Genomic Sequence Fragment Identification using Quasi-Alignment

Anurag Nagar
Southern Methodist University
Dallas, TX 75206
anagar@smu.edu

Michael Hahsler
Southern Methodist University
Dallas, TX 75206
mhahsler@smu.edu

ABSTRACT

Identification of organisms using their genetic sequences is a popular problem in molecular biology and is used in fields such as metagenomics, molecular phylogenetics and DNA Barcoding. These applications depend on searching large sequence databases for individual matching sequences (e.g., with BLAST) and comparing sequences using multiple sequence alignment (e.g., via Clustal), both of which are computationally expensive and require extensive server resources. We propose a novel method for sequence comparison, analysis, and classification which avoids the need to align sequences at the base level or search a database for similarity. Instead, our method uses alignment-free methods to find probabilistic quasi-alignments for longer (typically 100 base pairs) segments. Clustering is then used to create compact models that can be used to analyze a set of sequences and to score and classify unknown sequences against these models. In this paper we expand prior work in two ways. We show how quasi-alignments can be expanded into larger quasi-aligned sections and we develop a method to classify short sequence fragments. The latter is especially useful when working with Next-Generation Sequencing (NGS) techniques that generate output in the form of relatively short reads. We have conducted extensive experiments using fragments from bacterial 16S rRNA sequences obtained from the Greengenes project and our results show that the new quasi-alignment based approach can provide excellent results as well as overcome some of the restrictions of by the widely used Ribosomal Database Project (RDP) classifier.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and Genetics

General Terms

Algorithms

BCB '13, September 22 - 25, 2013, Washington, DC, USA

Keywords

Bioinformatics, Sequence Analysis, Alignment-Free, Sequence Classification, Sequence Identification

1. INTRODUCTION

Living beings show a wide variety and diversity of form. It is estimated that there are between 10-15 million species of life on earth [6, 9]. Identifying organisms and classifying them into the proper taxonomic hierarchy is a phenomenal problem and something that is impossible to be done manually through morphological methods [8] alone.

An alternative method, especially for bacterial species, is using genetic sequences to ascertain the identity of the organisms. This approach has received much attention in recent years due to the increasing availability of low cost sequencing facilities. In a traditional laboratory setting, entire genome of a single organism can be easily isolated and sequenced. Similarly, specific regions of the genomes and particular genes can be isolated and sequenced easily. Using these methods, it is feasible to create a genetic fingerprint of each species or related group of organisms. Various specific regions of the sequences, also known as genetic markers, are used for this purpose.

To analyze specific regions or entire genomes, it becomes necessary to use sequence similarity methods. A large set of sequences can be simultaneously compared using Multiple Sequence Alignment which is known to be NP-complete [22]. To make this type of analysis feasible, heuristics like progressive alignment (e.g., Clustal [17]) have been developed. Another tool for similarity search against a database of sequences is BLAST [15], which outputs shorter regions of high similarity between a query sequence and matched sequences in the database. However, all these methods are still computationally very expensive and require significant computational infrastructure.

Alignment-free methods [21] typically use word frequencies to represent a sequence, where words are subsequences of a fixed length. By comparing word frequency profiles rather than using multiple sequence alignment the computational complexity is greatly reduced. However, these methods consider sequences as "bags of words" and useful information such as location of words and their position-specific distribution is ignored.

In this paper we build on previous work on quasi-alignment [10, 13], which applies computationally very efficient position-sensitive word frequency analysis and data stream clustering to create compact and lightweight profiles of related genetic sequences and defines scoring functions to calculate the sim-

ilarity between sequences and profiles. The original method used the entire 16S rRNA sequence for finding similar regions and taxonomic classification. This approach assumed that the sequences had similar lengths and well-defined start and end points. In this paper, we extend the technique to the more general and also more difficult case of classifying sequence fragments of lengths commonly created by current Next-Generation Sequencing (NGS) technologies (between 200–500 bases).

The rest of the paper is organized as follows: in the next section we present a review of the Quasi-Alignment (QA) method and show how it can be used for two or more sequences. In Section 3, the extension of QA to shorter fragments of sequences is presented which will form the basis of the experiments described in section 4 of the paper. We compare our approach with the leading classifier and show how our method can classify even missing or incomplete data. We describe two sets of experiments in details and present a summary of the results. In Section 5, we present ideas for future work.

2. QUASI-ALIGNMENT

Quasi-alignment (QA) was first introduced in [10] and we will present a brief and updated overview here. QA uses the alignment-free method of word frequencies. First, we formally define word inside a biological DNA sequence and the corresponding word frequency.

Definition 1. A **word** w of length p is a string of p consecutive letters in a sequence S. It is also referred to as p-mer. The **word frequency** of w in S is the occurrence count f_w of w in the sequence.

Alignment-free methods such as word frequency distributions are an efficient alternative to traditional sequence alignment and can give an approximate idea of sequence similarity [21]. However, several research studies have shown that the nucleotide content varies across different areas of a sequence. For example, it is well known that certain regions have a high fraction of G and C bases and are known as GC-rich areas [12, 18]. Similarly, many other studies have shown that certain words are over-represented in specific regions of sequences [7]. Quasi-alignment divides the sequence into equal sized *segments* (by default of 100 bp) and can take into account the position dependent variation in word content and frequency. We define a *segment* inside a sequence as follows:

Definition 2. Given a sequences S of length L, a sequence **segment** $S_{i,l}$ is defined as a sub-sequence starting at position i and having length l where $l \leq L - i - 1$. The starting position i is also referred to as the *offset* of the sequence segment.

Segments are individually analyzed for word frequency distribution. In this analysis, we restrict ourselves to the alphabet consisting of the 4 nucleotide bases $\{A, C, T, G\}$. We refer to the word frequency distribution inside a segment a *Numerical Summarization Vectors (NSV)*, which is defined below.

Definition 3. Given a sequence segment $S_{i,l}$ and a fixed word size p, the **Numerical Summarization Vector (NSV)** for this segments is defined as $NSV_{i,l} = \{f_1, f_2, \ldots, f_{4^p}\}$ where each element f_j represents the count of one of the $j = 1, 2, \ldots, 4^p$ possible words in segment $S_{i,l}$. The order of the 4^p words in the vector is arbitrary but needs to stay consistent over all NSVs.

For example, if we count words of length 3, then there will be a total of 4^3 or 64 elements in the NSV vectors. Figure 1 depicts the process of creating segments and NSVs. For example, the segment $S_{1,10}$ in Figure 1 refers to the segment starting at position 1 and having length 10. The second step converts the segments into word frequency vectors by counting all possible contiguous words in each segment and creating a table of word frequencies. The process is straightforward and similar to ones used in various fields such as text analysis, networking, and other bioinformatics applications.

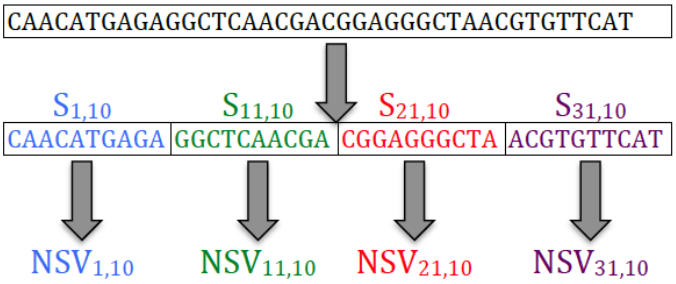

Figure 1: Process of creating Numerical Summarization Vectors (NSVs) from sequence segments

2.1 Pair-wise Quasi-Alignment of Segments

Segments from two different sequences can be compared using their edit distance [11], which is related to alignment and is computed using dynamic programming. In the area of approximate string matching, Ukkonen proposed to approximate the expensive computation of the edit distance between two strings by using q-grams (analog to words in sequences)[19]. First, q-gram profiles (which in our case are NSVs) are created and then the distance between the profiles is calculated using Manhattan distance. The Manhattan distance between two NSVs, x and y, is defined as:

$$d_{\text{Manhattan}}(x, y) = \sum_{i=1}^{4^p} |x_i - y_i| \quad (1)$$

Manhattan distance also has a particularly straightforward interpretation for NSVs. The distance counts the number of words by which two sequences differ which gives the following lower bound on the edit distance between the original sequences S_x and S_y:

$$d_{\text{Manhattan}}(x, y) \leq 2p\, d_{\text{Edit}}(S_x, S_y) \quad (2)$$

This relationship is easy to prove since each insertion, deletion or substitution in a sequences destroys at most p words and introduces at most p new words. Although, we can construct two completely different sequences with exactly the same NSVs (see [19] for a method to create such strings), we are typically interested in sequences of high similarity in which case $d_{\text{Manhattan}}(x, y)/(2p)$ gets closer to the edit distance.

This relationship can be used to determine a reasonable cut-off at which we determine that two segments are similar

enough to consider them to be potentially aligned, i.e., quasi-aligned. For example, we often use a segment size of 100 bases with words of size 3. If we want to quasi-align all segments where less than 5 bases differ, then Equation 2 gives us a threshold of 30.

Definition 4. Two segments S_x and S_y represented by the NSVs x and y are **quasi-aligned (QA)** if, and only if, $d_{\text{Manhattan}}(x, y) \leq 2p\delta$, where p is the word length used to create the NSVs and δ is the alignment threshold defined as the maximum edit distance allowed in two quasi-aligned segments.

An example is shown in Figure 2. Segments across different sequences having NSV distance less than the threshold value are joined together by arrows.

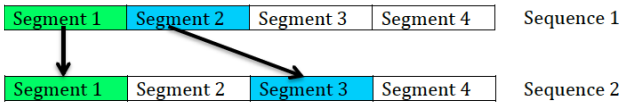

Figure 2: Two quasi-aligned segment pairs for two sequences.

A drawback of the described approach is that the sequences involved need to be split in the same position to make segments comparable. This is not a realistic assumption if we work with fragments and we will remove this restriction later in this paper.

2.2 Multiple Sequence Quasi-Alignment

Comparing all combinations of segments between several (potentially thousands) of sequences becomes quickly computationally expensive. However, we can use clustering to find sets of similar NSVs. Figure 3 shows the clustering process for sequence segments. The blue segments show similar frequency distribution and are clustered together in Cluster 1. Similarly, green segments have similar frequency distribution and are part of Cluster 2.

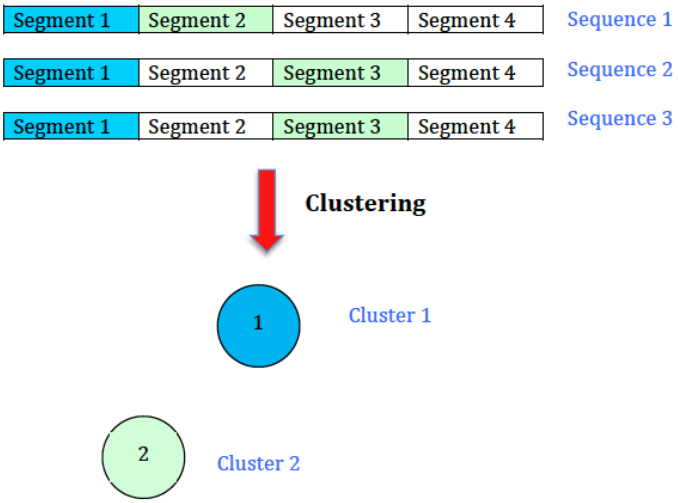

Figure 3: Clustering segments with similar word frequency distributions.

Although any clustering algorithm based on the distances between NSVs (see definition above) could be used, we suggest using high efficiency data stream clustering [2]. These algorithms are designed to cluster very large data sets using a single pass over the data and have only minimal memory overhead. The clustering algorithm used for this study is described in the open source **R** package *QuasiAlign*[14] and can be used for other Bioinformatics applications as well.

2.3 Expanding Quasi-alignments

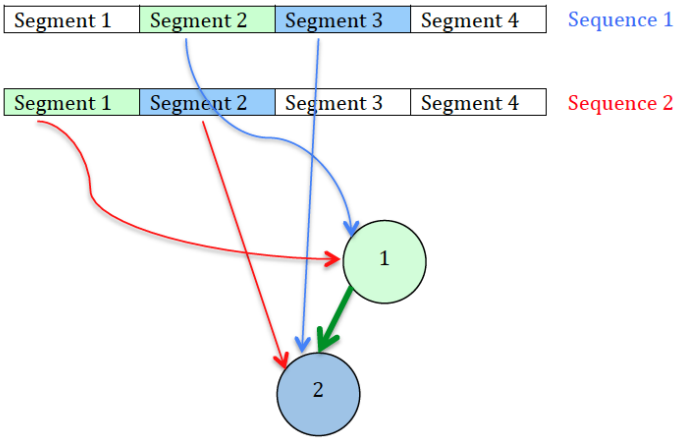

Figure 4: Expanding segment quasi-alignments to larger quasi-aligned areas.

Clusters represent similar segments across multiple sequences and provide useful local (quasi) alignment information. However, alignment might exist for subsequences larger than single segments. For example, Figure 4 shows two sequences, where two consecutive segments in Sequence 1 quasi-align with two consecutive segments in Sequence 2. In this case, the whole area of two segments should be considered a larger quasi-alignment. For pair-wise quasi-alignment this is trivial, however, for multiple sequence quasi-alignment this is more complicated. Our proposed solution is to record the order of segments in sequences while clustering. This is shown in the example in Figure 4 where after seeing contiguous segments clustered in clusters 1 and 2, an association relation between the clusters is marked by a directed arrow joining them. We record such order information for all quasi-aligned sequences with k clusters by counting how often one segment assigned to cluster j follows a segment in cluster i in matrix $\mathbf{C} = [c_{ij}]_{k \times k}$. Note, that by scaling the count matrix by dividing each row by the row sum will give estimates for the conditional probabilities to see a segment grouped in cluster i being followed by one in cluster j. The scaled count matrix $\mathbf{A} = [a_{ij}]_{k \times k}$ can be seen as a transition matrix of a discrete-time Markov Chain [16] with the clusters as states. This is important since Markov Chains are a theoretically very well understood mathematical model for which many properties and guarantees have been established.

We refer to a clustering and the associated count matrix as a genetic model or *GenModel* for short. This model stores in a very compact way (quasi) alignment information for sets of sequences which is similar to multiple sequence alignment. However, segments are clustered into a set of clusters which is typically much smaller than the number of segments and the order information is aggregated at the cluster level. This means that these models are very space efficient.

2.4 Scoring New Sequences against Models

In the previous section, we have introduced how to use the idea of quasi-alignment to create cluster models called GenModels which represent sets of quasi-aligned sequence segments as well as aggregated order information. For many applications it is important to evaluate if a new, unidentified sequence is similar i.e., aligns well with a set of known sequences. After converting the known sequences to GenModels, a new sequence can be scored against them as follows:

1. Find for each segment in the new sequence the best quasi-alignment in the GenModel. This is done by finding for each segment s_t the closest cluster in the model using the distance metric used to build the model. We record the distance for each segment.

2. Evaluate if the segment-wise quasi-alignment can be expanded. For each set of consecutive segments s_t and s_{t+1}, the strength of the transition in the model between states S_t and S_{t+1} is recorded.

3. The distances and transition strengths are aggregated into a single score.

There are many different ways the aggregated score can be calculated and we will only present two here. A full set of scoring methods is implemented and described the package QuasiAlign. A straight forward score for scoring sequences against Markov Chains is the product of transition probabilities along the new sequence. This scores for a new sequence with l segments is defined as:

$$S_{\text{product}} = \sqrt[l-1]{\prod_{i=1}^{l-1} a_{s(i),s(i+1)}} \tag{3}$$

where $s(i)$ is the cluster the i^{th} segment in the new sequence is assigned to, and a_{ij} are elements of the model's transition matrix $\mathbf{A}$.

Another, much simpler score can be obtained by just counting the number of transitions in the new sequence which are also present (supported) in the model.

$$S_{\text{supported\_transitions}} = \frac{1}{l-1} \sum_{i=1}^{l-1} \mathrm{I}(a_{s(i),s(i+1)}) \tag{4}$$

where $\mathrm{I}(v)$ is indicator function which is 0 for $v = 0$ and 1 otherwise.

As a concrete example, Figure 5 shows a sequence on the left and a model on the right. The transition matrix is visualized as a the arrows labeled with the transition probability in the graph. The closest clusters corresponding to each segment is shown by the blue arrows. For this case,

$$S_{\text{product}} = \sqrt[3]{1.00 * 0.20 * 0.40} = 0.431$$

and

$$S_{\text{supported\_transitions}} = \frac{1}{3}(1+1+1) = 1$$

.

For classification, we build a set of GenModels

$$M = \{m_1, m_2, \ldots, m_n\}$$

for the phylogenetic level p and having class labels

$$P = \{p_1, p_2, \ldots, p_n\}$$

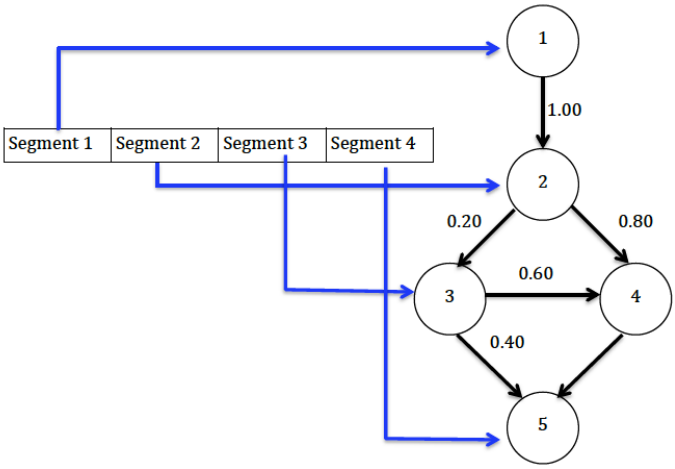

Figure 5: Scoring a new sequence against an existing model.

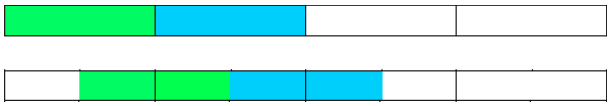

Figure 6: Two sequences with similar regions where quasi-alignment fails due to different offsets.

Then a test sequence S_{test} is scored against all models in M creating similarity scores $\{sim_1, sim_2, \ldots, sim_n\}$. We then assign the test sequence to class p_k where k is the index of the highest similarity score, i.e.,

$$sim_k = max(sim_1, sim_2, \ldots, sim_n) \tag{5}$$

3. QUASI-ALIGNMENT FOR FRAGMENTS

All previous research on quasi-alignment has focused on complete 16S rRNA sequences where creating consistent segments is easy since these sequences have a well defined starting and ending points. This means that cutting different sequences at the same offset will ensure that the segments cover the same area and are comparable. However, such an approach will not work well with random sequence fragments (such as those encountered in Next Generation Sequencing data) as they can have variable starting and ending points. Consider the example in Figure 6. The colored regions indicate aligned areas, but since the segments in the second sequence start at different offsets, quasi-alignment cannot find them.

To remedy this shortcoming we suggest to learn a model by considering each possible offset in the sequence. This will allow us to capture all possible segments of the given length irrespective of their position. As an example, consider a sequence of length 20 nucleotides that has been divided into 4 segments of size 5 each. This is shown in Figure 7. The first pass through the sequence is identical to before. We start at offset 0 and create models using the segments {ACTGG, CACTG, GTAAA, CGCGT}. In the enhanced algorithm, we make 4 more passes by creating segments at offset 1, 2, 3 and 4. The second pass through the sequence with offset 1 creates segments {CTGGC, ACTGG, TAAAC, GCGT} and so on.

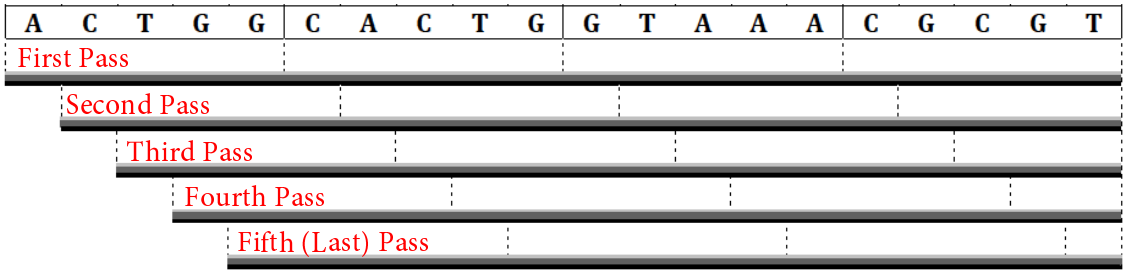

Figure 7: Segment creation for quasi-alignment of fragments.

Table 1: List of phylums used in experiments

Phylum	Counts
Thermotogae	445
Synergistetes	421
SAR406	336
Fibrobacteres	169
Deferribacteres	168
Chlorobi	408
Chlamydiae	216
Armatimonadetes	136

Effectively, a sequence of length L is treated as $L - l + 1$ different sequences of length l each with different starting points. Each sequence has one base striped off from the start and a new base appended at the end. This way we learn all possible segment cutting points and no matter at what position a fragment starts there will be a cluster in the GenModel that learned from sequences at the same position. Note, that this approach increases the time needed to create models by the constant factor of L. The size of the model will increase only be a much smaller degree, since many new segments will readily cluster into already existing clusters. Also, in order to score a new sequence, it only needs to be cut into segments using a single offset and not all L possible ones. Therefore, the additional computational burden only effects model creation which is not done often and can be executed as a batch job using more computational power.

4. EXPERIMENTS

For the experiments we use bacterial DNA sequences from the 16S rRNA gene, which are widely used for phylogenetic studies because it has remained relatively conserved over time [23]. The sequence data along with the complete phylogenetic hierarchy is available from the Greengenes [1] project. We used the unaligned format of the data which is available from the Greengenes website at `http://greengenes.lbl.gov`.

The sequences were parsed and efficiently loaded into SQLite databases using the R package BioTools. After that, we randomly selected sequences from 8 medium sized phylums having between 100–500 sequences. The list of phylums and their sequence counts is shown in Table 1. The same analysis can be done for any sized phylums, with larger ones requiring more processing time.

The sequences were then split up in the ratio 90/10 for training and test within each rank. For example, for the phylum "Synergistetes" having 421 total sequences, 90% or 379 were used for model creation and the remaining 10% or 42 were used for testing. Since we are interested in classifying fragments and not the entire sequences, we randomly extracted a fragment of fixed size (400 bp) and used it for classification against the trained models.

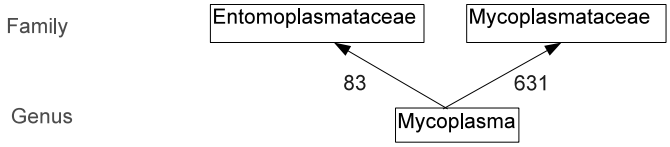

Figure 8: RDP classifier is not able to handle multiple inheritance within the taxonomic hierarchy.

4.1 Validation of Results

To validate our approach, we compared the class label prediction from Quasi-Alignment (QA) to the actual phylogenetic class label which is available in the Greengenes database. We also compared our performance against the RDP classifier, which is a popular tool in molecular biology and has been widely used for taxonomic identification and classification [3, 20]. It takes a Naïve Bayesian approach to classification by considering a feature space consisting of all words of length 8 in rRNA sequences. We used the interface to RDP available in the BioTools package [5] and it can be used to create a custom trained RDP classifier using the same set of sequences that were used for training the QA classifier.

4.1.1 Problems training the RDP Classifier

Although widely used, we discovered some problems when training the RDP classifier using data from Greengenes. First is the issue with a lower rank belonging to more than one upper rank, which we refer to as the *multiple inheritance* problem. RDP classifier can not handle this as it requires an exact hierarchical tree to be constructed as detailed in the instructions of RDP classifier download page at : `http://sourceforge.net/projects/rdp-classifier/files`. As an example, the genus *Mycoplasma* can belong to two families *Entomoplasmataceae* and *Mycoplasmataceae* with 83 and 631 sequences in each respectively. This is illustrated in Figure 8. RDP classification tree can not be constructed in such cases and manual pre-processing is required to clean the data and remove the sequences that have lower count at the family level. QA classifier can handle such a situation and is able to classify sequences with incomplete or missing taxonomic data and still achieve excellent prediction accuracy.

Table 2: Parameters used for the experiments

Parameter	Value
Segment Length	100
Clustering Threshold	30
Word Size	3
Test Fragment Size	400
Scoring Method	Supported Transitions

It has been estimated that almost 43% of sequences in the Greengenes database are not completely classified [4]. In such a scenario, the requirement of having a perfect hierarchy clearly laid out for the training data set is too restricting. We believe that our classifier does away with this limitation and can be used on a less perfect but much larger data set.

Another possible limitation of the RDP classifier is that it works best at the genus level (as mentioned in the README file of the classifier). Although the tree can be constructed down to the species level, it would require lot of pre-processing effort in cleaning up the data since a large percentage of species are unclassified.

4.2 Phylum Level Classification Results

For the first set of experiments, we used the 8 phylums listed in Table 1 and randomly split each phylum into training and test data using a ratio of 90/10. The models were constructed from the training dataset using the default values of all parameters as shown in Figure 2 and random fragments of fixed length from each test sequence were scored against each of the models to find the similarity score.

In the first set of experiments, we constructed models at the phylum level. We compared our results with those from the RDP classifier and also the actual classification obtained from the Greengenes database. The parameters used for creating the models are specified in Table 2. We chose random fragments from the test sequences and classified them using the QA and RDP classifiers. The RDP classifier was trained at the genus rank as suggested in the documentation.

The results are presented in Table 3 below. Some points need to be clarified here:

1. For the RDP classifier to work, a depth parameter has to be specified which refers to the depth of the hierarchy tree. By default, a tree is constructed upto the genus level as suggested in the documentation. In many cases, sequences have incomplete hierarchy and a tree can not be constructed. For example, phylum *SAR406* contains several sequences that have been classified only upto the order level and not below. In such cases, it is not possible to construct the complete hierarchy tree.

2. The somewhat lower performance of RDP can be due to the fact that sequences with any taxonomic rank missing upto the genus level have been removed. We could have constructed a partial taxonomic tree for each case, but that would require extensive pre-processing and cleaning of training data set every time. The point to be noted from this analysis is that QA can perform much better when there is uncertainty in the data.

3. We tried to filter out those sequences that had incomplete or ambiguous hierarchy as much as possible. However, this is not feasible in cases of real world sequences. Sometimes, a researcher might be interested in getting an idea of the taxonomic hierarchy even if partial information is available.

4. QA requires no pre-processing of data either due to incomplete hierarchy or because of the multiple inheritance problem. It can be used locally and can also store vital meta information about the models [13].

Results in Table 3 show that at the phylum level QA is able to outperform RDP and also does not require any preprocessing or cleaning of the sequence data. It can be run locally and does not require extensive server resources.

4.3 Species Level Classification Results

In the second round, we conducted experiments for identification of species from sequences. Previous work and classifiers such as RDP have focused mostly on genus level classification and does not provide an easy way to identify species from their sequence data. This can be due to the fact that it is hard to differentiate sequences from same or closely related genera. QA has a flexible approach to classification and can be tuned to classify at a finer level and differentiate species sequences.

We selected 10 random species from the Greengenes database that contained between 100–500 sequences and carried out testing on them using a training/test ratio of 90/10. We used the default values of the parameters in this case which are specified in Table 2.

The list of species along with their counts are in the first two columns of Table 4. The number of sequences used for testing are shown in the third column. In case of species classification, there were many ties in the similarity scores between the test sequences and the GenModels. This is expected as some species such as *Bacillus amyloliquefaciens* and *Bacillus anthracis* are closely related and belong to the same genus *Bacillus*. In cases of tied winners, we consider the sequence as correctly classified. This is reflected in the fourth column of the table. The last three columns give more information on ties and how many species are tied. It can be seen that in more than 30% of the cases we have been able to identify the clear winner and in more than 58% of the cases the correct classification was one of three tied species. Overall, in 99.56% of the cases a clear or tied winner was the correct classification. Therefore, this classification algorithm can serve as a good starting point for further analysis since more computationally expensive methods (e.g., multiple sequence alignment) can be limited to just a few potential species rather than the entire database.

5. DISCUSSION AND FUTURE WORK

We have presented a simple yet powerful technique for fragment identification and classification using the Quasi-Alignment approach. This method avoids the computationally expensive alignment process and is able to take a higher level view of sequences by looking at the word frequency distribution inside fixed length segments.

We carried out extensive experiments using open source packages and very modest computing infrastructure. Another useful feature of QA is that it can use incomplete or missing taxonomy information to construct models and classify unknown sequences in a fuzzy way. This is an improvement over existing classifiers such as RDP that require the

Table 3: Sequences used in Phylum Level Experiments and Comparison between QA and RDP

Species	Counts	Used for testing	QA Correct	QA % Correct	RDP Correct	RDP % Correct
Thermotogae	445	45	45	100%	45	100%
Synergistetes	421	43	42	97.67%	43	100%
SAR406	336	34	34	100%	28	82.35%
Fibrobacteres	169	17	17	100%	12	70.59%
Deferribacteres	168	17	17	100%	17	100%
Chlorobi	408	41	40	97.56%	34	82.93%
Chlamydiae	216	22	22	100%	22	100%
Armatimonadetes	136	14	13	92.86%	8	57.14%
Total	2299	233	230	98.71%	209	89.70%

Table 4: Sequences used in experiments for species classification

Species	Counts	Test Total	Correct	% Correct	%1 Winner	%2 Winners	%3 Winners
Eubacterium biforme	494	50	50	100%	4%	30%	44%
Neisseria meningitidis	484	49	49	100%	95.92%	4.08%	0%
Clostridium perfringens	480	48	48	100%	0%	6.25%	33.33%
Bacillus anthracis	475	48	48	100%	0%	0%	0%
Collinsella aerofaciens	449	45	45	100%	80%	17.78%	0%
Lactococcus lactis	434	44	44	100%	0%	11.36%	40.91%
Dialister invisus	427	43	42	97.67%	0%	11.63%	27.91%
Bacteroides plebeius	426	43	43	100%	100%	0%	0%
Ruminococcus gnavus	401	41	40	97.56%	24.39%	14.63%	21.95%
Bacillus amyloliquefaciens	397	40	40	100%	0%	2.50%	10%
Total	4467	451	449	99.56%	30.60%	9.98%	17.96%

taxonomy of all training sequences to be defined completely upto the desired classification level.

The performance of QA is excellent and can achieve high classification accuracy at all levels, from phylum down to the species level. In the experiments that we performed, our methods were able to correctly classify over 90% of the sequences in all cases using a ratio of 90/10 for training and test sequences. A higher training ratio further improves our performance.

In this paper, we have presented a preliminary analysis of using QA for fragment identification and classification. Our results have been very encouraging and we plan to develop a complete framework for identifying unknown sequences using the existing taxonomic knowledge from the Greengenes and other related sources. We also plan to build a web interface so that a wider audience can access this work.

6. ACKNOWLEDGEMENT

This research was partially supported by the National Human Genome Research Institute under grant no. R21HG005912 and by the National Science Foundation under grant no. IIS-0948893.

7. REFERENCES

[1] Greengenes Website – 16S rRNA Gene Database. http://greengenes.lbl.gov. [Online; accessed November-2012].

[2] C. Aggarwal, editor. *Data Streams – Models and Algorithms*. Springer, 2007.

[3] J. Cole, B. Chai, R. Farris, Q. Wang, A. Kulam-Syed-Mohideen, D. McGarrell, A. Bandela, E. Cardenas, G. Garrity, and J. Tiedje. The ribosomal database project (RDP-II): introducing myRDP space and quality controlled public data. *Nucleic Acids Research*, 35(suppl 1):D169–D172, 2007.

[4] T. DeSantis, P. Hugenholtz, N. Larsen, M. Rojas, E. Brodie, K. Keller, T. Huber, D. Dalevi, P. Hu, and G. Andersen. Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Applied and environmental microbiology*, 72(7):5069–5072, 2006.

[5] M. Hahsler and A. Nagar. *BioTools: Tools based on Biostrings (alignment, classification, database)*, 2013-05-01. R package version 0.0-1.

[6] P. Hammond. Species inventory. In *Global biodiversity*, pages 17–39. Springer, 1992.

[7] S. Hampson, D. Kibler, and P. Baldi. Distribution patterns of over-represented k-mers in non-coding yeast DNA. *Bioinformatics*, 18(4):513–528, 2002.

[8] P. D. Hebert, A. Cywinska, S. L. Ball, et al. Biological identifications through DNA barcodes. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(1512):313–321, 2003.

[9] V. H. Heywood et al. *Global biodiversity assessment.* Cambridge University Press, 1995.

[10] R. M. Kotamarti, M. Hahsler, D. Raiford, M. McGee, and M. H. Dunham. Analyzing Taxonomic Classification Using Extensible Markov Models. *Bioinformatics*, 26(18):2235–2241, 2010.

[11] V. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10, 1966.

[12] H. Miyata, H. Tsunoda, A. Kazi, A. Yamada, M. A. Khan, J. Murakami, T. Kamahora, K. Shiraki, and

S. Hino. Identification of a novel GC-rich 113-nucleotide region to complete the circular, single-stranded DNA genome of TT virus, the first human circovirus. *Journal of Virology*, 73(5):3582–3586, 1999.

[13] A. Nagar and M. Hahsler. A Novel Quasi-Alignment Based Method for Discovering Conserved Regions in Genetic Sequences. In *Proceedings of the IEEE BIBM 2012 Workshop on Data-Mining of Next-Generation Sequencing*. IEEE Computer Society Press, October 2012.

[14] A. Nagar and M. Hahsler. *QuasiAlign: Infrastructure for Quasi-Alignment of Genetic Sequences*, 2012-12-06. R package version 0.0-3.

[15] National Center for Biotechnology Information. BLAST - Basic Local Alignment Search Tool. `http://blast.ncbi.nlm.nih.gov/Blast.cgi`. [Online; accessed November-2012].

[16] E. Parzen. *Stochastic Processes*. Society for Industrial Mathematics, 1999.

[17] J. Thompson, D. Higgins, and T. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, 22(22):4673–4680, 1994.

[18] V. Torsvik and L. Øvreås. Microbial diversity and function in soil: from genes to ecosystems. *Current opinion in microbiology*, 5(3):240–245, 2002.

[19] E. Ukkonen. Approximate string matching with q-grams and maximal matches. *Theoretical Computer Science*, 92(1):191–211, 1992.

[20] C. Vilo and Q. Dong. Evaluation of the RDP Classifier Accuracy Using 16S rRNA Gene Variable Regions. *Metagenomics*, 1:1–5, 2012.

[21] S. Vinga and J. Almeida. Alignment-free sequence comparison-a review. *Bioinformatics*, 19(4):513–523, 2003.

[22] L. Wang and T. Jiang. On the complexity of multiple sequence alignment. *Journal of computational biology*, 1(4):337–348, 1994.

[23] C. R. Woese and G. E. Fox. Phylogenetic structure of the prokaryotic domain: the primary kingdoms. *Proceedings of the National Academy of Sciences*, 74(11):5088–5090, 1977.