Classifying HIV-1 Circulating Recombinant Forms

A. Steven Eliuk[1], **B. Keith Ruiter**[2], **and C. Pierre Boulanger**[1]
[1]Department of Computing Science,
Advanced Man Machine Interface Laboratory (AMMi), University of Alberta, Edmonton, Alberta, Canada
[2]University of Alberta, Edmonton, Alberta, Canada

Abstract— *The intent of the following paper is to expound on new algorithmic ideas that show marked improvement over formerly state-of-the-art functions in HIV-1 subtyping, such as those found in Wu et al. and NCBI. The paper identifies deficiencies in these older conceptions and sets forth, in a clear and simplistic manner, our improved methodology. The two main boons to the new method described below are the development and utilization of reference profiles and the increased recombination prediction accuracy due to increased branching options and redesigned replacement policies. There is also a new importance placed on absolute prediction accuracy, thus making room for a multitude of real-world possibilities.*

Keywords: Recombination, HIV-1 subtyping, statistical classification

1. Introduction

Human Immunodeficiency Virus type-1 (HIV-1) is incredibly adaptive and diverse. This diversity is caused by a high error rate during transcriptase and a likelihood of recombination [4]. Recombination is the process by which different pure subtypes recombine to form a new strain, in terms of HIV-1, a new circulating recombinant form (CRF) is generated. Understanding recombination, and correctly classifying the pure subtypes that define a CRF, gives the research community the means by which to correctly define the phylogeny of the virus. By understanding the evolution of the virus, the development of effective drug treatments and control vaccines could be possible. Lastly, by correctly classifying an HIV-1 CRF in a host, correct drug treatment could be established, if available for the CRF in question.

Techniques from [11] and [12] and those from NCBI [10], and others [5], [4] using sequence alignment have been very good at predicting the genetic subtypes for an HIV-1 strain, with Wu *et al.* obtaining 100-percent prediction accuracy. However, detection and classification of an HIV-1 CRF is very difficult [11] to attain. Algorithms, such as construction of top strings from *relative entropy*, in order to determine the subtypes of a CRF test sequence and that is proposed by [10], which uses NCBI sliding window to create BLAST similarity scores between reference and testing sequences, have preformed reasonably well (obtaining $\approx$ 87-percent and 77-percent prediction accuracy respectively). However one should note, in [11], that the prediction accuracy refers not to the number of correctly predicted pairs, but the number of correctly predicted subtypes. For example, take testing sequences CRF1-A1F1 and CRF1-A2G1. These two sequences have four subtypes, mainly A1, F1, A2, and G1, [11] only gives the accuracy in terms of correctly defined singles. In this case, a 50-percent prediction accuracy would represent classifying 2/4. Even though it is possible that CRF1-A1F1 was classified as A1 and D, likewise CRF1-A2G1 could be classified as B and G1. Results being 2/4 correctly identified (50-percent) but zero pairs correctly classified. In this paper, *absolute* prediction accuracy will refer to the metric of correctly classified *pairs*, and *relative* prediction accuracy will refer to the metric used in [11] of correctly classified subtypes. Obviously a correctly classified pair provides more information, but for comparative purposes with [11] we will list both *relative* and *absolute* prediction accuracy.

[11] shows great results in terms of *relative* prediction accuracy, achieving the noted 87-percent; however, testing for *absolute* accuracy (complete *pair* subtype match) results in 70-percent prediction accuracy, a remarkable difference. The novelties of our algorithm stem from a quicker implementation of [11] along with changes and improvements in both *relative* and, most importantly, *absolute* prediction accuracy. There are three new techniques implemented, all obtaining improvements in runtime and accuracy; however, all are based on the generation of top strings T, *relative entropy*, and Euclidean distance between reference sequences and test sequences, formally found in Wu *et al*.

The information below will: give a formal description of the methodologies used, the underlying algorithm, and the three subsections defining the main novelty of each algorithm; provide a small section describing the 42-reference sequences used for generation of top strings and the 91 CRF test sequences; a results section, showing results of T on accuracy; and lastly *relative* and *absolute* prediction accuracy of the baseline algorithm from [11] and the three new refinements.

2. Review

2.1 Nucelootide composition string selection in HIV-1 subtyping using whole genomes [11]

The techniques from Wu *et al.* are based on nucleotide composition string selection. This methodology was chosen by Wu *et al.* for a number of reasons. First, it requires

no foreknowledge of the genes being tested. Second, no compression is undertaken which results in fewer errors. Due to the fact that every composition string provides unequal amounts of information to the evolutionary distance calculation, Wu *et al.* noted that by selecting the most important composition strings, those that contribute the most evolutionary data, analysis of thousands of strings can be done in a very affordable manner. This nucleotide composition string selection is a highly effective way to assess HIV-1 recombination and evolution. By selecting the genes that contribute most information to the evolutionary process Wu *et al.* met with impressive results in predicting HIV-1 subtyping. The dataset utilized by Wu *et al.* was composed of 867 pure subtype HIV-1 strains and 331 recombinants. By setting the maximum number of strings at 500 and ensuring string length did not exceed 21, Wu *et al.* attained 100% leave-one-out subtyping accuracy while maintaining computational efficiency. To further test this methodology, Wu *et al.* blindly compared their results to three HIV-1 subtyping programs, again meeting with impressive results.

2.2 Top Strings

A string of nucleotides is generated from a reference sequence in an incremental fashion up to length-*K*. For example, take the nucleotide sequence AAGC, and length-*K* = 3, the strings constructed would be A, AA, AAG, A, AG, AGC, G, and GC. Notice that the maximum length string is three equaling length-*K*.

Each string generated from the reference sequences is scored based on *relative entropy* (or Killback-Leibler distance), Equation 1.

$$s(\alpha) = \sum_{i=1}^{n} |\pi(\alpha, i)| ln| \frac{\pi(\alpha, i)}{\Pi(\alpha)}|, \tag{1}$$

where $s(\alpha)$ = *relative entropy* of string α, *i* = genome *i*, *n* = number of whole genomes, $\pi(\alpha, i)$ = absolute composition value for string α in a given genome *i*, defined in Equation 2, and $\Pi(\alpha)$ = *unnormalized* background probability.

$$\pi(\alpha) = \frac{(p(\alpha) - q(\alpha))}{q(\alpha)} \tag{2}$$

where $\pi(\alpha)$ = absolute composition value, $p(\alpha)$ = probability of string α in a given genome, and $q(\alpha)$ = expected appearance of string α defined in Equation 3.

$$q(a_1 a_2 ... a_k) = \frac{(p(a_1 a_2 ... a_{k-1}) * p(a_2 a_3 ... a_k))}{p(a_2 a_3 ... a_{k-1})}, \tag{3}$$

where $p(a_1 a_2 ... a_{k-1})$ = probability of sub pattern a_1 to a_{k-1}, $p(a_2 a_3 ... a_k)$ = probability of sub pattern a_2 to a_k, and $p(a_2 a_3 ... a_{k-1})$ = probability of sub pattern a_2 to a_{k-1}.

2.3 Complete Composition Vector (CCV)

After the scoring and ranking of strings, the top T strings are used to compute a CCV. The vector always has T values and represents the composition values of the top strings in a given genome. Where the vector index i would represent the composition value of the i^{th} top ranked string. String selection and scoring is very important to this technique, with higher scores seeming to contain richer information [11], [12]. Generating the selected string composition vector is rather simple. If there are less than 500 strings, add the current string in question. If not, and the current string has a higher score, a larger absolute relative entropy, then the lowest score is replaced. This technique of only storing the richest 500 strings basically resolves all memory issues according to [11]. Once all the strings have been examined a 500-dimensional composition vector is built. For example, testing in [11] included the use of 42 reference whole genomes, 331 recombinant, and 825 pure subtype whole genomes. 500 top ranked strings were used, in turn producing a 500-dimensional composition vector. The technique was 100% successful in the subtyping of the 825 pure subtypes. Most importantly, the technique does not rely on prior knowledge about the genomic sequences.

2.4 Pair-wise distance

Given a pair of genomes, a and b, the distance between them can be represented as the Euclidean *distance* between their respective CCV's as seen in Equation 4.

$$distance = (\sum_{l=1}^{m} (a_l - b_l)^2))^{1/2} \tag{4}$$

2.5 Basic Local Alignment Search Tool (BLAST)

BLAST is a widely used method for comparison between nucleotide and protein sequences. It is used to determine relative relationships between test and reference strains [6]. BLAST is such an effective tool because of its speed and ease of use; however, it is victim to one downfall, namely that, because of its high speed, its optimality cannot be guaranteed in alignment. This large speedup, approximately fifty times faster than conventional optimal algorithms, is made possible by a simple heuristic. Using this heuristic ensures high computing speed while maintaining quality results and high accuracy. More information about the specifics of BLAST can be found at [6]. BLAST is a useful tool in the analysis of recombination in HIV, such as being able to compare a test strain against known reference strains using BLAST, in order to classify the test strain. After utilizing BLAST, the results can show a high probability of belonging to a certain clade, being recombinant, or being a pure subtype. If the results show the test strain belongs to a certain clade, a drug treatment that is specific to this clade can be administered for a more effective treatment.

2.6 National Centre for Biotechnology (NCBI) algorithm using Scored BLAST

NCBI, being considered a state-of-the-art institution [10] in recombination detection prior to 2007, utilizes a technique that uses a score based BLAST [6] pairwise alignment between overlapping segments. This alignment is carried out between a query sequence and a known reference sequence. The algorithm moves a window along the query sequence, processing each window segment separately while comparing each against the reference sequences using BLAST. BLAST returns a similarity score for each local alignment [10]. The reference sequence that matches with the highest similarity score is assigned for the local alignment. The process is repeated for each window and recorded. Once the comparisons are completed, if a single genotype is assigned to most segments, the query segment is considered a single genotype and classified accordingly. If multiple genotypes were recorded during local alignment and the percentage belonging to each genotype is higher than a predefined threshold. the query sequence is deemed recombinant. This process could easily be used to speculate the most probable breakpoint for recombination [10] because the location of divergence is easily seen when local alignment produces a new reference sequence and they match continually. The three parameters that govern the NCBI method are: the choice of window size, often experiment specific; the incremental step, defined as the amount the window is shifted along the sequence; and the similarity threshold, defined as the percentage of non-primary genotypes that can be recorded before recombination is considered, for a match. The NCBI method is impressively simple and the results it yields are among the best when detecting recombination. Tests of 48 reference sequences [10] were used to predict recombinant deterministic forms. NCBI was able to obtain a 73.4% prediction accuracy where later CCV tests only yielded 66.2% prediction accuracy using the same reference sequences. This method was able to accurately predict all but two CRF12BF strains, namely AY771588 and AY771589. The techniques of [11] were tested on the 91 strains that have deterministic recombinant forms and was able to determine 87.3% accuracy. Likewise, the 42 known reference sequences were used; however, 5000 top ranked strings were used vs. the 500 top ranked strings used in pure subtyping. The results were a substantial increase over those of NCBI.

2.7 Detecting subtypes in CRF

Difficulty arises when trying to compute the pure subtypes that make up a CRF. There is no guarantee the breakpoint is consistent and it likely varies. Therefore, Wu *et al*. suggests breaking a sequence into equal parts. At each testing, a consecutive number of parts are removed and the remaining concatenated together. For example, take a partitioning factor P = 50, a CRF genome would be broken into $\approx$ 180 nucleotides (9000 nucleotides / 50 = 180). A maximum l parts can be removed, $l \approx P/2$ seems to work well in empirical testing.

Given a partitioning factor of $P = 50$, and if $1 <= l <= 25$ parts can be removed, we would construct the following test strings.

$$
\begin{gathered}
l = 1, \\
s_1 = (p_2 ... p_{50}), \\
s_2 = (p_1, p_3 ... p_{50}), \\
... \\
s_{49} = (p_1 ... p_{48}, p_{50}), \\
s_{50} = (p_1 ... p_{49}). \\
l = 2, \\
s_1 = (p_3 ... p_{50}), \\
s_2 = (p_1, p_4 ... p_{50}), \\
... \\
s_{47} = (p_1 ... p_{47}, p_{50}), \\
s_{48} = (p_1 ... p_{48}). \\
. \\
. \\
. \\
l = 25, \\
s_1 = (p_{26} ... p_{50}), \\
s_2 = (p_1, p_{27} ... p_{50}), \\
... \\
s_{24} = (p_1 ... p_{24}, p_{50}), \\
s_{25} = (p_1 ... p_{25}).
\end{gathered}
$$

In all, 950 strings are constructed. For each test string the CCV is generated and the Euclidean distance between the test string and the reference CCVs are calculated, see Equation 4. The two reference sequences, that, when compared against the test sequence, produced the lowest scores are recorded. In all, 1900 reference sequences would be stored. The frequency of a reference sequence can be thought of as the amount of the test genome that belongs to a specific reference sequence; in turn, a specific pure subtype. The two reference sequences with the greatest frequency are reported as the two predicted pure subtypes of the test CRF sequence.

2.8 Conclusion

The base technique from Wu *et al.* is seen in many different areas of computer-based learning. The algorithm breaks down into a learning stage, a metric between learned top ranked strings and reference sequences; distance is then computed between test and reference sequences before the minimum distances between reference and test data is finally associated with the most probable match. Many enhancements are possible, such as using the ordering of top ranked strings as a weight metric. Giving a higher weight to the very best strings and decreasing accordingly as lower ranked strings are used. Likewise, different distance metrics can be used when comparing test to reference sequences and; furthermore, the metric used to score a string can be replaced with a variety of other metrics. As with most

Replacement Policy	
R	[A or G]
Y	[T or C]
K	[G or T]
M	[A or C]
S	[G or C]
W	[A or T]

Fig. 1: Nucleotide Replacement Policy, see [1]

Replacement Policy	
B	[C or G or T]
D	[A or G or T]
H	[A or C or T]
V	[A or C or G]
N	[A or C or G or T]

Fig. 2: Complex Nucleotide Replacement Policy, see [1]

learning techniques, the metrics or kernels used are often application or class-of-problem specific – more testing in this area is needed and enhancement in predicting CRFs is probable.

3. Methodologies

3.1 Nucleotide Replacement Policy - Alg. 1

The reference sequences used to construct the top strings T often contain questionable nucleotides. Frequently these nucleotides are ignored, as in Wu *et al.* However, by ignoring these nucleotides, it is possible that important strings or patterns could be lost. Algorithm 1 focuses on replacing these questionable nucleotides as seen in Figure 1, based on internationally agreed standards outlined in [1]. During string generation, when one of these questionable nucleotides is seen, it is replaced with two possible occurrences. Most importantly, because we are not incrementing the occurrence of substrings for the newly generated strings, the probability calculations are still accurate.

3.2 Complex Nucleotide Replacement Policy - Alg. 2

The reference sequences used to construct the top strings T often contain complex questionable nucleotides. These are nucleotides that have > 2 possible replacements. Likewise, we are never incrementing subpatterns of the newly formed strings so the probability distributions are still accurate. The replacement policy used can be seen in Figure 2, which are also based on internationally agreed standards [1]. For testing purposes, algorithm 2 also uses the simple replacement policy seen in the previous section.

Reference Distribution		
6	`subtype A`	4 A1 and 2 A2
4	`subtype B`	4 B1
4	`subtype C`	4 C1
3	`subtype D`	4 D1
8	`subtype F`	4 F1 and 4 F2
3	`subtype H`	3 H1
3	`subtype G`	3 G1
2	`subtype K`	2 K1
3	`subtype N`	3 N1
2	`subtype J`	2 J1
4	`subtype O`	4 O1

Fig. 3: Pure subtype distribution in 42 reference sequence database

3.3 Reference Profiles - Alg. 3

Creating the top strings T has a small disadvantage to strings or patterns seen in the same subtypes. For instance, say a string was seen in four pure subtype reference sequences. We would like a way to emphasize this occurrence, rather than the marginal increment it would get using the standard *relative entropy* calculation. In the simplest form, we combined the reference sequences into pure subtype profiles. In all, 13 reference profiles were constructed, representing 13 pure subtypes. This provided an increased *relative entropy* score for regularly seen strings/patterns in the same subtype. Reference profiles use both simple and complex nucleotide replacement policies as described in the previous sections.

4. Pure Subtype and CRF Databases

Although many techniques use simulated data, we believe using actual data is more realistic regarding the natural diversity found in HIV-1, in terms of recombination and pure subtype reference sequences. With this consideration in mind, we focus testing entirely on the datasets used in [11]. This makes comparison between algorithms easier and prior results from [11] can be examined directly. Lastly, the generation of good testing data is difficult to achieve. The issues surrounding data acquisition are mainly the complex nature of naturally occurring recombinant forms and how to simulate them. For instance, there is often multiple breakpoints in a strain and non-reciprocal exchange [7], [8], [9], which is very hard to reproduce. Therefore, we focus on test data previously classified and internationally used for recombinant form classification, mainly those found in [11].

4.1 Reference Pure Subtype Sequences

42 pure subtype reference sequences are used to construct the top ranked strings. The distribution of the 42 reference sequences can be seen in Figure 3.

Test CRF Distribution	
52	subtype A1 and G1
3	subtype A1 and B1
3	subtype D1 and F1
11	subtype B1 and C1
3	subtype C1 and D1
10	subtype B1 and F1
7	subtype B1 and G1
2	subtype A2 and D1

Fig. 4: 91 unique test sequences and respective compositions

4.2 CRF Test Sequences

91 deterministic CRF test sequences are used. These test sequences are well-documented and the respective pure subtypes are well-defined and accepted. The distribution of the 91 test sequences can be seen in Figure 4. Most importantly, not all pure subtypes are seen in the 91 test sequences; however, all pure subtypes are used during the training stage of the algorithm. Better results can be obtained by narrowing the training stage to only those reference sequences that are present in the CRF test sequences. However, the goal of the research is to construct a method to reliably predict recombinant forms (pure subtypes that define the CRF) from test sequences where there is no knowledge of the phylogeny of the sequence. Therefore, all pure subtype reference sequences are always used regardless of the specifics that may be known about the test data. Lastly, throughout all testing, the knowledge of what pure subtypes make up a given CRF is never used, only during verification of the prediction.

5. Results

Overall, some notably important results were obtained. Chiefly, a quicker runtime was realized, a limit has been found for top string count, and better prediction accuracy in terms of both *relative* and *absolute* accuracy for all algorithms was achieved.

5.1 Limits on number of Top Strings

Figures 5 and 6 clearly show that, as T grows from 0 to 5000, prediction accuracy steadily improves. As T grows larger than 5000 one can see accuracy, conversely, drops. These results counter suggestions in [11] that the greater the size of T the greater the knowledge contained in T.

5.2 *Relative* prediction accuracy

Previous results from [11] show 87-percent prediction accuracy and using [10] NCBI with a sliding window and BLAST comparative scores, obtained 77-percent prediction accuracy. Simple nucleotide replacement policy resulted in 88-percent prediction accuracy and complex nucleotide replacement policy resulted in 90-percent prediction accuracy. These results are rather impressive on their own and should

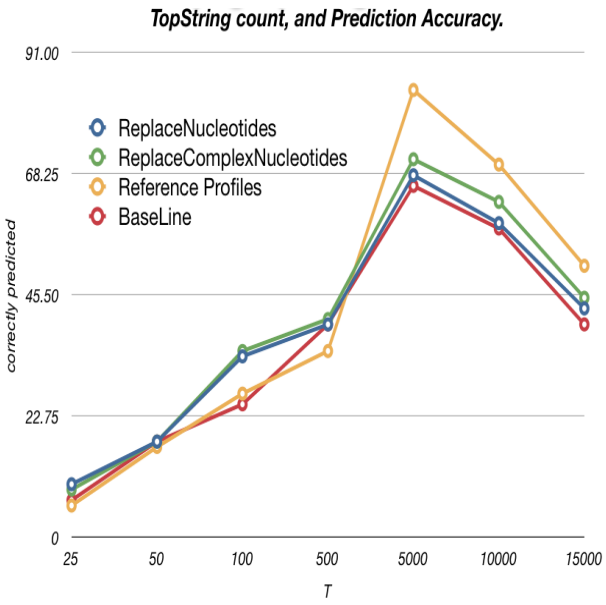

Fig. 5: Accuracy improves up to T $\approx$ 5000, decreases steadily as T > 5000.

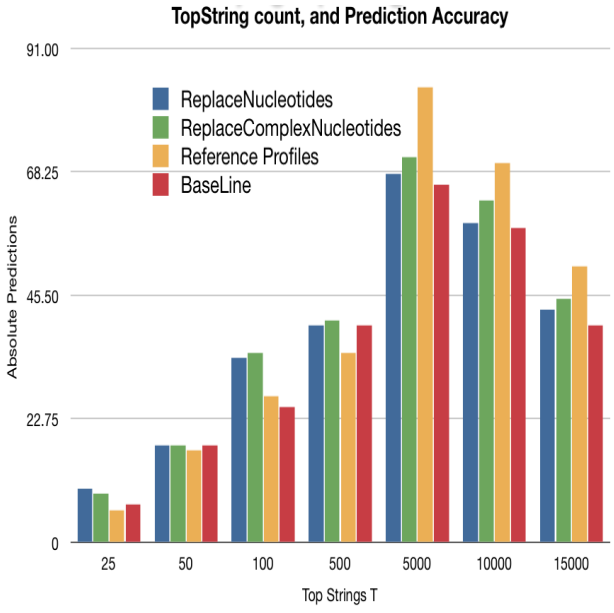

Fig. 6: Accuracy improves up to T $\approx$ 5000, decreases steadily as T > 5000.

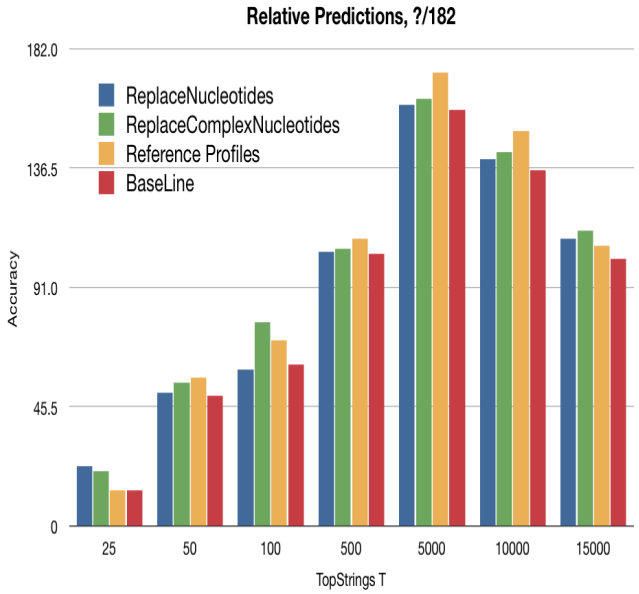

Fig. 7: *Relative* prediction accuracy in terms of 182 subtypes

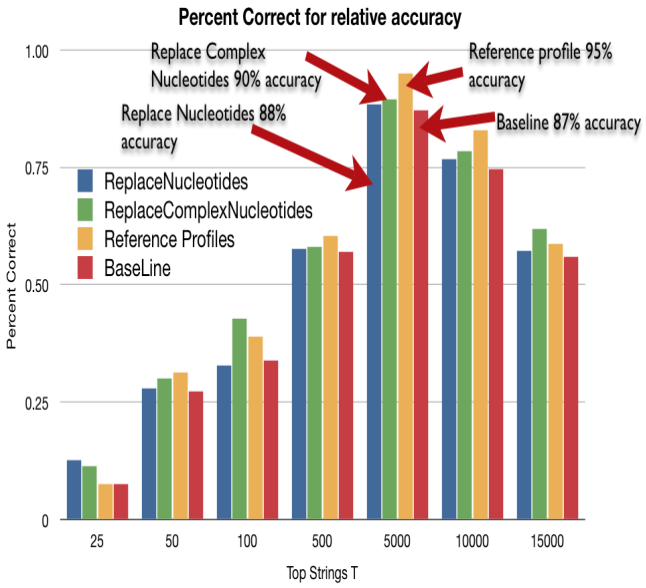

Fig. 8: *Relative* prediction accuracy percent correct.

not be overshadowed by the results from the third algorithm reference profiles. Obviously, there is knowledge contained in these areas of questionable nucleotides, the fact the top strings changed dramatically depending on the use of simple and complex nucleotide replacement policies show us this. However, the third algorithm shows extraordinary results, achieving 95-percent accuracy, see Figure 8. This is likely because when the *relative entropy* is calculated for a string, the strings are given a slight boost because they are are seen in the foreground distribution more than the background. The boost is only slight, but works well experimentally.

5.3 *Absolute* prediction accuracy

Absolute prediction accuracy is an important metric because it not only tells us how many pure subtypes we predicted correctly, but it also reports many correctly predicted pairs were obtained. This is ultimately the goal: predict the makeup of a CRF with high precision. Previous algorithms demonstrated only marginal accuracy, as in Wu et al., where even our simple and complex nucleotide replacement policies show a respectable gain in terms of *relative* accuracy, fare much better in terms of *absolute* accuracy. For instance, looking at Figures 9 and 10, we see the simple nucleotide replacement policy predicts three more pairs correctly, and complex nucleotide replacement predicts five more pairs correctly, when compared to the baseline algorithm that predicts only 66/91. These results show clearly, like that shown in *relative* prediction accuracy, that information is gained when using the replacement policies. This information results in new strings in our top strings list that were never available previously. *Absolute* prediction accuracy was never included

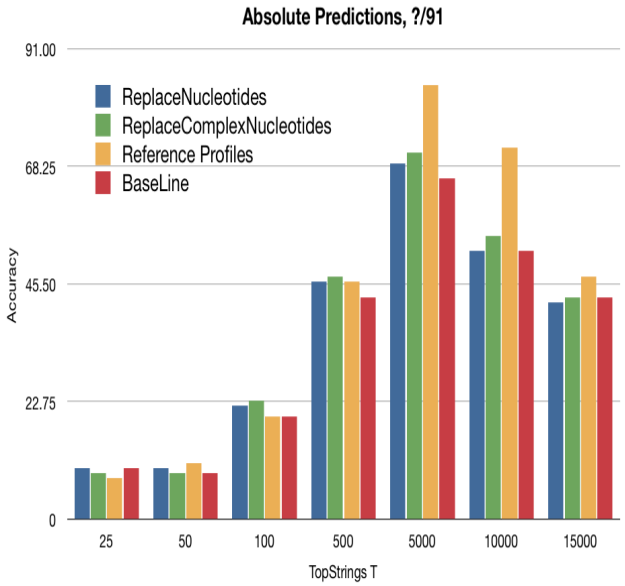

Fig. 9: *Absolute* prediction accuracy in terms of pairs correctly labelled.

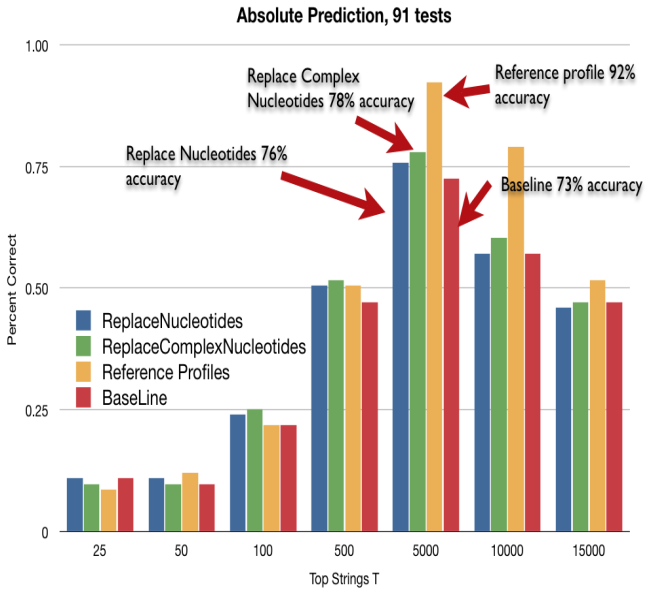

Fig. 10: *Absolute* prediction accuracy percent correct.

in [11] for NCBI tool and, in turn, are not included.

Reference profiles show a clear advantage and rather impressive results, with 84/91 correctly predicted pairs and 92-percent overall *absolute* prediction accuracy. Clearly this technique works well on the 91 deterministic CRF database.

5.4 Considerations

These results show our techniques perform well under the 42 reference sequences used and the 91 deterministic CRFs; however, there really is not much deterministic CRF data available. Even the database of 91 CRFs likely has some error and that could be in our benefit or not. In the future we hope to obtain more reliable datasets, not simulated data, but real-world CRFs that have been carefully examined to define the composition of pure subtypes. We hope the real-world data will further validate our method. Likewise, it is interesting to see that there is an upper bound to $T \approx 5000$ and counters previous thoughts that more strings would contain more information [11]. Some other results that were not included in this paper are, chiefly, the results of $T > 5000$ can be improved marginally if we restrict the length-k of strings to ≈ 14. However, this only resulted in a small improvement and therefore, was not formally displayed. One can infer that shorter length strings are more important in classification, even though longer strings often can show more information.

The ability to calculate the composition of the test sequence is very important, and because we are not limited to only two results per test sequence we can easily give composition based certainty that our algorithm uses for classification. We have seen many examples that show five or more base type ancestry. There is also the ability to do inter-clade analysis after the initial classification is done using the same algorithm. This is very important because one often wishes to know the composition outside of the reference profiles that were created by joining pure subtypes.

Lastly, all results are available online, see [2]. We invite any and all suggestions and also look forward to testing other research groups' data, whether HIV-1 specific or not.

Conclusion and FutureWork

[11] provides a novel starting point based on a general machine learning framework used in bioinformatics. We have shown a substantial increase in terms of both *relative* and *absolute* prediction accuracy in all of our algorithms. The goal of our research is to build a tool that gives high certainty results concerning the makeup of a CRF. These results can lead to more accurate HIV-1 phylogeny and the development of widely applicable treatments that are more adaptive to recombinant forms of HIV-1. Further testing is needed to validate the results in this paper, we will continue to refine our algorithm and as more deterministic and reliable data becomes available we hope to have a sound method for detection of recombination and classification or pure subtypes in the sequence.

References

[1] J. Biochem. Nomenclature for incompletely specified bases in nucleic acid sequences. *Biochem Journal*, 229:281Ð286, 1985.

[2] S. Eliuk. http://www.cs.ualberta.ca/ eliuk/recombinationresults/.

[3] S. Eliuk, P. Boulanger, and K. Kabin. Sunviz: A real-time visualization environment for space physics applications. In *ISVC '08: Proceedings of the 4th International Symposium on Advances in Visual Computing, Part II*, pages 1–11, Berlin, Heidelberg, 2008. Springer-Verlag.

[4] D. P. Martin, C. Williamson, and D. Posada. Rdp2: recombination detection and analysis from sequence alignments. *Bioinformatics*, 21(2):260–262, 2005.

[5] I. Milne, F. Wright, G. Rowe, D. Marshall, D. Husmeier, and G. McGuire. Topali: software for automatic identification of recombinant sequences within dna multiple alignments. *PubMed*, 20(11):1806–7, 2004.

[6] E Myers, S Altschul, W Gish, D Lipman, and W Miller. http://blast.ncbi.nlm.nih.gov/blast.cgi.

[7] D. Posada. Evaluation of methods for detecting recombination from dna sequences: Empirical data. *Mol Biol Evol*, 19(5):708–717, 2002.

[8] D. Posada and KA. Crandall. The effect of recombination on the accuracy of phylogeny estimation. *Journal of Molecular Evolution*, 54(3):396–402, 2002.

[9] D. Posada and KA. Crandall. Evaluation of methods for detecting recombination from dna sequences: Computer simulations. *Proc Natl Acad Sci USA*, 98(24):13757–62, 2002.

[10] Mikhail Rozanov, Uwe Plikat, Colombe Chappey, Andrey Kochergin, and Tatiana A. Tatusova. A web-based genotyping resource for viral sequences. *Nucleic Acids Research*, 32(Web-Server-Issue):654–659, 2004.

[11] Xiaomeng Wu, Zhipeng Cai, Xiu-Feng Wan, Tin Hoang, Randy Goebel, and Guohui Lin. Nucleotide composition string selection in hiv-1 subtyping using whole genomes. *Bioinformatics*, 23(14):1744–1752, 2007.

[12] Xiaomeng Wu, Xiu feng Wan, Gang Wu, Dong Xu, and Guohui Lin. Whole genome phylogeny via complete composition vectors, 2004.