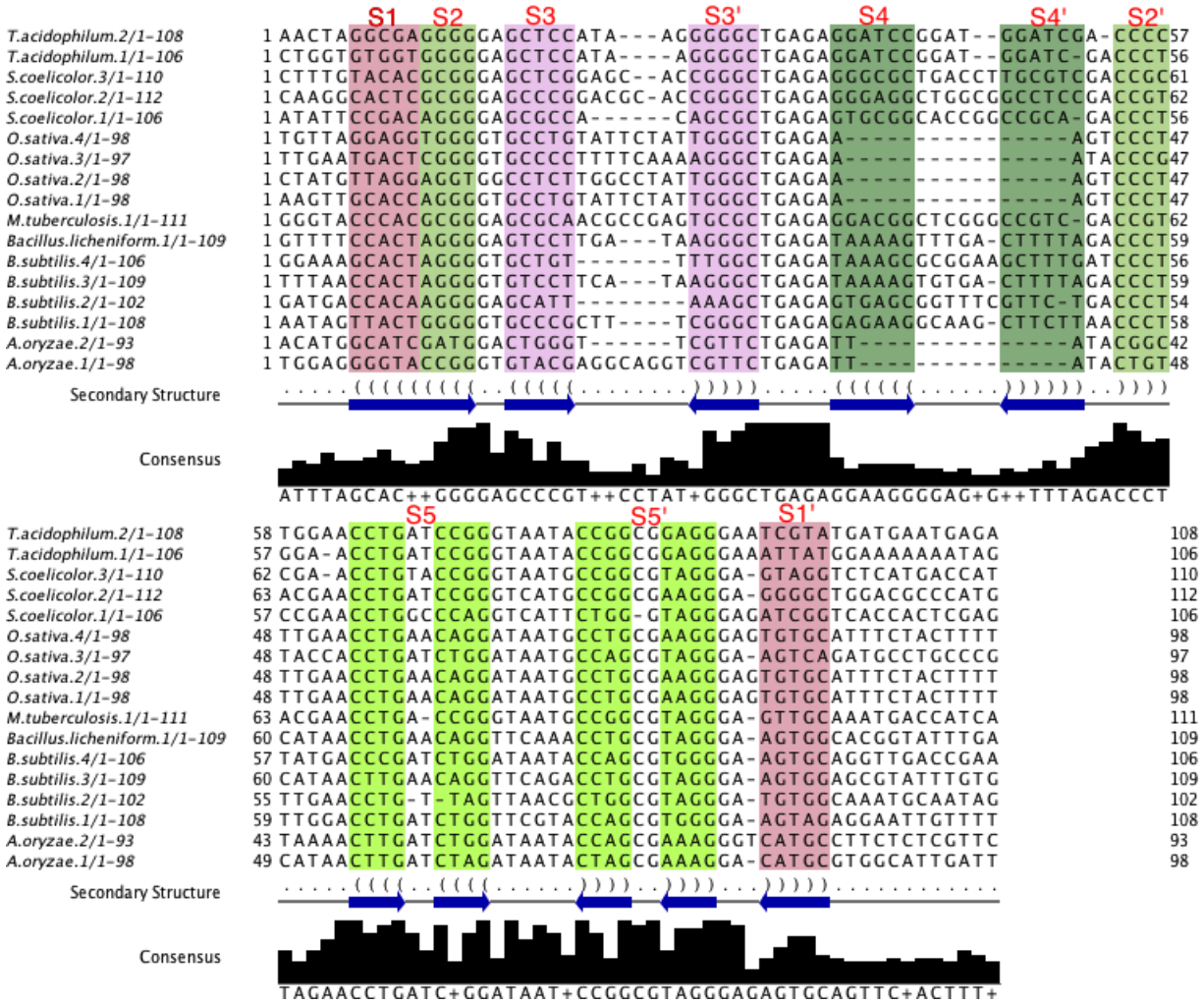


Figure 2.6: **Curated TPP riboswitch alignment from Rfam (ID: RF00059).** Species: *O. sativa* (Eukaryota, rice), *A. oryzae* (Eukaryota, fungi), *T. acidophilum* (Archaea), *M. tuberculosis* (Actinobacteria), *S. coelicolor* (Actinobacteria), *B. subtilis* (Firmicutes), *B. lincheniformis (Firmicutes).*

2.5.5 Testing for alternatively trained evolutionary models

The 17-vertebrate training dataset is an extensive set of 264 different ncRNA alignments. The resulting trained model could be too generalized because of the diversity of the families or too specific to eukaryotic species, but without an equally extensive training set from other domains it is hard to say. Obtaining a bacteria-specific training set is hard because the species tree is less well-defined and there are few ncRNAs present in a wide range of bacterial species. Nevertheless, we randomly sampled 50 bacterial species from the All-Species Living Tree Project [113], and used the 16S rRNA species tree (constructed using RAxML) and their 16S rRNA sequence alignment to train an alternative evolutionary model.

We ran concordance tests with the alternative model on 16S bacteria but did not see a significant difference between the 17-vertebrate GBPML model and the alternative one (Wilcoxon test p-value for SDD, nBSD, and RBD are all > 0.05). Though it is important to find the most proper training set that would yield the best evolutionary model, for the purpose of evaluating the value of structure information, we find the 17-vertebrate model to be sufficient.

2.6 Discussion

We showed that using structure information, a more realistic gap model, and a maximum likelihood approach improved phylogenetic tree inference. In our simulated dataset, treating paired columns separately from unpaired columns when calculating the likelihoods improved tree estimation. Entropy calculation showed that a paired column is more informative than an unpaired column but is not twice as informative because the pairings are dependent. In varying the alignment length, we showed that longer alignments have more information and improve tree estimation. We got mixed results in the rRNA concordance test, but overall GBPML was better when structure information was used and performed better in many cases than Pfold, DNAML-ϵ , and RAxML. On the manually curated TPP riboswitch dataset, GBPML clustered all the paralogs together and correctly clustered the eukaryotes as an outgroup.

Here we discuss some potential improvements for future work:

(1) A better indel model: The rRNA concordance test results show that a more complex indel model is needed for ncRNAs with a large number of insertions and deletions. The advantage of using structure information (GBPML) is reduced when total indel percentage $> 1\%$. Incorporating a gap model that allows for gap insertions like DNAML-ϵ will likely improve tree estimation, but may slow down computation even further.

(2) Technical and algorithmic speedup: The usability of GBPML would benefit from runtime speedup. Several technical tricks were described in RAxML to speed up likelihood calculation which we did not employ. We also limited the number of parallelized threads in subtree rearrangement to 5, which could be increased to as many CPUs as are available per machine. Algorithmically, the bottleneck is in the optimization of the tree branches after subtree rearrangement. We used numerical approximation to obtain the derivative and this was the most time-consuming step. We have not investigated whether there are numerical solutions to optimizing branch lengths given the non-reversibility of our gap model. It is also possible that when there are few indels in the alignment, we can heuristically ignore gaps as they would not change the likelihood much.

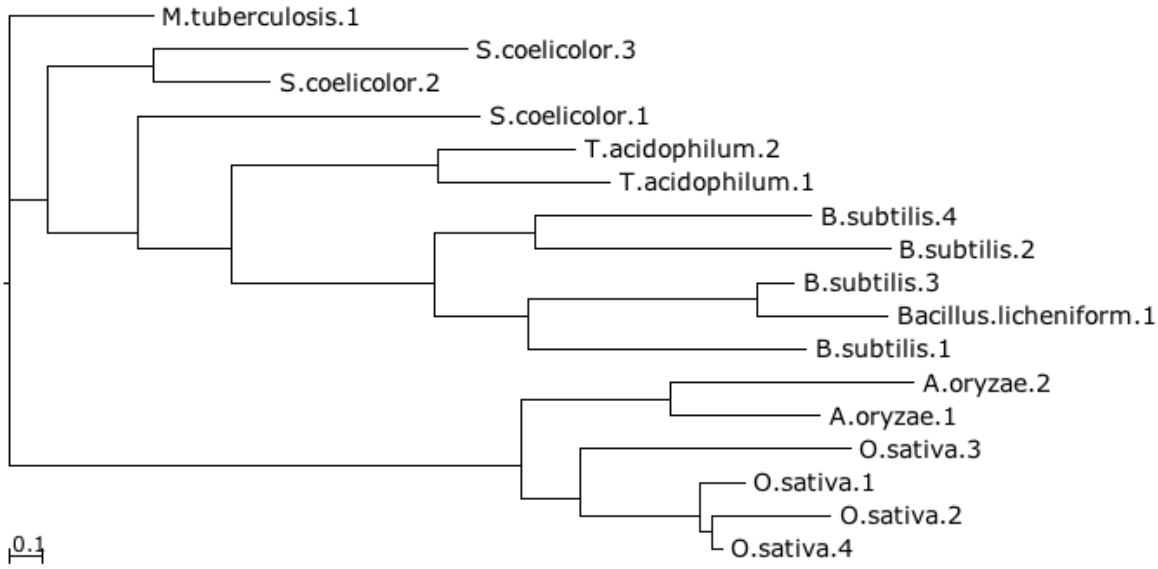
(3) Statistical confidence: Adding bootstrap values to the inferred trees and seeing how different starting trees affect the tree search would also improve GBPML. In this study, we used the Pfold neighbor joining tree as the starting tree. We did not test for different starting trees.

(4) Improved evaluation: Evaluation remains a main challenge in developing good tree inference methods. Simulated data is convenient because both the evolutionary model and true tree are known, but in most cases are oversimplified and provide overly positive results. Real-world data provide realistic evaluations but come with complications. First, only the seed alignments in Rfam are manually curated; the full alignments are automated and can contain false positives and misannotations. Second, many of the predicted ncRNA secondary structures have not been experimentally verified (e.g. through X-ray crystallography) and can provide false base

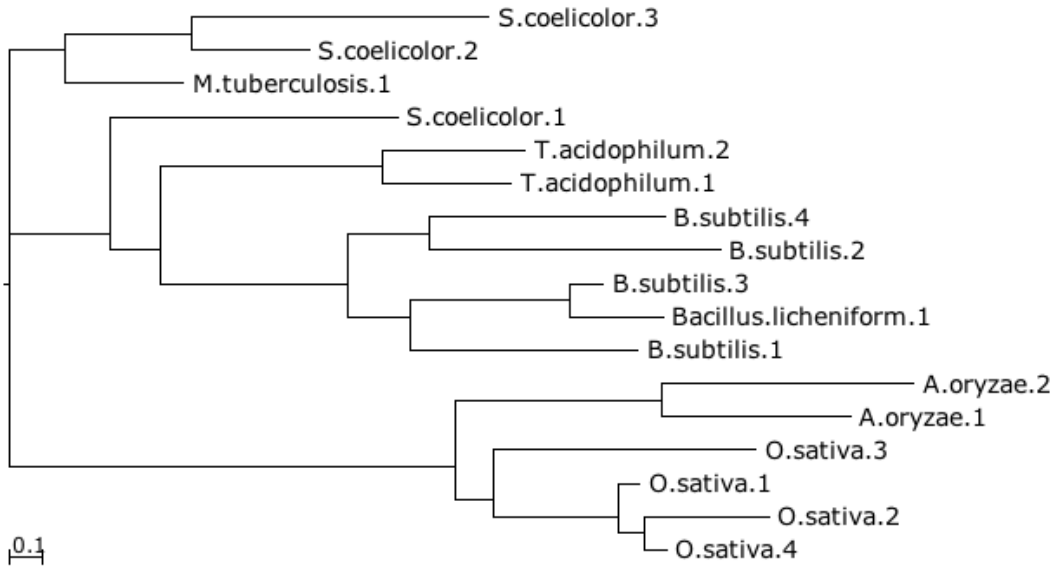
pairings. Finally, even when there is good alignment data, the true tree remains unknown. During the study, we looked at two alternative ways to get a good tree. One was to use the semi-curated species tree from the All-Species Living Tree project. We compared the common species between the LSU (23S) and SSU (16S) tree and found that they differed a lot (SDD: 50 for a 54-taxon tree), so neither could've been used reliably as a true tree. Another approach was to look at the downstream gene tree. For *cis*-regulatory ncRNAs like riboswitches that are directly upstream of their regulated genes, the phylogenetic tree for the ncRNA tree and the gene tree are likely to be the same. Unfortunately it is difficult to automatically extract downstream genes as a lot of the ncRNA sequences come from incomplete genomes with few gene annotations. In addition, inferring the gene tree adds another layer of unreliability and does not apply to all ncRNA families.

As mentioned in Introduction, accurate alignment and structure are not always obtainable, and this strongly affects tree estimation. It's a chicken-and-egg problem between alignment, structure, and tree. Knowing one or two of the components makes predicting the remainder easier. The original Sankoff algorithm and its variants such as stemloc [33] simultaneously predict alignment and structure and do not estimate the tree. EvoFold [73] predicts secondary structure using a given alignment and tree, and is largely applied to predicting ncRNA structures on multi-species alignments for which the phylogenetic tree topology is well understood (e.g. the UCSC 17-vertebrate alignment). Pfold takes a given alignment and predicts structure with inferred trees. All of the above are based on stochastic context free grammar models (SCFGs) that focus on structure and alignment prediction more than tree inference, and are largely driven by ncRNA discovery. On the other end, a completely different set of tree inference algorithms exists that were originally designed to infer trees from protein or genomic sequences. Maximum likelihood-based methods such as DNAML, RAxML, and FastTree, take an input alignment and output a tree without considering structures. The tree inference method we presented in this paper focused on improving tree inference with secondary structure information and a better gap model. To make it even more practical for cases where good alignment

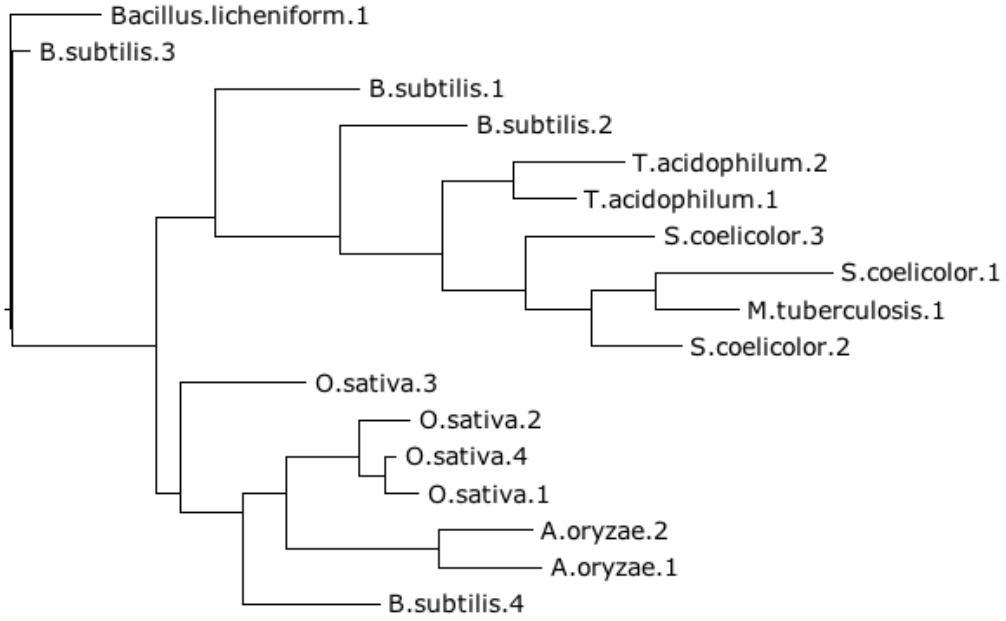
and structure don't exist, an approach where the alignment, structure, and tree are iteratively re-estimated through an EM algorithm would be the next step.



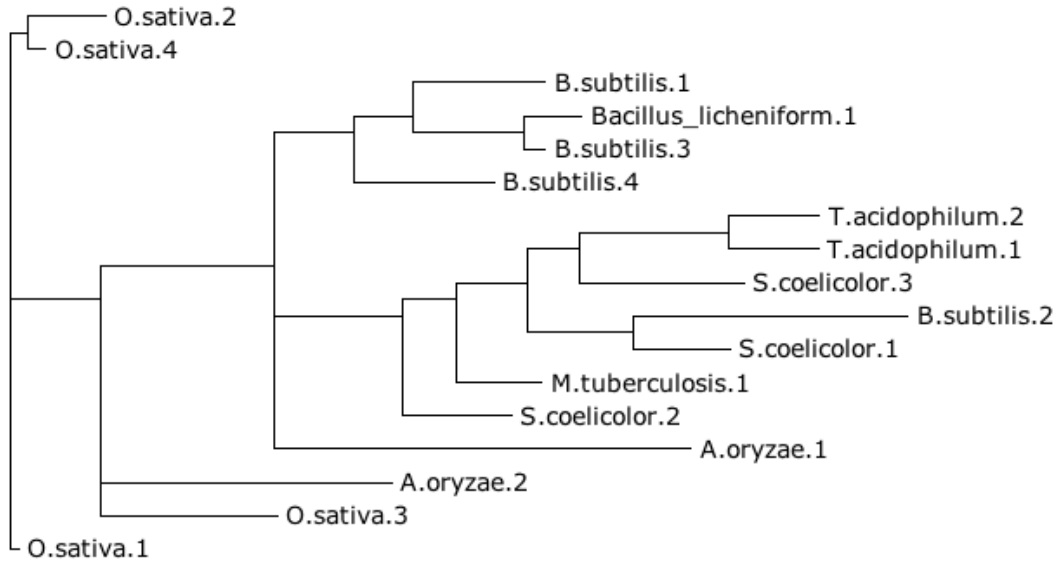
(a) GBPML.



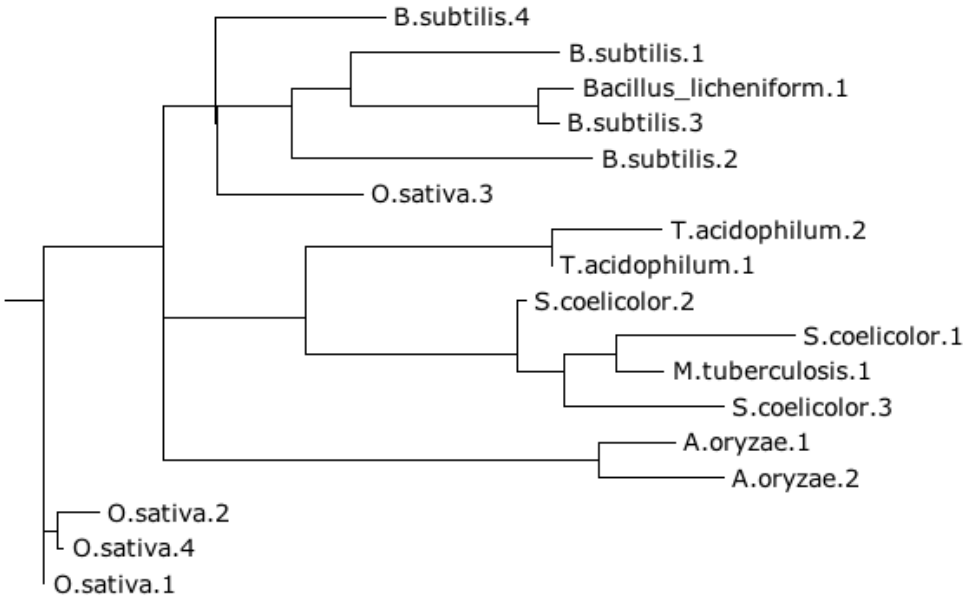
(b) GBPML_nobp.



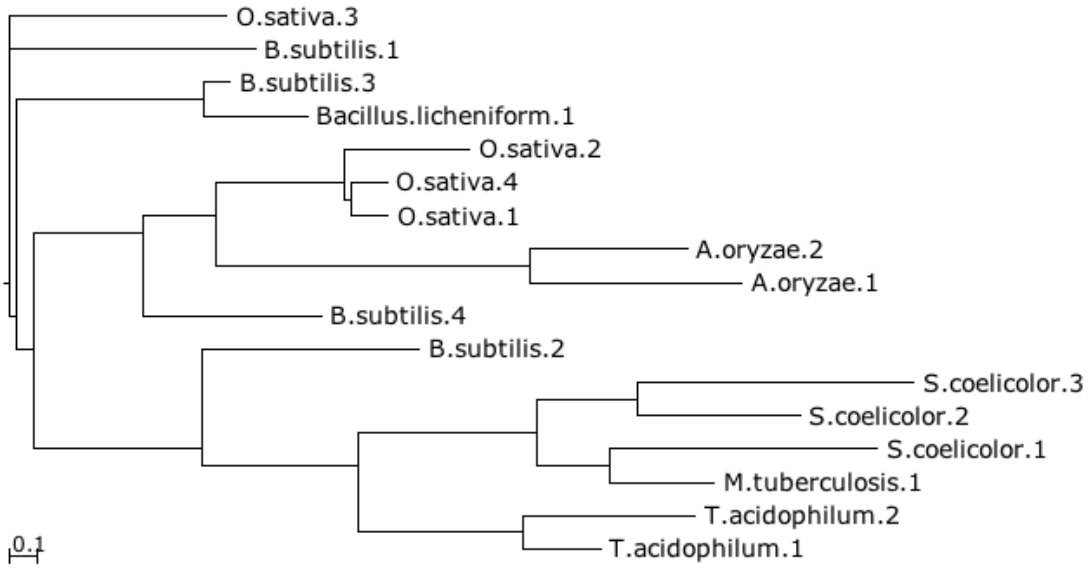
(c) GBPML, gaps treated as missing data.



(d) DNAML-ϵ .



(e) RAxML.



(f) Pfold.

Figure 2.7: **TPP riboswitch tree.** Species: *O. sativa* (Eukaryota, rice), *A. oryzae* (Eukaryota, fungi), *T. acidophilum* (Archaea), *M. tuberculosis* (Actinobacteria), *S. coelicolor* (Actinobacteria), *B. subtilis* (Firmicutes), *B. lincheniformis (Firmicutes).*

Chapter 3

MICROBIAL NUCLEOTIDE SIGNATURE: A PROFILING METHOD FOR THE HUMAN MICROBIOTA

3.1 Introduction

Billions of bacteria live in the gastrointestinal tract and their symbiotic relationship with the host greatly affects human health [44]. There are many ways to characterize the human gut microbiota. Sequencing the 16S rRNA gene is widely used for several reasons [16]: (1) it is present in all bacteria and archaea; (2) it has both conserved regions that can be used as universal primer targets and variable regions that are species-specific; (3) it has been extensively studied and comprehensive databases exist (Appendix B). By sequencing the 16S rRNA gene, one can characterize the diversity and abundance of microbial species in the gut and identify shifts associated with diseases.

Over the past few years, high-throughput sequencing of the 16S rRNA gene has shown associations between the gut microbiota and diseases such as obesity, diabetes, inflammatory bowel disease and cancer [44]. To be able to assess their role in human diseases, studies need to be done prospectively and across different populations with a large enough sample size to encompass the natural variation in the gut microbiota. Increasing sample size means potentially reducing the number of sequences per sample for cost efficiency, but it is not clear whether at low sequencing depth and with short reads that preclude accurate taxonomic identification, the microbial community could be sufficiently represented. In 2007, Liu et al. [50] looked at the effects of different read lengths and 16S rRNA primers using pyrosequencing and found that (1) the choice of primers affected the identifiability of a read even at read lengths $>$ 500 bp, and (2) read identifiability decreased with read length. Since they used pyrosequencing, they tested read lengths as short as 100 bp. For Illumina sequence,

100 bp is often the maximum read length—many samples have been sequenced with 30-50 bp reads. With lower sequence cost and higher read yield per run, Illumina is gradually becoming the favored platform.

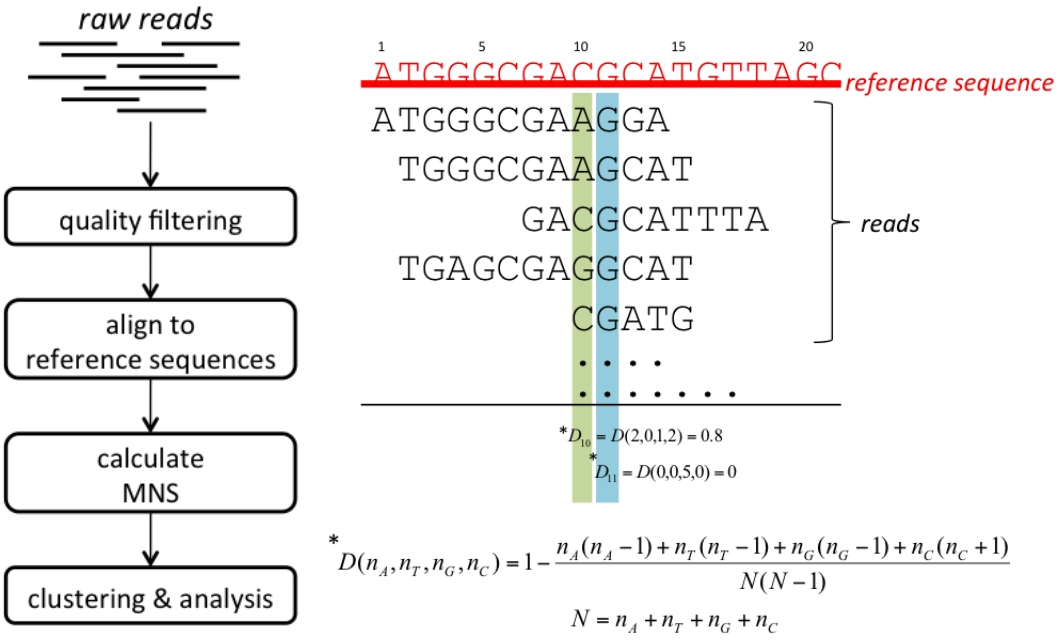
In this chapter, we develop a profiling method for short Illumina reads. We describe Microbial Nucleotide Signatures (MNS), a characterization of short reads based on nucleotide diversities. We apply MNS to a 51bp Illumina dataset and show that MNS can differentiate between the gut microbiota of 9 healthy individuals with as few as 40,000 reads per sample. While the eventual goal of gut microbiota studies is to find out what species are present, our approach provides a cheap way of characterizing samples that can be applied as a first-pass screening on large populations. Once differences between samples have been identified and grouped, samples can be selected from each group for further, deeper sequencing.

3.2 Methods

Figure 3.1 outlines the microbial nucleotide signatures process: reads are filtered by quality scores and length then aligned to a reference 16S rRNA gene database. A diversity index is calculated to represent the nucleotide diversity and evenness independently at each position in the sequenced 16S rRNA gene region. This method focuses on one variable region of the 16S rRNA gene but may be expanded to include the entire gene or other genes of interest. The total set of diversity indices—a 1-d vector—is the microbial nucleotide signature (MNS) of the sample.

In the following sections, we describe how the reference database is created (section 3.2.1), how reads are aligned (section 3.2.2), and how we calculate the MNS given the alignment (section 3.2.4). Initial read filtering is described in the Materials section (section 3.3).

The computational bottleneck here is aligning reads to the reference database. This step can be easily parallelized and be completed for an entire Illumina run (80 million reads) in several hours. After alignment is done, computing and clustering the MNSs takes only several minutes.



(a) MNS diagram. (b) MNS concept.

Figure 3.1: **Diagram and concept of microbial nucleotide signatures (MNS) analysis.** MNS is a vector of nucleotide diversity indices calculated independently for each position in the sequenced region. Here nucleotide diversity is calculated using Simpson's Index, although alternative indices such as the Entropy Index (see section 3.2.4) could also be used.

3.2.1 Creating a reference sequence database

The reference sequence database is a multiple sequence alignment of representative gut bacterial species. Accuracy of read placement is dependent on how well the database represents the gut microbiota. The more species we include in the database, the more likely a read will be correctly aligned. However, as the database size grows, both runtime and memory requirements for the aligners increase. We created a gut bacterial reference database by selecting non-redundant sequences from SILVA 104 that were extracted from fecal samples, which included both cultured and uncultured sequences. We obtained 87,295 16S rRNA gene sequences that were mostly Firmicutes and Bacteroidetes (the two most dominant phyla in the gut). Although creating this database from existing knowledge of the gut species may cause reads

from novel species to be misaligned, we used this approach to maximize alignment efficiency. In our own dataset, we were able to align 60-90% of the reads, indicating that our reference database is representative of the gut microbiota.

3.2.2 Aligning reads using BowTie and BLAST

BowTie [46] is a fast, memory-efficient aligner designed to align reads with very few mismatches to reference sequences. We run BowTie with the maximum number of mismatches allowed (3 mismatches) against the gut bacterial reference database. Because BowTie uses a greedy, randomized search algorithm to find non-exact matches, unless an exhaustive search is done, it may either fail to find the optimal match or not report a match at all. Since an exhaustive search is time-consuming, we set the parameter to `-k 1` to report the first database match for a given read. This less stringent criterion does not affect the MNS of a sample significantly (section 3.2.3). To recover the small percentage of reads with at most 3 mismatches but missed by BowTie, we align the remaining unaligned reads through the same database, using the slower but more sensitive BLAST [11]. We do not allow gaps in BLAST and only the highest scoring match is reported. Both BowTie and BLAST can be parallelized to reduce runtime. For a full Illumina dataset ($\sim$80 million reads), splitting the reads into 10 partitions takes at most several hours to complete. From the BowTie/BLAST output, we get an ungapped, base-to-base mapping of the read to a reference sequence; this mapping can then be extended to a universal, gapped, alignment for MNS calculation.

3.2.3 Testing the random seed effect in BowTie

Choosing a different random seed in BowTie can affect the microbial nucleotide signatures in two ways. First, reads that do not have perfect matches to the reference database may be missed; this is because BowTie does not exhaustively search for all possible near-matches and instead a random seed is used to randomly generate search outcomes. This causes some reads to be mistakenly reported as unaligned/non-

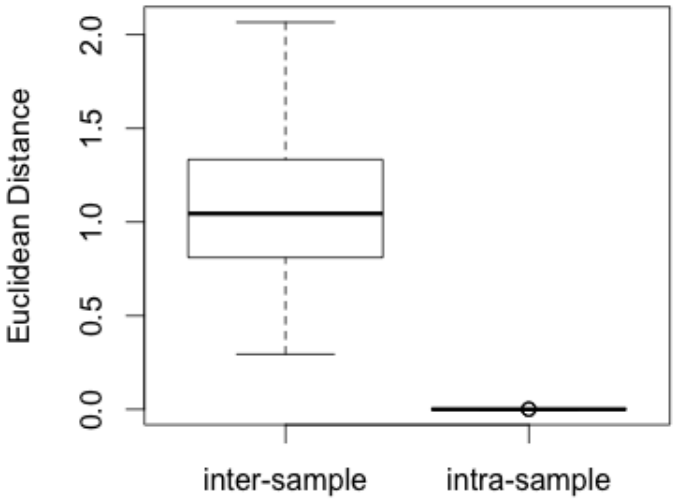


Figure 3.2: **The effect of random BowTie seeds on estimating the microbial nucleotide signatures.** (a) inter-sample: (Euclidean) distances between the MNSs of two different samples. (b) intra-sample: distances between the MNSs of 10 randomized BowTie runs of A +0. The distance between the MNS of two genuinely different samples was much larger than the distance between the MNSs of the same sample with different BowTie seeds. We concluded that BowTie's seed randomness has negligible effect on MNS.

mappable. Second, for a read that is mappable to more than one reference, we currently only report the first match to reduce runtime and output file size. Different reference sequences may map to the full alignment with subtle differences and this could affect diversity index calculations.

Here, we show that BowTie's randomness does not affect MNS significantly. We took one of the Illumina samples (A +0), which contained ∼1.8 million raw reads prior to quality filtering, and ran BowTie 10 times with different random seeds. We found that the small perturbations in the MNSs calculated from the 10 randomized BowTie runs were negligible compared to the true difference between two samples (Figure 3.2). The maximum intra-sample distance was 0.00059, well below the minimum observed inter-sample distance (0.2956).

3.2.4 Calculating microbial nucleotide signatures as a vector of Simpson or Entropy indices

Simpson's Diversity Index [87] is commonly used in ecology to measure both species diversity and evenness. *Diversity* refers to the number of different species and *evenness* refers to the distribution of the species in a community. The same concept is applied here to account for the diversity and evenness of nucleotides at each sequenced position in the 16S rRNA gene. The Simpson's Index D_i is the fraction of non-identical pairs of nucleotides in the i-th position of the alignment. It is calculated by:

$$D_i = 1 - \frac{\sum_{k=\{A,T,C,G\}} n_{k,i}(n_{k,i}-1)}{N_i(N_i-1)} \tag{3.1}$$

where $n_{k,i}$ is the number of nucleotides k observed at position i and N_i is the total number of nucleotides at position i. When N_i is large, we can approximate D_i by:

$$D_i \approx 1 - \sum_{k=\{A,T,C,G\}} (\frac{n_{k,i}}{N_i})^2 \tag{3.2}$$

The second term in D_i can be thought of as the sum of squared nucleotide proportions and is closer to 1 when the position is dominated by a certain nucleotide. Therefore, D_i is 0 when there is no diversity and increases with diversity (with a maximum of $\frac{3}{4}$ for the 4-letter nucleotide alphabet using the simplified formula). The total MNS of a sample is just a vector containing the D_i at each position, $D = \{D_1, D_2, D_3 \dots\}$.

Simpson's Index encapsulates both the abundance and diversity of nucleotides at each position. Another way to account for nucleotide diversity is to use relative entropy. Entropy is often used in information theory to measure the expected amount of information contained in a message. The Entropy Index E_i is calculated by:

$$E_i = \sum_{k=\{A,T,C,G\}} p_{k,i} \; log \frac{p_{k,i}}{q_k} \tag{3.3}$$

where $p_{k,i}$ is the nucleotide frequency of base k at the i-th position and q_k is the nucleotide frequency of base k across the entire sequenced region.

Once MNSs are calculated, pairwise distances between samples are calculated using Euclidean distance. Clustering is done using average-linkage hierarchical clustering.

3.3 Materials

3.3.1 Choice of hypervariable region to sequence

To evaluate which of the nine hypervariable regions is the one with the highest sequence identifiability, we took all sequences from our reference database, excised different hypervariable regions and calculated the percentage of excised sequences that were correctly classified by the RDP Classifier at the phylum and genus level (Table 3.1). V2, V3, V4, V7-8 all had comparable identifiability at the genus level, but V3 was shorter (155-210 bp as opposed to $>$ 200bp for the other three), so to increase sequencing depth with the same sequencing cost, we chose V3.

3.3.2 Study participants

Women and men were recruited from the Seattle, Washington area. Exclusion criteria included use of antibiotics during the 3-month period prior to sample collection and age less than 18 years. Fecal samples were collected from 9 participants (individual A-I) at two time points (3 months apart). The Institutional Review Board of the Fred Hutchinson Cancer Research Center, Seattle, WA approved all study procedures, and all participants provided written informed consent (IRB study number: 6096).

Region	*E. coli* range	Seq lengths (bp)	Number of sequences	Correctly assigned seqs (%)	
				phylum	genus
V1	8-120	100-140	29,417	65%	24%
V2	101-361	240-280	46,068	86%	63%
V3	338-534	155-210	46,851	86%	62%
V4	519-806	270-310	46,842	87%	65%
V5	787-926	120-160	46,851	86%	48%
V6	907-1073	140-180	46,851	86%	46%
V7-8	1054-1406	330-370	46,058	86%	66%
V9	1392-1507	90-130	9,315	93%	58%

Table 3.1: **Correct taxonomic assignment by RDP Classifier on different hypervariable regions in the 16S rRNA gene.** Regions were chosen to be mostly non-overlapping, each containing one or two variable regions. Coordinates are given relative to the 1542 bp *E. coli* K12 16S rDNA sequence. For hypervariable region definitions and common primers refer to Sundquist et al. [93].

3.3.3 Sample and DNA extraction

Fresh fecal samples were collected into fecal collection containers (Fisher Scientific, Fair Lawn, NJ) and subsamples were immediately placed in RNAlater. DNA was extracted using established protocols [48]. About 3g of feces were suspended and shaken in 3 ml RNAlater (Ambion, Austin, TX) immediately after defecation. Samples were then shipped to the lab and stored at -80°C until DNA extraction. Before DNA extraction, samples were homogenized by OMNI tissue homogenizer 115 (OMNI Inc., Marietta, GA) and divided into 300 μl aliquots, centrifuged at 16,000 xg with 300 μl phosphate buffered saline for 10 min and the supernatant containing RNAlater was discarded. Genomic DNA was extracted from fecal samples (QIAamp DNA Stool Mini Kit, QIAgen, Valencia, CA) with 1 min bead beating on setting 5.5 using a FastPrep system (MP Biomedicals, Solon, OH) [48].

3.3.4 PCR primers and conditions

The DNA of bacterial 16S rRNA genes were amplified with primer 330F (5′-ACT CCT ACG GGA GGC AGC AGT-3′) and 530R (5′-GTATTACCGCGGCTGCTGGCAC-3′) using PCR conditions as described in [2].

3.3.5 Library construction

Amplified sequences were finely fragmented using the published Covaris S2 protocol "DNA Shearing with microTubes (< 1.5kb fragments)" with a target base pair peak of 200 for 6 minutes (Covaris Inc., Woburn, MA). Following fragmentation, libraries were prepared following the protocol supplied with Illumina's Multiplexing Sample Preparation Oligonucleotide Kit (cat # PE-400-1001) scaled down for smaller DNA concentrations. The resulting end-adaptered DNA fragment population was subjected to massively parallel sequencing on an Illumina GA2 genome analyzer (Illumina, Inc., San Diego, CA)

3.3.6 Quality filtering

Reads were end-clipped using a Phred score cutoff of 2. Clipped reads with length < 30 bp were discarded and accounted for $< 1\%$ of the reads. Any remaining read with at least one base with Phred score < 10 was further removed. 70-80% of reads remained after the above steps (Table 3.2).

3.3.7 Read alignment

Reads were aligned using BowTie and BLAST to map all reads with at most 3 mismatches to the reference database. For alignment, we removed gaps from the reference database but kept a mapping of the ungapped (local) to gapped (global) positions. Reads whose first base aligned outside the V3 hypervariable region were discarded as they were likely to be contaminants. Reads whose first base aligned to the first or second position of V3 or whose last base aligned to the last or second to last position of V3 were also discarded because we observed an amplification bias that resulted in an abnormal excess of primer-containing reads (Table 3.3). Mapped reads were extended to the gapped alignment in preparation for MNS calculation.

3.3.8 Calculating microbial nucleotide signatures

We calculated diversity indices (either Simpson's or Entropy Index) using an "*E. coli* mask" on the sequenced region, i.e., we only included nucleotides that mapped to an *E. coli* reference nucleotide position in the SILVA database. We did this for practical reasons: the full alignment in the current SILVA database was very long (50,000 bp) due to long insertions (gaps) with respect to the *E. coli* reference. Most gut species had very few insertions with respect to *E. coli* and by excluding those insertions, we lost little information. Of the 87,295 reference sequences in SILVA, 86,447 had fewer than 5% non-*E. coli* position bases.

3.4 Results

3.4.1 Applying microbial nucleotide signatures to 9 healthy individuals

We sampled 9 healthy individuals (A-I) at two time points (3 months apart) and obtained a total of 20 samples (individual C had technical replicates). We PCR-amplified the V3 hypervariable region and sequenced the result using Illumina sequencing. We decided to sequence the V3 hypervariable region of the 16S rRNA gene because our analysis showed that it was most suitable for short read sequencing given its moderate length and high taxonomic identifiability (section 3.2.1). We obtained 1-7 million 51 bp reads per sample. After quality filtering and alignment, 20-50% of the reads remained (Table 3.2, Table 3.3). Of the total usable reads, less than 0.8% of the nucleotides mapped to non-*E. coli* positions.

Clustering based on MNSs (Simpson Index) showed that intra-individual differences were smaller than inter-individual differences. Samples from the same individuals always clustered closely (Figure 3.3a) and the distances within the same individual were smaller than distances between individuals (Figure 3.3b, *t*-test; *p*-value: $8\text{x}10^{-10}$). Since there were no changes in the individuals' health conditions, we expected samples to cluster by individuals.

Sample	**Binned**	**Clipped len $<$ 30**	**Low quality**	**BowTie**	**BLAST**	**Unaligned**
A +0	1,824,457	0.9%	22.56%	65.2%	20.4%	0.8%
A +3	3,573,651	0.6%	24.03%	60.8%	25.3%	1.4%
B +0	2,350,257	0.9%	22.95%	60.9%	25.4%	0.7%
B +3	2,582,226	0.6%	22.62%	62.3%	24.0%	1.3%
Ca +0	3,471,332	0.7%	20.92%	66.3%	21.8%	1.1%
Cb +0	4,664,796	0.7%	15.21%	79.8%	10.7%	0.7%
Ca +3	2,358,891	0.5%	25.75%	67.9%	19.2%	2.2%
Cb +3	4,285,065	0.5%	17.89%	78.2%	11.4%	1.5%
D +0	3,467,457	1.0%	38.44%	38.0%	43.0%	4.7%
D +3	967,973	0.7%	40.87%	39.1%	38.9%	6.2%
E +0	6,385,646	0.8%	20.81%	68.3%	18.7%	0.9%
E +3	3,244,355	0.6%	46.90%	36.6%	34.2%	2.1%
F +0	3,451,929	0.9%	23.34%	65.0%	18.5%	1.3%
F +3	3,254,432	0.7%	29.98%	52.5%	30.3%	1.9%
G +0	7,413,969	0.9%	33.84%	44.1%	38.9%	1.1%
G +3	3,067,727	0.6%	43.80%	36.8%	38.8%	1.9%
H +0	7,265,011	0.9%	17.54%	73.1%	15.7%	0.5%
H +3	4,486,313	1.0%	69.43%	19.9%	41.6%	3.4%
I +0	5,350,038	0.9%	37.87%	46.5%	30.1%	1.4%
I +3	5,651,154	0.7%	52.21%	33.2%	33.4%	3.1%

Table 3.2: **Quality filtering and alignment (part 1) of the reads to our reference database.** Reads were matched by barcode (2 mismatches allowed) then end-clipped with a Phred score cutoff of 2. We discarded reads of clipped length $<$ 30 bp. BowTie and BLAST were used to align reads with at most 3 mismatches to the reference database. Percentages listed in each column are with respect to the original number of reads (Binned).

Sample	**Binned**	**pre-V3**	**post-V3**	***E. coli* 338-339**	***E. coli* 485-486**	**Used for MNS**
A +0	1,824,457	3.4%	2.7%	15.3%	15.8%	39.34%
A +3	3,573,651	4.4%	3.5%	16.2%	13.5%	37.77%
B +0	2,350,257	4.8%	3.4%	13.7%	15.7%	38.55%
B +3	2,582,226	4.3%	3.5%	16.3%	14.1%	38.58%
Ca +0	3,471,332	4.3%	2.8%	16.9%	14.7%	39.68%
Cb +0	4,664,796	1.9%	1.9%	16.0%	15.5%	48.79%
Ca +3	2,358,891	3.6%	1.9%	18.8%	13.3%	36.15%
Cb +3	4,285,065	1.9%	1.6%	15.5%	10.9%	51.71%
D +0	3,467,457	6.1%	5.3%	9.6%	13.5%	26.06%
D +3	967,973	4.7%	5.1%	7.9%	10.6%	30.13%
E +0	6,385,646	4.0%	2.4%	18.2%	14.4%	39.39%
E +3	3,244,355	3.8%	4.6%	6.8%	9.9%	27.40%
F +0	3,451,929	3.4%	2.6%	16.2%	14.9%	38.66%
F +3	3,254,432	4.5%	4.3%	12.7%	12.3%	35.52%
G +0	7,413,969	6.0%	4.8%	11.1%	14.2%	29.16%
G +3	3,067,727	4.7%	5.0%	8.7%	10.0%	27.20%
H +0	7,265,011	3.4%	2.1%	19.6%	16.3%	40.16%
H +3	4,486,313	3.3%	4.2%	2.0%	3.3%	16.77%
I +0	5,350,038	3.9%	3.7%	9.1%	13.0%	31.53%
I +3	5,651,154	3.3%	4.1%	6.7%	7.9%	25.09%

Table 3.3: **Quality filtering (part 2) of the reads after alignment.** We discarded reads that mapped outside the V3 region or corresponded to *E. coli* positions 338, 339, 485, and 486. 16-51% reads remained for MNS calculation. Percentages listed in each column are with respect to the original number of reads (Binned).

3.4.2 MNS at different subsampling depths

Microbial nucleotide signatures differentiated samples when the effective sequencing depth (i.e., number of high-quality, aligned reads; see last column in Table 3.3) was on the order of millions. To see if MNS was equally informative when sequencing depth was lower, we randomly subsampled the pool of high-quality, aligned sequences for each individual, calculated the MNSs, and compared them to the original MNSs. Subsampling was repeated 100 times for each subsampling depth. As we drew more reads from the original pool, the subsampled MNSs approached the originals (Figure 3.4). Furthermore, the subsampled MNSs converged rather quickly. For example, sample A +0 originally had 1.8 million raw reads. After quality filtering, 39% or ∼0.7 million reads were used to create the MNSs and were drawn from to create our subsamples. At a subsample size of 5,120 reads, the Euclidean distance between the subsampled and original MNS was < 0.2, smaller than the distance between any pairs of samples shown in Figure 3.3b. This suggested a modest number of reads per sample sufficed to obtain a MNS that can differentiate between samples adequately. As proof, we generated a clustering for each of the subsampled MNSs to see how often we recovered the subtrees found in the full dataset. Figure 3.5 and Figure 3.6 show the frequency of the subtrees in the subsampled MNS overlaid on the original clustering. A frequency of 1.0 means that the subtree appeared in the cluster in 100% of the subsampling tests. At 40,960 reads, except for one subtree, both the Simpson Index MNS and the Entropy Index MNS resulted in the same clustering as their original MNSs. It appeared that Entropy Index was better at lower effective sequencing depths, which might have been because the Entropy Index accounted for the background nucleotide diversity (the denominator in the index calculation) and the Simpson Index did not.

3.4.3 Applying MNS clustering to other datasets

We applied our MNS clustering approach to two other datasets: the MetaHit data from Qin et al. [77] and the atherosclerosis patient samples from Omry et al. [42].

The MetaHit data is publicly available and consists of metagenomics reads (75bp, Illumina) from 60 Spanish and 40 Danish human samples. To extract the 16S rRNA gene reads, we ran BowTie with our reference database and calculated MNSs using the same procedures. We obtained an average of 33,000 reads per sample that were distributed across the entire 16S rRNA gene. Using the whole reference *E. coli* region to calculate MNS, the technical replicates clustered very well for both datasets (Figure 3.7). For the atherosclerosis dataset, we obtained the pyrosequencing data from the authors, which consisted of samples from feces, mouth, and plaque. We obtained an average of 5,000 sequences per sample. The sequenced region was the 16S rRNA V2 region (*E. coli* position 220-320), which we used to align and calculate MNS. Except for two samples, all samples clustered together by body site and not individuals (Figure 3.8).

3.5 Discussion

We showed that using microbial nucleotide signatures, microbial communities can be characterized even when assembly or taxonomic identification are not feasible. For long pyrosequencing reads, standard processing pipelines from RDP [100], Mothur [85], and Qiime [13] may be the right tools. For short, low-coverage, fragmentary reads, like the ones produced in our study and Qin et al., MNS can be used to differentiate samples between individuals—all that is required is a reference database and a read aligner.

We believe that MNS can be of great use in screening for gut microbiota differences between diseased and non-diseased individuals. Once differences between health and disease are noted, deeper sequencing and phylogenetic analysis could be performed to examine their functional signifcance. MNS may be particularly useful in diseases with diagnostic uncertainty or overlap. For example, inflammatory bowel disease (IBD) have an established correlation with the gut microbiota, but diagnostic subtyping (Crohn's versus ulcerative colitis) based on standard clinical testing is difficult because both diseases have overlapping clinical manifestations [26]. With MNS, we can classify IBD individuals into distinct subgroups and target them for further

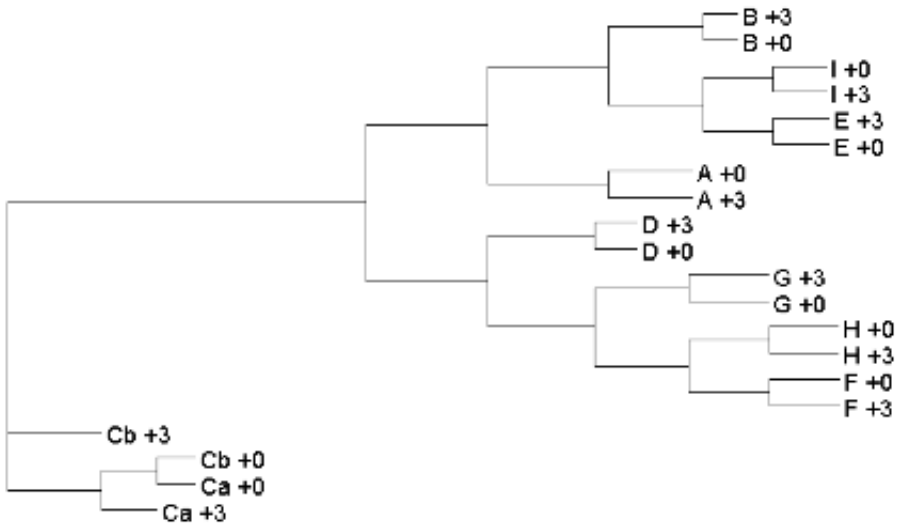
analysis. Similarly, MNS is a potential forensics tool, as a recent study showed that pyrosequencing data from biological stains could be used for forensic identification [10].

We showed, using repeated subsampling, that ∼40,000 51 bp reads was just as effective at differentiating among healthy individuals' gut microbiota as using millions of reads. This means that for single high-throughput sequencing runs, say, Illumina Hi-Seq which produces $>$ 100 million reads, with multiplexing one could sequence hundreds of individuals, reducing screening costs while assessing variation at the human population level.

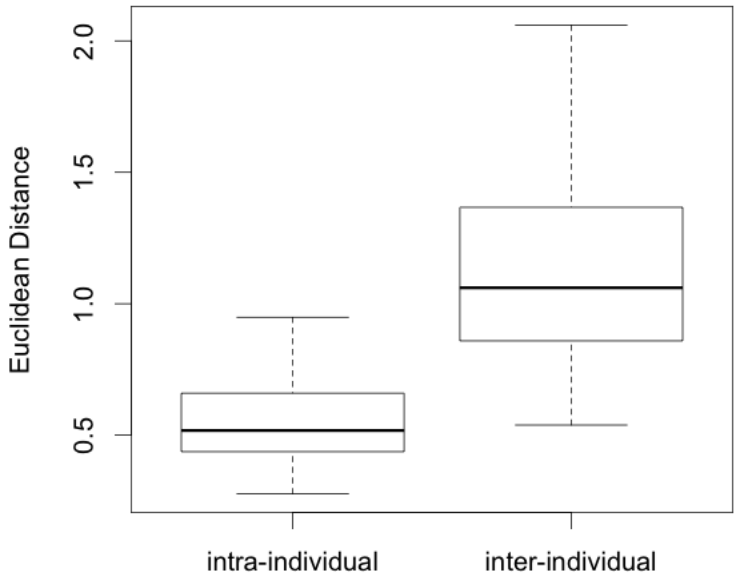
The number of sequences required to represent a microbial community differs depending on the sampled environment, the sequenced gene region, and the read length. In the study by Caporaso et al. [14], the authors showed that 2,000 reads (of 100 bp) was enough to represent environmental soil samples. In our study, we needed 40,000 51 bp reads to represent human gut microbiota. We speculate that the difference in the minimum number of reads is a result of read length and differences in phylogenetic diversity of the environments we are distinguishing. We applied our method to distinguish individual differences within the same environment (human gut, variation at species level) as opposed to comparing across different environments (gut, tongue, ocean, soil, etc, variation at phylum level). Since individual differences are more subtle than environmental differences, it is not surprising that we needed more reads.

The current microbial nucleotide signatures method requires that all samples be sequenced from the same region. For future work, we aim to develop methods that do not require sequencing from the same region. We would also like to extend the approach beyond the 16S rRNA gene. Though the 16S rRNA gene is ideal for identifying bacteria, it is often more important to look at the profile of a microbial community from the perspective of functionality, as different species can have the same set of active genes. However, functional genes are not as comprehensively studied as the 16S rRNA gene and lack comprehensive databases. Our method will have to deal with identifying novel genes, comparing homologs and develop ways to

account for the presence or absence of genes.

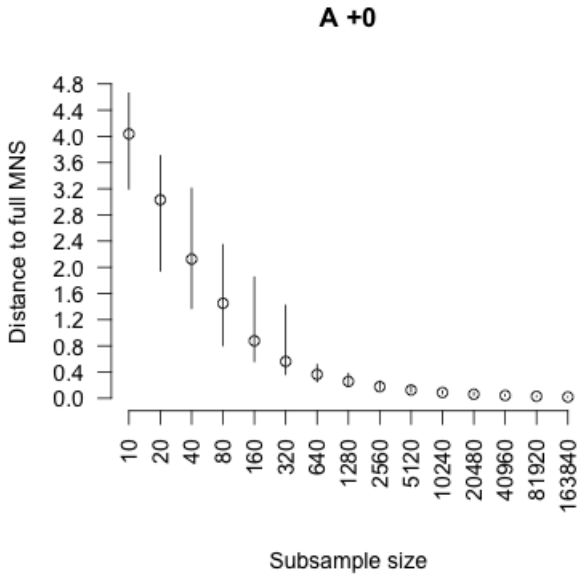


(a) **Clustering of 20 samples from 9 individuals (A-I) at two time points (+0, +3) using microbial nucleotide signatures (Simpson Index).** Ca/Cb are technical replicates. Samples from the same individuals all cluster closely together.

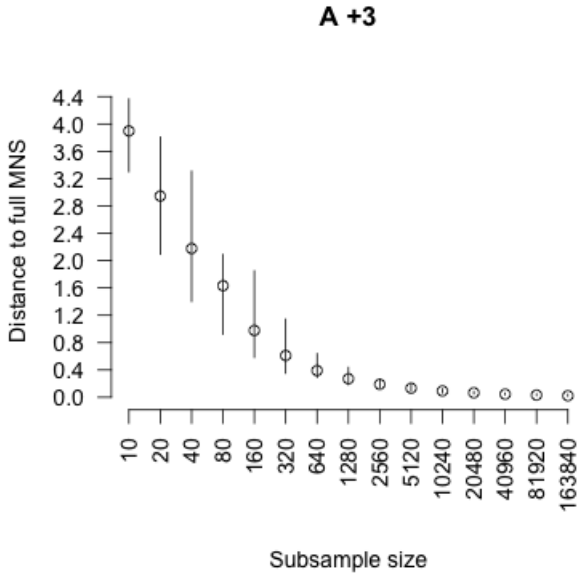


(b) **Intra- and inter-individual distances between 9 healthy individuals.** The mean was 0.54±0.18 for intra-individual samples and 1.14±0.35 for inter-individual samples. (*t-test p-value: $8\text{x}10^{-10}$).

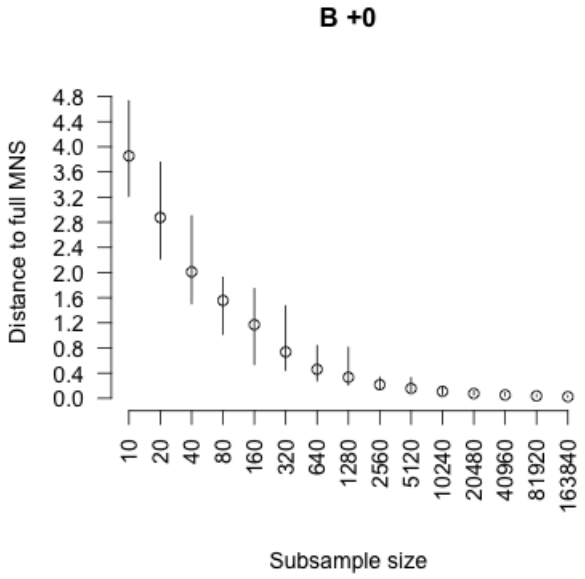
Figure 3.3: **Clustering of 20 samples from 9 individuals (A-I) at two time points (+0, +3) using MNS (Simpson Index).**


A +0
Distance to full MNS
4.8
4.4
4.0
3.6
3.2
2.8
2.4
2.0
1.6
1.2
0.8
0.4
0.0
10
20
40
80
160
320
640
1280
2560
5120
10240
20480
40960
81920
163840
Subsample size

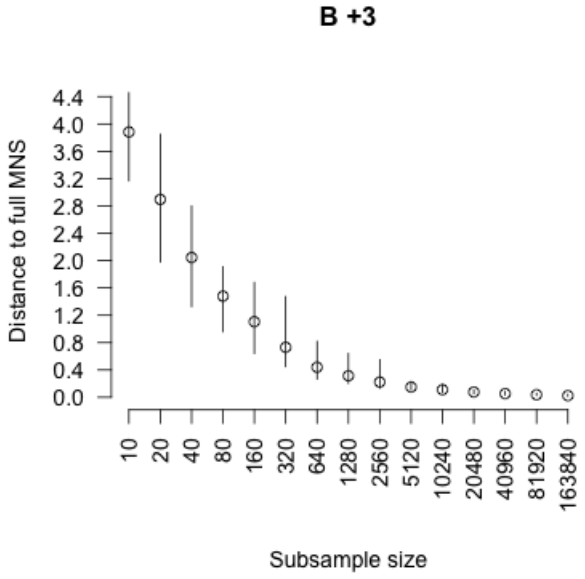
(a)


A +3
Distance to full MNS
4.4
4.0
3.6
3.2
2.8
2.4
2.0
1.6
1.2
0.8
0.4
0.0
10
20
40
80
160
320
640
1280
2560
5120
10240
20480
40960
81920
163840
Subsample size

(b)

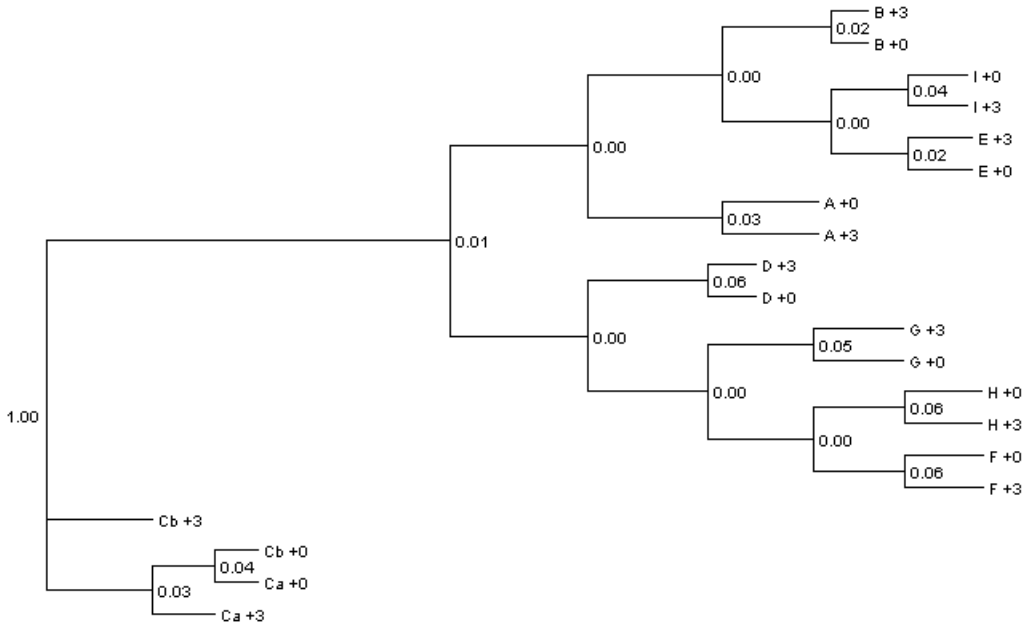


(c)

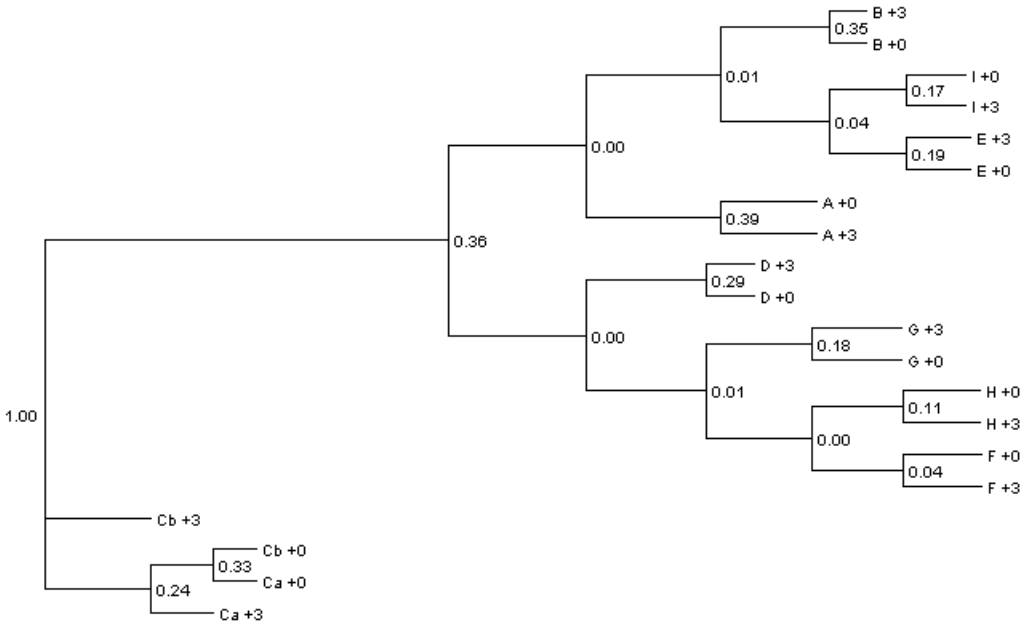


(d)

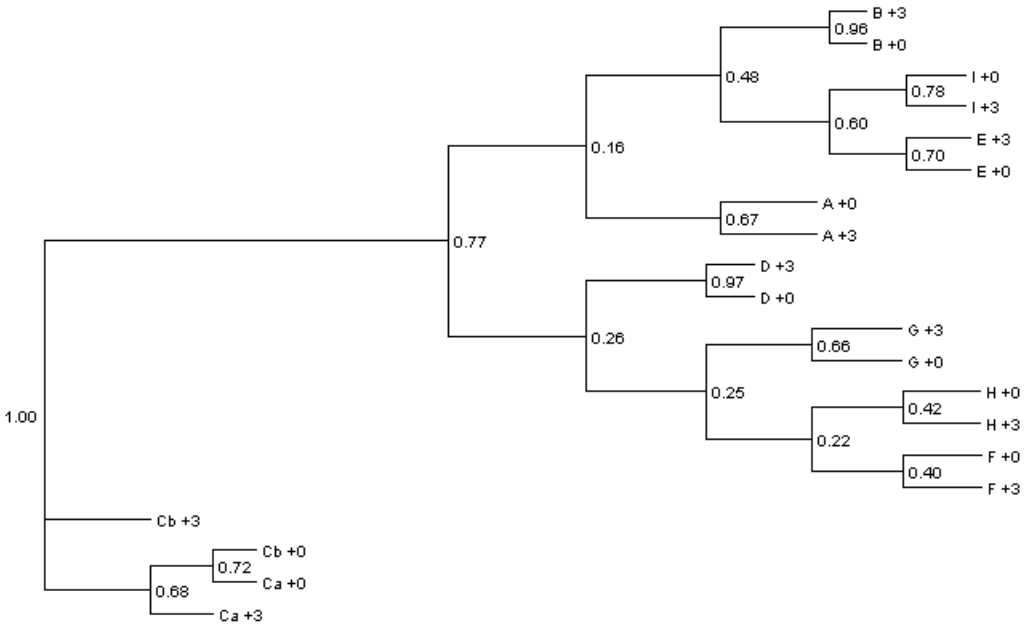
Figure 3.4: **Subsampled microbial nucleotide signature (MNS) rapidly approached the original MNS with increasing subsample size.** Means are drawn as circles and error bars are plotted based on 100 repetitions. Sample A and B shown only; remaining samples are similar.



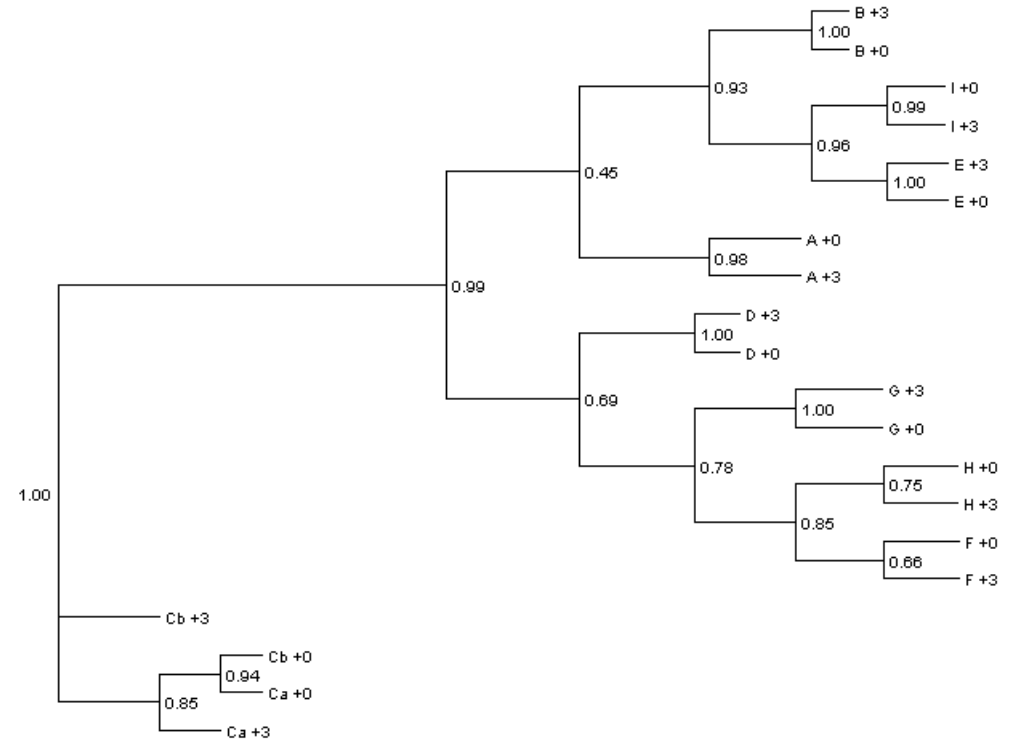
(a) Simpson index. Subsampling size = 10.



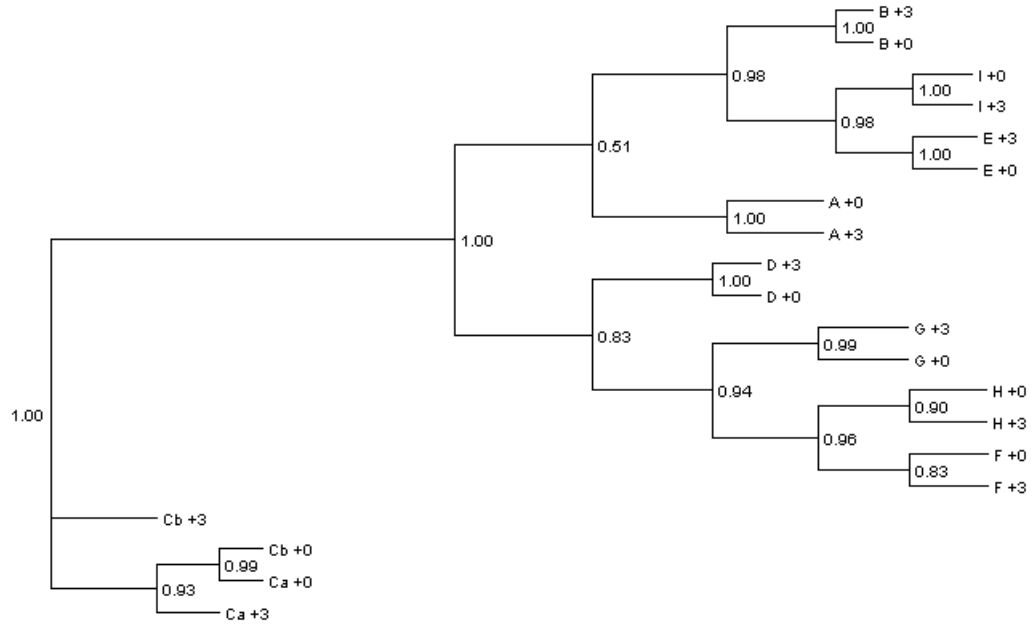
(b) Simpson index. Subsampling size = 160.



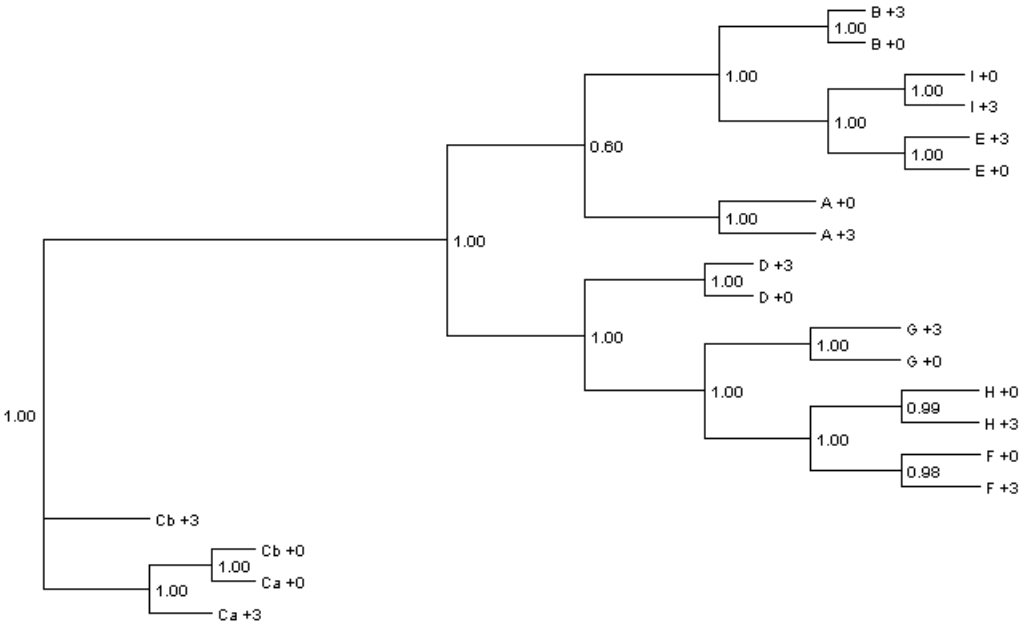
(c) Simpson index. Subsampling size = 1,280.



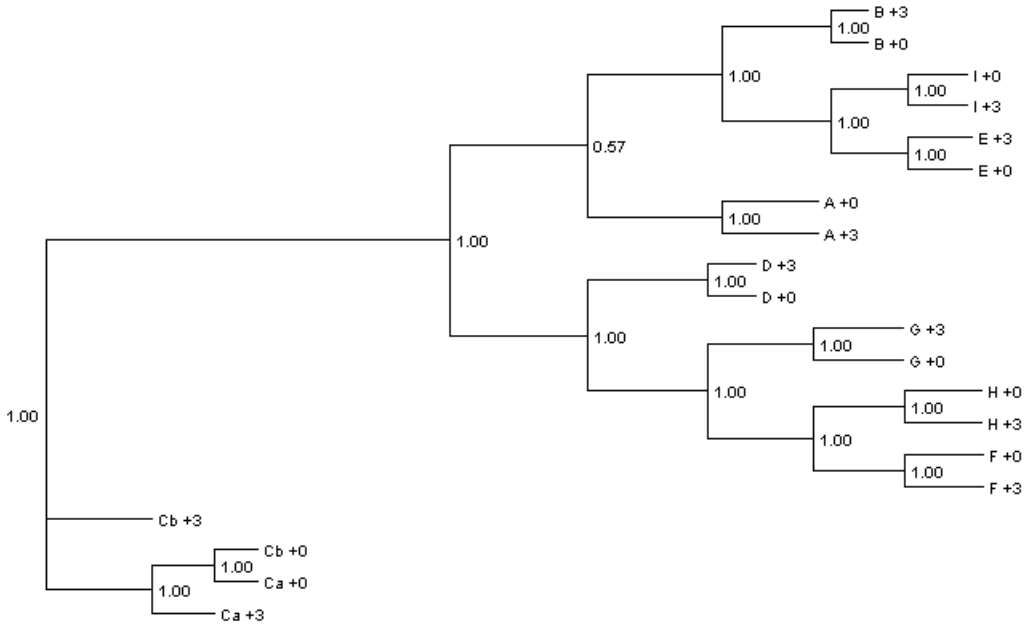
(d) Simpson index. Subsampling size = 5,120.



(e) Simpson index. Subsampling size = 10,240.

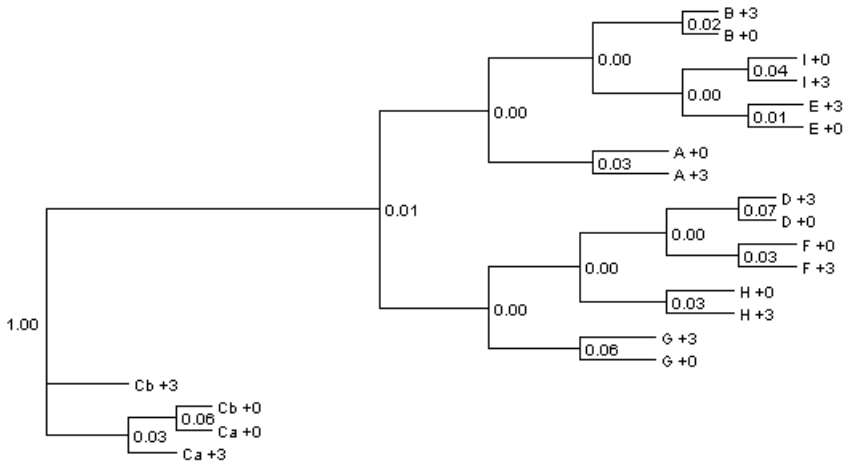


(f) Simpson index. Subsampling size = 40,960.

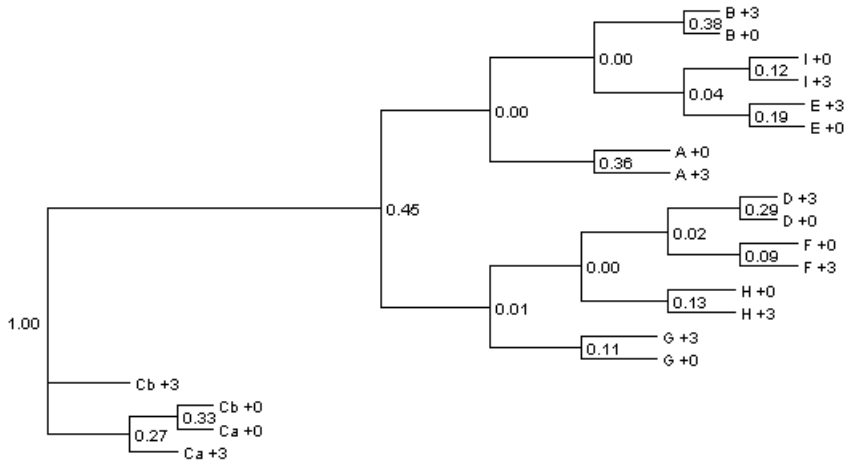


(g) Simpson index. Subsampling size = 81,920.

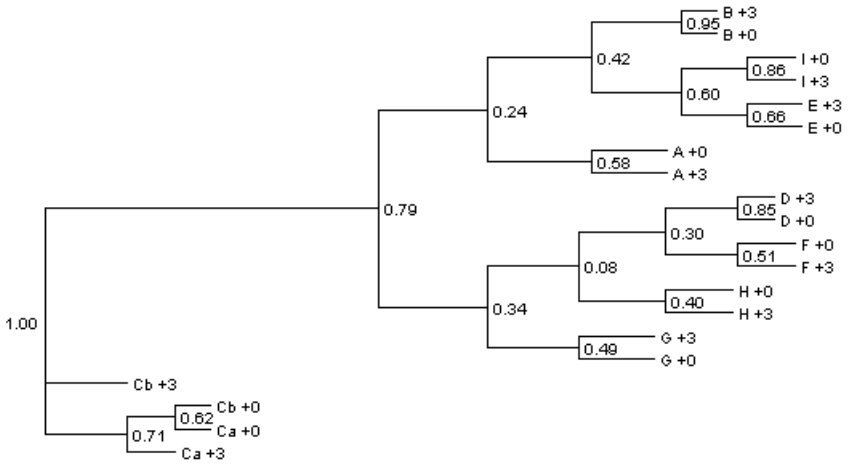
Figure 3.5: **Clustering of 9 healthy individuals using Simpson Index at different subsampling depth.** Numbers at internal nodes are the fraction of 100 random subsamples in which the partition (subtree) appeared in the clustering.



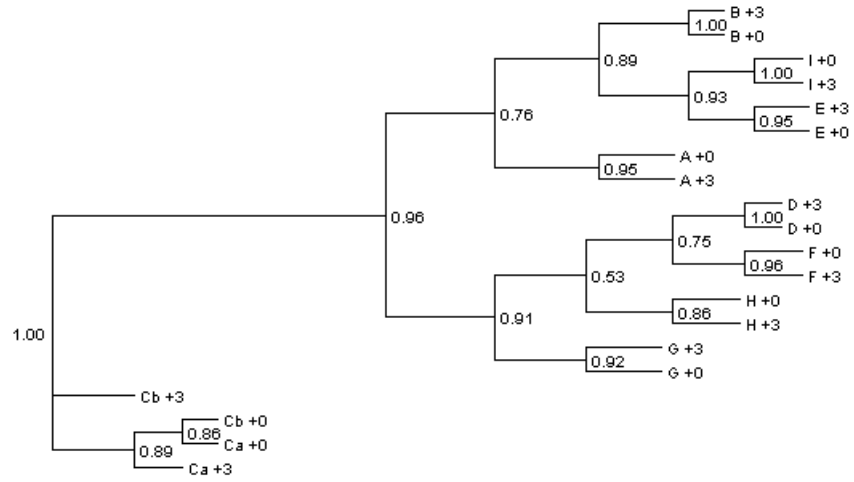
(a) Entropy Index. Subsampling size = 10.



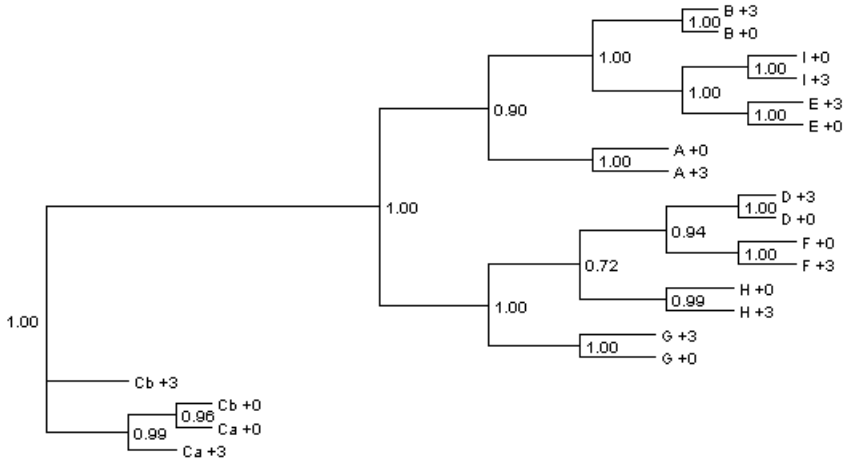
(b) Entropy Index. Subsampling size = 160.



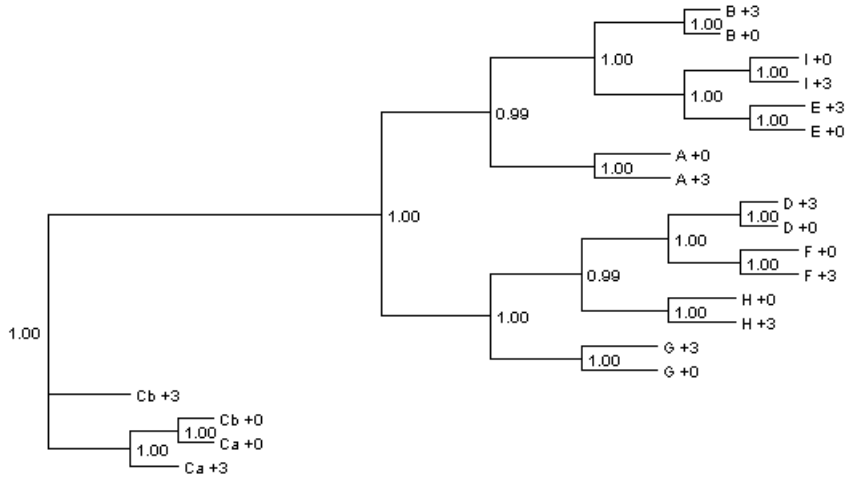
(c) Entropy Index. Subsampling size = 1,280.



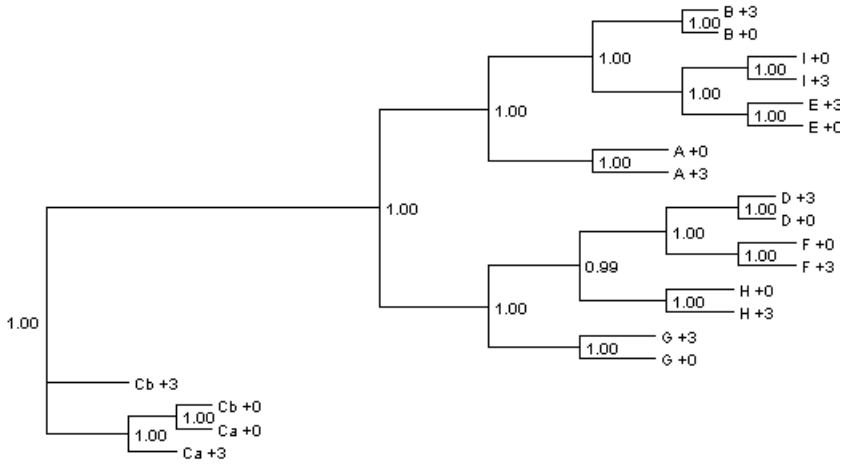
(d) Entropy Index. Subsampling size = 5,120.



(e) Entropy Index. Subsampling size = 10,240.

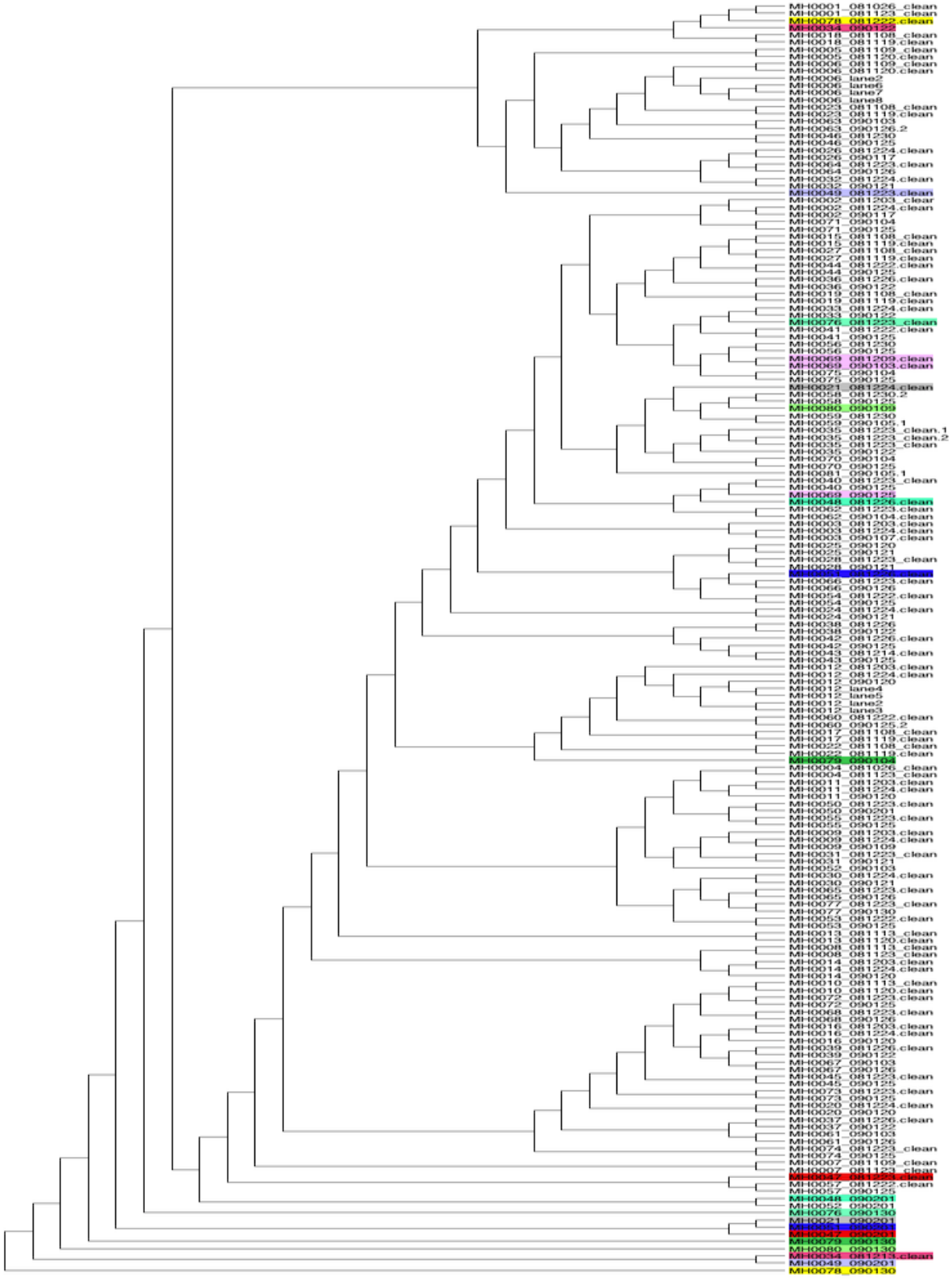


(f) Entropy Index. Subsampling size = 40,960.

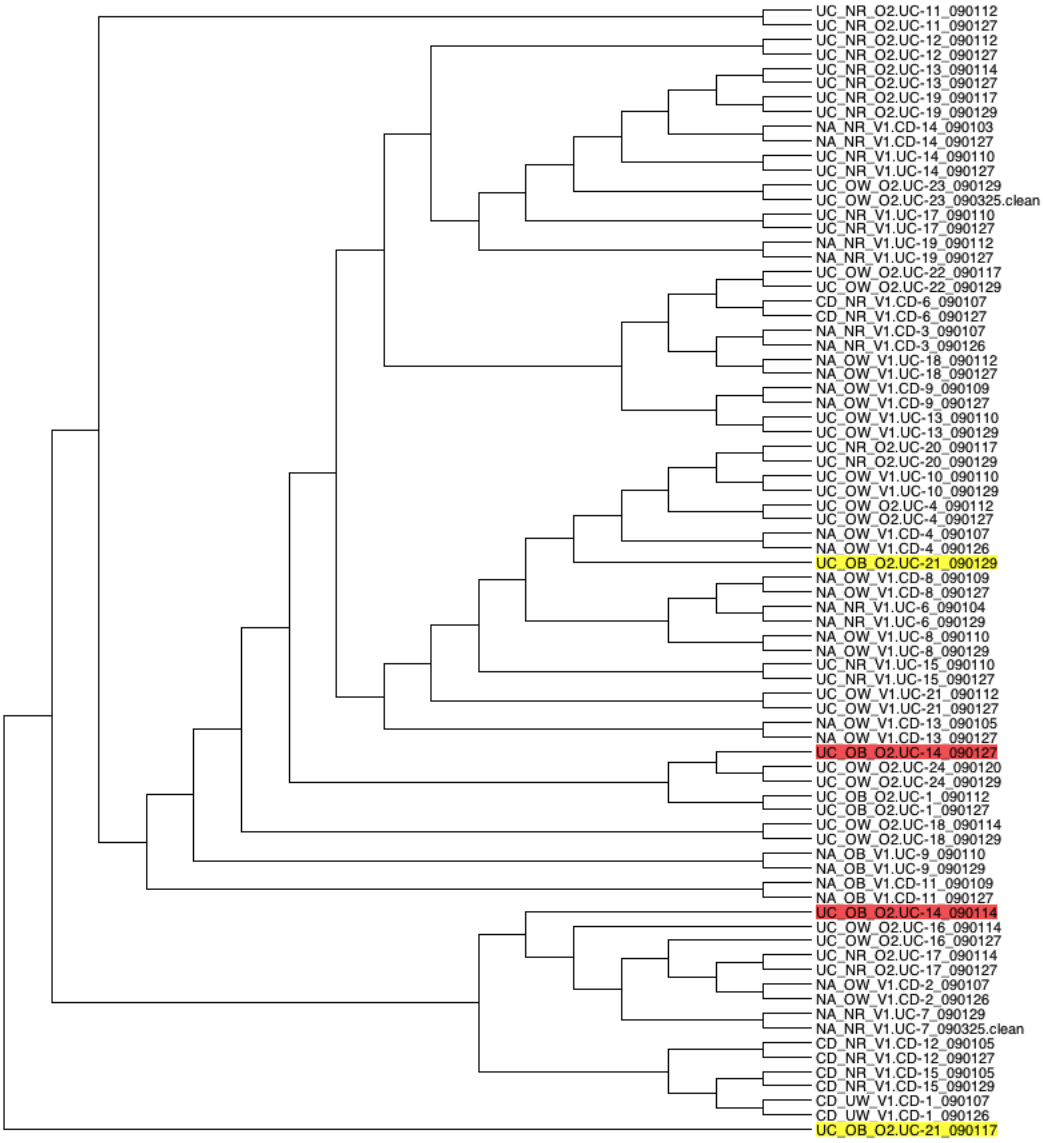


(g) Entropy Index. Subsampling size = 81,920.

Figure 3.6: **Clustering of 9 healthy individuals using Entropy Index at different subsampling depth.** Numbers at internal nodes are the fraction of 100 random subsamples in which the partition (subtree) appeared in the clustering.



(a) Qin et al. Danish-only samples, date separated.



(b) Qin et al. Spanish-only samples, date separated.

Figure 3.7: **Clustering of the (a) Danish and (b) Spanish samples from Qin et al. using MNS (Entropy Index).** 16S rRNA reads were extracted by running BowTie against our reference database. Replicate samples that were not clustered together are colored. Average nucleotide coverage per read position was 500-1000.

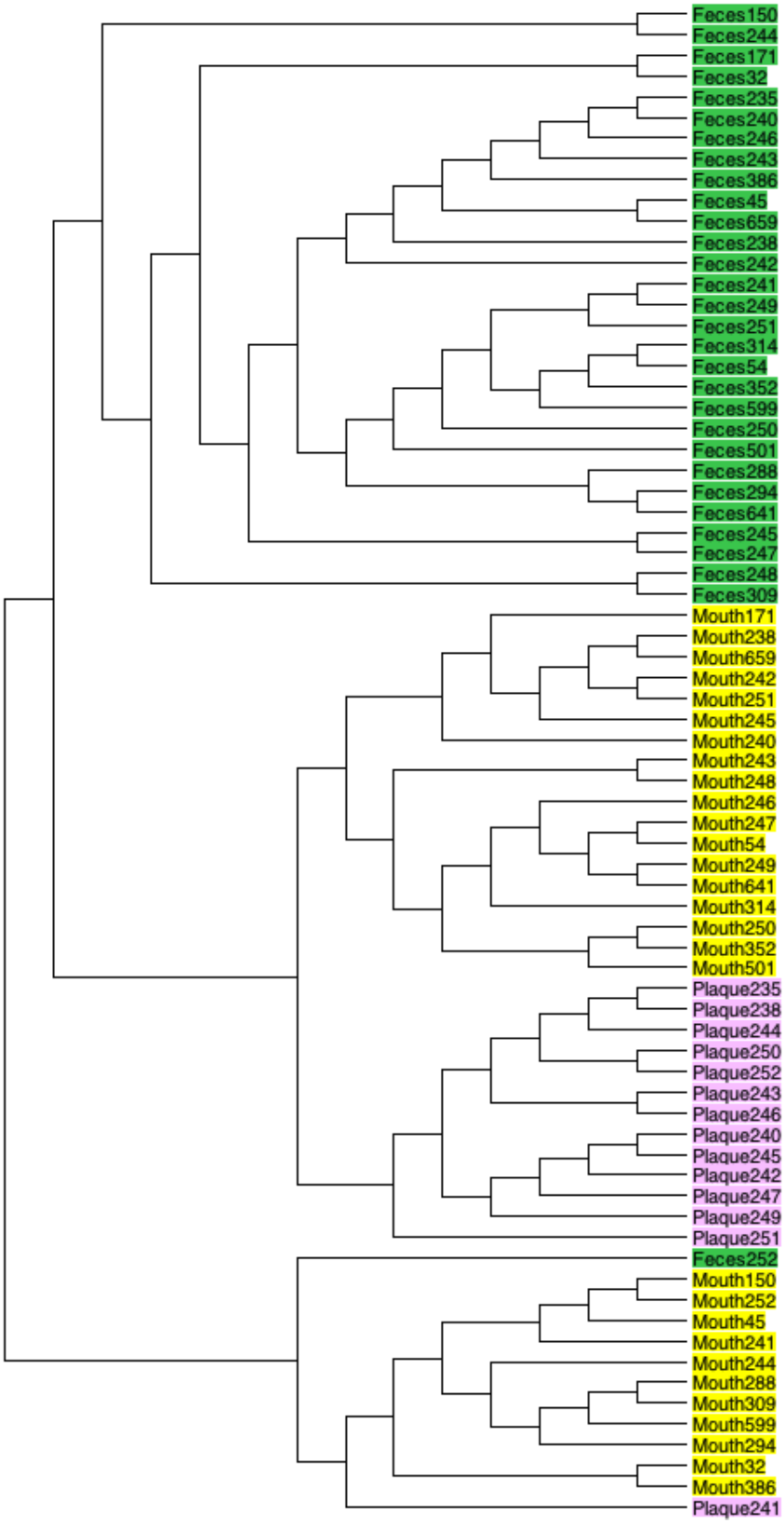


Figure 3.8: **Clustering of samples of atherosclerosis patients from Omry et al. using MNS (Entropy Index).** Samples are colored by body site.

Chapter 4

DEALING WITH SEQUENCING BIAS AND ERRORS IN ILLUMINA SHORT READS

4.1 Introduction

Sequencing bias and errors are two important issues in all applications of high-throughput sequencing. Sequencing bias refers to the misrepresentation of the original distribution of molecules in the samples. For Illumina sequencing, bias can occur at two stages: PCR amplification during library preparation and cluster amplification during Illumina's sequencing-by-synthesis process. Sequencing bias leads to misinterpretation of data: (1) in de novo sequencing, low or no coverage of genomic regions hinders assembly; (2) in RNA-seq experiments, uneven coverage leads to incorrect estimation of transcript abundance; (3) in 16S rRNA human gut microbiota studies, sequencing bias results in a skewed representation of the microbial community [54].

Sequencing errors refer to the erroneous reporting of bases and are caused by biological and bioinformatic errors. For example, the *Taq* DNA polymerase used in PCR library construction has a natural error rate of 0.01% [94]. More often, though, the errors come from incorrect base calling due to low signal intensities, successive bases of the same nucleotide (homopolymers), or formation of secondary structures. Recent studies found that Illumina sequencing errors are not uniformly distributed and are more error-prone for specific patterns (section 4.3). At the time of this writing, the Illumina sequencing error rate is below 1% [61]. Depending on the type of application, however, error correction might not be necessary. For transcript abundance estimation in RNA-seq, this rate is low enough that most reads are mapped back to the reference genome unambiguously. On the other hand, error correction is essential for SNP calling on rare variants [115].

In section 4.2, we describe previous studies on how to measure and reduce PCR amplification bias. In section 4.3, we describe previous studies on identifying and correcting systematic errors in Illumina sequencing. Unlike the chemical and procedural solutions proposed for reducing PCR amplification bias, error correction methods are purely computational. In section 4.4 and 4.5, we motivate and present findings from our own work. We added fixed proportions of two foreign bacterial sequences to two sets of human gut microbial samples (standard additions experiment) and sequenced the samples with Illumina HiSeq. Using the foreign bacterial sequences as ground truth, we identified sequencing bias and error patterns. Finally, in section 4.6, we propose and evaluate an error correction method for Illumina short reads.

4.2 Sequencing bias due to PCR amplification

Since the days of Sanger sequencing, PCR amplification has been an essential step in library preparation; this is because most imaging systems cannot detect the fluorescent signal of a single event and template amplification is required [62]. The central PCR amplification steps are: denaturation, annealing of primers to the template, and elongation through DNA polymerase. Double stranded DNA are held together by hydrogen bonds formed between complementary bases, and since the binding energy between a GC pair is much stronger than an AT pair, all three steps are subject to bias introduced by differences in sequence base compositions. Kanagawa et al. [36] found that having GCs in degenerate primer positions resulted in higher amplification efficiency. Aird et al. [3] showed that using the standard Illumina library preparation protocols, sequences with GC composition below 12% and above 56% were dramatically less amplified at as few as ten PCR cycles. A likely explanation is that GC-rich sequences take longer to denature, so they are amplified less efficiently. For AT-rich sequences, the cause could be poor annealing (weaker binding due to only two instead of three hydrogen bonds between AT pairs). When the authors decreased temperature ramp rates—which increased denaturation time—more GC-rich sequences were amplified, but AT-rich sequences were even less amplified than before. Adding an additional enzyme (AccuPrime Taq HiFi, Introvigen) to the mix-

ture and lowering the thermocycler temperature slightly increased AT-rich sequence amplification. None of the experiments resulted in a completely even representation of sequences at all GC%, however, and the authors concluded that more careful selection of PCR enzymes and additives might help with amplifying the extreme AT-rich sequences (GC% $< 10\%$) that were still 10- to 100-fold less amplified than others. In a later study, Oyola et al. screened a variety of additives and enzymes and found that the PCR additive (tetramethylammonium chloride) improved amplification on extremely AT-rich regions [72].

To conclude, PCR amplification bias is caused by base composition differences in both primer and substrates and can be lessened through proper selection of enzymes, thermocycler settings, and number of cycles. It should be noted that the Illumina sequencing step itself also involves amplification and is subject to the same kind of bias, but it is not clear what can be done from the user end to reduce it.

4.3 Sequencing error due to Illumina sequencing technology

Sequencing error continues to affect sample estimates despite improvements in sequencing technology. If errors at each position are independent and occur at the same rate, they can be modeled with a single parameter. Zagordi et al. [114, 115] used this approach to find rare variants in HIV genes. To find the number of quasispecies (clusters of haplotypes), they used a Dirichlet process mixture to model the uncertainty of a read originating from a particular haplotype. The probability of observing the read given the haplotype assignment was then computed as the probability of observing each base with uniform error rate. They used Gibbs sampling to sample the posterior distribution of the Dirichlet process mixture model and calculated the posterior probability of the haplotypes.

Unfortunately, modeling sequencing errors in Illumina with a single error rate does not capture the error patterns that have been observed. Studies have shown that the error rate increases with cycle number due to fading fluorescence intensity, decreasing purity of nascent strands in a cluster, and accumulation of residual dyes [39]. A gradually increasing error rate is not difficult to parameterize if it can be

fitted to some distribution. However, Nakamura et al. found that the errors were additionally influenced by sequence patterns and base composition [70]. Using publicly available data and their own data, they analyzed four bacterial genomes with Illumina GAII (read length 70-100bp), and found that pattern-specific errors occurred most frequently with (1) GGC, or (2) long inverted repeats. The authors were not certain why GGC was the dominant error-inducing pattern, but suspected it was due to the preference of DNA polymerase. A long inverted repeat is a stretch of the template sequence that contains two closely located and perfectly complementary bases. Secondary structures formed by the folding of DNA single strands can inhibit nucleotide elongation and cause folded sequences to go out of phase. Together GGC and long inverted repeats accounted for 90% of the errors.

As corroborating evidence that pattern-specific errors may be a true phenomenon in Illumina sequencing, Meacham et al. [61] looked at methyl-Seq data sequenced with Illumina GAIIX (76bp paired reads). They found that GGT, not GGC, was the most frequent error pattern. Furthermore, they observed a dominating proportion ($\sim$80%) of T to G miscalls (e.g., GGT would be reported as GGG). While Nakamura et al. did not observe such a pattern, their study provided some explanation as to why GGT would be miscalled as GGG: they found that mismatched bases often matched the immediately preceding or the second preceding reference base.

To conclude, Nakamura et al. and Meacham et al. found that:

- Sequencing errors increased with number of cycles
- Sequence-specific patterns, such as GGX or long inverted repeats, induced systematic errors
- Read strand may or may not have been a factor in sequencing error (Meacham et al. found correlations, Nakamura et al. did not)

4.4 Why sequencing bias and errors matter for 16S rRNA sequencing

As we have shown in Chapter 3, accurate characterization of the human microbiome is important for studying human health and disease risks. Shifts in the abundance and evenness of microbial species both within and between samples can be indicators of diet, genetic, and disease risk factors. It is therefore important to quantify both the absolute and relative abundance of species in the samples. Sequencing errors introduce erroneous diversity estimates in the reads and can lead to overestimation in the number of species. For example, Degnan & Ochman [19] sequenced a mock community consisting of just the 16S rRNA gene of *E. coli* K-12 with Illumina GAIIX. They found many error-containing reads that drove up the number of 100% sequence identity clusters (or, operational taxonomic units, OTUs) to several thousands, when the expected number of OTUs should have been exactly 2. When they iteratively removed OTUs that were less than 1% abundant (an approach suggested in [14]), they were able to reduce the number of OTUs down to 2. This arbitrary cutoff might work in some cases, but one would be discarding precious information on more rare species. Rare species are often important drivers in some disease states and should not be easily overlooked [95].

In the following sections, we describe our findings on the effects of sequencing bias and errors from our own case study. We designed a standard additions experiment, adding two foreign bacterial sequences to human gut microbial samples with varying concentrations. The experiment allowed us to answer: (1) Do we recover the same proportion of foreign bacteria as was added? (2) What is the detection limit? (3) Are quality scores (Phred scores) indicative of base errors? (4) Are there systematic errors in Illumina HiSeq? We then present a method for correcting errors that resulted in more accurate species numbers.

Mycoplasma pneumoniae AF132740.1.1487
GC content = 42% Length = 201 bp
ACTCCTACGGGAGGCAGCAGTAGGGAATTTTTCACAATGAGCGAAAGCTTGATGGAGCAATGCCGCGTG
AACGATGAAGGTCTTTAAGATTGTAAAGTTCTTTTATTTGGGAAGAATGACTTTAGCAGGTAATGGCTA
GAGTTTGACTGTACCATTTTGAATAAGTGACGACTAACTATGTGCCAGCAGTCGCGGTAATAC

Deinoccocus radiodurans AF289090.1.1455
GC content = 56% Length = 186 bp
ACTCCTACGGGAGGCAGCAGTTAGGAATCTTCCACAATGGGCGCAAGCCTGATGGAGCGACGCCGCGTG
AGGGATGAAGGTTTTCGGATCGTAAACCTCTGAATCTGGGACGAAAGAGCCTTAGGGCAGATGACGGTA
CCAGAGTAATAGCACCGGCTAACTCCGTGCCAGCAGCCGCGGTAATAC

Table 4.1: **DNA sequence of the V3 region of the 16S rRNA gene of *M. pneumoniae* and *D. radiodurans*.** Primer locations are marked in gray. Mismatch to the primer is marked in red.

4.5 Case study: using standard additions in human gut microbial samples

4.5.1 Standard additions experimental design

We added known proportions of foreign bacterial 16S rRNA sequence to two of the samples (denoted Sample Set 1 and 2) from a cohort study (section 4.5.2). Prior to PCR amplification, the amount of DNA in each of the two samples was determined, and the foreign bacterial sequences were added so that it was 0.001%, 0.0032%, 0.01%, 0.032%, or 0.1% of the total amount of sample DNA. For the choice of foreign bacteria, we selected one high GC% (*Deinococcus Radiodurans*) and one low GC% bacterium (*Mycoplasma Pneumoniae*) that are not found in the human gut. We focused on the V3 region of the 16S rRNA gene because it contains the most phylogenetic information and is short enough to have read pairs overlap (section 3.2.1). Table 4.1 shows the V3 hypervariable region of the 16S rRNA gene of the two foreign bacteria: the *D. radiodurans* sequence is 186 bp long with a GC% of 56%; the *M. pneumoniae* sequence is 201 bp long with a GC% of 42%. Note that, for the Illumina HiSeq 100 bp paired-end reads used in this study, 201 bp is actually too long for paired-end overlap, but some of our read pairs did not include the entire

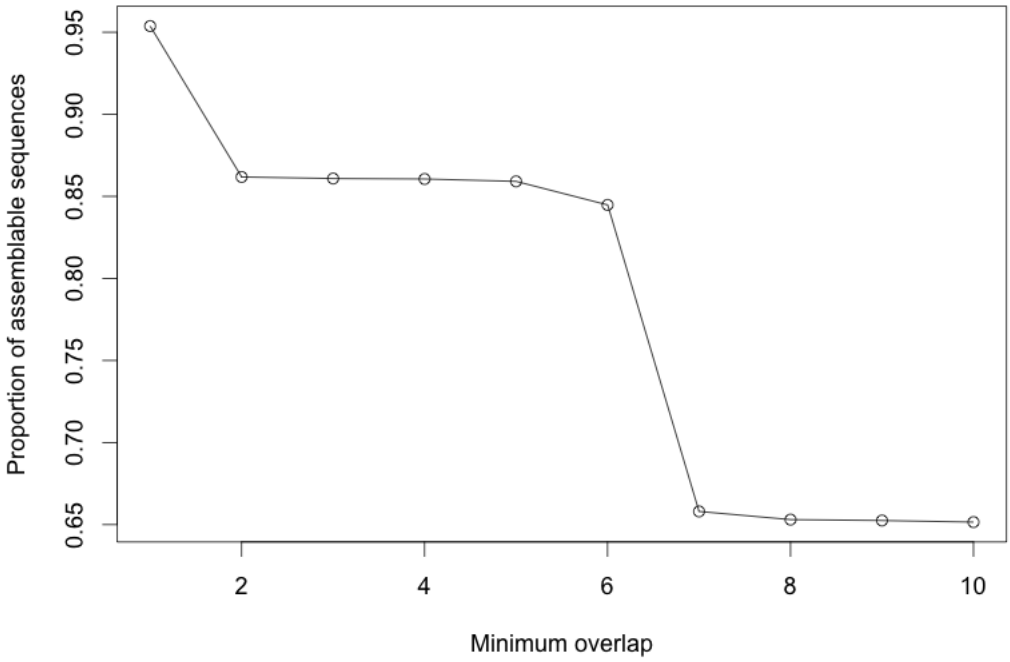


Figure 4.1: **Proportion of assemblable sequences at different minimum overlap lengths using 100 bp paired-end reads.** Gut bacterial sequences were manually selected from SILVA 104. At each minimum overlap length, a sequence is assemable if its length is $\leq$ (2 x 100 - *overlap*).

primer region (see Results section). To see if 100 bp paired-end reads have sufficient overlap for most gut bacterial sequences in the V3 region, we plotted the proportion of assemblable sequences at different minimum overlap lengths. Using the SILVA 16S rRNA database [75] (version 104), we manually selected 87,295 bacterial sequences that were extracted from mammalian fecal samples (using keyword search on sample sources annotated in the SILVA database). 85% of the gut bacterial sequences have V3 region lengths <= 194, which means a minimum overlap of 6 would successfully assemble them (Figure 4.1).

4.5.2 Study participants

Samples used in this study were collected from a cohort previously recruited from Group Health (GH), a health management organization in Washington State. Women were recruited based on criteria relevant to the parent study, an analysis of demographic, anthropometric, diet and lifestyle factors associated with soy isoflavone-metabolizing phenotypes [5]. Premenopausal women, ages 40-45, were selected by breast density score on their most recent GH screening mammogram in order to obtain a distribution of women across BIRADS scores. Women were ineligible if they: currently used hormone therapy (HT) or oral contraceptives (OC), or used them in the 6 mo before their screening mammogram; had a history of, or current, breast cancer; used tamoxifen or raloxifene; had breasts or ovaries removed; had a diagnosis of gastrointestinal disorders such as Crohn's disease or UC, or gastrointestinal surgery within 10 y of their most recent mammogram; currently used antibiotics or used them in the 12 mo preceding the sampling date; or were allergic to soy beans or soy protein. A total of 203 eligible women attended an initial study clinic visit during the parent study. All study procedures were approved by the Institutional Review Boards of the Fred Hutchinson Cancer Research Center (FHCRC) and GH, and all participants provided written informed consent.

4.5.3 Sample preparation and sequencing

Microbial genomic DNA was extracted from fecal samples using QIAgen stool mini kit (QIAgen, Valencia, CA) modified from Li et al. [48]. Bacterial 16S rRNA was amplified with V3 region primers 330F (5′-ACT CCT ACG GGA GGC AGC AGT-3′) and 530R (5′-GTG CCA GCA GCC GCG GTA ATAC-3′).

For the standard additions, we purchased *M. pneumoniae* and *D. radiodurans* gDNA (only the 16S rRNA sequence, accession number AF132740 and AF289089) from ATCC and reconstituted DNA in TE buffer. We then mixed the two gDNA from *M. pneumoniae* and *D. radiodurans* in equal amount and added it to fecal gDNA from Sample Set 1 and 2. The final concentrations were 0.1%, 0.032%, 0.01%,

0.0032%, and 0.001% of the total gDNA. The mixtures were then amplified using PCR.

To determine the appropriate cycle number, we amplified six of the cohort samples using qPCR on ABI-7900HT (ABI, Inc., Foster City, CA). Each 10μl mixture contained: 5 2X Invitrogen SYBR Green Mix, 0.1 μl of 10 μM forward and reverse primers, and 2.3 μl of DNA. Cycling conditions were: 2 min at 50°C, 10 min of denaturation at 95°C, 30 cycles of: [15 sec at 95°C, 42 sec at 60°C] with melting curve test. All qPCR samples were done in duplicates and mean values were used for further calculation. The curve test (Ct) values for each sample were determined using SDS software (ABI, Inc., Foster City, CA). The Ct values varied between 19.7 and 21.9. If Ct values of two samples differed by n, the baseline gDNA concentrations should differ by 2^n. We then diluted the original gDNA samples to the concentration so that Ct values approximated 22.5. In this case, the amplification phase remained linear until 28 cycles. Thus, we determined 28 cycles to be the appropriate number for PCR on all EBB samples.

For PCR, each 50 μl reaction mixture contained: 5 μl 10X Buffer B (Qiagen), 6 μl 25 mM $MgCl_2$, 10 mM of each dNTP, 0.5 μl of 10 muM forward and reverse primers, 1 U of Taq polymerase (Fisher Scientific), and 3.5 μl of DNA. Cycling conditions were: 3 min of denaturation at 95°C, 28 cycles of: [30 sec at 95°C, 30 sec at 55°C, 12 sec at 72°C], then a final 7 min extension step at 72°C. PCR products were further purified using QIAquick 96-well purification kit (Qiangen, Inc., Valencia, CA) using the manufacturer's protocol to remove unincorporated nucleotides and primers. DNA was quantified by determining absorption of samples at 260 nm and purified DNA was submitted for sequencing.

PCR amplified samples were sent to the John Stamatoyonnopoulos lab. Libraries were prepared following the protocol supplied with Illlumina's TruSeq DNA Prep Kit. TruSeq v1 adapters were used. A total of 85 samples (including the standard additions samples) were multiplexed using 6-mer barcodes and loaded onto the flowcells at 8 pM concentrations. Sequencing was done on an Illumina HiSeq machine (Illumina, Inc., San Diego, CA) with Illlumina's TruSeq SBS V3 Kit kits. The PhiX

control lane was run on a separate lane (lane 8) than the samples.

4.5.4 Results

We sequenced two sets of standard additions samples on 3 different lanes and obtained 4-7 million 100 bp read pairs after quality filtering (Table 4.2). The quality filtering steps included matching sample barcodes, detecting and removing primer matches, and removing low quality reads based on Phred score (section 4.6.2.1). We extracted *M. pneumoniae* and *D. radiodurans* read pairs using the criteria: (1) both orientations were $\geq 90\%$ similar to *M. pneumoniae* or *D. radiodurans*, (2) RDP Classifier (version 2.3, trained with GreenGenes 2011 taxonomy) correctly classified the read pairs with confidence score $\geq 50\%$. *M. pneumoniae* and *D. radiodurans* reads were only found in the samples to which they were added (no cross-sample contamination).

Addition %	ID	Set	Barcode	Lane	Total reads	MP reads	DR reads
0.001%	DS21061	1	ATCACG	6	5,120,011	44 (0.0009%)	38 (0.0007%)
	DS21066	2	GCCAAT	6	7,050,383	0 (0%)	0 (0%)
0.0032%	DS21060	1	CTTGTA	5	4,456,853	57 (0.0013%)	72 (0.0016%)
	DS21109	1	CTTGTA	7	4,305,442	90 (0.0021%)	68 (0.0016%)
	DS21065	2	ACAGTG	6	7,508,043	0 (0%)	0 (0%)
0.01%	DS21059	1	GGCTAC	5	4,769,683	497 (0.0104%)	189 (0.0040%)
	DS21064	2	TGACCA	6	5,431,036	470 (0.0087%)	328 (0.0060%)
0.032%	DS21058	1	TAGCTT	5	5,402,865	1,090 (0.0202%)	645 (0.0119%)
	DS21063	2	TTAGGC	6	5,306,906	2,377 (0.0448%)	796 (0.0150%)
0.1%	DS21057	1	GATCAG	5	4,804,522	5,454 (0.1135%)	1,466 (0.0305%)
	DS21062	2	CGATGT	6	6,124,291	8,430 (0.1376%)	3,098 (0.0506%)

Table 4.2: **Run information on the standard addition samples.** Two foreign bacterial 16S rRNA sequences, *M. pneumoniae* (MP) and *D. radiodurans* (DR) were added to two human gut microbiome samples in varying concentrations. The two samples are denoted Set 1 and 2. For Set 1, the concentration 0.0032% has two technical replicates (DS21060, DS21109). For each sample, we show the set number, barcode, lane, and total number of quality read pairs obtained. The last two columns show the number of recovered MP and DR read pairs and its relative proportion to the total number of quality read pairs (in parenthesis). Ideally, the recovered relative proportions should be the same as the added proportions (first column in Table). The discrepancy between the added and recovered MP/DR proportions is plotted in Figure 4.2.

4.5.4.1 ***D. radiodurans is less efficienctly amplified than M. pneumoniae***

The last two columns of Table 4.2 show the numbers and relative proportions of extracted *M. pneumoniae* (MP) and *D. radiodurans* (DR) reads. For both sample sets, we recovered at least one read at added proportions above 0.01%. Note that even though two of the Set No.2 samples had more total reads (DS21066, DS21065), we were still not able to recover the foreign bacteria at 0.0032% or 0.001%. This suggests that the detection limit for Illumina samples is 0.01%. The recovery slopes for *M. pneumoniae* are slightly above 1, while the recovery slopes for *D. radiodurans* are 0.51 and 0.3 (Figure 4.2). A possible explanation for this is PCR and Illumina amplification bias (section 4.2). Recall that Aird et al. found that sequences with $\geq$ 56% GC content were significantly less efficiently amplified [3], and the GC% of the 16S rRNA gene V3 region for *D. radiodurans* is exactly 56%. Even though we controlled for PCR amplification bias by reducing the number of cycles (28 cycles), we probably did not fully eliminate it; we also had no control over Illumina's amplification bias.

Our results indicate that amplification bias will likely lead to inaccurate estimation of absolute species abundances. To what extent GC% affects Illumina HiSeq is something we cannot conclude by using only two bacterial species. For the study of human gut microbiome, however, this may be less of a concern if the GC% of the sequenced bacteria are homogenous: the majority of the gut bacterial sequences from SILVA 104 had a GC content between 50-56% (Figure 4.3).

4.5.4.2 ***Quality scores are mildly indicative of base errors***

Next, we looked at whether Phred scores are indicative of base errors. We plotted the average Phred score at each read position (excluding primer positions) for all correct and erroneous bases (Figure 4.4). A base was erroneous if it did not match the corresponding base from its source sequence. 19,130 out of 4,019,793 bases were erroneous (error rate: 0.0047, or 0.47%). The average Phred scores decreased with read position, as expected, and the average Phred scores for erroneous bases were

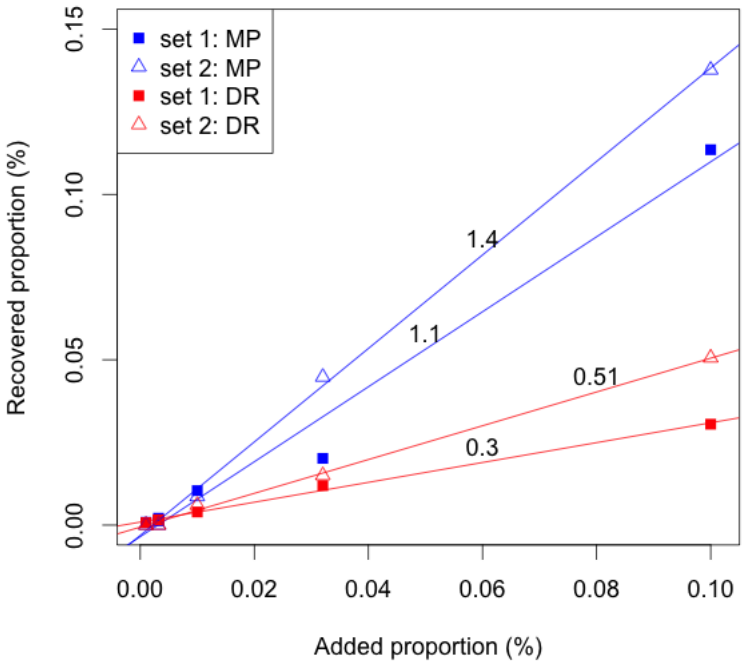


Figure 4.2: **Proportion of added versus recovered *M. pneumoniae* (MP) and *D. radiodurans* (DR) sequences.** Recovered proportions for the low GC *M. pneumoniae* were higher than the GC-rich *D. radiodurans* . Slope and standard deviation for the sets are: 1.1±0.08 (Set 1, MP), 1.4±0.029 (Set 2, MP), 0.3±0.009 (Set 1, DR), 0.51±0.014 (Set 2, DR).

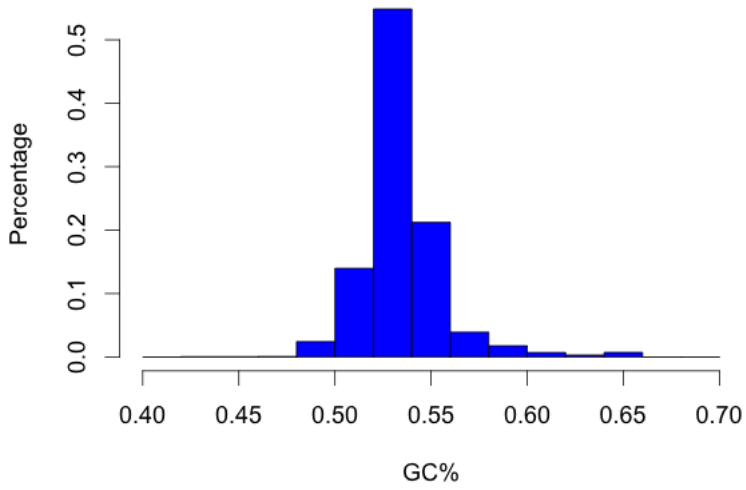


Figure 4.3: **GC% of gut-related bacterial 16S rRNA gene sequences.** We selected 87,295 bacterial sequences extracted from mammalian fecal samples from the SILVA database and computed the GC% for the entire 16S rRNA gene region.

lower than that of correct bases. This suggests a potential reduction in sequencing errors using Phred scores and we can think of two possible ways of filtering: (1) discard all reads with one or more bases of Phred score below some threshold; (2) simply mark low quality bases as erroneous.

Applying approach (1), we tallied the number of MP/DR read pairs that remained after using Phred score cutoffs of 0, 10, 20, 25, and 30 (Table 4.3). No read pairs remained if we used a score cutoff of 35. At score cutoff 10, 15, 20, and 25, we were able to recover MP/DR read pairs for all samples, however their relative proportions varied. The number of DR reads dropped more dramatically than MP, with the relative proportions reduced by roughly a factor of 2 at cutoff 25. This may again be associated with Illumina having trouble sequencing GC-rich sequences, resulting in lower Phred scores for DR reads.

We tested the alternative approach of marking bases below a Phred score cutoff as erroneous. The tradeoff is that we also marked some correct bases as incorrect simply because they fell below the cutoff (Figure 4.5). At cutoff 25, we identified $\sim$50% of the erroneous bases and misclassified $\sim$0.05% of correct bases. At cutoff 35,

we have detected 70% of the erroneous bases, yet also marked the same percentage of correct bases! The Matthews Correlation Coefficient (MCC), which is calculated using $(TP \times TN - FP \times FN)/\sqrt{((TP+FP)(TP+FN)(TN+FP)(TN+FN))}$, depicts the tradeoff between maximizing true positives (TPs, correct bases above cutoff) and true negatives (TNs, erroneous bases below cutoff) and minimizing false positives (FPs, erroneous bases above cutoff) and false negatives (FNs, correct bases below cutoff). Using approach (2) means we get to keep more sequences, but may complicate the use of processing software such as Mothur [85] and Qiime [13], which do not yet include tools that handle marked base errors.

In summary, a Phred score cutoff of up to 25 may be reasonable for detecting species at abundances as low as 0.01%. However, GC-rich sequences are disproportionately overfiltered at higher Phred score cutoffs. Alternatively, it is possible to use Phred scores to mark erroneous bases, but this results in incorrectly marking correct bases. Finally, neither approach fully eliminates base errors. We show in later sections (and Table 4.5) that even with a Phred score cutoff of 25, the numbers of unique (100% similarity) MP/DR sequences were way above 1. Only by relaxing the similarity cutoff down to 97% or 95% did the number of clusters/OTUs reflect the ground truth.

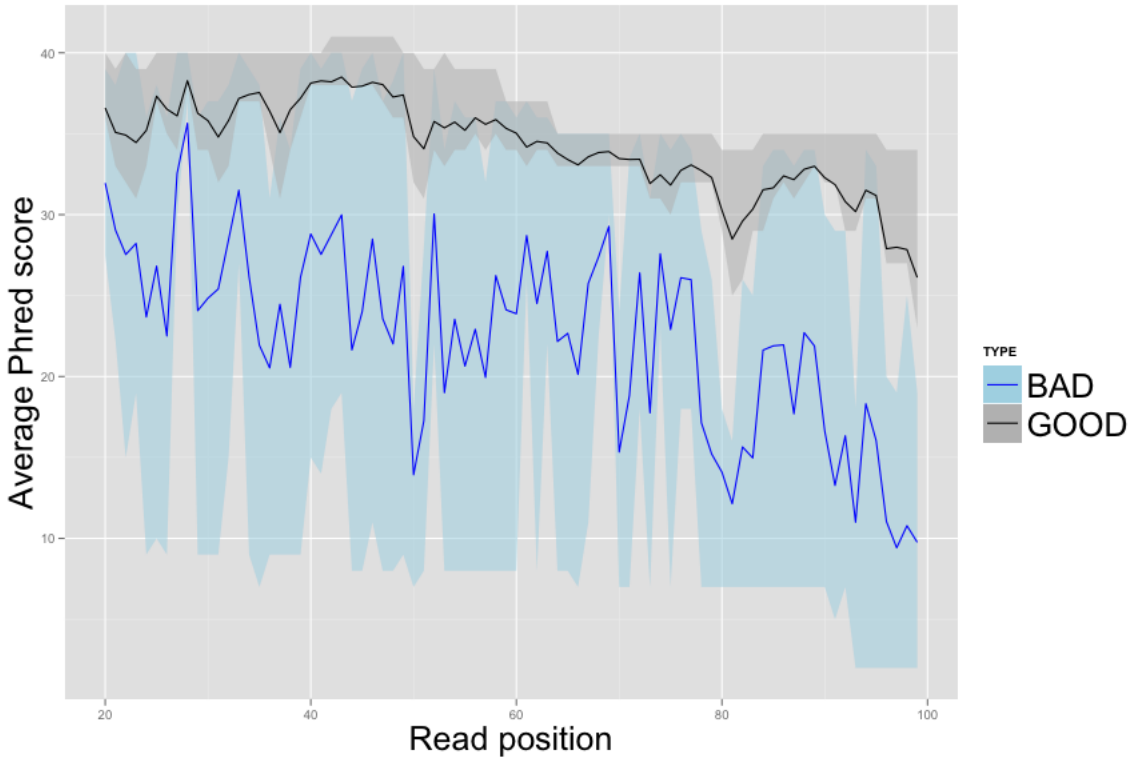


Figure 4.4: **Average Phred score per position of correct and erroneous bases.** Primers have been removed and are not shown here. The average Phred score of erroneous bases (blue line) was lower than the average Phred score of correct bases (black line). Shades represented the 25% and 75% quantile. Phred scores appeared to be gradually decreasing with read position. Sharp declines in average Phred scores corresponded to elevated base error rates around positions 22-23, 71-72, and 81 as plotted in Figure 4.8.

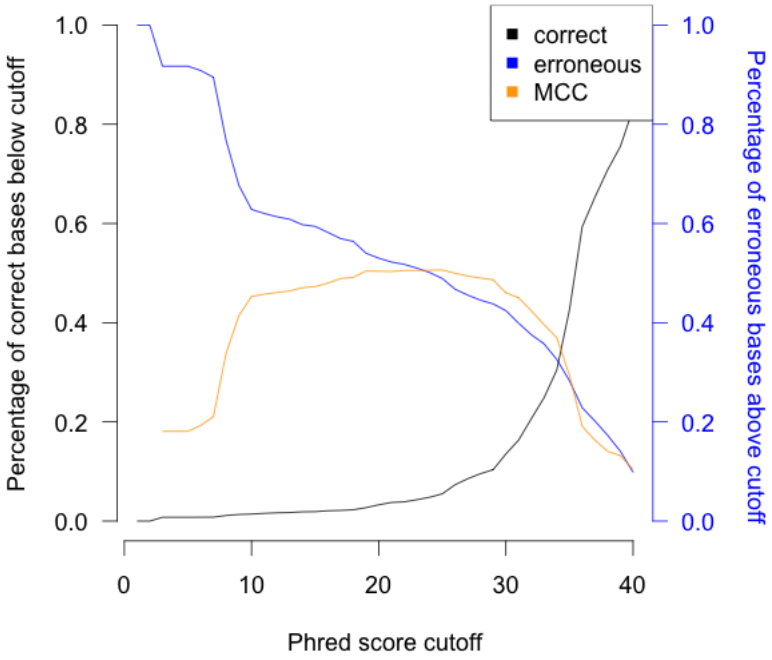


Figure 4.5: **The percentage of erroneous bases above Phred score cutoff (blue, false positive) and percentage of correct bases below Phred score cutoff (black, false negative).** As we increased the Phred score cutoff, we identified more erroneous bases but also misclassified an increasing portion of correct bases. The Matthews Correlation Coefficient (orange, MCC) depicts the tradeoff.

Added Proportion	Sample	Type	Phred score cutoff				
			0	10	20	25	30
0.001%	DS21061	MP	44 (0.0009%)	19 (0.0007%)	7 (0.0006%)	2 (0.0004%)	0 (0%)
		DR	38 (0.0007%)	13 (0.0005%)	9 (0.0008%)	4 (0.0009%)	0 (0%)
	DS21066	MP	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
		DR	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
0.0032%	DS21060	MP	57 (0.0013%)	33 (0.0014%)	12 (0.0010%)	4 (0.0007%)	0 (0%)
		DR	72 (0.0016%)	35 (0.0014%)	15 (0.0012%)	3 (0.0005%)	0 (0%)
	DS21109	MP	90 (0.0021%)	24 (0.0015%)	4 (0.0007%)	3 (0.0015%)	0 (0%)
		DR	68 (0.0016%)	18 (0.0011%)	5 (0.0009%)	2 (0.0010%)	0 (0%)
	DS21065	MP	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
		DR	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
0.01%	DS21059	MP	497 (0.0104%)	229 (0.0088%)	101 (0.0082%)	50 (0.0089%)	0 (0%)
		DR	189 (0.0040%)	89 (0.0034%)	36 (0.0029%)	11 (0.0020%)	0 (0%)
	DS21064	MP	470 (0.0087%)	238 (0.0093%)	95 (0.0090%)	46 (0.0113%)	0 (0%)
		DR	328 (0.0060%)	159 (0.0062%)	56 (0.0053%)	20 (0.0049%)	0 (0%)
0.032%	DS21058	MP	1,090 (0.0202%)	564 (0.0192%)	251 (0.0176%)	141 (0.0211%)	0 (0%)
		DR	645 (0.0119%)	316 (0.0108%)	116 (0.0081%)	29 (0.0043%)	0 (0%)
	DS21063	MP	2,377 (0.0448%)	1,149 (0.0469%)	450 (0.0472%)	261 (0.0735%)	0 (0%)
		DR	796 (0.0150%)	337 (0.0138%)	103 (0.0108%)	30 (0.0084%)	0 (0%)
0.1%	DS21062	MP	8,430 (0.1376%)	4,217 (0.1461%)	1,666 (0.1442%)	926 (0.2122%)	9 (0.0731%)
		DR	3,098 (0.0506%)	1414 (0.0490%)	456 (0.0395%)	138 (0.0316%)	4 (0.0325%)
	DS21057	MP	5,454 (0.1135%)	2,822 (0.1074%)	1,179 (0.0935%)	636 (0.1096%)	2 (0.0094%)
		DR	1,466 (0.0305%)	698 (0.0266%)	264 (0.0209%)	89 (0.0153%)	3 (0.0141%)

Table 4.3: **Number and relative proportion (in parenthesis) of MP and DR read pairs using different Phred score cutoffs.** For each standard additions sample and phred score cutoff, we discarded all reads where one or more bases have Phred scores below the cutoff. At Phred score cutoff 35, there were no MP or DR reads. At Phred score cutoffs below 35, the number of DR read pairs dropped more dramatically than MP. We highlighted several cells in yellow where the relative proportion changed dramatically depending on the Phred score cutoff used. For example, for sample DS21057 (0.1% addition), the relative proportion of DR read pairs went from 0.03% to 0.01% with a Phred score cutoff of 0 and 25.

Figure 4.6: **Sequencing cycle versus reference position.** Our reads didn't always start at the first base of the primer region. We use *sequencing cycle* to refer to the position on the read and *reference position* to refer to the position on the MP/DR sequence.

4.5.4.3 ***Sequencing errors in the primer region***

In this section, we examine base errors in the primer region. Due to possible primer degradation and incorrect adapter trimming, our reads did not always begin with the first base of the primer region. 60% of the MP/DR reads started at primer position 1, 20% at position 2, with about 1-2% at position 3, 4...14, which was our cutoff for primer matching. For clarity, we use the term *sequencing cycle/read position* to refer to the position on the read and *reference position* to refer to the position on the MP/DR sequence (Figure 4.6).

We calculated error rates in the primer region[1] and found no difference between MP and DR reads (Figure 4.7). This was expected as the primer sequences were identical. On the forward primer, the error rate was consistently higher at the 3rd and 4th position with consistent error conversion rates in both MP and DR (Figure 4.7): for position 3 (ref base T), the error conversion was { T>N: 30%, T>C: 22-25%, T>G: 35-39%, T>A: 6-7% }; for position 4 (should be C) it was { C>N: 14%, C>G: 10%, C>A: 51-53%, C>T: 22% }. Note that, the 12th position of the reverse primer was a mismatch to the MP sequence: It was G in the primer and A in MP; we found that 99% of the bases at this position were Gs. This was expected

[1]Base errors were excluded if (1) there were deletions in the primer region, or (2) it was the 12th position in the reverse primer region for MP because the reference base was a mismatch to the primer.

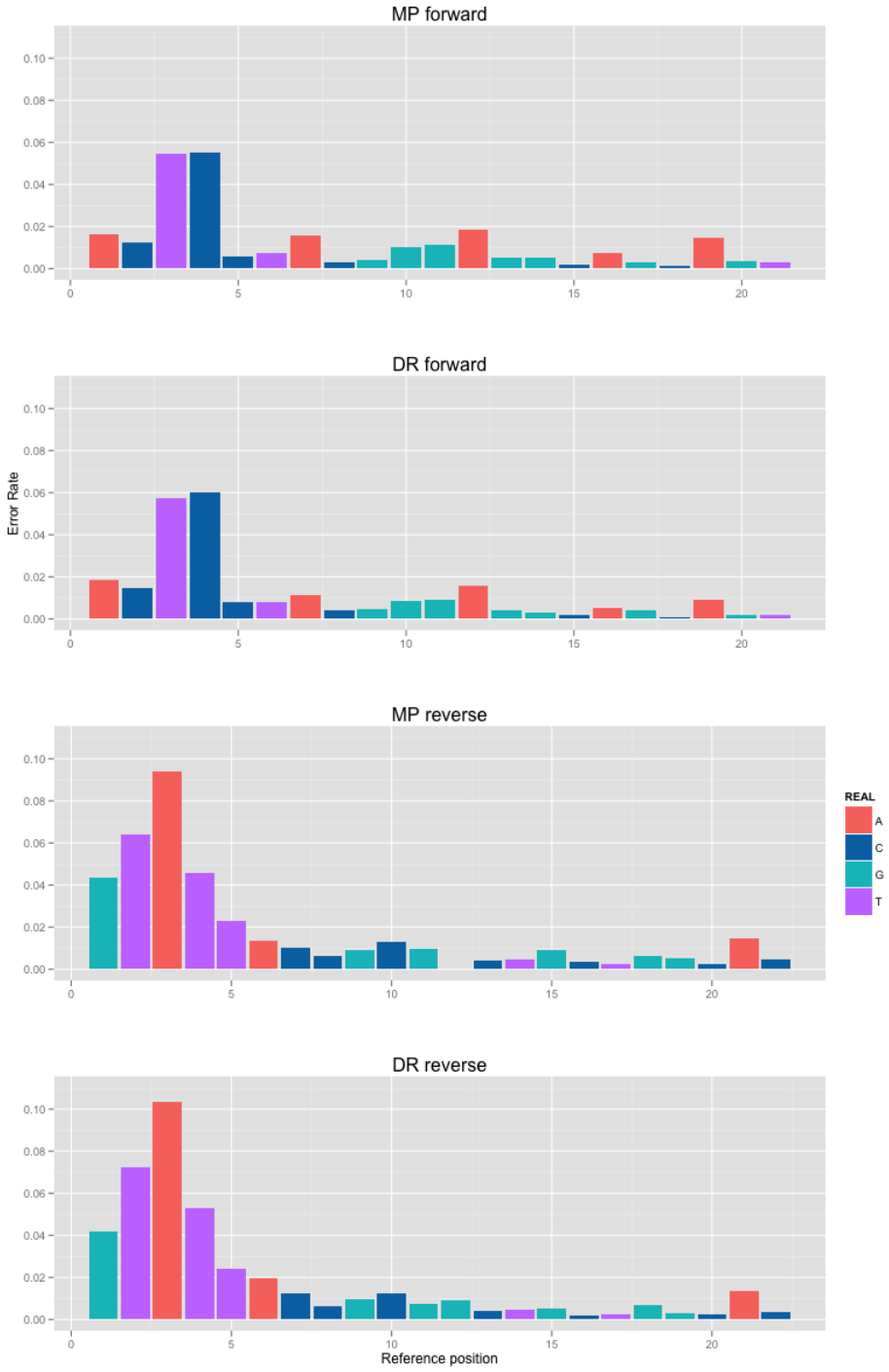


Figure 4.7: **Error rates in the primer region.** The error rate at each position is (*Number of erroneous bases*/*Total number of bases*). Length of forward primer: 21 bp. Length of reverse primer: 22 bp. The stack color at each position indicates the reference base. The 12^{th} position of the *M. pneumoniae* reverse primer region was a mismatch to the primer (in the primer: G, in MP: A) and was not considered a base error if the observed base was an A or G.

due to sequences being amplified via PCR. In total, we observed 14,534/1,022,207 base errors (error rate: 1.4%) in the primer region[2]. In the following sections, we excluded primers from all analyses.

4.5.4.4 ***Sequencing errors in the non-primer region***

We calculated the error rate at each sequencing cycle ($\frac{Number\ of\ incorrect\ bases\ at\ cycle\ i}{Total\ number\ of\ bases\ at\ cycle\ i}$) (Figure 4.8). This included all reads, MP or DR, forward or reverse, for each sample, with the primers excluded. The first set was sequenced on lane 5 and the second set was sequenced on lane 6. We did not find the error rates to be differently distributed across samples and lanes (t-tests were all insignificant except for between DS21063 and DS21064, p-value: 0.003; data not shown otherwise). We observed sharp spikes in error rates around cycle 22-23, 71-72, and 81, which mapped back to mostly As and Ts on the reference sequence (Figure 4.9). We saw a higher conversion of A/T to C/G than C/G to A/T (Figure 4.10), consistent with the findings of Nakamura et al. We did not find GGC or GGT to be particularly error-prone as Nakamura did. However, this may be due to the limited set of sequences we investigated—GGC and GGT were both underrepresented triplets in the V3 region of *M. pneumoniae* and *D. radiodurans*. Nakamura et al. and Meacham et al. looked at genome-wide human data and it was likely that all possible triplets were more evenly represented in the sequenced regions. Another possibility is that the GGC/GGT pattern was no longer present in Illumina HiSeq. We did not find an association between inverted repeats and base errors.

4.5.5 Conclusion

To summarize, we found sequencing errors to be associated with Phred scores, nearby bases and sequencing cycle/read position. In the following section, we introduce an error correction method for reducing sequencing errors and show that it improves OTU clustering results in our own and published datasets.

---

[2]This excludes the 12-th position A>G error in MP reverse reads

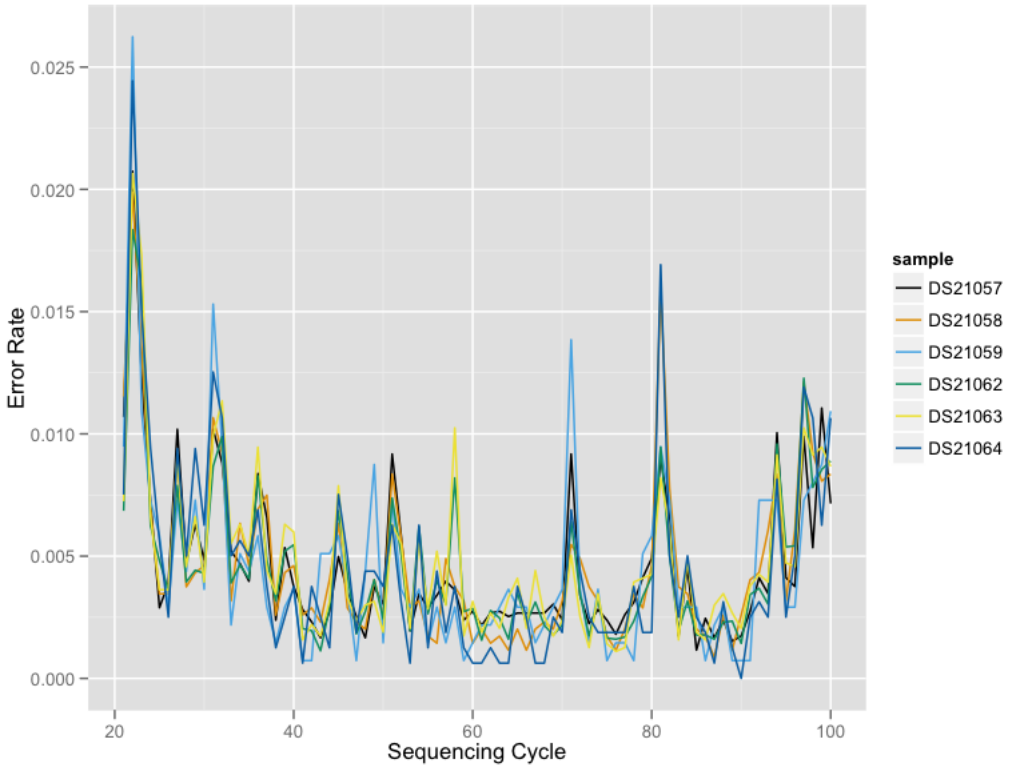


Figure 4.8: **Error rate at each cycle/read position.** We excluded samples with too few recovered read pairs (DS21055, DS21056, DS21060, DS21061, DS21065, DS21066). Samples used here included all reads, *M. pneumoniae* or *D. radiodurans*, forward or reverse, with the primers excluded.

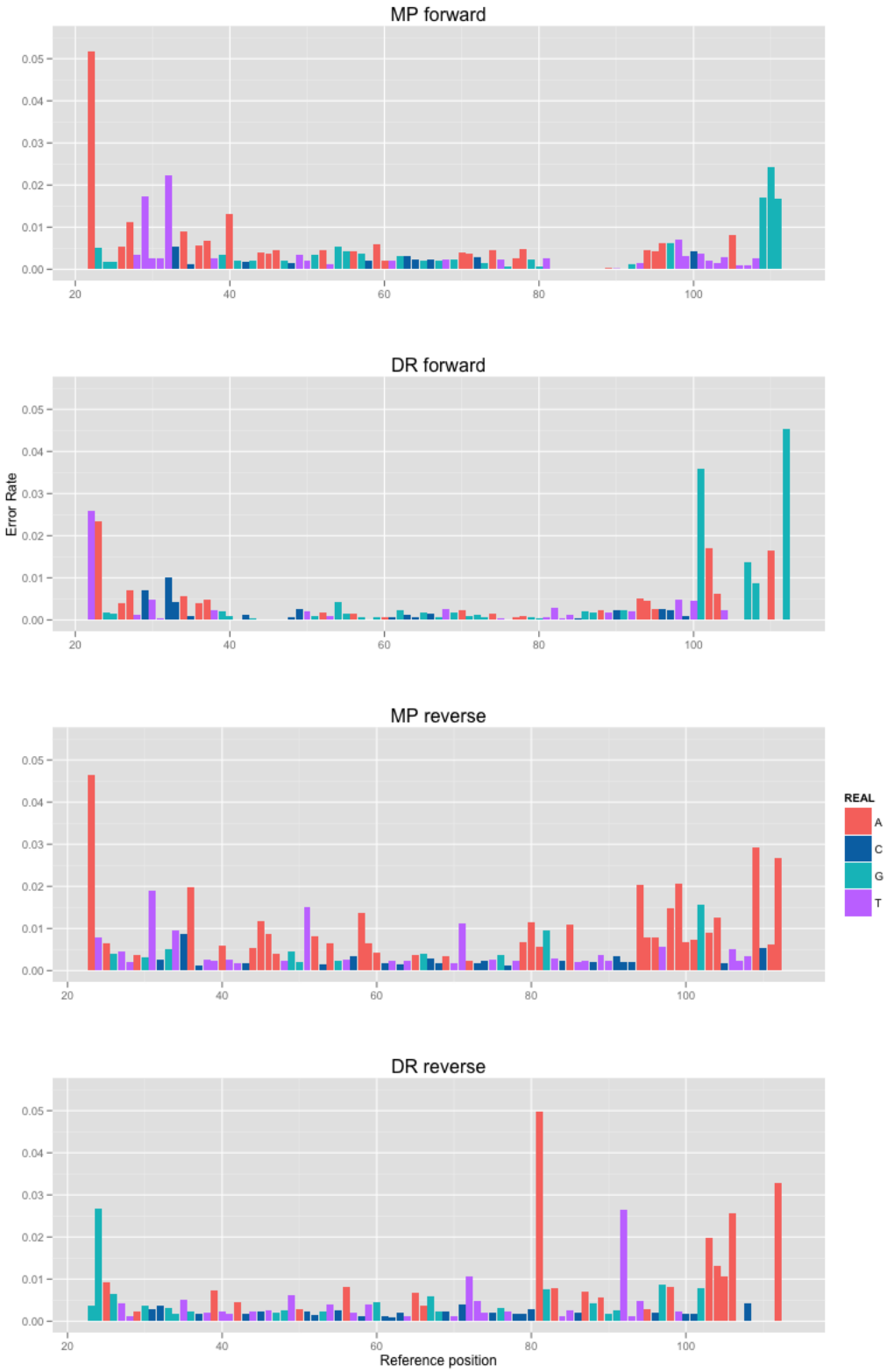


Figure 4.9: **Error rates in the non-primer region.** The stack color indicates reference base (also refer to Table 4.1). The highest error rates occured at bases immediately proceeding the primers and near the 3' end. A/Ts had higher error rates than C/Gs.

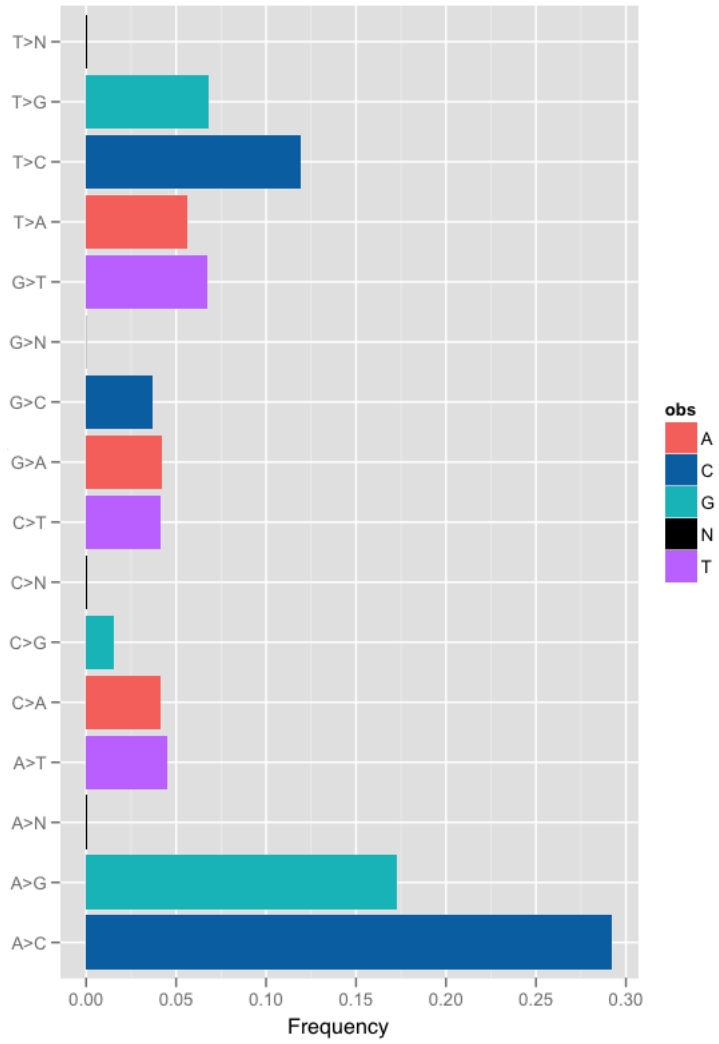


Figure 4.10: **Frequency of base error conversions (*expected* > *observed*) in our standard additions samples.** The errors were tallied over all MP/DR read pairs, both read orientations, from all samples. We observed a higher incidence of A/T to C/G than C/G to A/T.

4.6 ErrCor: an error correction method for reducing systematic errors in Illumina sequencing

4.6.1 Outline

We developed an error correction pipeline called ErrCor to detect and correct base errors. We trained a Support Vector Machine (SVM) classifier based on sequencing cycle, Phred score, the target base and its preceding two bases (section 4.6.3). Our classifier is similar to the work of Meacham et al. [61]), who trained a linear regression model using similar features for detecting rare SNPs in human datasets. Our problem is different from SNP calling, however, in that not only do we need to determine whether a base is correct or erroneous, but if it is erroneous, we need to know what the true base is. Training a multi-class classifier is one potential way of determining the correct base, but since we only have a small set of training data, we instead leverage the fact that the majority of the reads sequenced from the same species will either be 100% correct or have only a handful of errors. The intuition is as follows: If we see two sequences A and B that are only different by, say, two positions, and our classifier calls both positions incorrect on either A or B, we can "correct" the bases by merging A and B together (section 4.6.4). For paired-end data, we apply error correction to forward and reverse reads independently, then assemble them using an overlap-finding algorithm (section 4.6.5). Finally, assembled reads are clustered into OTUs (section 4.6.6).

4.6.2 Methods

4.6.2.1 Initial quality filtering of raw reads

Figure 4.11 depicts the ErrCor pipeline. The initial filtering phase is dataset-specific. For our dataset, we discarded all reads that did not match sample barcodes (up to 2 mismatches allowed) or matched the contaminant database (phiX, human chromosome M, adapters). We then retained pairs that satisfied the following: (1) one read matched the 330F primer and the other matched the 530R primer, each match having no more than 2 mismatches and 1 deletion, and (2) the expected number of

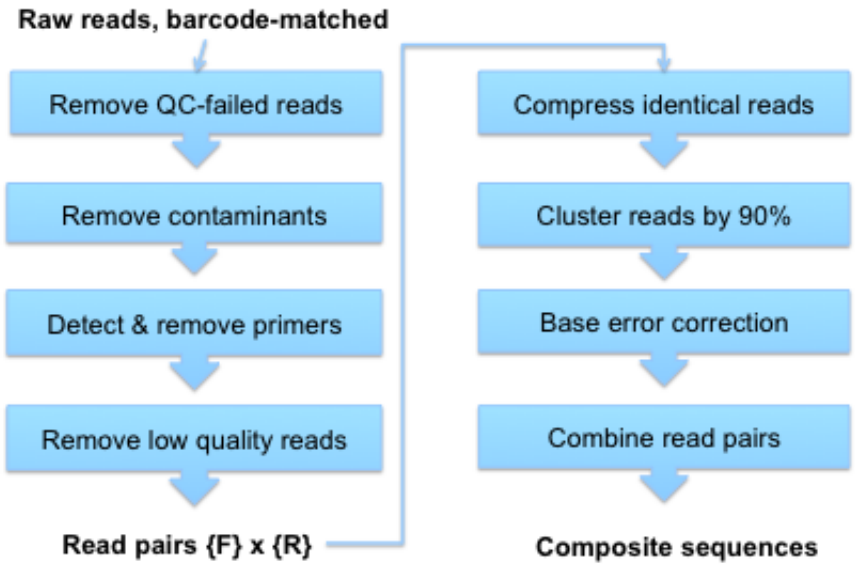


Figure 4.11: **Flowchart for processing paired-end Illumina reads.**

erroneous bases (based on Phred scores) was less than 10 for both reads. Step (1) removed ∼10% of the read pairs. Step (2) removed an additional 10-15%.

4.6.3 ErrCor model training and prediction

We use the `ksvm` function from the R library `kernlab` (version 0.9-14). To train the model, we use the bound constraint SVM classification (C-BSVC) with a Gaussian kernel (RBFDOT) with the following parameters:

```
ksvm(Class~, data=<data>, type="C-bsvc", C=10, kpar=list(sigma=0.1),
                prob.model=TRUE, kernel="rbfdot", cross=10)
```

This outputs a model which is saved as an R object and used to predict the bases as correct ('+') or incorrect ('-'). We use 5 features: $(cycle, p_0, b_{-2}, b_{-1}, b_0)$ for the

SVM classifier, where p_0 is the Phred score of the target base, and b_{-2}, b_{-1}, b_0 are the two preceding and target bases.

4.6.4 Iterative error correction

We describe below the iterative error correction steps:

STEP 1: COLLAPSING READ PAIRS INTO SUPER READS The input reads could contain tens of thousands of different species. It is impractical and inefficient to apply error correction to all of them as a single group. We use 90% OTU clustering to group reads so that each group is of manageable size and likely to contain the same species. Typically, species are defined at 97% sequence identities, and we do not expect to see more than 7% sequencing error after our initial filtering steps. We compress reads that are either identical or are perfect subsequences of another using Qiime's trie clustering (Qiime version 1.5.0 [13]). Each compressed read is a "super read", consisting of (the longest) read sequence, read count, and an average Phred score at each position. Compressing reads reduces error correction accuracy, however it takes significantly more time to apply the model to every single base in the dataset. Instead, we compress reads to reduce runtime and modify our algorithm to use the super read size to improve accuracy.

STEP 2: ITERATIVE ERROR CORRECTION Within each input cluster from step 1, we first sort the super reads in decreasing order by size. For a pair of super reads, position i is conflicting if the bases are different at that position. The position is "correctable" if it satisfies one of the conditions listed in Table 4.4. The super read sizes are used to put trust in the more abundant super read. If the less abundant super read has the correct base, we will only trust it if both super reads have counts < 10 or the count ratio between the larger and smaller super reads is less than 10. A pair of super reads is merged if all differing bases are correctable. To merge super reads, we correct all differing positions according to the output in Table 4.4, combine the read sizes, and re-calculate the average (size-weighted) Phred scores of non-differing positions. We start from the most abundant super reads and iteratively

*input: base b_1, base b_2, size s_1, size s_2	
Correctable Condition	**Output**
b_1 == 'N' OR b_2 == 'N'	N
!isCorrect(b_1) AND !isCorrect(b_2)	N
isCorrect(b_1) AND !isCorrect(b_2)	b_1
!isCorrect(b_1) AND isCorrect(b_2) AND trustworthy(s_1, s_2)	b_2
*definition: trustworthy(s_1, s_2) = $s_1 < 10$ OR $\frac{s_1}{s_2} < 10$	

Table 4.4: **Satisfying conditions for correctable bases.** It is always assumed that $s_1 \geq s_2$.

merge pairs of super reads until no merging can be done.

STEP 3: PAIRED-END READS ASSEMBLY After step 2, we obtain corrected super reads for forward and reverse reads. We keep track of the original read pairs, so if read pairs (F_1, R_1), (F_2, R_2), (F_3, R_3) are grouped into super reads $SF_1 = \{F_1, F_3\}$, $SF_2 = \{F_2\}$ for forward and $SR_1 = \{R_1\}$, $SR_2 = \{R_2, R_3\}$ for reverse, we run assembly on all combinations $SF_1 \times SR_1$, $SF_2 \times SR_2$, $SF_1 \times SR_2$ (section 4.6.5). The output is a single FASTQ file of error-corrected, assembled sequences.

4.6.5 Assembling paired-end reads

To assemble overlapping paired-end reads, we implement an algorithm similar to the ones described in Rodrigue et al. (SHERA [81]) and Miller et al. (EMIRGE [66]). There are two components to assembling overlapping reads: finding the best overlap (alignment) and calculating the Phred score (error probability).

Let the true base at position i be n, then the probability of observing base b with error probability e is:

$$\Pr(b \mid n, e) = \begin{cases} (1-e) & \text{if b=n} \\ e \times p_{n,b} & \text{otherwise} \end{cases} \tag{4.1}$$

where $p_{n,b}$ is often assumed to be $\frac{1}{3}$, i.e. a fixed probability of reporting one of the other three bases if there is an error. Let b_1, b_2, e_1, e_2 be the pairs of reported bases and error probabilities at an overlapping position. Let f_x denote the prior

base frequencies ($\sum_{x=A,T,C,G} f_x = 1$). The error probability we want to calculate then becomes:

$$1 - \Pr(n \mid b_1, b_2, e_1, e_2) = 1 - \frac{f_n \Pr(b_1 \mid n, e_1) \Pr(b_2 \mid n, e_2)}{\sum_{x=A,T,C,G} f_x \Pr(b_1 \mid x, e_1) \Pr(b_2 \mid x, e2)} \tag{4.2}$$

Both SHERA and EMIRGE set $p_{n,b}$ to $\frac{1}{3}$. Our implementation allows both a user-provided set of frequencies and a fixed rate of $\frac{1}{3}$. For prior base frequencies, our program estimates them from the input data.

To find the best overlap between read pairs, we report the longest overlap such that the number of mismatches ≤ 2 and the overlap length ≥ 7 bp. For each overlapping base, we choose the consensus base to be the one with the smaller error probability and calculate the Phred scores[3] using Equation 4.2. Our approach is similar to SHERA, which reported the highest scoring alignment using a match/mismatch scoring scheme. EMIRGE, on the other hand, used BowTie against a reference database for read mapping. BowTie is useful when the reads do not overlap significantly, but is slower if the reference database is large and will fail to align reads not represented in the database. We apply both the faster overlap-finding algorithm and the BowTie-based assembly algorithm in our evaluation. For the BowTie reference database, we downloaded the SILVA SSURef database (version 108) which contains 618,442 16S rRNA sequences. The resulting BowTie index is 1GB. BowTie (version 0.12.7 [46]) is run as follows:

```
bowtie -chunkmbs 2000 -n 3 -p 10 -y -l 10 -q -fr -1 <fq1> -2 <fq2>
```

4.6.6 OTU Clustering

To cluster assembled sequences into OTUs, we run Qiime (version 1.5.0) with the command:

```
pick_otus.py -m uclust -s <similiarity cutoff>.
```

[3]We did not use the post-assembly Phred scores in our evaluations but only provided them as part of our program output.

When grouping reads into super reads, we use the additional parameter `-m trie`, which collapses identical sequences and sequences which are subsequences of other sequences. The similarity cutoffs we use are 90%, 95%, 97%, 99%, and 100%.

4.7 Results

4.7.1 Applying ErrCor to our standard addition dataset

The goal of error correction is to reduce errors so that both the number of OTUs and their representative sequences (often chosen as the most abundant sequence) reflect the ground truth. After error correction, we expect fewer OTUs. Table 4.7 and 4.8 show the number of OTUs after error correction using different OTU size cutoffs (0 and 10). At size cutoff=10, we excluded OTUs with 10 or fewer sequences. Compared with results from no error correction (Table 4.5, 4.6), we noticed several improvements. First, the percentage of assembled sequences increased from 97% to 99%-100%. This was because (a) there were fewer errors, and (b) the iterative merging of overlapping super reads increased paired-end overlaps. For example, the two reads in Figure 4.6 would be merged into a longer read, extending the 3' end. The *M. pneumoniae* (MP) sequence (including primer region) was 201 bp long, so most 100 bp paired-end reads did not overlap and was only assembled through reference-based alignment (BowTie). This was why the overlap finding algorithm had low assembled rates in non-error corrected reads for MP but not for the 186-bp DR (Table 4.5, 4.6)—here, the small percentage of assembled MP sequences (16-19%) came from reads that began at primer positions ≥ 2 (or more depending on the criterion for overlap). With iterative merging, reads were merged into super reads on both read orientations, and the overlaps became significant enough for the overlap-finding algorithm to succeed. For MP with Phred score cutoff 0 and 100% similarity, before error correction, the most abundant OTU had a size of 6,158 (no base error, overlap length 4 bp); after error correction, the most abundant OTU had a size of 15,094 (no base error, overlap length 24 bp).

In some cases, uncorrected or misclassified bases increased the number of OTUs.

For MP, at Phred score cutoff 25 and 97% similarity, the number of OTUs before error correction was 1 (2,017 sequences) but increased to 6 (2,056 sequences) after error correction. For the latter, OTU sizes were 2041, 6, 4, 2, 2, 1 (data not shown). Using no cutoff (size=0) and an arbitrary non-zero cutoff (size=10), we showed that many small OTUs came from reads with remaining errors.

Nevertheless, with error correction, the number of OTUs decreased while the number of assembled and clustered sequences increased. We also saw a drop in the total error rate (Figure 4.12). For MP, at Phred score cutoff 0, 100% similarity and OTU size cutoff 0, the error rates before and after error correction were 0.43% and 0.20%. For DR, the error rate went from 0.30% to 0.17%.

Phred cutoff	Type	Total	Method	Assembled	Number of OTUs (of size > 0)				
					100%	99%	97%	95%	90%
0	MP	18,510	BowTie	17,988 (97%)	3,298	2,017	247	22	1
			overlap	3,020 (16%)	730	415	33	7	1
	DR	6,700	BowTie	6,627 (98%)	767	399	8	1	1
			overlap	6,685 (99%)	799	417	13	2	1
10	MP	9,295	BowTie	9,065 (97%)	956	406	1	1	1
			overlap	1,582 (17%)	281	94	6	2	1
	DR	3,079	BowTie	3,049 (99%)	282	63	1	1	1
			overlap	3,079 (100%)	300	72	3	1	1
15	MP	7,972	BowTie	7,763 (97%)	839	357	1	1	1
			overlap	1,357 (17%)	251	76	6	2	1
	DR	2,556	BowTie	2,528 (98%)	242	45	1	1	1
			overlap	2,556 (100%)	260	54	3	1	1
20	MP	3,765	BowTie	3,676 (97%)	467	201	1	1	1
			overlap	741 (19%)	148	43	4	1	1
	DR	1,060	BowTie	1,048 (98%)	125	15	1	1	1
			overlap	1,060 (100%)	136	22	3	1	1
25	MP	2,069	BowTie	2,017 (97%)	307	131	1	1	1
			overlap	378 (18%)	86	30	4	1	1
	DR	326	BowTie	320 (98%)	50	9	1	1	1
			overlap	326 (100%)	56	12	3	1	1

Table 4.5: **No error correction: Number of *M. pneumoniae* (MP) and *D. radiodurans* (DR) OTUs using different filtering and clustering criteria.** Reads were first filtered at different Phred score cutoffs (0, 10, 15, 20, 25). We then assembled the sequences using either (a) BowTie against a reference database, or (b) an overlap-finding algorithm without a reference database. Assembled sequences were clustered at different OTU similarity cutoffs. The overlap-finding algorithm had low assembly rate for MP (16-19%) but not DR sequences (99-100%) because the MP sequence is longer than the 100 bp paired end reads; it could only assemble the small percentage of MP reads that started at primer positions ≥ 2 and had overlap between the read pairs. The true number of MP and DR OTUs should be exactly 1 each.

Phred cutoff	Type	Total	Method	Assembled	Number of OTUs of size > 10 (Number of sequences)				
					100%	99%	97%	95%	90%
0	MP	18,510	BowTie	17,988 (97%)	100 (12,775)	93 (15,382)	2 (17,597)	1 (17,964)	1 (17,988)
			overlap	3,020 (16%)	15 (1,934)	2 (2,472)	3 (2,969)	1 (3,002)	1 (3,020)
	DR	6,700	BowTie	6,627 (98%)	27 (5,203)	4 (6,141)	1 (6,619)	1 (6,627)	1 (6,627)
			overlap	6,685 (99%)	28 (5,211)	5 (6,170)	2 (6,673)	1 (6,682)	1 (6,685)
10	MP	9,295	BowTie	9,065 (97%)	24 (7,149)	18 (8,151)	1 (9,065)	1 (9,065)	1 (9,065)
			overlap	1,582 (17%)	4 (1,130)	1 (1,443)	1 (1,557)	1 (1,577)	1 (1,582)
	DR	3,079	BowTie	3,049 (99%)	8 (2,542)	3 (2,982)	1 (3,049)	1 (3,049)	1 (3,049)
			overlap	3079 (100%)	8 (2,542)	4 (2,999)	2 (3,077)	1 (3,079)	1 (3,079)
15	MP	7,972	BowTie	7,763 (97%)	17 (6,112)	11 (6,980)	1 (7,763)	1 (7,763)	1 (7,763)
			overlap	1,357 (17%)	4 (978)	1 (1,240)	1 (1,338)	1 (1,352)	1 (1,357)
	DR	2,556	BowTie	2,528 (98%)	6 (2,103)	2 (2,473)	1 (2,528)	1 (2,528)	1 (2,528)
			overlap	2,556 (100%)	6 (2,103)	2 (2,477)	2 (2,554)	1 (2,556)	1 (2,556)
20	MP	3,765	BowTie	3,676 (97%)	6 (2,926)	4 (3,361)	1 (3,676)	1 (3,676)	1 (3,676)
			overlap	741 (19%)	4 (560)	1 (684)	1 (732)	1 (741)	1 (741)
	DR	1,060	BowTie	1,048 (98%)	1 (849)	2 (1,030)	1 (1,048)	1 (1,048)	1 (1,048)
			overlap	1,060 (100%)	1 (849)	2 (1,031)	1 (1,053)	1 (1,060)	1 (1,060)
25	MP	2,069	BowTie	2,017 (97%)	4 (1,586)	2 (1,829)	1 (2,017)	1 (2,017)	1 (2,017)
			overlap	378 (18%)	1 (264)	1 (341)	1 (371)	1 (378)	1 (378)
	DR	326	BowTie	320 (98%)	1 (256)	1 (307)	1 (320)	1 (320)	1 (320)
			overlap	326 (100%)	1 (256)	1 (307)	1 (321)	1 (326)	1 (326)

Table 4.6: **No error correction: OTUs containing 10 or more sequences**. Same as Table 4.5 except that OTUs containing ≤ 10 sequences were discarded. We show the number of OTUs of size > 10 and the total number of sequences contained in these OTUs (in parenthesis). For example, with Phred score cutoff 0, we obtained 18,510 read pairs, 3,020 of which were assembled by our overlap-finding algorithm. From Table 4.5 we see that this resulted in 730 OTUs; however only 15 of these OTUs contained more than 10 sequences, and together these size > 10 OTUs constitute 1,934 out of the 3,020 assembled sequences.

Phred cutoff	Type	Total	Assembled	Number of OTUs (of size > 0)				
				100%	99%	97%	95%	90%
0	MP	18,510	18,464 (99%)	740	447	37	6	1
	DR	6,700	6,695 (99%)	287	163	8	1	1
10	MP	9,295	9,265 (99%)	397	199	13	2	1
	DR	3,079	3,079 (100%)	186	89	6	2	1
15	MP	7,972	7,933 (99%)	358	124	9	2	1
	DR	2,556	2,556 (100%)	166	41	5	1	1
20	MP	3,765	3,749 (99%)	276	129	10	2	1
	DR	1,060	1,060 (100%)	82	20	3	2	1
25	MP	2,069	2,056 (99%)	178	41	6	2	1
	DR	326	326 (100%)	44	39	3	1	1

Table 4.7: **With error correction: Number of MP and DR OTUs.** Same filtering and assembly procedure as in Table 4.5 and Table 4.6. Only the overlap-finding assembly algorithm is shown here (the BowTie results are identical). For each similarity cutoff, the number of OTUs is shown. For example, using Phred cutoff 25, 2,056 out of 2,069 MP error-corrected read pairs were assembled and resulted in 178 100% OTUs; by contrast, without error correction and at the same Phred score cutoff, we obtained 307 OTUs (Table 4.5).

Phred cutoff	Type	Total	Assembled	Number of OTUs of size > 10 (Total number of seqs)				
				100%	99%	97%	95%	90%
0	MP	18,510	18,464 (99%)	88 (17,123)	47 (17,730)	7 (18,398)	3 (18,460)	1 (18,464)
	DR	6,700	6,695 (99%)	15 (6,011)	13 (6,296)	5 (6,681)	1 (6,695)	1 (6,695)
10	MP	9,295	9,265 (99%)	39 (8,315)	45 (8,767)	4 (9,239)	2 (9,265)	1 (9,265)
	DR	3,079	3,079 (100%)	5 (2,704)	4 (2,839)	2 (3,061)	1 (3,072)	1 (3,079)
15	MP	7,972	7,933 (99%)	31 (7,044)	23 (7,608)	3 (7,925)	1 (7,931)	1 (7,933)
	DR	2,556	2,556 (100%)	4 (2,234)	2 (2,431)	1 (2,540)	1 (2,556)	1 (2,556)
20	MP	3,765	3,749 (99%)	5 (3,201)	5 (3,384)	3 (3,734)	1 (3,743)	1 (3,749)
	DR	1,060	1,060 (100%)	1 (908)	1 (1,006)	2 (1,058)	1 (1,059)	1 (1,060)
25	MP	2,069	2,056 (99%)	1 (1,720)	2 (1,962)	1 (2,041)	1 (2,054)	1 (2,056)
	DR	326	326 (100%)	1 (267)	1 (268)	1 (322)	1 (326)	1 (326)

Table 4.8: **With error correction: OTUs containing more than 10 sequences.** Same as Table 4.7 except that OTUs containing ≤ 10 sequences were discarded. We show the number of OTUs of size > 10 and the total number of sequences contained in these OTUs (in parenthesis).

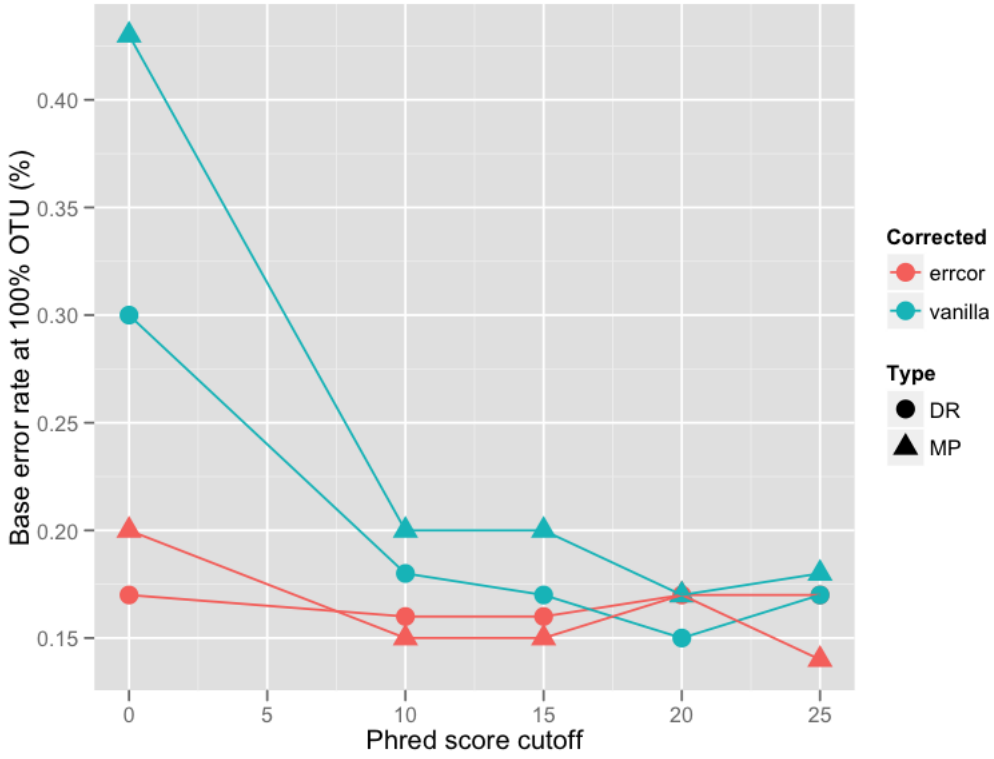


Figure 4.12: **Total error rate before and after error correction.** Shown here is the total error rate ($\frac{\text{Total number of erroneous bases}}{\text{Total number of bases}}$) using 100% OTU clustering and OTU size cutoff 0. At Phred score cutoff 0, the total error rate dropped from 0.43% (MP) and 0.30% (DR) to 0.20% (MP) and 0.17% (DR). *vanilla*: without error correction. *errcor*: with error correction.

4.7.2 Applying ErrCor to Ochman et al. Illumina dataset

We downloaded the raw Illumina FASTQ files from Ochman et al. [19]. The dataset consisted of 9 samples from two different runs (run 1: 75 bp paired-end Illumina GAIIx, run 2: 100 bp paired-end Illumina GAIIx). Samples came from three distinct datasets: (1) only *E. coli* K-12 (MG1655); (2) a mixture of 19 bacterial strains; or (3) fecal sample from a single lab mouse. The primers used were on the 16S rRNA V6 hypervariable region; 3 sets of primer pairs were used: 967F&1046R (denoted S_1V6), 970F&1050R (denoted S_2V6), and 917F&1061R (denoted LV6). We used

dataset (1) and (2) as the true sequences. Dataset (1) was run with the three different primer pairs and is denoted Ecoli_1, Ecoli_2, Ecoli_3. Dataset (2) was run with two different primer pairs and is denoted Mix_1 and Mix_2. We filtered the read pairs by barcode (perfect match required), primer (1 mismatch allowed, no indels), and low Phred quality (expected number of erroneous bases < 10). Primers were removed. For training data, we used only read pairs from Ecoli_1. Since *E. coli* MG1655 contains two distinct operons (Table 4.9), we retained read pairs only if its assembled sequence mapped to one of the two *E. coli* operons with less than 5 base errors. From the remaining read pairs we selected 2,000 correct and 2,000 erroneous bases for training. The trained classifier had an accuracy of 90% for correct bases and 73% for erroneous bases.

We evaluated the *E. coli*-only dataset by the number of assembled sequences that were 100% identical to one of the two *E. coli* operons before and after error correction. The two samples that were not used to train the classifier (Ecoli_2, Ecoli_3) had more *E. coli* sequences after error correction (Table 4.10). The number of correct *E. coli* sequences dropped for Ecoli_1 due to a single misclassification: In the Ecoli_1 sample, the most abundant cluster (size 340,607) had a single base error G>A. It resulted from the two most abundant super reads differing at that position. The more abundant super read (size 331,025) had the correct base G but the less abundant super read (size 66,706) had the erroneous base A. Similar to our own dataset, there was a high frequency of A>G conversion (Figure 4.13). The trained classifier misclassified the correct base as erroneous and vice versa, and since the super read size ratio was below 10, the erroneous base was accepted as the new base in the merged super read. This suggests that a more complicated scheme for deciding what super reads are trustworthy might have better results than our simplified ratio test.

Dataset (2) contained 19 bacterial strains[4], many of which shared identical or

[4]The 19 strains are: *Bacillus subtilis*, *Bacillus cohnii*, *Bacillus clarkii*, *Bacillus gibsonii*, *Staphylococcus intermedius*, *Staphylococcus vitulinus*, *Staphylococcus arlettae*, *Streptococcus suis*, *Streptococcus cristatus*, *Agrobacterium vitis*, *Agrobacterium radiobacter*, *Agrobacterium tumefaciens*, *Escherichia fergusonii*, *E. coli* K-12, 4 *Salmonella enterica* strains (14028S, SARB49, SARB1,

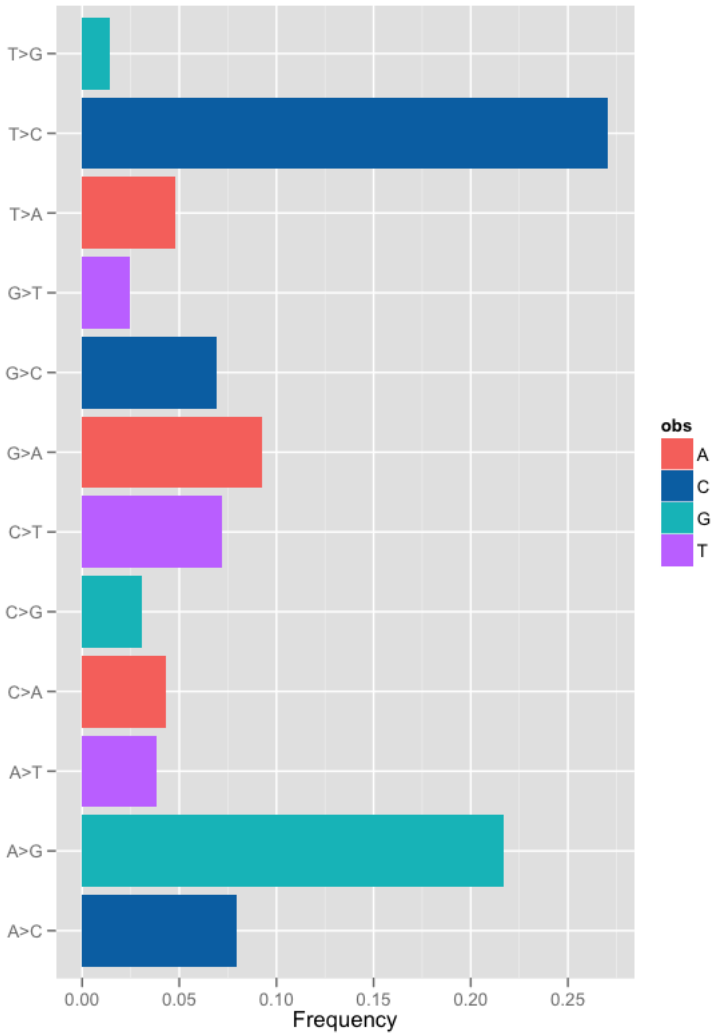


Figure 4.13: **Frequency of base error conversions from Ecoli_1 in Ochman et al.** We used reads from Ecoli_1 to train the classifier.

***E. coli* K-12|part of rrnA;rrnB;rrnC;rrnD;rrnE;rrnG operon**
TGGTCTTGACATCCACGGAAGTTTTCAGAGATGAGAATGTGCCTTCGGGAACCGTGAGAC
***E. coli* K-12|part of rrnH operon**
TGGTCTTGACATCCACAGAACTTTCCAGAGATGGATTGGTGCCTTCGGGAACTGTGAGAC

Table 4.9: **DNA sequencing of the V6 region of the 16S rRNA genes of *E. coli*K-12 (substr. MG1655).** Primer regions are not shown. There are two distinct V6 sequences, one is present in 6 rRNA operons and the other only in *rrnH*. Base differences are marked in red.

highly similar sequences in the 16S rRNA gene V6 region. We could not accurately map the assembled sequences (before or after error correction) unambiguously to the ground truth sequences. Instead, we BLAST-ed them against the ground truth sequences and counted the number of sequences that had at least one hit above 97% or 100% similarity. An OTU was considered a "good" OTU if its representative sequence (the most abundant sequence in the cluster) had a good BLAST hit. We then tallied the total number of sequences present in these good OTUs. Ideally, error correction should decrease the number of good OTUs but maintain or increase the total number of good sequences. With error correction, the number of OTUs was smaller, but we also had fewer good sequences (Table 4.11). When the non-error corrected (*vanilla*) sequences were subsampled to the same size as the number of good error-corrected sequences, the number of good OTUs were still at least 1.5 times more than with error correction. In other words, at the same number of good sequences, error correction resulted in smaller number of clusters. A high incidence of G>A and T>C was observed, suggesting that the classifier was prone to classifying G/Ts as errors when they were in fact correct. Despite our error correction efforts, the number of OTUs remained well above the ground truth number: 19.

SARB63), and *Xyllela fastidiosa*.

Phred cutoff	Sample	Total	Corrected?	
			vanilla	*errcor*
0	Ecoli_1	481,219	401,837	14,161
	Ecoli_2	313,385	273,078	283,375
	Ecoli_3	296,811	266,090	271,477
25	Ecoli_1	370,128	318,563	274,876
	Ecoli_2	213,831	194,289	197,371
	Ecoli_3	206,067	191,468	194,527

Table 4.10: **Assembled *E. coli* sequences that were 100% correct.** *vanilla*: without error correction. *errcor*: with error correction. The approach that resulted in more correct sequences is highlighted in yellow. Ecoli_2 and Ecoli_3 had more *E. coli* sequences after error correction. The low number for Ecoli_1 after error correction was due to the largest cluster having a single misclassified base. The three samples were run with different barcode-primer pairs. Ecoli_1: ATG-S_1V6, Ecoli_2: AGC-S_2V6, Ecoli_3: CCAT-S_2V6.

Phred cutoff	Sample	Total	BLAST hit	Good OTUs (Number of seqs)	
				vanilla	*errcor*
0	Mix_1	698,317	97%	3,987 (631,862)	1,037 (552,928)
			100%	932 (552,455)	370 (298,836)
	Mix_2	221,892	97%	2,056 (204,154)	537 (203,436)
			100%	525 (175,038)	215 (181,588)
25	Mix_1	448,816	97%	2,532 (409,156)	756 (394,165)
			100%	598 (361,839)	223 (265,952)
	Mix_2	136,799	97%	1,258 (127,543)	380 (126,467)
			100%	345 (109,827)	156 (106,078)

Table 4.11: **Number of OTUs (and associated total number of sequences) that had a BLAST hit to one or more of the 19 bacterial strains with hit threshold $\geq$ 97% or 100%.** *vanilla*: without error correction. *errcor*: with error correction. Since many of the 19 bacterial strains were highly similar, we could not map the assembled sequences accurately back to them. Instead we evaluated our method based on (a) how many OTUs were 97% or 100% similar to one or more of the 19 sequences, and (b) how many sequences were included in those OTUs. Error correction resulted in fewer assembled sequences but also fewer OTUs. The two samples were run with different barcode-primer pairs. Mix_1: CAG-S_1V6, Mix_2: ATT-S_2V6. Phred score cutoff: 0 and 25.

4.8 Discussion

In this chapter, we reviewed the sources of sequencing bias and errors. Using standard additions in human gut microbial samples, we were able to detect species at as low as 0.01%. We found that the GC-rich *D. radiodurans* was less efficiently amplified than the low GC *M. pneumoniae*, which is likely caused by sequencing bias due to base composition difference. We found sequencing errors to be non-uniform and associated with sequencing cycle, Phred score, and preceding bases. As a result, we developed an error correction pipeline called ErrCor to identify and reduce errors. We show that ErrCor reduced base errors and decreased the number of OTUs towards the ground truth.

Having an accurate OTU number (and of course, the correct representative sequences) impacts the downstream analyses of microbial communities. Without error correction, we observed $>$ 3000 MP clusters when we knew there was exactly 1 sequence! This grossly overestimated the amount of species diversity. With error correction, we observed 100 MP clusters—still not close to 1, but at least 10 times closer. In addition, error corrected OTUs present a smaller input set to downstream analyses, and this translates to shorter runtime. We did not find base errors to affect taxonomic identification: In our dataset, RDP Classifier correctly classified all assembled MP/DR reads (180-220 bp) without error correction.

In many ways, eliminating base errors is already implemented throughout our processing pipeline: the initial barcode binning, primer matching and using Phred score cutoffs all contribute to removing errors from the dataset. However these methods remove error by discarding reads when parts of the read could still be informative. Trimming low quality read ends, a commonly used method in pyrosequencing, is not applicable to Illumina reads that are much shorter. Assembling read pairs through overlap, as we presented here, can only eliminate errors on the overlapped region. With ErrCor, we reduce base errors while keeping as many usable reads as possible.

One caveat with our classification approach is that we need training data that comes from the same sequencing run. With different versions of sequencing machines

and chemistry used, it is unlikely that the error models can be used across different runs. Furthermore, the features associated with building the error model are not easily identified. With better feature selection we might improve classification accuracy, but we would also need more data and it is not always feasible to do standard additions. For future work, we plan to examine whether training data could be "inferred" from datasets: If we have knowledge of what is in the sample, we can identify base errors if the read matches a reference sequence with some mismatches. Another area of potential improvement is better use of the super reads. In ErrCor we compressed Phred score information in super reads and used a simple ratio test to add confidence to our classification outcome. In some cases, more errors were introduced due to misclassification. A probabilistic model that considers the likelihood of two super reads originating from the same sequence given the Phred scores, super read sizes and the error model will likely reduce misclassification even further.

BIBLIOGRAPHY

[1] J. Abello, P. M. Pardalos, and M. G.C Resende. On maximum clique problems in very large graphs. *AT&T labs Reserrch Technical Report: TR98*, 32, 1998.

[2] S. Ahmed, G. T. Macfarlane, A. Fite, A. J. McBain, P. Gilbert, and S. Macfarlane. Mucosa-associated bacterial diversity in relation to human terminal ileum and colonic biopsy samples. *Applied and Environmental Microbiology*, 73:7435–7442, September 2007.

[3] D. Aird, M.G. Ross, W.S. Chen, M. Danielsson, T. Fennell, C. Russ, D.B. Jaffe, C. Nusbaum, and A. Gnirke. Analyzing and minimizing PCR amplification bias in illumina sequencing libraries. *Genome Biol*, 12(2):R18, 2011.

[4] J. Alexandros Stamatakis. *(RAxML) Distributed and Parallel Algorithms and Systems for Inference of Huge Phylogenetic Trees based on the Maximum Likelihood Method*. Duv, 2006.

[5] Charlotte Atkinson, Katherine M Newton, Erin J Aiello Bowles, Mellissa Yong, and Johanna W Lampe. Demographic, anthropometric, and lifestyle factors and dietary intakes in relation to daidzein-metabolizing phenotypes among premenopausal women in the united states. *The American Journal of Clinical Nutrition*, 87(3):679–687, March 2008. PMID: 18326607.

[6] J. E. Barrick. New RNA motifs suggest an expanded scope for riboswitches in bacterial genetic control. *Proceedings of the National Academy of Sciences*, 101(17):6421–6426, April 2004.

[7] Jeffrey E Barrick, Narasimhan Sudarsan, Zasha Weinberg, Walter L Ruzzo, and Ronald R Breaker. 6S RNA is a widespread regulator of eubacterial RNA polymerase that resembles an open promoter. *RNA (New York, N.Y.)*, 11(5):774–784, May 2005. PMID: 15811922.

[8] K. F. Block, M. C. Hammond, and R. R. Breaker. Evidence for widespread gene control function by the ydaO riboswitch candidate. *Journal of Bacteriology*, 192(15):3983–3989, May 2010.

[9] Ronald R Breaker. Prospects for riboswitch discovery and analysis. *Molecular cell*, 43(6):867–879, September 2011. PMID: 21925376.

[10] B Brenig, J Beck, and E Schütz. Shotgun metagenomics of biological stains using ultra-deep DNA sequencing. *Forensic Science International. Genetics*, 4(4):228–231, July 2010. PMID: 20457050.

[11] Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L Madden. BLAST+: architecture and applications. *BMC Bioinformatics*, 10:421, 2009. PMID: 20003500.

[12] Jamie J Cannone, Sankar Subramanian, Murray N Schnare, James R Collett, Lisa M D'Souza, Yushi Du, Brian Feng, Nan Lin, Lakshmi V Madabusi, Kirsten M Müller, Nupur Pande, Zhidi Shang, Nan Yu, and Robin R Gutell. The comparative RNA web (CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BMC Bioinformatics*, 3:2, 2002. PMID: 11869452.

[13] J Gregory Caporaso, Justin Kuczynski, Jesse Stombaugh, Kyle Bittinger, Frederic D Bushman, Elizabeth K Costello, Noah Fierer, Antonio Gonzalez Peña, Julia K Goodrich, Jeffrey I Gordon, Gavin A Huttley, Scott T Kelley, Dan Knights, Jeremy E Koenig, Ruth E Ley, Catherine A Lozupone, Daniel McDonald, Brian D Muegge, Meg Pirrung, Jens Reeder, Joel R Sevinsky, Peter J Turnbaugh, William A Walters, Jeremy Widmann, Tanya Yatsunenko, Jesse Zaneveld, and Rob Knight. QIIME allows analysis of high-throughput community sequencing data. *Nature Methods*, 7(5):335–336, April 2010.

[14] J Gregory Caporaso, Christian L Lauber, William A Walters, Donna Berg-Lyons, Catherine A Lozupone, Peter J Turnbaugh, Noah Fierer, and Rob Knight. Global patterns of 16S rRNA diversity at a depth of millions of sequences per sample. *Proceedings of the National Academy of Sciences of the United States of America*, 108 Suppl 1:4516–4522, March 2011. PMID: 20534432.

[15] T. R. Cech. The RNA worlds in context. *Cold Spring Harbor Perspectives in Biology*, February 2011.

[16] J. E. Clarridge. Impact of 16S rRNA gene sequence analysis for identification of bacteria on clinical microbiology and infectious diseases. *Clinical Microbiology Reviews*, 17(4):840–862, October 2004.

[17] J. R. Cole, Q. Wang, E. Cardenas, J. Fish, B. Chai, R. J. Farris, A. S. Kulam-Syed-Mohideen, D. M. McGarrell, T. Marsh, G. M. Garrity, and J. M. Tiedje. The ribosomal database project: improved alignments and new tools for rRNA analysis. *Nucleic Acids Research*, 37(Database):D141–D145, January 2009.

[18] Michael D Dambach and Wade C Winkler. Expanding roles for metabolite-sensing regulatory RNAs. *Current Opinion in Microbiology*, 12(2):161–169, April 2009. PMID: 19250859.

[19] Patrick H Degnan and Howard Ochman. Illumina-based analysis of microbial community diversity. *The ISME Journal*, 6(1):183–194, January 2012. PMID: 21677692.

[20] Richard Durbin. *Biological sequence analysis probalistic models of proteins and nucleic acids*. Cambridge University Press, Cambridge, UK; New York, 1998.

[21] Sean R. Eddy. NON-CODING RNA GENES AND THE MODERN RNA WORLD. *Nature Reviews Genetics*, 2(12):919–929, December 2001.

[22] Manel Esteller. Non-coding RNAs in human disease. *Nature reviews. Genetics*, 12(12):861–874, December 2011. PMID: 22094949.

[23] J Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution*, 17(6):368–376, 1981. PMID: 7288891.

[24] J Felsenstein. PHYLIP (Phylogeny inference package) version 3.6. *Distributed by the author. Department of Genome Sciences, University of Washington, Seattle.*, 2006.

[25] J Felsenstein and G A Churchill. A hidden markov model approach to variation among sites in rate of evolution. *Molecular Biology and Evolution*, 13(1):93–104, January 1996. PMID: 8583911.

[26] Daniel N Frank, Allison L St Amand, Robert A Feldman, Edgar C Boedeker, Noam Harpaz, and Norman R Pace. Molecular-phylogenetic characterization of microbial community imbalances in human inflammatory bowel diseases. *Proceedings of the National Academy of Sciences of the United States of America*, 104(34):13780–13785, August 2007. PMID: 17699621.

[27] Eva K Freyhult, Jonathan P Bollback, and Paul P Gardner. Exploring genomic dark matter: a critical assessment of the performance of homology search methods on noncoding RNA. *Genome research*, 17(1):117–125, January 2007. PMID: 17151342.

[28] GM Garrity, TG Lilburn, JR Cole, SH Harrison, Jean Euzeby, and BJ Tinball. The taxonomic outline of bacteria and archaea. http://www.taxonomicoutline.org/index.php/toba/about, 2007.

[29] Melinda S. Gerdeman, Tina M. Henkin, and Jennifer V. Hines. Solution structure of the bacillus subtilis t-box antiterminator RNA: seven nucleotide bulge characterized by stacking and flexibility. *Journal of Molecular Biology*, 326(1):189–201, February 2003.

[30] W Gish. WU-BLAST, 2009.

[31] Julia B Greer and Stephen John O'Keefe. Microbial induction of immunity, inflammation, and cancer. *Frontiers in Physiology*, 1:168, 2011. PMID: 21423403.

[32] Tony Gutschner and Sven Diederichs. The hallmarks of cancer: A long non-coding RNA point of view. *RNA biology*, 9(6), June 2012. PMID: 22664915.

[33] I. Holmes. Accelerated probabilistic inference of RNA structure evolution. *BMC Bioinformatics*, 6(1):73, 2005.

[34] S. M. Huse, J. A. Huber, H. G. Morrison, M. L. Sogin, D. M. Welch, et al. Accuracy and quality of massively parallel DNA pyrosequencing. *Genome Biol*, 8(7):R143, 2007.

[35] Dennis Maletich Junqueira, Rúbia Marília de Medeiros, Maria Cristina Cotta Matte, Leonardo Augusto Luvison Araújo, Jose Artur Bogo Chies, Patricia Ashton-Prolla, and Sabrina Esteves de Matos Almeida. Reviewing the history of HIV-1: spread of subtype b in the americas. *PloS one*, 6(11):e27489, 2011. PMID: 22132104.

[36] Takahiro Kanagawa. Bias and artifacts in multitemplate polymerase chain reactions (PCR). *Journal of Bioscience and Bioengineering*, 96(4):317–323, 2003. PMID: 16233530.

[37] C. Kanz. The EMBL nucleotide sequence database. *Nucleic Acids Research*, 33(Database issue):D29–D33, December 2004.

[38] Ilene Karsch-Mizrachi, Yasukazu Nakamura, and Guy Cochrane. The international nucleotide sequence database collaboration. *Nucleic Acids Research*, 40(Database issue):D33–37, January 2012. PMID: 22080546.

[39] Martin Kircher, Udo Stenzel, and Janet Kelso. Improved base calling for the illumina genome analyzer using machine learning strategies. *Genome biology*, 10(8):R83, 2009. PMID: 19682367.

[40] B Knudsen and J Hein. (Pfold 1999) RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics (Oxford, England)*, 15(6):446–454, June 1999. PMID: 10383470.

[41] Bjarne Knudsen and Jotun Hein. (Pfold 2003) pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Research*, 31(13):3423–3428, July 2003. PMID: 12824339.

[42] Omry Koren, Aymé Spor, Jenny Felin, Frida Fåk, Jesse Stombaugh, Valentina Tremaroli, Carl Johan Behre, Rob Knight, Björn Fagerberg, Ruth E Ley, and Fredrik Bäckhed. Microbes and health sackler colloquium: Human oral, gut, and plaque microbiota in patients with atherosclerosis. *Proceedings of the National Academy of Sciences of the United States of America*, October 2010. PMID: 20937873.

[43] M K Kuhner and J Felsenstein. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Molecular Biology and Evolution*, 11(3):459–468, May 1994. PMID: 8015439.

[44] Johanna W Lampe. The human microbiome project: getting to the guts of the matter in cancer epidemiology. *Cancer Epidemiology, Biomarkers & Prevention: A Publication of the American Association for Cancer Research, Cosponsored by the American Society of Preventive Oncology*, 17(10):2523–2524, October 2008. PMID: 18842991.

[45] C Lanave, G Preparata, C Saccone, and G Serio. A new method for calculating evolutionary substitution rates. *Journal of Molecular Evolution*, 20(1):86–93, 1984. PMID: 6429346.

[46] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3):R25, 2009. PMID: 19261174.

[47] Nadja Larsen, Finn K Vogensen, Frans W J van den Berg, Dennis Sandris Nielsen, Anne Sofie Andreasen, Bente K Pedersen, Waleed Abu Al-Soud, Søren J Sørensen, Lars H Hansen, and Mogens Jakobsen. Gut microbiota in human adults with type 2 diabetes differs from non-diabetic adults. *PloS One*, 5(2):e9085, 2010. PMID: 20140211.

[48] Fei Li, Meredith A J Hullar, and Johanna W Lampe. Optimization of terminal restriction fragment polymorphism (TRFLP) analysis of human gut microbiota. *Journal of Microbiological Methods*, 68(2):303–311, February 2007. PMID: 17069911.

[49] W T Liu, T L Marsh, H Cheng, and L J Forney. Characterization of microbial diversity by determining terminal restriction fragment length polymorphisms of genes encoding 16S rRNA. *Applied and Environmental Microbiology*, 63(11):4516–4522, November 1997. PMID: 9361437.

[50] Z. Liu, C. Lozupone, M. Hamady, F. D. Bushman, and R. Knight. Short pyrosequencing reads suffice for accurate microbial community analysis. *Nucleic Acids Research*, 35(18):e120–e120, August 2007.

[51] Y. Loewenstein, E. Portugaly, M. Fromer, and M. Linial. Efficient algorithms for accurate hierarchical clustering of huge datasets: tackling the entire protein space. *Bioinformatics*, 24(13):i41–i49, June 2008.

[52] Catherine Lozupone, Manuel E Lladser, Dan Knights, Jesse Stombaugh, and Rob Knight. UniFrac: an effective distance metric for microbial community comparison. *The ISME Journal*, 5(2):169–172, February 2011. PMID: 20827291.

[53] W. Ludwig. ARB: a software environment for sequence data. *Nucleic Acids Research*, 32(4):1363–1371, February 2004.

[54] Daniel MacLean, Jonathan D. G. Jones, and David J. Studholme. Application of 'next-generation' sequencing technologies to microbial genetics. *Nature Reviews Microbiology*, February 2009.

[55] D.R. Maddison, K. S. Schulz, and W. P. Maddison. Tree of life web project. http://www.tolweb.org/tree/, 2012.

[56] M. Mandal. A glycine-dependent riboswitch that uses cooperative binding to control gene expression. *Science*, 306(5694):275–279, October 2004.

[57] Maumita Mandal, Benjamin Boese, Jeffrey E Barrick, Wade C Winkler, and Ronald R Breaker. Riboswitches control fundamental biochemical pathways in bacillus subtilis and other bacteria. *Cell*, 113(5):577–586, May 2003.

[58] Phillip J McCown, Wade C Winkler, and Ronald R Breaker. Mechanism and distribution of glmS ribozymes. *Methods in molecular biology (Clifton, N.J.)*, 848:113–129, 2012. PMID: 22315066.

[59] Daniel McDonald. Greengenes taxonomy 2011 | second genome. http://www.secondgenome.com/go/2011-greengenes-taxonomy/, 2011.

[60] Daniel McDonald, Morgan N Price, Julia Goodrich, Eric P Nawrocki, Todd Z DeSantis, Alexander Probst, Gary L Andersen, Rob Knight, and Philip Hugenholtz. An improved greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. *The ISME journal*, 6(3):610–618, March 2012. PMID: 22134646.

[61] F. Meacham, D. Boffelli, J. Dhahbi, D. Martin, M. Singer, and L. Pachter. Identification and correction of systematic error in high-throughput sequence data. *BMC Bioinformatics*, 12(1):451, 2011.

[62] M. L. Metzker. Sequencing technologies — the next generation. *Nature Reviews Genetics*, 11(1):31–46, 2009.

[63] Irmtraud M Meyer and István Miklós. SimulFold: simultaneously inferring RNA structures including pseudoknots, alignments, and trees using a bayesian MCMC framework. *PLoS Computational Biology*, 3(8):e149, August 2007. PMID: 17696604.

[64] Michelle M Meyer, Tyler D Ames, Daniel P Smith, Zasha Weinberg, Michael S Schwalbach, Stephen J Giovannoni, and Ronald R Breaker. Identification of candidate structured RNAs in the marine organism 'Candidatus pelagibacter ubique'. *BMC Genomics*, 10:268, 2009. PMID: 19531245.

[65] Michelle M Meyer, Ming C Hammond, Yasmmyn Salinas, Adam Roth, Narasimhan Sudarsan, and Ronald R Breaker. Challenges of ligand identification for riboswitch candidates. *RNA biology*, 8(1):5–10, February 2011. PMID: 21317561.

[66] C. S. Miller, B. J. Baker, B. C. Thomas, S. W. Singer, and J. F. Banfield. EMIRGE: reconstruction of full-length ribosomal genes from microbial community short read sequencing data. *Genome biology*, 12(5):R44, 2011.

[67] Rebecca K Montange and Robert T Batey. Structure of the s-adenosylmethionine riboswitch regulatory mRNA element. *Nature*, 441(7097):1172–1175, June 2006. PMID: 16810258.

[68] Steve Mount. blastn parameters for noncoding queries. http://stevemount.outfoxing.com/Posting0004.html, 2005.

[69] Ali Nahvi, Jeffrey E Barrick, and Ronald R Breaker. Coenzyme b12 riboswitches are widespread genetic control elements in prokaryotes. *Nucleic Acids Research*, 32(1):143–150, 2004. PMID: 14704351.

[70] K. Nakamura, T. Oshima, T. Morimoto, S. Ikeda, H. Yoshikawa, Y. Shiwa, S. Ishikawa, M. C. Linak, A. Hirai, H. Takahashi, M. Altaf-Ul-Amin, N. Ogasawara, and S. Kanaya. Sequence-specific error profile of illumina sequencers. *Nucleic Acids Research*, 39(13):e90–e90, May 2011.

[71] Eric P Nawrocki, Diana L Kolbe, and Sean R Eddy. Infernal 1.0: inference of RNA alignments. *Bioinformatics (Oxford, England)*, 25(10):1335–1337, May 2009. PMID: 19307242.

[72] S.O. Oyola, T.D. Otto, Y. Gu, G. Maslen, M. Manske, S. Campino, D.J. Turner, B. Maclnnis, D.P. Kwiatkowski, H.P. Swerdlow, et al. Optimizing illumina next-generation sequencing library preparation for extremely AT-biased genomes. *BMC genomics*, 13(1):1, 2012.

[73] Jakob Skou Pedersen, Gill Bejerano, Adam Siepel, Kate Rosenbloom, Kerstin Lindblad-Toh, Eric S. Lander, Jim Kent, Webb Miller, and David Haussler. Identification and classification of conserved RNA secondary structures in the human genome. *PLoS Computational Biology*, 2(4):e33, 2006.

[74] Morgan N Price, Paramvir S Dehal, and Adam P Arkin. FastTree 2–approximately maximum-likelihood trees for large alignments. *PloS one*, 5(3):e9490, 2010. PMID: 20224823.

[75] Elmar Pruesse, Christian Quast, Katrin Knittel, Bernhard M. Fuchs, Wolfgang Ludwig, Jorg Peplies, and Frank Oliver Glockner. SILVA: a comprehensive online resource for quality checked and aligned ribosomal RNA sequence data compatible with ARB. *Nucleic Acids Research*, 35(21):7188–7196, December 2007.

[76] E. Puerta-Fernandez, J. E. Barrick, A. Roth, and R. R. Breaker. Identification of a large noncoding RNA in extremophilic eubacteria. *Proceedings of the National Academy of Sciences*, 103(51):19490–19495, December 2006.

[77] Junjie Qin, Ruiqiang Li, Jeroen Raes, Manimozhiyan Arumugam, Kristoffer Solvsten Burgdorf, Chaysavanh Manichanh, Trine Nielsen, Nicolas Pons, Florence Levenez, Takuji Yamada, Daniel R Mende, Junhua Li, Junming Xu, Shaochuan Li, Dongfang Li, Jianjun Cao, Bo Wang, Huiqing Liang, Huisong Zheng, Yinlong Xie, Julien Tap, Patricia Lepage, Marcelo Bertalan, Jean-Michel Batto, Torben Hansen, Denis Le Paslier, Allan Linneberg, H Bjørn Nielsen, Eric Pelletier, Pierre Renault, Thomas Sicheritz-Ponten, Keith Turner, Hongmei Zhu, Chang Yu, Shengting Li, Min Jian, Yan Zhou, Yingrui Li, Xiuqing Zhang, Songgang Li, Nan Qin, Huanming Yang, Jian Wang, Søren Brunak, Joel Doré, Francisco Guarner, Karsten Kristiansen, Oluf Pedersen, Julian Parkhill, Jean Weissenbach, Peer Bork, S Dusko Ehrlich, and Jun Wang. A human gut microbial gene catalogue established by metagenomic sequencing. *Nature*, 464(7285):59–65, March 2010. PMID: 20203603.

[78] M. Rattray. Phase: a software package for PHylogenetics and sequence evolution. http://www.bioinf.man.ac.uk/resources/phase/, 2007.

[79] Elena Rivas and Sean R Eddy. (DNAML-e) probabilistic phylogenetic inference with insertions and deletions. *PLoS Computational Biology*, 4(9):e1000172, 2008. PMID: 18787703.

[80] Victoria L. Robinson. Rethinking the central dogma: Noncoding RNAs are biologically relevant. *Urologic Oncology: Seminars and Original Investigations*, 27(3):304–306, May 2009.

[81] Sébastien Rodrigue, Arne C. Materna, Sonia C. Timberlake, Matthew C. Blackburn, Rex R. Malmstrom, Eric J. Alm, and Sallie W. Chisholm. Unlocking short read sequencing for metagenomics. *PLoS ONE*, 5(7):e11840, July 2010.

[82] Adam Roth, Wade C Winkler, Elizabeth E Regulski, Bobby W K Lee, Jinsoo Lim, Inbal Jona, Jeffrey E Barrick, Ankita Ritwik, Jane N Kim, RÃijdiger Welz, Dirk Iwata-Reuyl, and Ronald R Breaker. A riboswitch selective for the queuosine precursor preQ1 contains an unusually small aptamer domain. *Nature Structural & Molecular Biology*, 14(4):308–317, March 2007.

[83] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, 4(4):406–425, 1987.

[84] Eric W Sayers, Tanya Barrett, Dennis A Benson, Stephen H Bryant, Kathi Canese, Vyacheslav Chetvernin, Deanna M Church, Michael DiCuccio, Ron Edgar, Scott Federhen, Michael Feolo, Lewis Y Geer, Wolfgang Helmberg, Yuri Kapustin, David Landsman, David J Lipman, Thomas L Madden, Donna R Maglott, Vadim Miller, Ilene Mizrachi, James Ostell, Kim D Pruitt, Gregory D Schuler, Edwin Sequeira, Stephen T Sherry, Martin Shumway, Karl Sirotkin, Alexandre Souvorov, Grigory Starchenko, Tatiana A Tatusova, Lukas Wagner, Eugene Yaschenko, and Jian Ye. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 37(Database issue):D5–15, January 2009. PMID: 18940862.

[85] Patrick D Schloss, Sarah L Westcott, Thomas Ryabin, Justine R Hall, Martin Hartmann, Emily B Hollister, Ryan A Lesniewski, Brian B Oakley, Donovan H Parks, Courtney J Robinson, Jason W Sahl, Blaz Stres, Gerhard G Thallinger, David J Van Horn, and Carolyn F Weber. Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and Environmental Microbiology*, 75(23):7537–7541, December 2009. PMID: 19801464.

[86] S. E. Seemann, J. Gorodkin, and R. Backofen. Unifying evolutionary and thermodynamic information for RNA folding of multiple alignments. *Nucleic Acids Research*, 36(20):6355–6362, September 2008.

[87] Edward H Simpson. Measurement of diversity. *Nature*, 163:688, 1949.

[88] A. Stamatakis. Phylogenetic models of rate heterogeneity: A high performance computing perspective. In *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, pages 8–pp, 2006.

[89] Alexandros Stamatakis. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics (Oxford, England)*, 22(21):2688–2690, November 2006. PMID: 16928733.

[90] N. Sudarsan. An mRNA structure in bacteria that controls gene expression by binding lysine. *Genes & Development*, 17(21):2688–2697, October 2003.

[91] Narasimhan Sudarsan, Jeffrey E Barrick, and Ronald R Breaker. Metabolite-binding RNA domains are present in the genes of eukaryotes. *RNA (New York, N.Y.)*, 9(6):644–647, June 2003. PMID: 12756322.

[92] Jeet Sukumaran and Mark T Holder. DendroPy: a python library for phylogenetic computing. *Bioinformatics (Oxford, England)*, 26(12):1569–1571, June 2010. PMID: 20421198.

[93] Andreas Sundquist, Saharnaz Bigdeli, Roxana Jalili, Maurice L Druzin, Sarah Waller, Kristin M Pullen, Yasser Y El-Sayed, M Mark Taslimi, Serafim Batzoglou, and Mostafa Ronaghi. Bacterial flora-typing with targeted, chip-based pyrosequencing. *BMC Microbiology*, 7(1):108, 2007.

[94] Kenneth R. Tindall and Thomas A. Kunkel. Fidelity of DNA synthesis by the thermus aquaticus DNA polymerase. *Biochemistry*, 27(16):6008–6013, August 1988.

[95] Harold Tjalsma, Annemarie Boleij, Julian R. Marchesi, and Bas E. Dutilh. A bacterial driver–passenger model for colorectal cancer: beyond the usual suspects. *Nature Reviews Microbiology*, 10(8):575–582, June 2012.

[96] Elfar Torarinsson, Zizhen Yao, Eric D Wiklund, Jesper B Bramsen, Claus Hansen, Jørgen Kjems, Niels Tommerup, Walter L Ruzzo, and Jan Gorodkin. Comparative genomics beyond sequence-based alignments: RNA structures in the ENCODE regions. *Genome Research*, 18(2):242–251, February 2008. PMID: 18096747.

[97] Huei-Hun Tseng, Zasha Weinberg, Jeremy Gore, Ronald R Breaker, and Walter L Ruzzo. Finding non-coding RNAs through genome-scale clustering. *Journal of Bioinformatics and Computational Biology*, 7(2):373–388, April 2009. PMID: 19340921.

[98] Peter J Turnbaugh, Micah Hamady, Tanya Yatsunenko, Brandi L Cantarel, Alexis Duncan, Ruth E Ley, Mitchell L Sogin, William J Jones, Bruce A Roe, Jason P Affourtit, Michael Egholm, Bernard Henrissat, Andrew C Heath, Rob Knight, and Jeffrey I Gordon. A core gut microbiome in obese and lean twins. *Nature*, 457(7228):480–484, January 2009. PMID: 19043404.

[99] Adrienne X Wang, Walter L Ruzzo, and Martin Tompa. How accurately is ncRNA aligned within whole-genome multiple alignments? *BMC Bioinformatics*, 8:417, 2007. PMID: 17963514.

[100] Qiong Wang, George M Garrity, James M Tiedje, and James R Cole. Naive bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Applied and Environmental Microbiology*, 73(16):5261–5267, August 2007. PMID: 17586664.

[101] Kaifa Wei, Yanfeng Chen, Juan Chen, Lingjuan Wu, and Daoxin Xie. Evolution and adaptation of hemagglutinin gene of human H5N1 influenza virus. *Virus Genes*, 44(3):450–458, January 2012.

[102] Zasha Weinberg, Jeffrey E Barrick, Zizhen Yao, Adam Roth, Jane N Kim, Jeremy Gore, Joy Xin Wang, Elaine R Lee, Kirsten F Block, Narasimhan Sudarsan, Shane Neph, Martin Tompa, Walter L Ruzzo, and Ronald R Breaker. Identification of 22 candidate structured RNAs in bacteria using the CMfinder comparative genomics pipeline. *Nucleic Acids Research*, 35(14):4809–4819, 2007. PMID: 17621584.

[103] Zasha Weinberg, Jonathan Perreault, Michelle M Meyer, and Ronald R Breaker. Exceptional structured noncoding RNAs revealed by bacterial metagenome analysis. *Nature*, 462(7273):656–659, December 2009. PMID: 19956260.

[104] Zasha Weinberg, Joy X Wang, Jarrod Bogue, Jingying Yang, Keith Corbino, Ryan H Moy, and Ronald R Breaker. Comparative genomics reveals 104 candidate structured RNAs from bacteria, archaea, and their metagenomes. *Genome Biology*, 11(3):R31, 2010. PMID: 20230605.

[105] J. J. Werner, O. Koren, P. Hugenholtz, T. Z. DeSantis, W. A. Walters, J. G. Caporaso, L. T. Angenent, R. Knight, and R. E. Ley. Impact of training sets on classification of high-throughput bacterial 16s rRNA gene surveys. *The ISME Journal*, 6(1):94–103, 2011.

[106] William B Whitman. Bergey's manual trust. http://www.bergeys.org/, 2012.

[107] Wade Winkler, Ali Nahvi, and Ronald R. Breaker. Thiamine derivatives bind messenger RNAs directly to regulate bacterial gene expression. *Nature*, 419(6910):952–956, October 2002.

[108] Wade C Winkler, Smadar Cohen-Chalamish, and Ronald R Breaker. An mRNA structure that controls gene expression by binding FMN. *Proceedings of the National Academy of Sciences of the United States of America*, 99(25):15908–15913, December 2002. PMID: 12456892.

[109] C Workman and A Krogh. No evidence that mRNAs have lower folding free energies than random sequences with the same dinucleotide distribution. *Nucleic Acids Research*, 27(24):4816–4822, December 1999. PMID: 10572183.

[110] Zizhen Yao. *Genome scale search of noncoding RNAs: Bacteria to Vertebrates*. PhD thesis, University of Washington, Department of Computer Science & Engineering, 2008.

[111] Zizhen Yao, Jeffrey Barrick, Zasha Weinberg, Shane Neph, Ronald Breaker, Martin Tompa, and Walter L Ruzzo. A computational pipeline for high-throughput discovery of cis-regulatory noncoding RNA in prokaryotes. *PLoS Computational Biology*, 3(7):e126, July 2007. PMID: 17616982.

[112] Zizhen Yao, Zasha Weinberg, and Walter L Ruzzo. CMfinder–a covariance model based RNA motif finding algorithm. *Bioinformatics (Oxford, England)*, 22(4):445–452, February 2006. PMID: 16357030.

[113] Pablo Yarza, Michael Richter, Jörg Peplies, Jean Euzeby, Rudolf Amann, Karl-Heinz Schleifer, Wolfgang Ludwig, Frank Oliver Glöckner, and Ramon Rosselló-Móra. The all-species living tree project: A 16S rRNA-based phylogenetic tree of all sequenced type strains. *Systematic and Applied Microbiology*, 31:241–250, September 2008.

[114] Osvaldo Zagordi, Lukas Geyrhofer, Volker Roth, and Niko Beerenwinkel. Deep sequencing of a genetically heterogeneous sample: local haplotype reconstruction and read error correction. *Journal of computational biology: a journal of computational molecular cell biology*, 17(3):417–428, March 2010. PMID: 20377454.

[115] Osvaldo Zagordi, Rolf Klein, Martin Däumer, and Niko Beerenwinkel. Error correction of next-generation sequencing data and reliable estimation of HIV quasispecies. *Nucleic Acids Research*, 38(21):7400–7409, November 2010. PMID: 20671025.

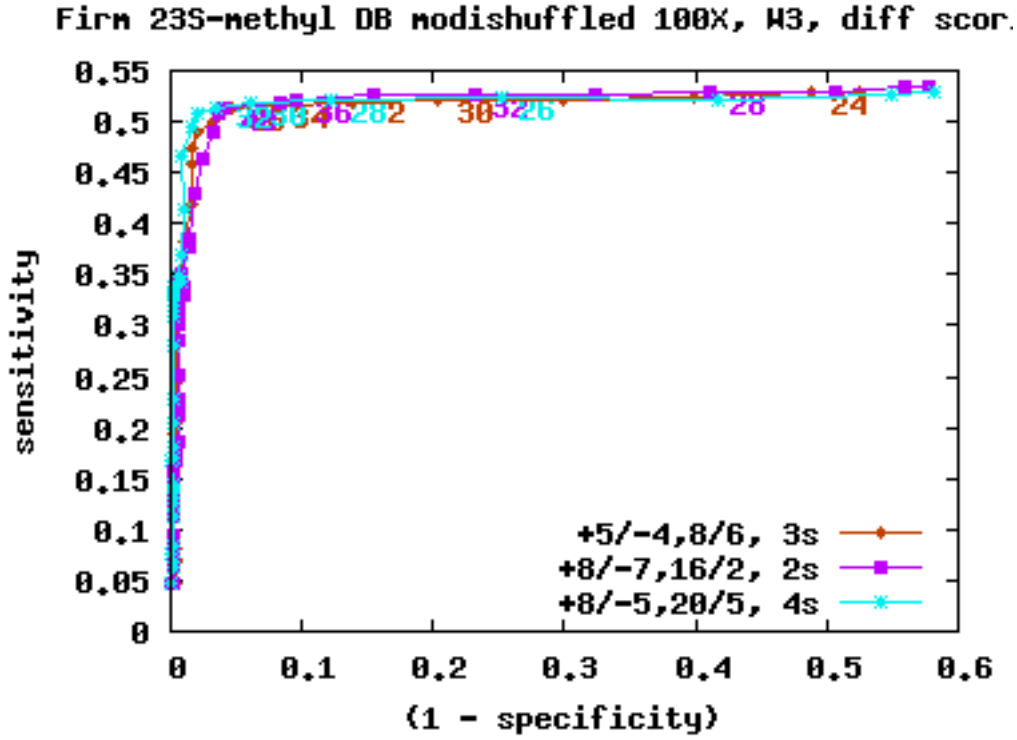
[116] Nikolay Zenkin. Hypothesis: Emergence of translation as a result of RNA helicase evolution. *Journal of Molecular Evolution*, April 2012.

[117] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B: fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560, December 1997.

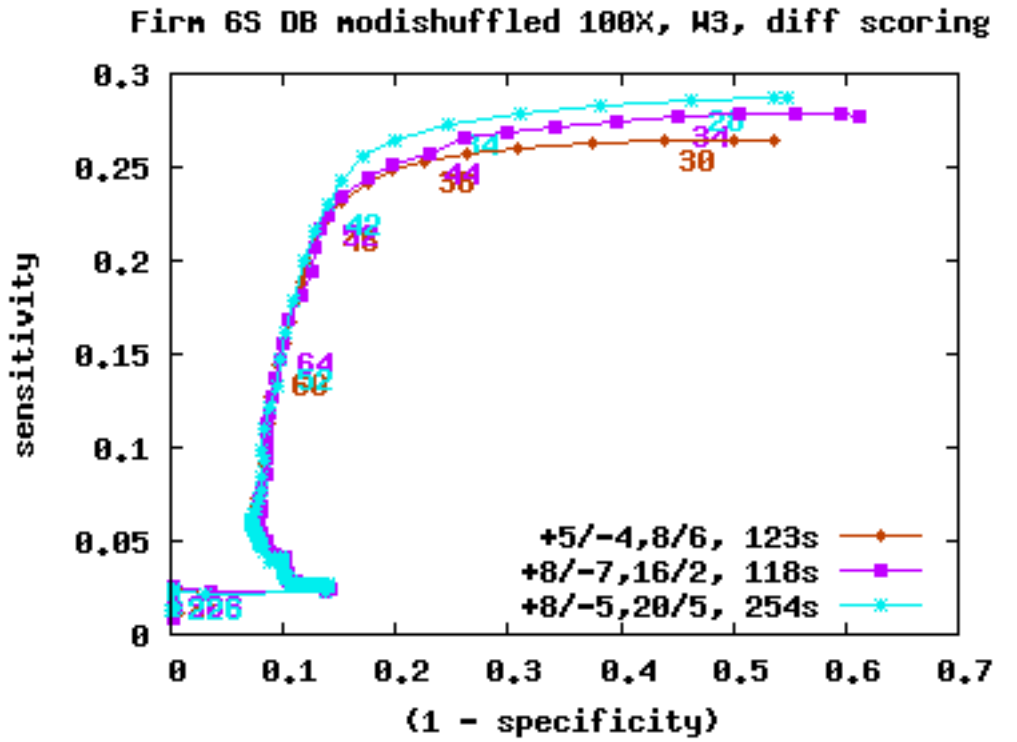
Appendix A

CHOOSING BLAST PARAMETERS FOR NCRNA DISCOVERY

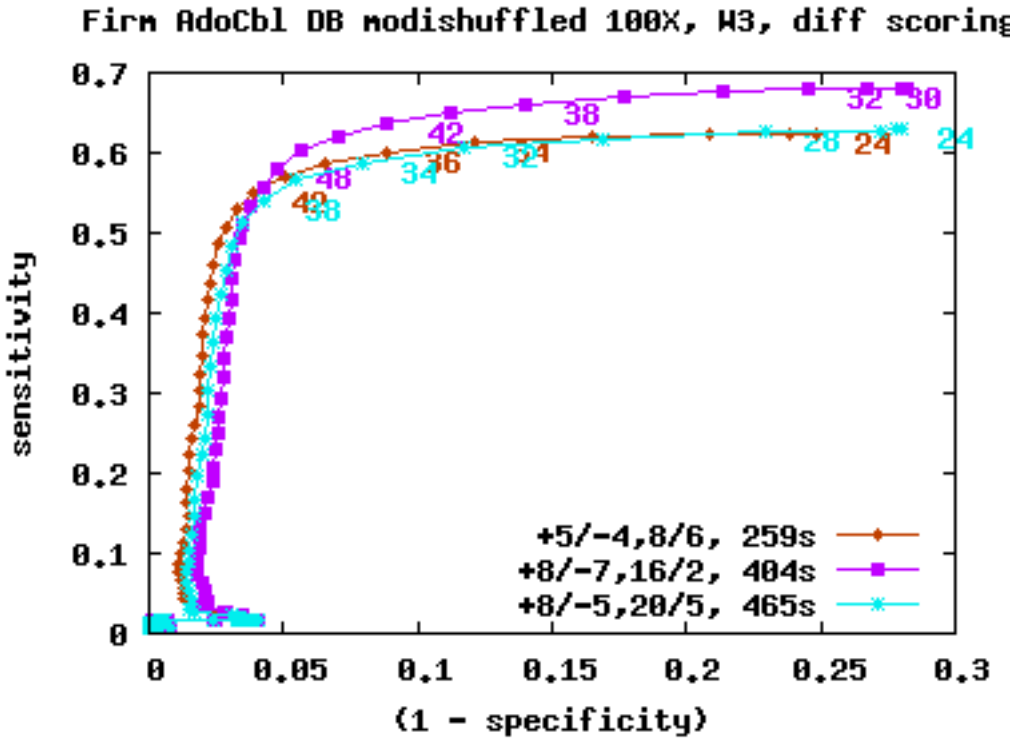
To choose the best BLAST parameters for finding homologous ncRNA sequences, we tested three different BLAST parameter sets (suggested in [68]) against different ncRNA families. For each ncRNA family, we created a database consisting of all Firmicutes sequences from the family and randomly shuffled them 100 times while preserving dinucleotide frequencies; it is common in ncRNA methods to use di-shuffled rather than mono-shuffled sequences to avoid creating artificially dissimilar sequences [27, 109]. At different BLAST score cutoffs, we calculated the tradeoff between sensitivity ($\frac{TP}{TP+FN}$) and specificity ($\frac{TP}{TP+FP}$) (Figure A.1). At higher BLAST score cutoffs, we recovered fewer ncRNAs (lower sensitivity) but reduced false hits (higher specificity). For some families (ex: 23S-methyl, GEMM), different BLAST scores yielded the same RoC curve. For others (ex: AdoCbl, glycine, ykkC), the parameter set {+8/-7, -16/-2} had higher sensitivity. We used WU-BLAST instead of NCBI-BLAST because WU-BLAST allows variable match/mismatch and gap open/extend scoring schemes, whereas NCBI-BLAST only provides a fixed set of parameter sets. For seed length, we chose the smallest number allowed in WU-BLAST (-W 3). For E-value cutoff, we chose a relaxed cutoff (-E 2) to allow for variable score cutoff in homology graph construction.



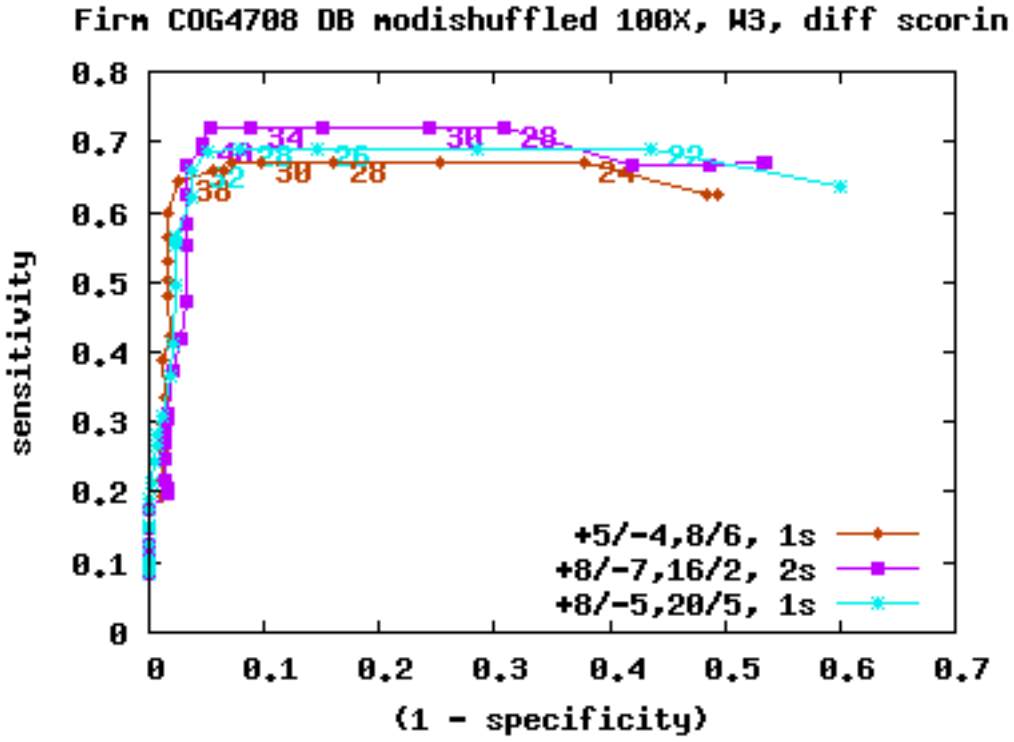
(a) 23S-methyl



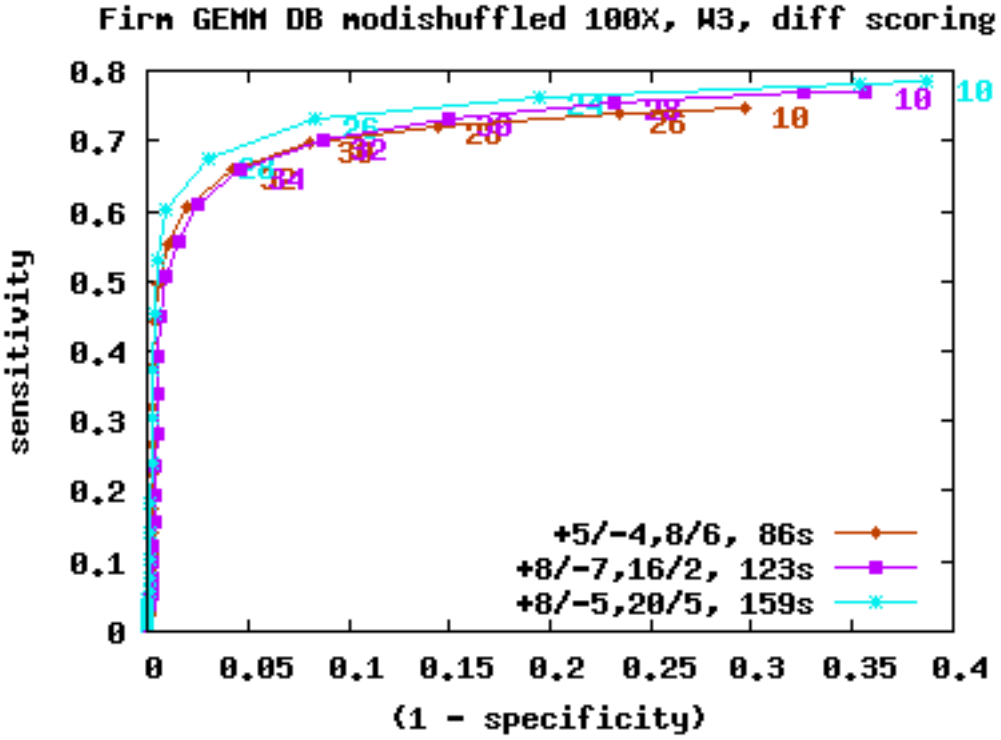
(b) 6S



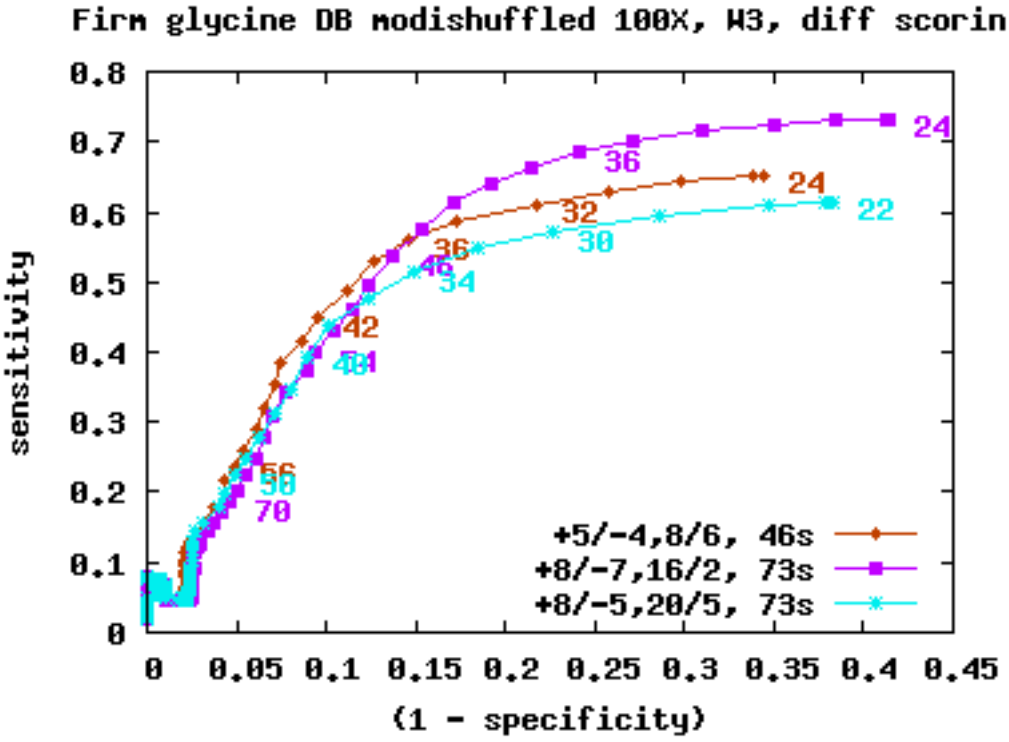
(c) AdoCbl



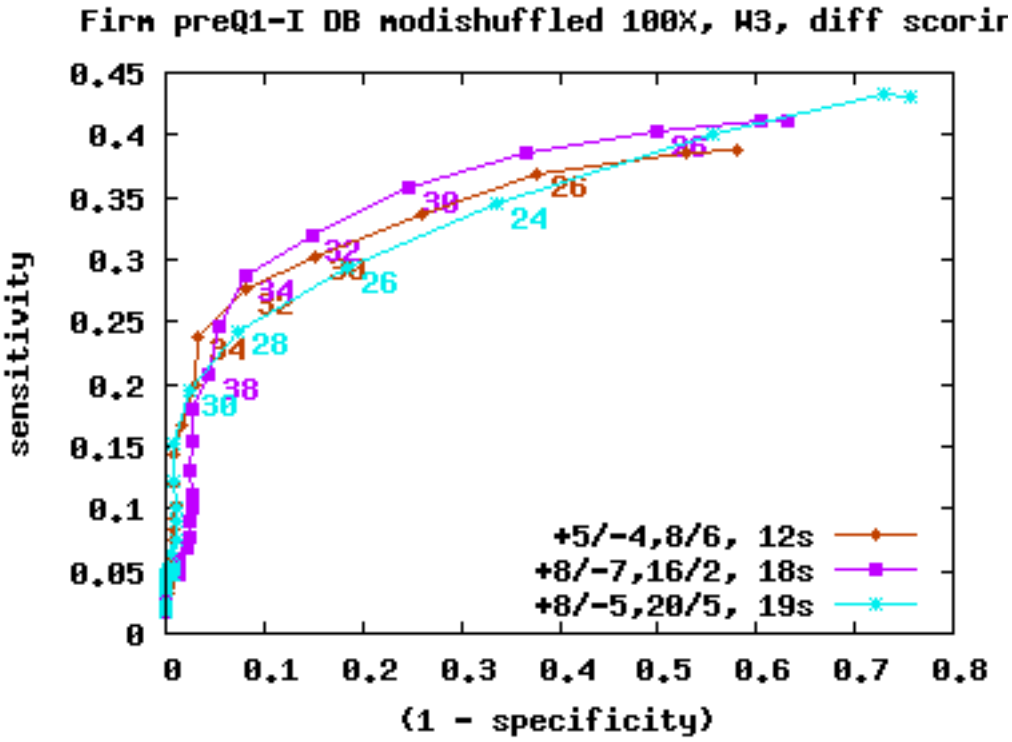
(d) COG4708



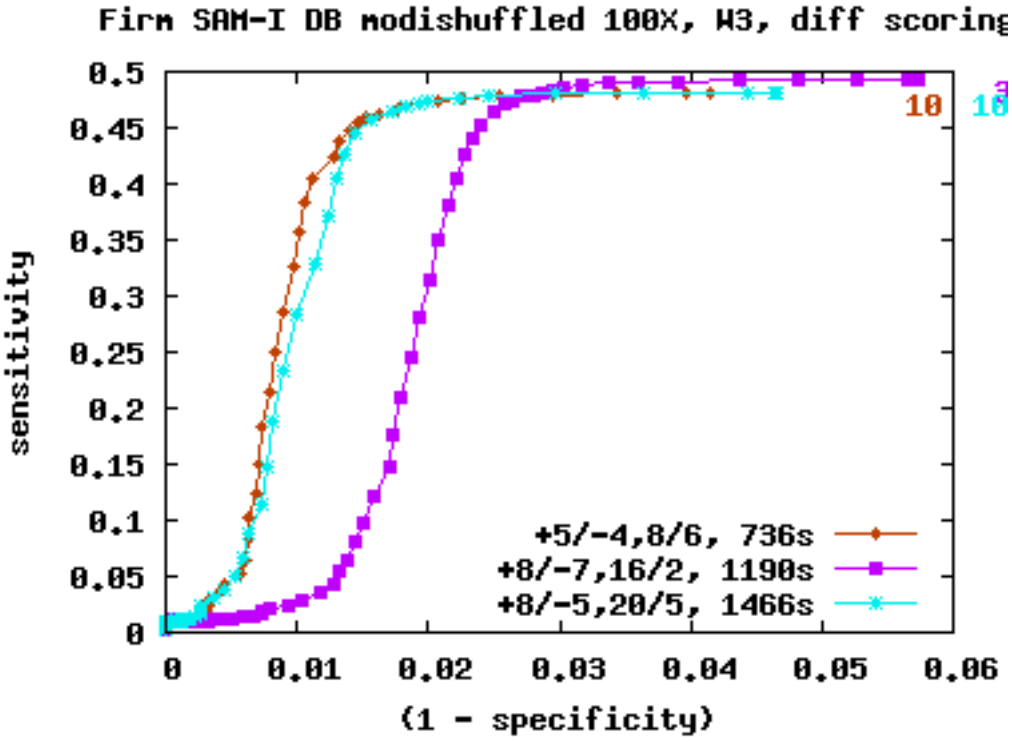
(e) GEMM



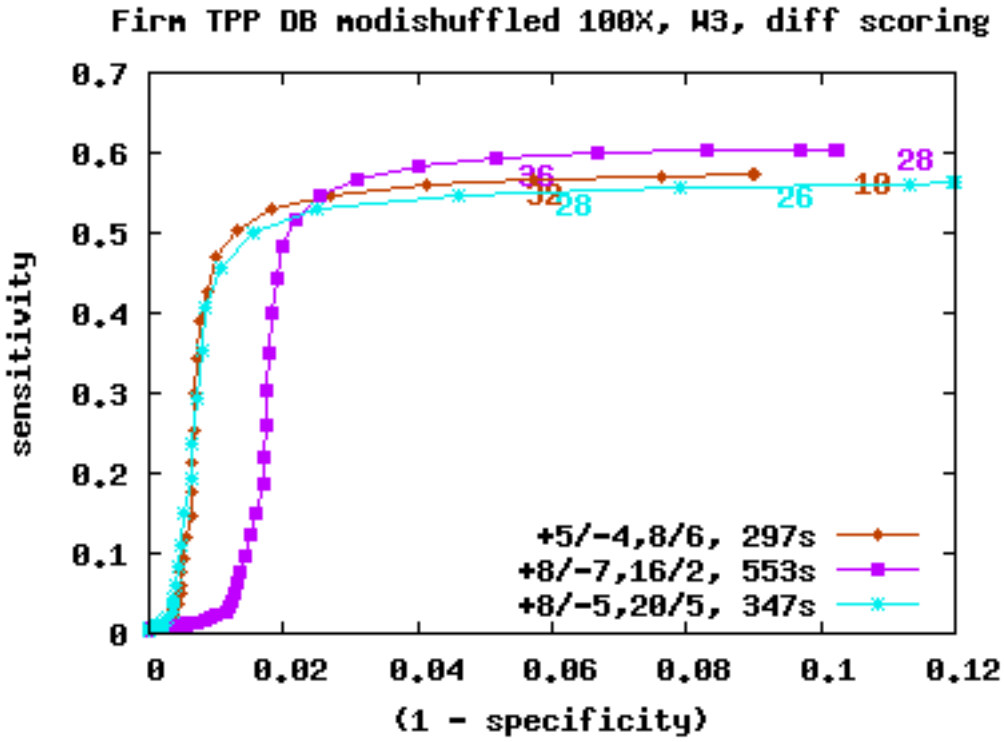
(f) glycine



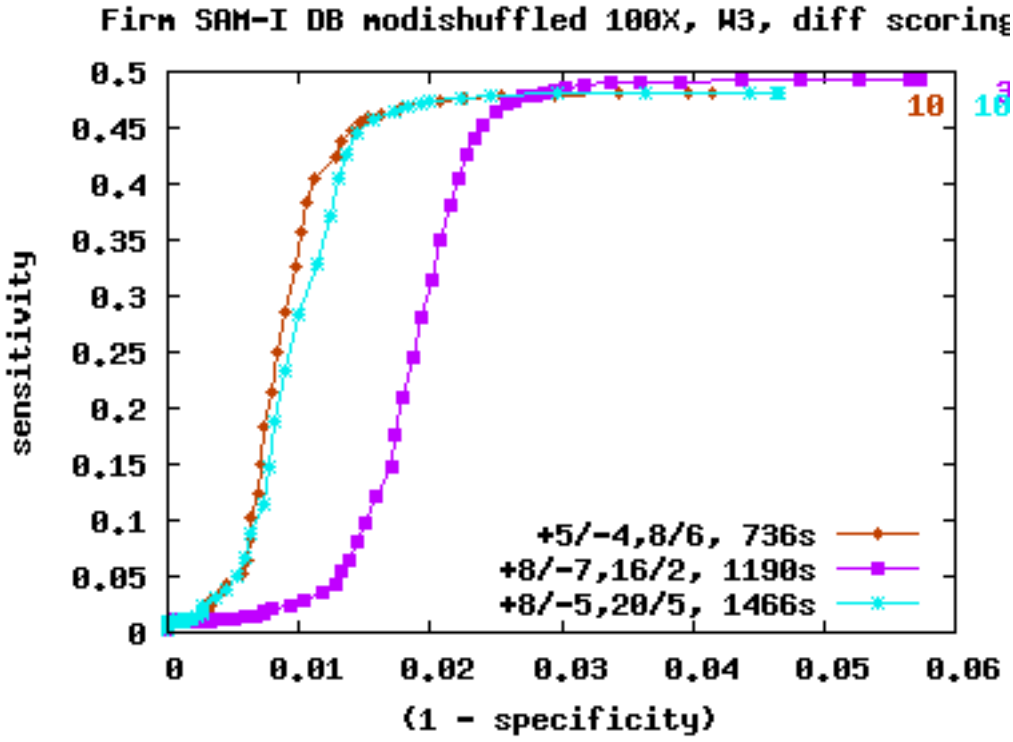
(g) PreQ1-I



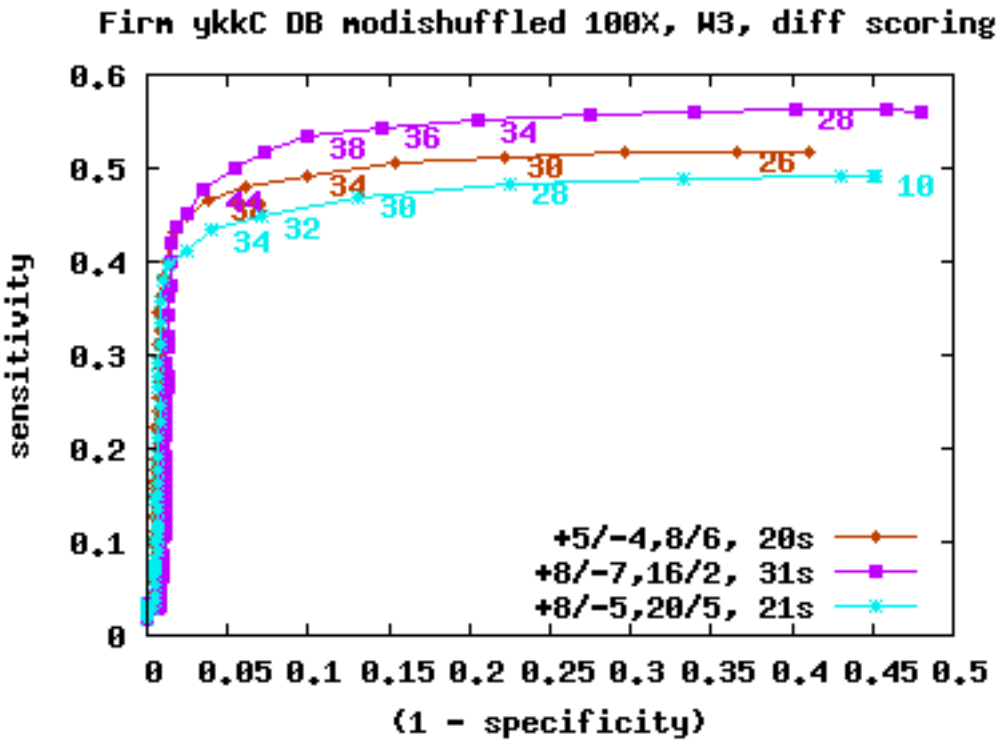
(h) SAM-I


Firm TPP DB modishuffled 100X, W3, diff scoring
sensitivity
(1 - specificity)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.02
0.04
0.06
0.08
0.12
+5/-4,8/6, 297s
+8/-7,16/2, 553s
+8/-5,20/5, 347s
36
32
28
26
28
10

(i) TPP


Firm SAM-I DB modishuffled 100X, W3, diff scoring
sensitivity
(1 - specificity)
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.01
0.02
0.03
0.04
0.05
0.06
+5/-4,8/6, 736s
+8/-7,16/2, 1190s
+8/-5,20/5, 1466s
10

(j) SAM-I



(k) ykkC

Figure A.1: **RoC curve of different BLAST parameters.** For each ncRNA family, we created a database consisting of real ncRNA sequences and 100 times more di-shuffled sequences. We calculated sensitivity and specificity at different BLAST score cutoffs. Three parameter sets {*match/mismatch, gap open/extend*} were tested and are shown here. Selective BLAST score cutoffs are shown in the RoC curves. Runtime (in seconds) are indicated next to the figure legends.

Appendix B

16S RRNA DATABASES

Currently, there are three major 16S rRNA databases: SILVA [75], RDP [17], and Greengenes [60].

B.1 SILVA

SILVA [75] curates both 16S rRNA (SSU) and 23S rRNA (LSU) in all three domains of life. Its versions correspond with the release versions of EMBL [37], from which they continuously extract SSU and LSU sequences using keywords. Sequences are aligned to a curated seed alignment using their own SINA aligner [75] and checked for quality. Taxonomic information is provided using EMBL Taxonomy, Greengenes, and RDP whenever available. SILVA also provides an in-house taxonomy that is a manually curated and revised taxonomy based on Bergey's Trust [106]. Metadata such as publication name, authors, origin of material (e.g., human feces, blood) are also given. The above information, together with the phylogenetic tree, is encapsulated in a .arb file that can be read by the software ARB[53]. With ARB, users can retrieve existing sequences, import new sequences into the database (tree & alignment), manipulate the sequence alignment, and export results. The SILVA website also provides similar functions. Compared to RDP and Greengenes, SILVA has the most comprehensive metadata associated with the sequences. The usage of ARB with SILVA, however, has several drawbacks: ARB is only available in Linux, consumes a lot of memory so performance quickly degrades with the number of sequences imported, and because it is not open source, the various functionalities it provides are not easily scripted. An increasingly common approach now is to use the SILVA sequences with more scriptable software, such as Mothur [85] and Qiime [13].

B.2 RDP

RDP [17] curates 16S rRNA sequences in bacteria and archaea. It obtains rRNA sequences from the International Nucleotide Sequence Databases (INSD: GenBank/EMBL/DDBJ [38]). For the most recent version (RDP 10), sequences are aligned to a structured model based on a seed alignment by Gutell et al. [12] using the Infernal aligner [71]. Taxonomic information is based on The Taxonomic Outline of Bacteria and Archaea (TOBA [28]) and Bergey's Trust [106]. The RDP website allows users to search, browse, and download. Guide tree, aligned sequences, and metadata are all available through the download. RDP does not allow sequences to be imported, like one can do with a SILVA/ARB database. Rather, a naive Bayes classifier trained on sequences in the database (RDP Classifier [100]) can be used to assign taxonomy to unknown sequences. Because RDP Classifier is available as a stand-alone package and can be retrained by user generated data, it is now included in the Qiime package as one of the taxonomy assignment tools.

B.3 Greengenes

The most recent version of the Greengenes taxonomy [60] is constructed using a dramatically different approach from SILVA and RDP. 16S rRNA sequences were downloaded from NCBI, checked for sequence quality, and aligned to Gutell et al.'s seed model using Infernal [71]. Then, a *de novo* tree was constructed using FastTree [74], an approximate maximum likelihood tree inference program. To populate the taxonomy on the *de novo* tree, they developed a program called tax2tree that does the following: (1) assign informative taxonomy names to the tree tips; (2) for each internal taxa (e.g. genus Bacteroidetes), assign it to the best internal node using a criterion that balances between precision and recall (F-measure). After step (2), internal nodes that do not have an assignment are polyphyletic. To resolve this, they use a backfilling procedure where they fill in internal nodes with missing assignment if all descendants share a common taxon name. Finally, the tree was manually curated to fill in all remaining information, which were mostly candidate phyla that were not

well-annotated in NCBI.

The resulting Greengenes taxonomy is very different from that of RDP and SILVA in that not only does it use a computationally inferred tree, it allows for polyphyletic groups.

All three taxonomy databases use different sequence sources, have different quality filtering criteria, and use different taxonomy naming systems. Which is more correct? The issue here is complicated. Taxonomy systems such as TOBA and Bergey's Trust are not just based on the 16S rRNA gene, but can be based on phenotypic traits and (partial or whole) genomic sequences. With more bacterial genomes being sequenced and studied, the systems are constantly under revision. Polyphyletic groups can arise as a result of true biological variation, incomplete or inaccurate sequence information, or different interpretations of sequence evolution. The Greengenes approach uses a purely computational approach to assess the phylogenetic relatedness of bacteria. It is only based on 16S rRNA sequences and does not include information on phenotypic traits or previously vetted taxonomic systems. The authors cite that Werner et al. [105] showed that Greengenes classified more sequences than RDP and SILVA. However, this field is rapidly evolving and it is still early to make a final judgement.

VITA

Huei-Hun Elizabeth Tseng was born in Taiwan. She obtained her B.S. degree in Computer Science from National Taiwan University in 2004. In 2012, she graduated with a Doctor of Philosophy in Computer Science & Engineering from the University of Washington.