CSA-X: Modularized Constrained Multiple Sequence Alignment

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
T.M. Rezwanul Islam

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Abstract

Imposing additional constraints on multiple sequence alignment (MSA) algorithms can often produce more biologically meaningful alignments. Hence, various constrained multiple sequence alignment (CMSA) algorithms have been developed in the literature, where researchers used anchor points, regular expressions, or context-free-grammars to specify the constraints, wherein alignments produced are forced to align around segments that match the constraints.

In this thesis, we propose CSA-X, a modularized program of constrained multiple sequence alignment that accepts constraints in the form of regular expressions. It uses an arbitrary underlying multiple sequence alignment program to generate alignments, and is therefore modular. The name CSA-X refers to our proposed program generally, where the letter X is substituted with the name of a (non-constrained) multiple sequence alignment algorithm which is used as underlying MSA engine in the proposed program. We compare the accuracy of our program with another constrained multiple sequence alignment program called RE-MuSiC that similarly uses regular expressions for constraints. In addition, comparisons are also made to the underlying MSA programs (without constraints).

The BAliBASE 3.0 benchmark database is used to assess the performance of the proposed program CSA-X, other MSA programs, and CMSA programs considered in this study. Based on the results presented herein, CSA-X outperforms RE-MuSiC, and scores well against the underlying alignment programs. It also shows that the use of regular expression constraints, if chosen well, created from the least conserved region of the correct alignments, improves the alignment accuracy. In this study, ProbCons and T-Coffee are used as the underlying MSA programs in CSA-X, and the accuracy of the alignments are measured in terms of Q score and TC score. On average, CSA-X used with constraints identified from the least conserved regions of the correct alignments achieves results that are 17.65% more for Q score, and 23.7% more for TC score compared to RE-MuSiC. In fact, CSA-X with ProbCons (CSA-PC) achieves a higher score in over 97.9% of the cases for Q score, and over 96.4% of the cases for TC score. In addition, CSA-X with T-Coffee (CSA-TCOF) achieves a higher score in over 97.7% of the cases for Q score, and over 94.8% of the cases for TC score. Furthermore, CSA-X with regular expressions created from the least conserved regions of the correct alignments achieves higher accuracy scores compared to standalone ProbCons and T-Coffee. To measure the statistical significance of CSA-X results, the Wilcoxon rank-sum test and Wilcoxon signed-rank test are performed, and these tests show that CSA-X results for the least conserved regular expression constraint sets from the correct BAliBASE 3.0 alignments are significantly different than those from RE-MuSiC, ProbCons, and T-Coffee.

Acknowledgements

At first, I would like to thank **Allah**, the most gracious, the most merciful. He has blessed me with lots of bounties that I am unable to count. One of them is being a student in the University of Saskatchewan under the supervision of Dr. Ian McQuillan. I am extremely thankful to Dr. McQuillan who basically introduced me to the vast dominion of research, showed constant patience in finding a suitable research question to answer, and developing a solution for the question in an iterative fashion, which takes lots of modifications and revisions.

I am thankful to Dr. Tony Kusalik, who brings a wonderful smile each day, and is also a great researcher. I highly appreciate your valuable comments and feedback on my thesis.

I would like to thank all the members of my advisory committee Dr. Kusalik, Dr. Keil, and my external adviser, Dr. FangXiang Wu.

I also want to thank all the fellow members in the Bioinformatics lab: Dr. Brett Trost (Yes Brett, you have defended your PhD. successfully, now we can call you with such a title!), Stephen Johnson, Lingling Jin, Sowgat Ibne Mahmud, Farhad Maleki, Nazifa Azam Khan, Kimberly MacKay, Katie Ovens, Daniel Hogan, and Mathew Shannon.

Special thanks to Gwen Lancaster for everything you do for us — the graduate students.

I would like to thank the Department of Compute Science, University of Saskatchewan for providing me with funding through the DCSS program.

Finally, I would like to thank all of my teachers, my friends, my family, my uncles, my aunts, my cousins and of course especially, Maniza Abedin Chowdhury!

Contents

List of Tables

List of Figures

List of Abbreviations

BAliBASE	Benchmark Alignment dataBASE
BLAST	Basic Local Alignment Search Tool
BLOSUM	Block Substitution Matrix
CDD	Conserved Domain Database
CFG	Context Free Grammar
CMSA	Constrained Multiple Sequence Alignment
COBALT	Constraint Based Alignment Tool
CS	Column Score
DNA	Deoxyribonucleic Acid
GST	Glutathione S-transferase
mRNA	Messenger Ribonucleic Acid
MSA	Multiple Sequence Alignment
MUSCLE	Multiple Sequence Comparison by Log-Expectation
PAM	Accepted Point Mutation
PDGF	Platelet-Derived Growth Factor
Q	Quality
RE-MuSiC	Multiple Sequence Alignment with Regular Expression Constraints
RECSA	Regular Expression Constrained Sequence Alignment
RNA	Ribonucleic Acid
rRNA	Ribosomal Ribonucleic Acid
SARS	Severe Acute Respiratory Syndrome
SH3	SRC homology 3
SP	Sum-of-Pairs
T-Coffee	Tree-based Consistency Objective Function for alignment Evaluation
TC	Total Column
tRNA	Transfer Ribonucleic Acid

Chapter 1

Introduction

Multiple sequence alignment (MSA) is a fundamental tool for phylogenetic studies, computational biology, prediction of functional residues, and prediction of secondary structure of proteins [39]. A large number of programs have been developed for multiple sequence alignment. Pais et al. [38] recently surveyed different MSA programs in terms of accuracy and computational time. Most of the state-of-the-art MSA programs such as ProbCons [15], T-Coffee [37], MAFFT [27] and ClustalW [51] are fully automated, and they allow the user to adjust a limited number of parameters. So these programs are used without explicitly taking into account any additional information regarding the sequences, such as functional or structural annotations. But often expert users have information regarding e.g., active site residues, intramolecular disulphide bonds, enzyme activities, and conserved motifs [50]. Hence, having a program that can use additional information, either manually entered, or created automatically from additional annotations, can improve the accuracy of alignments.

Constrained multiple sequence alignment (CMSA) [49] is a generalization of the MSA problem [7] that allows expert users to use knowledge regarding the sequences involved in the form of constraints, with a view to achieving more biologically meaningful alignments. For example, Du and Lin [16] showed that ClustalW [51] does not align common patterns and similar structures found in sequences consistently. Because of this, Tsai et al. [50] proposed MuSiC, a web server, that allows constrained alignment of sequences. But many biologically important motifs such as those listed as regular expressions in the PROSITE [25] database cannot be formulated as constraints according to the convention followed by MuSiC [8]. To solve this issue, Arslan [5] and Chung et al. [9] introduced alignment algorithms that accept regular expression constraints. Then, Chung et al. proposed RE-MuSiC [8], an extension to their previous work [9], to support multiple sequences and multiple constraints. In that work, they used sequence motifs found in PROSITE as regular expression constraints to improve the quality of alignments. However, there are some limitations of RE-MuSiC as well, as it does not allow the use of certain quantification operators such as Kleene star (*) and Kleene plus (+) in regular expression constraints. Thus, only a subset of regular expressions can be used as input, plus some other limitations are discussed in Section 5.1. Arslan [6] also proposed sequence alignment programs guided by Context Free Grammars (CFG), but this is only limited to pairwise sequence alignment. Moreover, Morgenstern et al. [35] developed DIALIGN, a web server, that can accept user defined anchor points as constraints. But this approach has some drawbacks; for example, user specified anchor points may conflict with one another. To avoid such a conflict, DIALIGN allows users to prioritize anchor points and then it aligns all anchor points that are consistent with each other.

Studies done by Notredame et al. [37], Katoh et al. [27], Edgar [17] used benchmark databases to assess the accuracy of their MSA programs. For RE-MuSiC, such a comparison is not available. As suggested by Kumar and Alan [28], less accurate MSA results cause negative effects on complex modular program pipelines that use MSA program as an underlying tool. Hence, it is important to select the best MSA program available in terms of accuracy.

Since typical MSA programs do not accept constraints, this study proposes a new algorithm and application program for CMSA. The proposed program, CSA-X, accepts regular expression constraints and creates a MSA that forces sections that match the regular expression to align. Furthermore, it is possible to indicate with an extended regular expression syntax that certain sections of the sequences that match corresponding sections of the regular expression be forced to align. Thus, it is possible to force characters matching the same section of a regular expression to align. CSA-X is a modularized program that uses an underlying MSA program, and because of this reason, it is possible to replace the underlying MSA program with another; perhaps, an improved program. In addition, this study compares the performance of CSA-X, RE-MuSiC and X, where X is the underlying MSA program in the proposed tool, with respect to the BAliBASE 3.0 [52] benchmark database, and assesses their accuracy in terms of Q score and TC score — two metrics developed to assess the quality of multiple sequence alignments [17] — and measures statistical significance of the results.

In short, this study introduces an extensible modularized tool, CSA-X, that creates more accurate constrained multiple sequence alignments compared to the other CMSA algorithm implementation that uses regular expression constraints. Furthermore, it also shows that if constraints are chosen appropriately, such as from the least conserved region of the correct alignments, CSA-X gives better results than the underlying MSA programs.

Layout of this Thesis: Chapter 2 of this thesis introduces the background study, Chapter 3 provides the goals and objectives of the research, Chapter 4 describes the CSA-X algorithm, Chapter 5 discusses the tools used to assess CSA-X and other programs considered in this study, Chapter 6 compares CSA-X performance to RE-MuSiC and other underlying MSA programs used in CSA-X. Lastly, Chapter 7 presents the concluding remarks and future research directions.

Chapter 2

Background

2.1 Biological Sequences

There are two types of biological sequences that are of particular interest in this thesis — nucleic acid sequences and protein sequences.

2.1.1 Nucleic Acid Sequences

A nucleic acid sequence is a chain of nucleotides, where each nucleotide contains a nitrogenous base, a phosphate group and a sugar. In nucleotides, mainly, five nitrogenous bases (containing at least one nitrogen atom, and showing the chemical properties of the base) are found. They are adenine (A), guanine (G), thymine (T), cytosine (C), and uracil (U). Figure 2.1 depicts the chemical composition of these bases.

The sugar in a nucleotide is a 5-carbon carbohydrate, either ribose or deoxyribose sugar. A deoxyribose sugar lacks one oxygen atom, hence its name contains the term *deoxy*. Figure 2.2 illustrates the chemical composition for ribose and deoxyribose sugar. If the sugar in a nucleotide is ribose, it is known as a ribonucleotide, and if the sugar is deoxyribose, then it is known as a deoxyribonucleotide.

Deoxyribonucleic acid (DNA) is a sequence of deoxyribonucleotides, while ribonucleic acid (RNA) is a sequence of ribonucleotides. Such sequences are usually represented with sequences of the letters A, G, T, C for DNA, and A, G, U, C for RNA.

DNA stores the genetic information needed for cellular growth, multiplication, and function in all living organisms. According to the double helical DNA model, DNA contains two single stranded chains of nucleotides, where each nucleotide base forms a hydrogen bond with a complementary base on the other strand. There is directionality to single-stranded DNA, where one end is known as the $5'$ end and the other is the $3'$ end. Unlike DNA, RNA is often a single stranded sequence of nucleotides. Different types of RNAs exist in cells such as messenger RNA (mRNA), transfer RNA (tRNA), ribosomal RNA (rRNA), and other small RNAs [41]. Figure 2.3 shows an example of the double stranded and single stranded structure of DNA and RNA respectively.

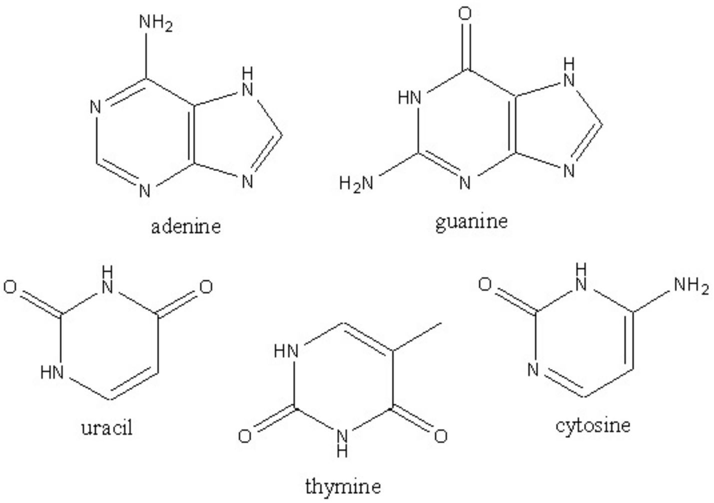


Figure 2.1: Nitrogenous bases found in nucleotides. Picture taken from `http://upload.wikimedia.org/wikipedia/commons/b/bf/Nitrogenous_bases.jpg`.

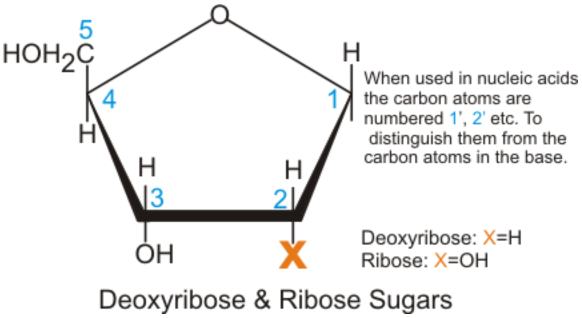

When used in nucleic acids the carbon atoms are numbered 1', 2' etc. To distinguish them from the carbon atoms in the base.
Deoxyribose: X=H
Ribose: X=OH
Deoxyribose & Ribose Sugars

Figure 2.2: Deoxyribose and ribose sugars. Picture taken from `http://upload.wikimedia.org/wikipedia/commons/d/d1/Ribose_deoxyribose.png`.

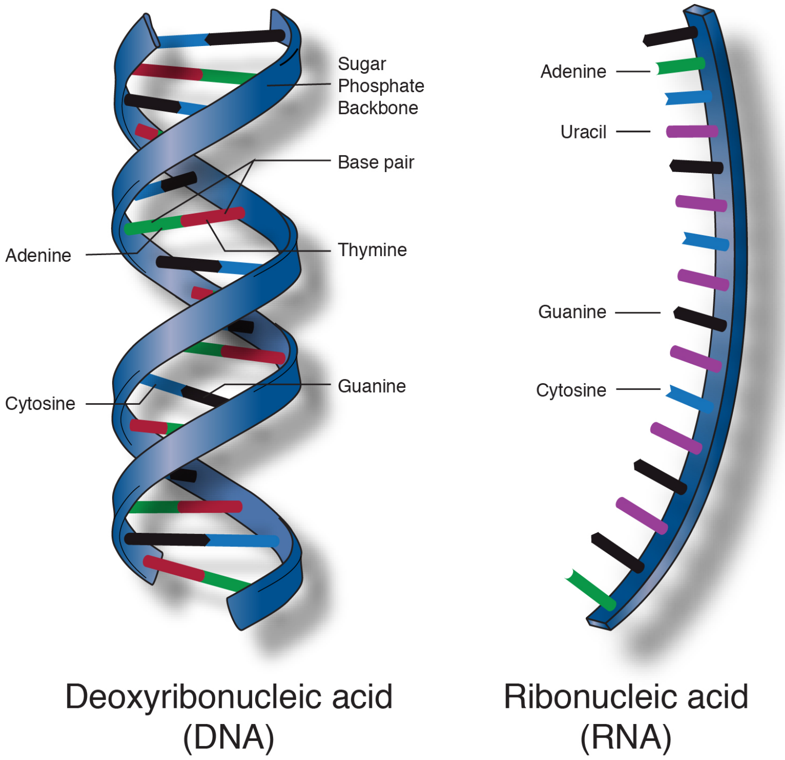


Figure 2.3: DNA and RNA. Image credit: Darryl Leja, NHGRI, `http://www.genome.gov/dmd/previews/85209_large.jpg`.

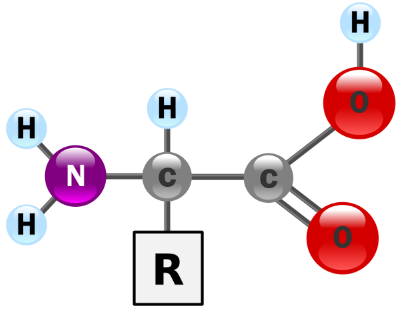


Figure 2.4: General structure of amino acid. Picture taken from `http://upload.wikimedia.org/wikipedia/commons/c/ce/AminoAcidball.svg`.

2.1.2 Protein Sequences

Amino acids are molecules that contain an amine (-NH2), a carboxyl group (-COOH) and a variable side chain, denoted by R. Figure 2.4 represents a general structure of amino acid. Mainly 20 amino acids are found in proteins. Table 2.1 contains different amino acid names and their representing 3 letter and 1 letter codes. Proteins are chains of amino acid residues linked with peptide bonds that perform different tasks in cells by acting as enzymes to catalyze reactions, serving in a structural role or co-operating in transport of materials within and between cells [41]. The sequence of amino acids in a protein is known as the primary structure. For example, the primary structure of the globin protein in *Sabella spallanzanii* is represented as (using the one letter codes):

```
MFRFALLCAFVADASAEGCCSMEDRQEVLNAWEALWSAEYTGRRVMIAQAAFQKLFEKAPDSKALFTRVNVDNIGSPQFR
AHCIRVTNGFDTIINMAFDTDVLEELLTHLGNQHTKYQGMRAAYLTHFRESFAEILPQAIPCFNTAAWNRCITAMQDKIG
ASLAA
```

2.2 Obtaining Biological Sequences

The process of experimentally determining biological sequences is known as sequencing. Then protein sequencing involves determining the sequence of amino acids in proteins, which is an essential tool to study the structures and functions of proteins in living organisms. There are mainly three techniques to identify proteins: direct protein sequencing [43], gel electrophoresis [20], and mass spectrometry [13]. Likewise, DNA sequencing refers to the process of identifying the order of nucleotides present in DNA. There are different techniques for gene sequencing such as Sanger sequencing, and more recently next-generation sequencing. Mardis [33] presented a detailed discussion about next generation DNA sequencing techniques.

2.3 Some Biological Events

2.3.1 DNA Replication and Mutation

Cell division is an important biological process that causes growth in living organisms. Usually each cell replicates its DNA before cell division. DNA replication refers to the process of creating two identical DNAs from a double-stranded single DNA. This process is quite accurate, and through this process the same sequence of nucleotides gets replicated. As an example, in the case of human cell division, DNA replicates and passes the same sequence of 3 billion nucleotides to the newly created cell [42]. But sometimes errors occur in the process, such as if a polymerase enzyme inserts a wrong nucleotide, or too many or too few nucleotides. Most of these errors are corrected by different repairing processes. But some replication errors get through the repairing process and generates a different sequence. This process is known as mutation and because of this process, altered sequences get copied in the next generation through cell division.

Amino Acid Names	**3-Letter Code**	**1-Letter Code**
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartic acid	Asp	D
Cysteine	Cys	C
Glutamic acid	Glu	E
Glutamine	Gln	Q
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanine	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

Table 2.1: 20 standard amino acid names with 3 letter and 1 letter codes.

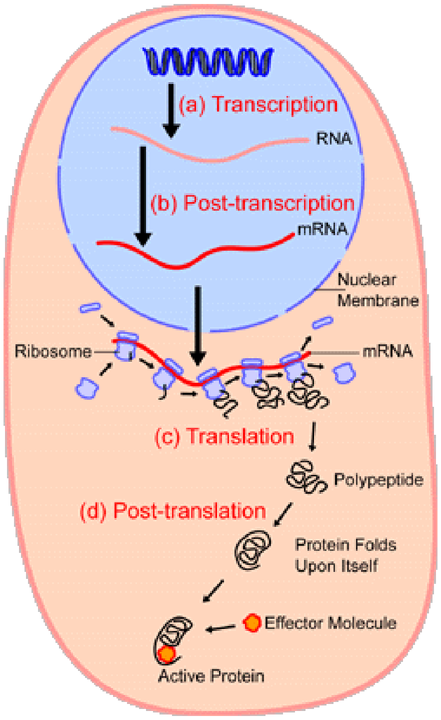


Figure 2.5: Process of synthesizing proteins. Picture taken from `http://upload.wikimedia.org/wikipedia/commons/0/09/Proteinsynthesis.png`.

2.3.2 Protein Synthesis

Protein synthesis is the process of creating protein sequences by using the *blueprint* stored in DNA. This process includes two main phases — transcription [10] and translation [11]. In the transcription phase, RNA is synthesized from an unwound DNA template. This is a complex process, where an RNA polymerase enzyme catalyzes the process of creating an RNA from a DNA template. This newly created RNA strand is known as messenger RNA (mRNA), which directs the synthesis of the protein chain. After that, a post-transcription phase is initiated, and the mRNA moves from the cell's nucleus to the cytoplasm. In the cytoplasm, the translation phase is sometimes initiated, where the mRNA binds with ribosomes, the sites of protein synthesis. Through another complicated procedure, proteins are created from RNA. This translation process is guided by the actions of transfer RNAs, ribosomes, and many soluble proteins. Figure 2.5 illustrates the different phases involved in production of proteins in cells. Finally, in the post-translation phase, the protein folds, and binds with a small molecule known as effector molecule, which can involve chemical modifications altering its properties.

2.4 Sequence Alignment

Sequence alignment is a well-known technique to compare biological sequences, where the involved sequences are written in such a manner that conserved residues (portion of the sequences that remained unchanged during evolution) or conserved nucleotides appear in the same column. Consider n sequences, $S_1, S_2, \ldots, S_n$, and let '-' be a new symbol (the gap symbol). Then a sequence alignment of $S_1, S_2, \ldots, S_n$ is a matrix with n rows, where the columns in row i contains S_i with the gap symbol inserted into it arbitrarily many times. As an example, a sequence alignment of two nucleotide sequences $S_1 =$ `TAGCCGCACGTATTAAC` and $S_2 =$ `TAGTCGTATAATCACGTATTACC` is represented as follows:

```
TAGCCG------CACGTATTAAC
TAGTCGTATAATCACGTATTACC
```

A match occurs in a column between two sequences if the two characters of those sequences are identical. A mismatch occurs if two characters are different. A gap occurs if a character is aligned with a gap. No column can contain gap symbols only. If two sequences are involved in alignment, then it is known as pairwise sequence alignment. If more than two sequences are involved then it is referred to as multiple sequence alignment (MSA). Section 2.5 presents a detailed discussion on different pairwise and MSA techniques.

2.4.1 Necessity of Sequence Alignment

Comparing gene sequences or protein sequences provides information on relatedness. If two sequences are related, This means they evolved from a common ancestor, and they are known as homologs. Relatedness among sequences suggests possible common function, and it also allows for identification of common regions. These common regions provide important information regarding motifs (short subsequence of amino acids that perform specific functions) that are conserved throughout evolution. In fact, sequence alignment is a fundamental tool for predicting protein structure, inferring the function of a novel gene or protein, identifying protein domains, and constructing phylogenetic trees. Then, high amounts of similarity present in a sequence alignment can provide evidence that sequences are homologous.

Structure Prediction

Protein structure modelling of a target protein (a protein whose structure is unknown) based on the sequence similarity to proteins of a known structure is a well-known method for determining protein structures. This technique is known as comparative protein structure modelling. 3D structures of proteins in a family is more conserved compared to amino acid sequences of them [31]. Hence, if a target sequence shows sequence similarity with other sequences of known structure, then the structure of the unknown sequence can be inferred. Lots of protein structure prediction tools were developed based on a comparative modelling strategy, and a detailed discussion about them can be found in [34].

Function Prediction

The function of an unknown gene or protein sequence can be inferred by comparing it with other known sequences. As an example, in 1984, scientists compared cancer causing ν-sis oncogene with other known

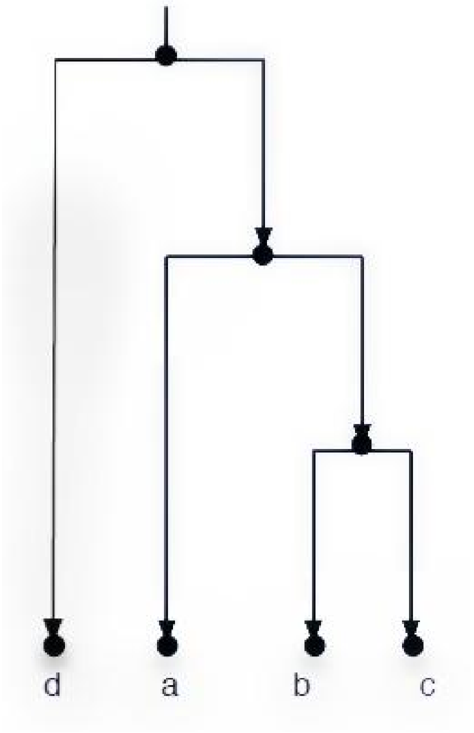


Figure 2.6: Phylogenetic tree where a, b, c, d represent different organisms.

sequences at that time, and they found similarity with platelet-derived growth factor (PDGF), which is responsible for cell growth. From that, scientists predicted cancer can be caused because of a normal gene becoming activated at the wrong time [26].

Domain Identification

A protein domain refers to a structural and functional unit in protein sequences. As an example, the SRC homology 3 domain (SH3) is responsible for protein-protein interactions. Construction of protein profiles provides information on how likely a residue is to appear at a certain position in sequences. It allows for the identification of conserved functional domains in sequences. To construct protein profiles, an alignment among the sequences is often used.

Phylogeny

Phylogeny is the study of evolutionary relationships among taxa, individuals or molecules. Usually this relationship is presented in the form of a tree where each leaf represents a taxon, or a specific molecule. To construct a phylogenetic tree using distance based approaches, a sequence alignment is performed among sequences. Then an inferred tree is constructed based on sequence information [41]. Figure 2.6 shows a phylogenetic tree. ClustalW [51] is an example of a well-known suite with a tool to construct phylogenetic trees.

2.5 Sequence Alignment Techniques

2.5.1 Pairwise Sequence Alignment

A pairwise sequence alignment is the alignment of two sequences. Sequences can be aligned globally or locally. A global alignment is an alignment of sequences across their entire lengths. In contrast, a local alignment is a global alignment of any substring (part of a string where the order of each character is preserved compared to the original string) of the first sequence with a substring of the second sequence from the involved sequences. As an example, let S_1 = `CCCCGGG` and S_2 = `CTTTCCCC` be two nucleotide sequences. Then a local alignment between them is as follows:

```
CCCC
CCCC
```

On the other hand, a global alignment between S_1 and S_2 is represented as:

```
CCCC-GGG
CTTTCCCC
```

Global Alignment: the Needleman-Wunsch Algorithm

In 1970, Needleman and Wunsch described a global pairwise sequence alignment technique using dynamic programming, which is well-known as the Needleman-Wunsch algorithm [36]. This approach reports the best alignment (using a predefined scoring scheme) out of all possible alignments. Each possible alignment has a score associated, and the algorithm finds out the alignment with the best score. The algorithm has two steps: (1) computing the similarity matrix and (2) finding an optimal alignment.

Let S_1 = `ACATCG` and S_2 = `ATTCA` be two sequences. To generate the alignment between these two sequences, at first, the similarity matrix is calculated as follows:

Computing The Similarity Matrix: The similarity matrix for the sequences S_1 and S_2 is a $(|S_1|+1) \times (|S_2|+1)$ matrix, where $|S_x|$ represents length of the sequence. As the length of the sequences S_1 and S_2 are 6 and 5 respectively in the example above, hence, the similarity matrix for them is a 7×6 matrix. Table 2.2 shows the similarity matrix for the sequences. In order to compute the similarity matrix, a scoring scheme is needed. The simplest scoring scheme considers 3 scenarios corresponding to each position of a hypothetical alignment:

1. match,
2. mismatch,
3. insertion/deletion.

A match occurs when two characters in the same column of an alignment are identical, while a mismatch indicates that two characters do not match. An insertion/deletion (indel) occurs when there is a gap in one sequence, perhaps caused by a character that got inserted or deleted during evolution. In the simplest scoring scheme, usually, if there is a match a positive score is awarded, known as the match score. For a mismatch, either zero or a negative number is given, called the mismatch score. For a gap, usually a negative score is given, known as the gap penalty. For example, to fill the similarity matrix in Table 2.2, the match score, mismatch score and gap penalty are considered as 1, -1 and -1 respectively. At the beginning, the top left cell in the similarity matrix is filled by zero. Then, the first row and first column of the similarity matrix are filled by the multiples of the gap penalty. Then, every cell in the matrix is filled by computing the score; the score for the cell $M(i, j)$ in the matrix is computed as follows:

$$M(i,j) = max \begin{cases} M(i-1, j-1) + s(x_i, y_i) \\ M(i-1, j) - g \\ M(i, j-1) - g \end{cases} \tag{2.1}$$

where i satisfies $1 \leq i \leq |S_2| + 1$ and j satisfies $1 \leq j \leq |S_1| + 1$, and g represents the gap penalty. Here, $s(x_i, y_i)$ is a function that returns the match score if x_i and y_i are identical, otherwise it returns the mismatch score, where x_i is a character in sequence S_1 and y_i is a character in sequence S_2.

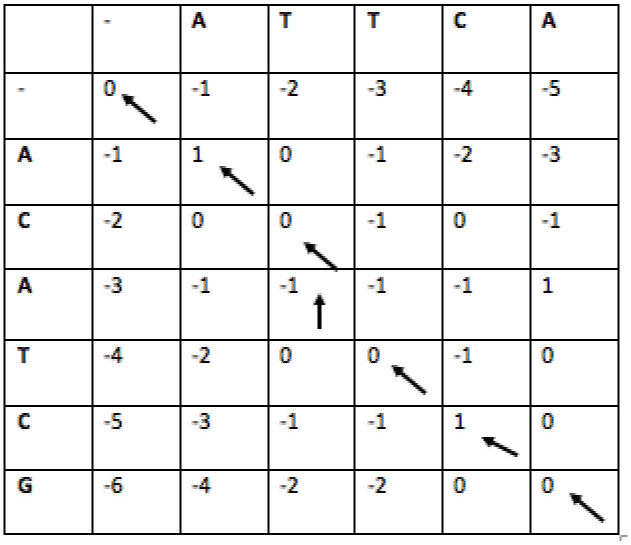
	-	A	T	T	C	A
-	0	-1	-2	-3	-4	-5
A	-1	1	0	-1	-2	-3
C	-2	0	0	-1	0	-1
A	-3	-1	-1	-1	-1	1
T	-4	-2	0	0	-1	0
C	-5	-3	-1	-1	1	0
G	-6	-4	-2	-2	0	0

Table 2.2: Similarity matrix for the sequences S_1 and S_2.

```
AT-TCA

ACATCG
```

Figure 2.7: An optimal alignment for the sequences S_1 and S_2.



	-	A	T	T	C	A
-	0 ↖	-1	-2	-3	-4	-5
A	-1	1 ↖	0	-1	-2	-3
C	-2	0	0 ↖	-1	0	-1
A	-3	-1	-1 ↑	-1	-1	1
T	-4	-2	0	0 ↖	-1	0
C	-5	-3	-1	-1	1 ↖	0
G	-6	-4	-2	-2	0	0 ↖

Figure 2.8: Traceback matrix for the sequences S_1 and S_2.

Identifying an Optimal Alignment: To determine an optimal alignment, a traceback matrix is calculated from the similarity matrix, where (in one possible implementation of the algorithm) arrows (←, ↑ or, ↖) are added in each cell of the similarity matrix to indicate the last position of the best alignment. Figure 2.8 shows certain arrows of the traceback matrix corresponding to an optimal alignment constructed from the similarity matrix in Table 2.2. The traceback matrix is traversed from the bottom-right cell following the arrows to determine an optimal alignment for the sequences S_1 and S_2, which is represented in Figure 2.7.

Local Alignment: the Smith-Waterman Algorithm

Smith and Waterman proposed a local sequence alignment algorithm to identify common molecular substrings [46]. For pairwise local sequence alignment, all possible substrings of the sequences are considered for the alignment, and the algorithm finds the alignment of the substrings that achieves the best score. The Smith-Waterman algorithm computes the similarity matrix in the same manner as the Needleman-Wunsch algorithm with one exception: whenever a computed score for a given cell is less than zero, then zero is entered in the matrix instead of the negative score. To identify an optimal local sequence alignment, the traceback matrix is computed in same manner as in the Needleman-Wunsch algorithm, but instead of traversing the matrix from the lower right corner, it is traversed from the highest value in the matrix. The traceback ends whenever a cell with value zero is encountered.

2.5.2 Multiple Sequence Alignment

In the case of pairwise sequence alignment, two sequences are involved where as for MSA more than two sequences are involved in the process. Both pairwise sequence alignment and MSA provides important biological information, but MSA is the more sensitive method compared to pairwise sequence alignment. Indeed, Park et al. show that sequence comparisons using multiple sequence alignment technique detects almost three times more homologs compared to pairwise methods [40]. Different methods for multiple sequence alignment have been proposed in the literature. Pevsner classifies these approaches into 5 categories [41]:

1. exact methods,
2. progressive alignment,
3. iterative approaches,
4. consistency-based methods,
5. structure-based methods.

Exact Methods

Exact multiple sequence alignment methods determine an optimal alignment out of all possible alignments. In this approach, the pairwise sequence alignment technique proposed by Needleman-Wunsch [36] for global multiple sequence alignment, and Smith-Waterman for local sequence alignment are extended to determine an optimal alignment for the multiple sequences. However, the dimension of the dynamic programming matrix (similarity matrix) in the Needleman-Wunsch algorithm must be expanded depending on the number of sequences. To compute this matrix for the pairwise sequence alignment of two sequences of length n and m, $O(mn)$ time is required. But if the number of sequences rises to N and if all the sequences are of length L, then the time complexity for computing the matrix becomes $O(L^N)$. As a result of such exponential time complexity, heuristic approaches are often used instead of exact alignment techniques.

Progressive Alignment

Feng and Doolittle popularized the idea of using a progressive alignment technique for generating multiple sequence alignment [18]. This technique is incorporated in different MSA programs, and one of them is ClustalW [51]. In ClustalW, at first pairwise alignments between all pairs of sequences are computed. For n sequences in the input dataset, $\frac{n(n-1)}{2}$ pairwise alignments are generated. For every pairwise alignment a distance score is computed, and each of these scores are stored in a distance matrix. Next, this distance matrix is used to generate a guide tree. According to this tree the two most similar (according to the pairwise distances calculated) sequences are aligned at first, and then sequences are added progressively one by one using the guide tree from most to least similar to produce a multiple sequence alignment.

Iterative Approach

In the progressive alignment technique, if an error occurs during the initial alignment, it is not corrected. The iterative alignment approach was introduced to overcome this shortcoming, where an initial potentially suboptimal multiple sequence alignment is generated using the progressive alignment technique, and later this alignment is improved through a series of iterations until a certain criteria is met. Several MSA programs follow this strategy such as MUSCLE [17], MAFFT [27] etc.

Consistency-based Methods

Consistency-based techniques such as ProbCons [15], MAFFT [27] etc. use the iterative approach of multiple sequence alignment. Different MSA programs use different methods to generate pairwise alignments, as an example, ProbCons uses a pair hidden Markov model for this purpose. In the consistency based technique, if a pairwise alignment of sequences X and Z is such that character x_i from the sequence X aligns with character z_k from sequence Z, and also if character y_j from sequence Y aligns with character z_k from sequence Z, then consistency-based techniques imply that x_i and y_j should align. In ProbCons, such additional information is used to "reestimate the match quality scores" [15]. ProbCons computes the guide tree based on the expected accuracy scores of pairwise alignments. Expected accuracy of a pairwise alignment is the ratio of number of correctly aligned letter-pairs to the length of the shorter sequence. Then, ProbCons aligns the sequences following the sequence order in the guide tree. Finally, it enters into the iterative refinement step to produce a more refined multiple sequence alignment.

Structure-based Methods

Tertiary structure of proteins evolve more slowly compared to the primary structure. For example, human beta globin and myoglobin share the same structure, and are considered homologus, but show very little sequence identity [41]. Hence, researchers attempted to improve the quality of alignments integrating structural information regarding the sequences. PRALINE [45] and Expresso [3] are examples of such programs. In Expresso, close homologs of the input sequences are identified, and then their structures are used to guide the multiple sequence alignment process.

2.6 Alignment Scores

2.6.1 Sum-of-Pairs Score

The sum-of-pairs function [30] is a way to calculate the score of a multiple sequence alignment. In this approach, all the columns in an alignment are considered independent. The score is the summation of the score for each column in the alignment. If an alignment of l sequences has a character in the i-th column of the j-th sequence stored in variable C_i^j, then the score for column can be computed as:

$$C_i = \sum_{a=1}^{l} \sum_{b=a+1}^{l} S(C_i^a, C_i^b), \tag{2.2}$$

where S is a function that indicates the score of aligning C_i^a with C_i^b — the two parameterized characters (including the gap symbol); this score is usually obtained from a scoring matrix, which is described in detail in Section 2.6.2. Hence, the sum-of-pairs score, SP, for the alignment A can be written as:

$$SP(A) = \sum_{i=1}^{N} C_i, \tag{2.3}$$

where N is the number of columns in the alignment.

2.6.2 Scoring Matrices

When protein sequences are aligned in the Needleman-Wunsch algorithm, each alignment is assigned a score based on the particular match, mismatch or insertion/deletion of the residues, and the algorithm reports an alignment with the best score. Then, scoring matrices are more general, whereby there is a score for aligning each character with each other character or aligning each character with a gap symbol. It is common to use a matrix where each entry gives a score for substituting one character with another calculated from the frequency with which each character can be substituted with each other character throughout evolution. Different scoring matrices exist for scoring nucleotide sequences and protein sequences.

Scoring matrix for nucleotide sequences: As nucleotide sequences consist of only 4 nucleotides, the scoring scheme is usually simpler. Commonly for nucleotides, positive and negative scores are awarded for matches and mismatches respectively. In BLAST (Basic Local Alignment Search Tool), the match score is $+5$ and the mismatch score is -4 by default.[1].

Scoring matrix for protein sequences: Two commonly used protein scoring matrices are PAM (Accepted Point Mutation) [14] and BLOSUM (Block Substitution Matrix) [22]. In the case of a PAM matrix, each entry in the matrix is calculated from the probability of replacing the corresponding amino acid in the column by the corresponding amino acid in the row over a defined evolutionary interval. Various PAM matrices exist, such as PAM1, PAM100, PAM250 etc. For the PAM1 matrix, the evolutionary interval is one PAM, which is calculated from sequences where 1% of the residues differ. Dayhoff et al. computed the PAM1 matrix based on the alignment of highly similar amino acid sequences, which makes it more suitable for the alignment of closely related sequences. To explore the relatedness of distantly related proteins, variants of the PAM1 matrix such as PAM100, PAM250 also exist, where PAMx is calculated mathematically by extrapolating PAM1 from one PAM time unit to x time units. However, Henikoff, G and Henikoff, J.S. proposed the BLOSUM matrix from the alignment of distantly related proteins. Different variants of BLOSUM matrices exist such as BLOSUM45, BLOSUM50, BLOSUM62, BLOSUM90 etc. High value BLOSUM matrices are suitable for high similarity protein sequences, while low value matrices are suitable for more divergent sequences. Scores in the BLOSUM matrix are derived from observations of the frequencies of substitutions in blocks of local alignments in related proteins. BLOSUM62 matrix was created using sequences sharing no more than 62% identity.

2.7 Benchmark Databases

Different techniques for sequence alignments exist in the literature. To evaluate the efficiency of different multiple sequence alignment techniques, several benchmark databases such as BAliBASE [54], IRMBASE [48], PREFAB [17] etc. have been proposed. These databases contain input datasets for multiple sequence alignment, and the reference alignments for the corresponding input datasets. Usually, these reference alignments are computed using the three-dimensional structure of the sequences created from x-ray crystallography [41]. The alignments produced by MSA programs are compared with the reference alignments in the benchmark databases, and such comparisons are made in terms of different scores proposed by the researchers. Edgar proposed two fundamental ways to assess multiple sequence alignments [17]. They are known as Q (Quality) score and TC (Total Column) score. A detailed discussion about these scores can be found in Section 5.3. Thompson et al. [53] defined the sum-of-pairs score (SPS) and the column score (CS) to assess the quality of multiple sequence alignment (this is referred to simply as the sum-of-pairs score in [53] but we use the terminology of ratio to disambiguate with the sum-of-pairs score of an individual alignment as defined in Section 2.6.1). Suppose, an alignment A contains N sequences and M columns where the i-th column in the alignment is denoted as $A_{i1}, A_{i2}, A_{i3}, \ldots, A_{iN}$. Then the score, S_i, for the i-th column can be computed as follows:

$$S_i = \sum_{j=1, j\neq k}^{N} \sum_{k=1}^{N} P_{ijk}, \tag{2.4}$$

where $P_{ijk} = 1$ if the residues A_{ij} and A_{ik} are aligned in the reference alignment as well; otherwise $P_{ijk} = 0$ is assigned. Now the score for the alignment A is:

$$S_A = \sum_{i=0}^{M} S_i, \tag{2.5}$$

then, the SPS ratio for the entire alignment is:

$$R(A) = \frac{S_A}{S_B}, \tag{2.6}$$

where B is the reference alignment. Further, the column score (CS) for the alignment is defined as follows:

$$CS(A) = \frac{\sum_{i=1}^{M} C_i}{M}, \tag{2.7}$$

where $C_i = 1$ if the i-th column in the alignment is identical to the i-th column in the reference alignment.

2.7.1 BAliBASE

BAliBASE [54] is a manually refined database for testing and comparing multiple sequence alignments. This database is categorized into different groups based on different criteria such as sequence similarity, presence of N/C terminal extensions, large internal insertions etc. The alignments in BAliBASE are manually refined according to the structural superposition. Until May 2015, 3 versions of BAliBASE have been released and the latest version is BAliBASE 3.0 [52]. BAliBASE 3.0 benchmark database is categorized into 5 groups: RV1X, RV20, RV30, RV40, and RV50. RV1X contains two subgroups: RV11 and RV12. RV11 includes divergent sequences having less than 20% identity, RV12 contains medium divergent sequences having 20–40% identity, RV20 contains sequences from same family with greater than 40% identity and also contains orphan sequences (distant members of the family having a common fold but sharing less than 20% identity), RV30 contains sequences from different subfamilies, where members of the same subfamily share more than 40% identity but two members from different subfamilies share less than 20% identity, and RV40 and RV50 contain sequences with large N/C-terminal extensions and internal insertions respectively.

2.7.2 PREFAB

Edgar proposed PREFAB (Protein Reference Alignment Benchmark) [17] for assessing the quality of different protein sequence alignment programs. As of May 2015, 4 versions of this benchmark database have been introduced. The latest version 4.0 contains 1682 input datasets. This database is created using a fully automated protocol from the FSSP database [24]. The protocol is such that at first, two protein sequences are aligned using a structural method; their homologs are identified using PSI-BLAST [2], and only the hits having more than 80% identity are considered. Among these hits for each query sequence, randomly 24 sequences are selected and combined with the queries to form a dataset. Later this dataset is aligned with another MSA program. Next the alignment accuracy for the query sequence pairs are computed by comparing them with their structural alignment from the FSSP database and the structural alignment generated by the program CE aligner [44]. Edgar included only those pairs and their homologs to form a dataset that had a considerable amount of agreement between the aforementioned comparison methods to minimize structural ambiguity.

2.7.3 IRMBASE

According to Subramanian et al. [48] the BAliBASE benchmark database suites contain sequences with high sequence similarity which makes them biased towards global MSA programs. Hence Subramanian et al. proposed IRMBASE (Implanted Rose Motifs Base) [48], a benchmark database for local multiple protein alignment. This database contains a set of artificial random sequences, which are used to create simulated sequences according to a stochastic rule that makes insertions, deletions, and substitutions at random spots. As a result, a simulated phylogenetically related sequence family is obtained from the stochastic process whose correct multiple alignments are known. Then a group of artificially conserved sequence motifs are created and inserted at random spots in the sequences. In this manner, 3 reference sets of artificial sequences are created in the IRMBASE 1.0 benchmark database, called ref1, ref2, and ref3, where sequences in each reference set contain one, two, and three motifs respectively.

2.8 Constrained Multiple Sequence Alignment

Most of the multiple sequence alignment algorithms proposed in the literature — such as T-Coffee [37], ProbCons [15], MUSCLE [17] etc. — are fully automated, and they generate multiple sequence alignments based on a some kind of predefined algorithm. Sometimes, however, expert users or existing databases possess important information such as functional or structural annotations regarding the sequences. But these automated MSA algorithms do not provide any mechanism to integrate expert users' knowledge, or additional information regarding the sequences. Researchers have proposed different tools that accept expert users' or additional information as a constraint on the alignment. MuSiC [50], RE-MuSiC [8], DIALIGN [35] are examples of such programs that allow the user to specify their information as constraints.

2.8.1 RECSA

Aslan [5] proposed an algorithm RECSA (Regular Expression Constrained Sequence Alignment) for pairwise sequence alignment that accepts a single regular expression constraint. RECSA determines an alignment of the two sequences with the maximum alignment score such that the alignment of the involved sequences matches a particular regular expression constraint. Later, Arslan extended his work to support multiple sequences and developed an algorithm for the alignment of multiple sequences which are required to contain a given sequence of regular expression constraints [4]. As far as we know, there is no implementation of this algorithm.

2.8.2 MuSiC

Psai et al. proposed MuSiC, a web server for constrained multiple sequence alignment [50]. It forces the alignment of specified nucleotides/residues with each other in columns. It accepts input sequences in FASTA format, allows choosing a scoring matrix from the list of available matrices, and penalizes using an affine gap penalty scheme. The constraints are specified as a string of characters where multiple constraints are separated by commas. There is a 'ratio" field, ϵ in MuSiC, which is used to compute the maximum number of allowable mismatches in the aligned segments. As an example, Tsai et al. showed the alignment of SARS-TW1, a coronavirus responsible for Severe Acute Respiratory Syndrome (SARS), with porcine epidemic diarrhea virus, human coronavirus, porcine transmissible gastroenteritis virus, bovine coronavirus, and mouse hepatitis virus using MuSiC, where $\epsilon = 0.5$ was used, and together with specific short nucleotide sequences used as constraints. The resulting alignment showed the specified constraints with at most one mismatch per segment.

2.8.3 RE-MuSiC

Chung et al. introduced RE-MuSiC for constrained multiple sequence alignment that accepts regular expression constraints [8]. It identifies the regions matched by the regular expression and generates an alignment where these matching regions get aligned. RE-MuSiC allows the specification of complex constraints for multiple sequence alignment. As an example, the PROSITE database includes information on conserved motifs, which are expressed in a form of regular expression. These motifs can be used as constraints in RE-MuSiC, and hence can be incorporated into the alignments, potentially improving their quality. The regular expression constraints in RE-MuSiC are formulated using the PROSITE pattern format [25]. As an example, Chung et al. showed the alignment of GSTs (Glutathione S-transferase) using RE-MuSiC with Casein Kinase II Phosphorylation site, using `[ST]-x(2)-[DE]` as a constraint, and the region identified by the constraint is aligned in the resultant alignment. In this example, this enforces that there is a region that aligns with either S or T in the same column of all sequences, followed by two occurrences of any character, followed by either D or E in every sequence.

2.8.4 DIALIGN

DIALIGN [35] is another tool for constrained multiple sequence alignment. It accepts constraints in the form of user defined anchor points, and generates an alignment by aligning the user specified anchor points wherever possible. An anchor point corresponds to equal length segments of the two sequences, and is defined by five coordinates: the first sequence, the second sequence, starting position in the first sequence, starting position in the second sequence, and the length of the segments. Users are also able to assign a priority to the anchor point which is a criterion to select the anchor point if there exists a conflicting set of anchor points.

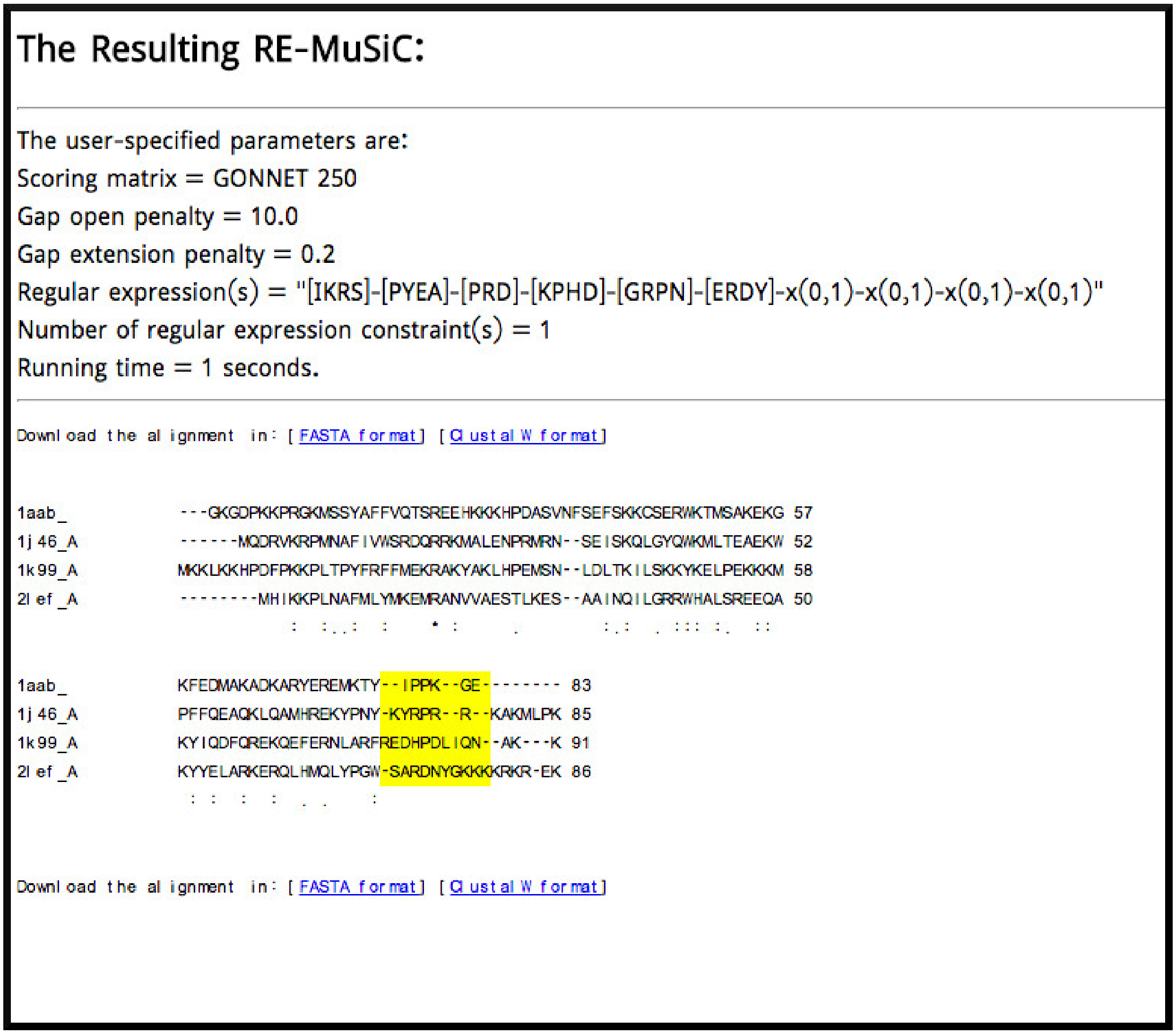
2.8.5 COBALT

COBALT (Constraint Based Alignment Tool) [39] is a framework for multiple alignment of protein sequences. COBALT does not explicitly allow the user to specify constraints on the input sequences; rather, it automatically generates constraints by searching the conserved domain database (CDD) [32] and the protein motif database PROSITE [25], and uses these constraints to improve the quality of the generated alignments.

Similar tools to COBALT have been proposed by other researchers such as Comet and Henry [12] who proposed a tool that extends the traditional Smith and Waterman algorithm and awards a score when the alignment of patterns from the PROSITE database are made. Similarly Du and Lin [16] proposed a pattern constrained algorithm that searches for pattern constraints in the PROSITE, Blocks+ [23], and eBLOCKs [47] databases.

2.9 Limitations of Existing CMSA Programs

The programs discussed in Section 2.8 have certain limitations. Papadopoulos et al. [39] used the additional biological information in the conserved domain database (CDD) and the PROSITE to improve the quality of multiple sequence alignments. But this information cannot be used directly in a CMSA program such as in MuSiC. For this reason, Arslan [5] proposed RECSA (Regular Expression Constrained Sequence Alignment) that accepts regular expression constraints for pairwise sequence alignment, and later he extended it to support multiple sequences. But according to Chung et al. [8] these algorithms by Arslan do not find the portion of the alignment that satisfies the constraints and only reports the alignment score without showing the complete alignment. So Chung et al. proposed a tool called RE-MuSiC (Multiple Sequence Alignment with Regular Expression Constraints) [8] for constrained multiple sequence alignment that accepts regular expression constraints. This tool allows the use of the conserved motifs described by regular expression such as those found in the PROSITE database to be used as constraints. But RE-MuSiC does not allow the use of certain quantification operators such as Kleene star (*), Kleene plus (+), and others in regular expression constraints. Thus, only a strict subset of regular expressions can be used as input. Moreover, RE-MuSiC does not always align certain parts of the regular expression matches in a desirable fashion. For example, RE-MuSiC used with the BB11001 dataset from the BAliBASE 3.0 benchmark database and with constraint `"[IKRS]-[PYEA]-[PRD]-[KPHD]-[GRPN]-[ERDY]-x(0,1)-x(0,1)-x(0,1)-x(0,1)"` produces an alignment where the residues `I`, `K`, `R`, and `S` are in different columns, as their algorithm matches the entire regular expression to each sequence, and then aligns entire matching segments. If the user wishes instead that anything matching the `[IKRS]` portion of the regular expression be aligned together in the same column, this cannot be done with RE-MuSiC. Figure 2.9 shows the generated alignment by RE-MuSiC for this aforementioned dataset and the constraint. Furthermore, the DIALIGN [35] web server accepts user defined anchor points as constraints for multiple sequence alignment, but does not accept regular expression constraints.



The Resulting RE-MuSiC:

The user-specified parameters are:
Scoring matrix = GONNET 250
Gap open penalty = 10.0
Gap extension penalty = 0.2
Regular expression(s) = "[IKRS]-[PYEA]-[PRD]-[KPHD]-[GRPN]-[ERDY]-x(0,1)-x(0,1)-x(0,1)-x(0,1)"
Number of regular expression constraint(s) = 1
Running time = 1 seconds.

Download the alignment in: [FASTA format] [ClustalW format]

```
1aab_             ---GKGDPKKPRGKMSSYAFFVQTSREEHKKKHPDASVNFSEFSKKCSERWKTMSAKEKG 57
1j46_A            ------MQDRVKRPMNAFIVWSRDQRRKMALENPRMRN--SEISKQLGYQWKMLTEAEKW 52
1k99_A            MKKLKKHPDFPKKPLTPYFRFFMEKRAKYAKLHPEMSN--LDLTKILSKKYKELPEKKKM 58
2lef_A            --------MHIKKPLNAFMLYMKEMRANVVAESTLKES--AAINQILGRRWHALSREEQA 50
                             :  :..:  :    * :     .        :.:  . ::: :.  ::

1aab_             KFEDMAKADKARYEREMKTY--IPPK--GE--------- 83
1j46_A            PFFQEAQKLQAMHREKYPNY-KYRPR--R--KAKMLPK 85
1k99_A            KYIQDFQREKQEFERNLARFREDHPDLIQN--AK---K 91
2lef_A            KYYELARKERQLHMQLYPGW-SARDNYGKKKRKR-EK 86
                   : :  :  :  . .    :
```

Download the alignment in: [FASTA format] [ClustalW format]

Figure 2.9: RE-MuSiC alignment of BB11001 dataset from BAliBASE 3.0 with the regular expression constraint `[IKRS]-[PYEA]-[PRD]-[KPHD]-[GRPN]-[ERDY]-x(0,1)-x(0,1)-x(0,1)-x(0,1)`.

Chapter 3

Objectives

Several tools for constrained multiple sequence alignment have been proposed in the literature such as MuSiC [50], RE-MuSiC [8], RECSA [5], DIALIGN [35] etc. Among these tools only RECSA and RE-MuSiC accept regular expression constraints, and they have some limitations as discussed in Section 2.9. This thesis proposes a novel technique for constrained multiple sequence alignment, with three main research objectives. The first is to introduce a modular program for constrained multiple sequence alignment called CSA-X that accepts regular expression constraints. This goal is described in more detail in Section 3.1. The second is to conduct a benchmarking study for the proposed program CSA-X with existing tools, which is described in detail in Section 3.2. Finally, Section 3.3 describes the third objective, which is to investigate whether CSA-X can improve the accuracy of alignments, versus not using any constraints, with various types of regular expressions.

3.1 CSA-X: A Modular Multiple Sequence Alignment Program

This section describes the motivations behind creating the proposed constrained multiple sequence alignment program CSA-X. CSA-X is a modular program that accepts regular expression constraints and uses an arbitrary multiple sequence alignment program to generate portions of the alignments. RECSA [5] and the other algorithm for the constrained alignment of multiple sequences [4] by Arslan are not available as implemented programs. Furthermore, Chung et al. [8] wrote in their paper that these algorithms only report the alignment score and not the complete alignment. The tool proposed by Chung et al. RE-MuSiC, does report full alignments and scores, but accepts only a subset of regular expressions as input, and does not always align the regions matched by the regular expression constraint, as discussed in detail in Section 2.9. This study proposes a tool that allows to use arbitrary regular expressions including the use of the quantification operators such as Kleene star and Kleene plus which cannot be used in the other tools proposed in the literature. The CMSA programs proposed in the literature do not guarantee the alignment of the sub-patterns of the regular expression constraint. CSA-X is designed in such a way that the user can enforce that the regions matched by the sub-patterns of the regular expression constraints get aligned. Moreover, CSA-X is a modular tool that uses an underlying program to generate alignments, which allows the changing of the underlying multiple sequence alignment program if more accurate or more efficient programs, or specialized programs appropriate for a given problem become available.

3.2 Benchmarking Study

To assess the accuracy of MSA programs different benchmark databases have been proposed in the literature such as BAliBASE 3.0 [52], IRMBASE [48], PREFAB [17] etc. Notredame et al. [37], Katoh et al. [27], and Edgar [17] used benchmarking studies to assess the efficiency of their proposed multiple sequence alignment tools. However, such a benchmarking study does not exist for the constrained multiple sequence alignment program RE-MuSiC. This study conducts a benchmarking study for the proposed tool CSA-X and RE-MuSiC, and as CSA-X is a modular tool that uses an underlying program to generate alignments — this study includes benchmarking results for the underlying MSA programs as well. Furthermore, to ensure that differences in accuracy between alignments generated by CSA-X and other algorithms using various quality metrics are not by chance, this study also conducts and presents the results of statistical significance testing.

3.3 Enhancing the Accuracy of Multiple Sequence Alignment Programs with Constraints

Using sequences involved in the alignment, sometimes, expert users can have information regarding active site residues, and conserved motifs — such additional information can improve the generated alignments by MSA programs [50]. Our objective is to test whether CSA-X can improve the accuracy of the alignments if an expert user provides necessary information regarding the sequences. To test this, CSA-X is provided with constraints identified from the correct alignments of the benchmark database. Constraints are identified from both the most conserved regions and the least conserved regions, and they are supplied to the CSA-X program to compare the accuracy of alignments versus other applications. A detailed discussion on how the constraints are generated for CSA-X can be found in the Section 5.2.

Chapter 4

Methods

The proposed CSA-X is a modular program that accepts constraints in the form of regular expressions over an extended syntax, where the symbol # has additional special meaning. This symbol can be placed in multiple spots in the regular expression. Intuitively, if the regular expression matches each of the sequences to be aligned, the sections that match between two consecutive # symbols are aligned together. At first, the regular expression matches for each entire sequence are identified. Then the sections between # symbols of a sequence are aligned with corresponding sections of the other sequences, and thus the entire regular expression matched segments are aligned. CSA-X constraints can be formulated without using any # symbols as well, but, if the # symbols are used for constructing the regular expression constraint, then it provides additional guidance by which the matching subwords are aligned. These regular expressions extended with the # symbol are referred to as hash-augmented regular expression.

4.1 Hash-augmented Regular Expressions

Formally, a hash-augmented regular expression can be defined inductively:

- every regular expression is a hash-augmented regular expression,
- if R and S are two hash-augmented regular expressions, then R#S is a hash-augmented regular expression.

From this definition, every hash-augmented regular expression can be written in the following form, for some $n \geq 1$:

$$R_1 \# R_2 \# \ldots \# R_n,$$

where $R_1, R_2, \ldots, R_n$ are regular expressions. A hash-augmented regular expression is a way of specifying a regular expression constraint in CSA-X. It is constructed in such a way that the sub-patterns in the regular expression can be separated by # symbols. As an example, consider a dataset having 3 protein sequences as follows:

$$\begin{aligned} S &= s_1 s_2 \cdots s_n \\ R &= r_1 r_2 \cdots r_m \\ T &= t_1 t_2 \cdots t_p \end{aligned}$$

where s_i, r_j, t_k are individual residues, for all $1 \leq i \leq n$, $1 \leq j \leq m$, $1 \leq k \leq p$. If a user would like to align segments $S_{2,5} = s_2s_3s_4s_5$, $R_{4,7} = r_4r_5r_6r_7$ and $T_{5,8} = t_5t_6t_7t_8$ with each other in a column-by-column fashion, then these segments can be converted to a single hash-augmented regular expression as follows:

$$[s_2r_4t_5]\#[s_3r_5t_6]\#[s_4r_6t_7]\#[s_5r_7t_8]. \tag{4.1}$$

Here, the square brackets use the standard Perl regular expression syntax, and refers to a single character listed or contained in the list (that is, any of the characters in parentheses can match). The # symbols provide additional control by giving information regarding the sections to align in columns, and ensures that the residues s_2, r_4, or t_5 followed by s_3, r_5, or t_6 followed by s_4, r_6, or t_7 followed by s_5, r_7, or t_8 get aligned in columns. If the # symbols are not used in Equation 4.1, then CSA-X constructs the best alignment which could be of the entire segment of S, R, and T that matches the regular expression, which possibly could for example, align as follows:

$$s_2s_3s_4s_5 -- \; --$$

$$-- \; -- r_4r_5r_6r_7$$

$$-- \; -t_5\, t_6t_7\, t_8-$$

For the case of hash-augmented regular expressions, between every two # symbols, there is a syntactically correct regular expression [1] (by the definition of hash-augmented regular expressions). As an example, suppose `(AC#TT)C#A` is a hash-augmented regular expression constraint for an input dataset of protein sequences. But this expression is not a valid CSA-X hash-augmented regular expression, because the left side of the first # symbol contains `‘(AC’` and right side contains `‘TT)C’` which are not syntactically correct (they are not regular expressions).

It should be noted that if a hash-augmented regular expression matches multiple sequences, then each matching segment must match the same number of hash symbols, since they cannot be placed inside any quantification operator by the definition of hash-augmented regular expressions.

4.2 Algorithm

The constrained multiple sequence alignment program, CSA-X, accepts constraints in the form of hash-augmented regular expressions. Consider, an input dataset that contains a list of N sequences $S_1, S_2, S_3, \ldots, S_N$, and a user has additional information regarding the sequences in the form of a hash-augmented regular expression $R = R_1\#R_2\#R_3\# \cdots \#R_M$, where M indicates the number of sub-patterns in the expression. CSA-X takes the dataset and the hash-augmented regular expression as input, and generates an output alignment by aligning the regular expression matched segments together. The process that CSA-X uses to generate such an alignment can be described using the following high-level steps:

[1] By syntactically correct, we mean that its syntax implies that it is defined to be a regular expression.

1. At first, CSA-X attempts to match the hash-augmented regular expression constraint, R to each sequence S_x of the input dataset, where $1 \leq x \leq N$. If the regular expression matches exactly once in each sequence S_x, then CSA-X determines a list of positions (for each x, $1 \leq x \leq N$) $l_x^0, l_x^1, l_x^2, \ldots, l_x^M$, where $0 \leq l_x^0 \leq l_x^1 \leq l_x^2 \leq \ldots \leq l_x^M \leq |S_x| + 1$, whereby regular expression R_j matches between positions l_x^{j-1} and $l_x^j - 1$, for each j, $1 \leq j \leq M$ (if $l_x^{j-1} = l_x^j$ then R_j matches the empty string). It should be noted that if CSA-X does not find any regular expression match on the input sequences or it finds matches for a strict subset of sequences in the dataset, then it returns the alignment of the input dataset using the underlying MSA program without using the regular expression.

2. In this step, CSA-X generates alignments for each of the matched sections of the sequences. That is, it aligns the subwords $S_1(1, l_1^0-1), \ldots, S_N(1, l_N^0-1)$, then aligns subwords $S_1(l_1^{j-1}, l_1^j-1), \ldots, S_N(l_N^{j-1}, l_N^j-1)$ for every j, $1 \leq j \leq M$, and then aligns subwords $S_1(l_1^{M+1}, |S_1|), \ldots, S_N(l_N^{M+1}, |S_N|)$. Then it concatenates each of these alignments together in order. Indeed as the constraints in CSA-X are specified using a hash-augmented regular expression, it generates alignments by decoding information from the specified constraints.

   Intuitively, the formalism of this step means that CSA-X identifies the segments that match the entire expression R for each sequence S_x in the previous steps. In addition, at the same time on each of the matched segments, it also identifies the subsections that match the sub-patterns $R_1, R_2, R_3, \ldots, R_M$ consecutively in the hash-augmented regular expression. Then, CSA-X aligns each matching sub-pattern separately (including the parts that match before the first matching sub-pattern, and the parts that match after the final sub-pattern has ended), using the underlying MSA program X to generate alignments, and then merges the generated alignments together to produce a complete alignment.

3. However, if it finds multiple regular expression matches on a single sequence, then it generates all possible combinations of the matched-segment datasets by selecting each regular expression match of the sequence separately. As an example, suppose a hash-augmented regular expression R, matches the sequence S_t at two spots, and hence has two matched segments $S_t(a, b)$ and $S_t(i, j)$ where, $1 \leq a \leq b \leq |S_t|$, $1 \leq i \leq j \leq |S_t|$, and $i > b$. If the rest of the sequences match the hash-augmented regular expression exactly at one spot, then CSA-X would create two alignments, one where the matching occurs between $S_t(a, b)$ with the single matches on the other sequences, and the other one is where the matching occurs between $S_t(i, j)$ with the single matches on the other sequences. Then the algorithm determines the alignment that has the highest sum-or-pairs score, and returns the alignment with the highest score.

It should be noted that, CSA-X only allows a single regular expression as input, although, multiple regular expressions can be combined into a single regular expression by joining them with quantifiers such as $R.^*S$, where R and S are two regular expressions, "." represents any character match, and "*" is Kleene star.

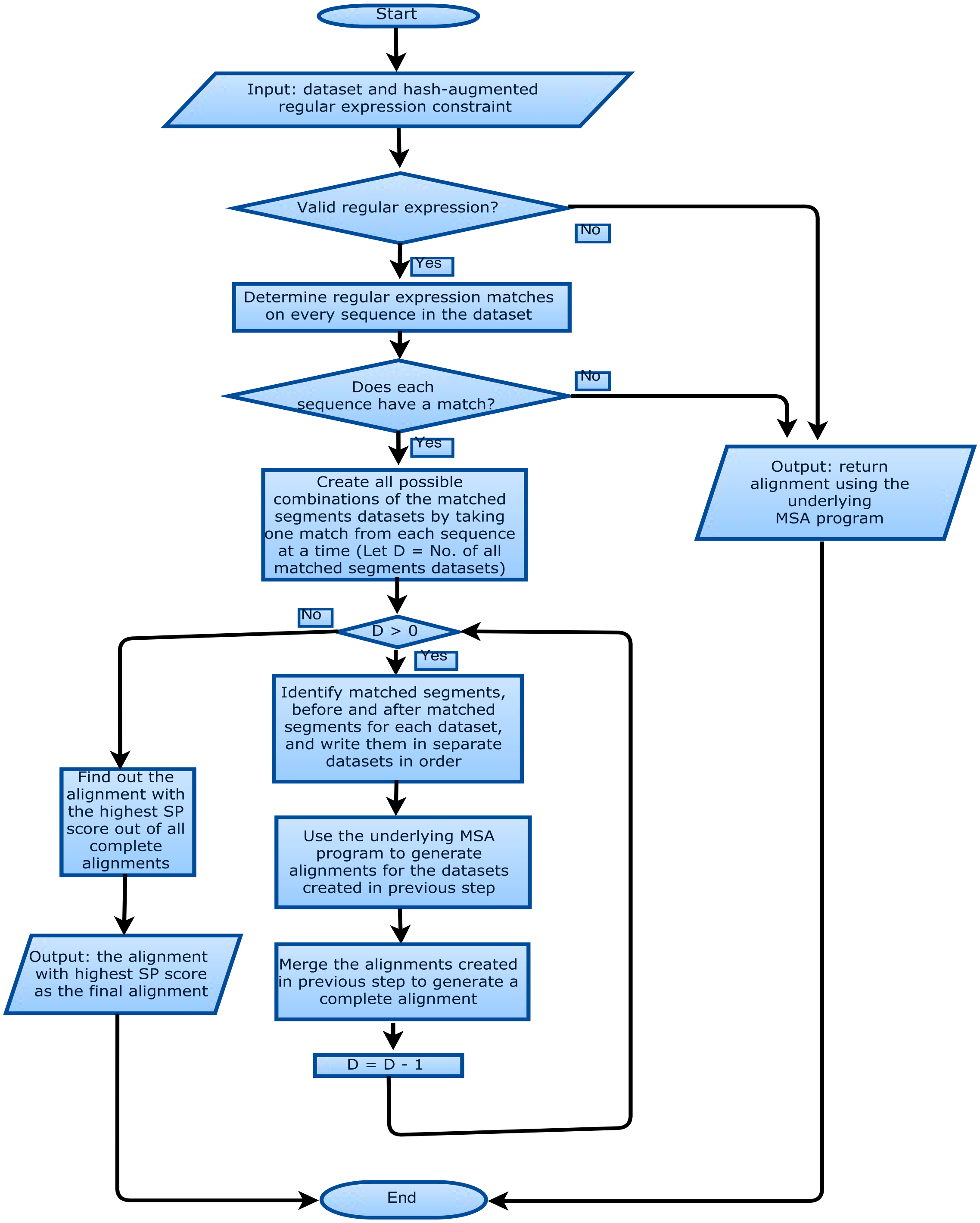

Start
Input: dataset and hash-augmented regular expression constraint
Valid regular expression?
No
Yes
Determine regular expression matches on every sequence in the dataset
Does each sequence have a match?
No
Yes
Create all possible combinations of the matched segments datasets by taking one match from each sequence at a time (Let D = No. of all matched segments datasets)
Output: return alignment using the underlying MSA program
No
D > 0
Yes
Identify matched segments, before and after matched segments for each dataset, and write them in separate datasets in order
Find out the alignment with the highest SP score out of all complete alignments
Use the underlying MSA program to generate alignments for the datasets created in previous step
Output: the alignment with highest SP score as the final alignment
Merge the alignments created in previous step to generate a complete alignment
D = D - 1
End

Figure 4.1: Flowchart of CSA-X algorithm.

4.3 Example

CSA-X is a modularized tool, which accepts hash augmented regular expression constraints. Conserved motifs for different protein sequences are listed in the PROSITE database in the form of regular expressions, which can be used as constraints to improve the biological accuracy of the alignments in different CMSA programs. Our implementation of the CSA-X algorithm accepts a hash-augmented regular expression constraint, and converts it to a Perl regular expression. But the format of specifying regular expression constraints are different in PROSITE and in CSA-X, hence a format conversion is required to use the motifs from PROSITE as a constraint in CSA-X. Furthermore, CSA-X allows the use of different quantification operators such as Kleene star, Kleene plus etc. in the hash-augmented regular expression constraint specification. For example, the TATA-binding protein plays a vital role in the activation of eukaryotic genes. PROSITE (PDOC00303) lists the consensus for the signature pattern of the TATA-binding protein as follows:

```
Y-x-[PK]-x(2)-[IF]-x(2)-[LIVM](2)-x-[KRH]-x(3)-P-[RKQ]-x(3)-L-[LIVM]-F-x-
[STN]-G-[KR]-[LIVMA]-x(3)-G-[TAGL]-[KR]-x(7)-[AGCS]-x(7)-[LIVMF].
```

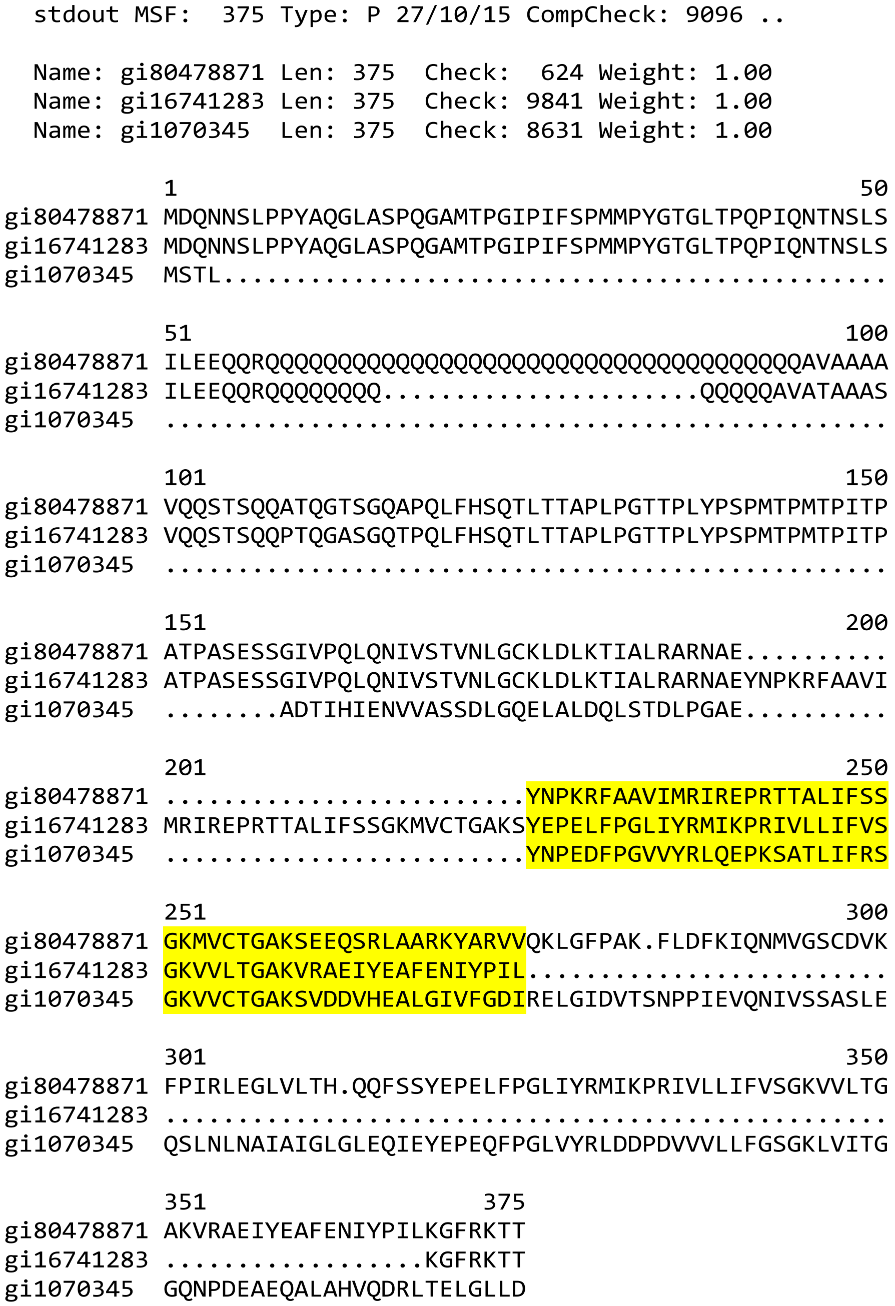
For the alignment of different TATA box proteins the above mentioned consensus pattern can be used as a constraint. For instance, if one would like to align TATA box proteins found in human (gb AAI09054.1), rat (gb AAH16476.1), and a microorganism called *Halobacterium salinarum* (emb CAA63691.1) using CSA-X, then the format of this consensus pattern can be as follows:

```
Y.[PK]..[IF]..[LIVM]{2}.[KRH]...P[RKQ]...L[LIVM]F.[STN]G[KR][LIVMA]...G
[TAGL][KR].......[AGCS].......[LIVMF].
```

To illustrate the use of a more general quantification operator, the segment "[LIVM]{2}" in regular expression constraint can be replaced by "[LIVM]+". It is also possible to add hash symbols to force matching sections of the regular expression to align, as follows:

```
Y.[PK]..[IF]..[LIVM]+.[KRH]#...P[RKQ]...L[LIVM]F.[STN]G[KR][LIVMA]#...G
[TAGL][KR].......[AGCS].......[LIVMF].
```

Figure 4.2 shows the alignment generated by CSA-X, where the region identified by the regular expression is aligned in columns (highlighted). For this alignment ProbCons is used as the underlying alignment tool.



```
  stdout MSF:  375 Type: P 27/10/15 CompCheck: 9096 ..

  Name: gi80478871 Len: 375  Check:  624 Weight: 1.00
  Name: gi16741283 Len: 375  Check: 9841 Weight: 1.00
  Name: gi1070345  Len: 375  Check: 8631 Weight: 1.00

           1                                               50
gi80478871 MDQNNSLPPYAQGLASPQGAMTPGIPIFSPMMPYGTGLTPQPIQNTNSLS
gi16741283 MDQNNSLPPYAQGLASPQGAMTPGIPIFSPMMPYGTGLTPQPIQNTNSLS
gi1070345  MSTL..............................................

           51                                             100
gi80478871 ILEEQQRQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQAVAAAA
gi16741283 ILEEQQRQQQQQQQQ......................QQQQQAVATAAAS
gi1070345  ..................................................

           101                                            150
gi80478871 VQQSTSQQATQGTSGQAPQLFHSQTLTTAPLPGTTPLYPSPMTPMTPITP
gi16741283 VQQSTSQQPTQGASGQTPQLFHSQTLTTAPLPGTTPLYPSPMTPMTPITP
gi1070345  ..................................................

           151                                            200
gi80478871 ATPASESSGIVPQLQNIVSTVNLGCKLDLKTIALRARNAE..........
gi16741283 ATPASESSGIVPQLQNIVSTVNLGCKLDLKTIALRARNAEYNPKRFAAVI
gi1070345  ........ADTIHIENVVASSDLGQELALDQLSTDLPGAE..........

           201                                            250
gi80478871 .........................YNPKRFAAVIMRIREPRTTALIFSS
gi16741283 MRIREPRTTALIFSSGKMVCTGAKSYEPELFPGLIYRMIKPRIVLLIFVS
gi1070345  .........................YNPEDFPGVVYRLQEPKSATLIFRS

           251                                            300
gi80478871 GKMVCTGAKSEEQSRLAARKYARVVQKLGFPAK.FLDFKIQNMVGSCDVK
gi16741283 GKVVLTGAKVRAEIYEAFENIYPIL.........................
gi1070345  GKVVCTGAKSVDDVHEALGIVFGDIRELGIDVTSNPPIEVQNIVSSASLE

           301                                            350
gi80478871 FPIRLEGLVLTH.QQFSSYEPELFPGLIYRMIKPRIVLLIFVSGKVVLTG
gi16741283 ..................................................
gi1070345  QSLNLNAIAIGLGLEQIEYEPEQFPGLVYRLDDPDVVVLLFGSGKLVITG

           351                   375
gi80478871 AKVRAEIYEAFENIYPILKGFRKTT
gi16741283 ..................KGFRKTT
gi1070345  GQNPDEAEQALAHVQDRLTELGLLD
```

Figure 4.2: CSA-X alignment of TATA box proteins, where the highlighted regions indicate the sections matched by the regular expression constraint.

Chapter 5

Assessment

Performance of CSA-X is assessed using the BAliBASE 3.0 benchmark database, and compared with the results of RE-MuSiC — the other CMSA program that takes regular expression constraints as input. Further, the performance of CSA-X with T-Coffee version 11.00.8cbe486 is compared to T-Coffee itself, and similarly with ProbCons version 1.12. Since CSA-X is a modular tool, the underlying MSA program can be changed to achieve better alignments. The study conducted by Pais, FSM et. al [38] showed that ProbCons, T-Coffee, Probalign and MAFFT achieve higher accuracy than other MSA tools considered in their study. In this study of assessment, ProbCons and T-Coffee are used as the underlying MSA algorithms in CSA-X (although other programs can be used with CSA-X as well, these are the only two used for the purposes of assessment). Whenever CSA-X uses ProbCons, it is referred to as CSA-PC, and for T-Coffee, it is called CSA-TCOF.

5.1 Benchmark Database

RE-MuSiC generates erroneous alignments for some datasets in the BAliBASE 3.0 benchmark database, where the length of the sequences are not equal and sometimes the resulting alignments contain wildcard characters. For example, RE-MuSiC run with BB20003 dataset from BAliBASE 3.0 using default settings and the following constraint

```
[IMLVFATWYQGKH]-[VDLMFISAYENGQCH]-[PSKEGRQAFDTLNWIVH]-[DTNKQAEHFGRSMCIP]-
[VIAGNDKPLMRHTSYEF]-[VQIPKGSEARLDYNF]-x(0,1)-x(0,1)-x(0,1)-x(0,1)
```

returns an alignment, where the length of the sequences are different for headers 1o20_A, PROA_STRTR and PROA_STRPN. Furthermore, for header PROA_STRPN, the sequence in the output alignment contains wildcard characters. Hence, the *working database* for this study is defined as being created from BAliBASE 3.0 including those datasets for which RE-MuSiC produces non-erroneous alignments. BAliBASE 3.0 is classified into several groups; namely RV11, RV12, RV20, RV30, RV40, and RV50. Each of these groups include datasets having full length sequences (filenames start with BB) and short truncated sequences (filename starts with BBS) except RV40. RV11 includes divergent equidistant sequences having less than 20% identity, but RV12 contains datasets where sequences have 20 – 40% identity. Datasets in RV20 contain sequences that are from the same family and share more than 40% identity, and there also exist orphan sequences (distant members of the family having a common fold and share less than 20% identity). RV 30 contains sequences from different subfamilies, where members of the same subfamily share more than 40% identity but two members from different subfamilies share less than 20% identity. RV40 and RV50 contain sequences with large N/C-terminal extensions and internal insertions respectively. In the working database for this study, there are 76 datasets from RV11, 84 datasets from RV12, 6 datasets from RV20, 6 datasets from RV30, 17 datasets from RV40 and 11 datasets from RV50, in total 200 datasets. Due to the erroneous alignments from RE-MuSiC, the working database therefore contains a total of 200 datasets from BAliBASE 3.0 out of 386 datasets. Out of these 200 datasets, 98 datasets contain short truncated sequences. To compare the performance of CSA-X with RE-MuSiC and other programs, this working database is used in this study (we will additionally consider the difference in results between programs when including those datasets not in the working dataset).

5.2 Constraints

To identify the effects of constraints on generated alignments, two sets of regular expression constraints are created from the correct alignments of the BAliBASE 3.0 benchmark database. One set of regular expression constraints are created from the most conserved region of the correct alignments having some maximum number of gaps. Another set is constructed from the least conserved region of the correct alignments with a maximum number of gaps. All of these constraints are automatically generated using a Perl script, which uses reference alignment files from BAliBASE 3.0, identifies the most conserved regions and the least conserved regions for the alignments, and generates the regular expression constraints. The length of the regular expression constraints and the maximum number of gaps allowed per sequence are also used as input for this script. Based on these information, the script generates regular expression constraints to be used in CSA-X and RE-MuSiC.

The idea behind this approach, is to identify the effects of constraints on multiple sequence alignment. Usually, expert users possess information about the sequences involved in the alignment process. They align the sequences using a MSA program, and after that, they correct the alignment based on their knowledge if it is not reflected in the generated alignment. If they have knowledge that a specific segment in a sequence should be aligned with some other segment in another sequence, it is possible to encode their knowledge beforehand and pass this information to MSA program. If this information is passed to a CMSA programs like RE-MuSiC and CSA-X, this study investigates which of the two programs achieves a higher score. To better understand the effects of constraints, they were chosen from both the most conserved region and the least conserved region of the correct alignments. In this study, conserved regions are identified in terms of sum-of-pair (SP) score, where higher SP score indicates the most conserved region and lower SP score indicates the least conserved region.

For this study, the regular expression constraints are generated to be of length 12 with a maximum of one gap per sequence. The length of regular expression constraint is chosen to be 12 to avoid multiple matches. To make a fair comparison between CSA-X and RE-MuSiC, both are tested on the same sets (most and least conserved) of regular expression constraints (and therefore, all regular expressions tested do not have the quantifiers * or + as these do not work with RE-MuSiC). Furthermore, CSA-PC with these regular expressions are compared to ProbCons without using any regular expressions at all (and similarly with T-Coffee) to gauge the potential improvements that using regular expressions as constraints can provide. This depends on whether the regular expressions are created from highly conserved or lesser conserved regions. Although this part of the assessment is done using the correct alignments to construct the regular expressions, it is only being used to see if regular expressions can possibly improve quality, depending on the type of regular expression. A thorough test of common regular expressions used by expert users together with a test to see if they improve alignment quality is beyond the scope of this thesis. However, for comparing RE-MuSiC to CSA-X, such regular expressions are equally favourable to both programs, and is therefore a useful method of comparison.

5.3 Accuracy Measurement

To measure the accuracy of considered programs in this study, two scores, Q score (Quality Score) and TC score (Total Column Score) are computed. Edgar [17] defined Q score as a ratio between the number of correctly aligned pairs by the algorithm being tested to the number of residue pairs in the reference alignment, which is same as the sum-of-pairs score defined by Thompson et. al [53]. TC score is the number of correctly aligned columns, divided by the number of columns in the reference alignment, which is also same as the column score (CS) defined by Thompson et al [53]. Program qscore, available at `http://drive5.com/qscore/` is used to compute these scores.

5.4 Statistical Analysis

In an experiment where sample data is collected from a population, an observed effect may be due to sampling error. To eliminate the idea that the difference is merely by chance, statistical significance tests are conducted. Researchers working in the area of multiple sequence alignment mainly use two tests to measure the statistical significance of the data; one is the Friedman rank test [19] and another one is the Wilcoxon signed-rank test [55]. Notredame et al. [37], Gotoh [21] used the Wilcoxon signed-rank test in their study to assess the statistical significance of their test results. On the other hand, Edgar [17], Batzoglou et al. [15] used the Friedman rank test. In this work, Wilcoxon signed-rank test and Wilcoxon rank-sum test are used to measure statistical significance. If two samples are paired, Wilcoxon signed-rank test is used. Otherwise, Wilcoxon rank-sum test is used. A brief introduction to these techniques are as follows:

5.4.1 Wilcoxon Signed-Rank Test

Wilcoxon signed-rank test is a non-parametric test, which means for this method the sample population need not be normally distributed, and the two sample datasets must be paired i.e. one dataset should be directly related to a specific observation in other dataset. In this method, two hypotheses are made; one is the null hypothesis, and the other one is the alternative hypothesis. Here "the two matched pairs come from populations with equal medians" — is assumed as the null hypothesis and "the two matched pairs come from populations with non-equal medians" — is assumed as the alternative hypothesis. Then in this test a score known as a "p-value" is computed, and based on that it is determined whether the null hypothesis is rejected [55].

5.4.2 Wilcoxon Rank-Sum Test

Wilcoxon rank-sum test is also a non-parametric test like the Wilcoxon signed-rank test. This test is applied if the two sample datasets are independent. Here, "the two samples come from populations with equal medians" — is considered as the null hypothesis and "the two samples come from populations with different medians" — is considered as the alternative hypothesis. Similarly for this test, a "p-value" is computed, and based on that it is determined whether the null hypothesis is rejected [55].

Note that in this study, "no significant difference between the datasets of the considered programs" — is considered as the null hypothesis, and if the p-value is less than 0.05 then the null hypothesis is rejected. Lesser p-value indicates more statistical significance.

5.5 Parameter Settings

Standalone ProbCons and T-Coffee are used with the default parameter settings (performing a comparison by systematically varying all parameters with every program is beyond the scope of this work). The same parameter settings of ProbCons and T-Coffee are used in CSA-PC and CSA-TCOF respectively. RE-MuSiC is run with default gap extension and gap open penalty. CSA-PC, CSA-TCOF, and RE-MuSiC are provided with the equivalent set of regular expression constraints. As the format of specifying regular expression constraints in CSA-X and RE-MuSiC is different, equivalent regular expression constraint sets are used for these programs.

Chapter 6
Results

6.1 Accuracy Results

In this study, ProbCons and T-Coffee are used as the underlying MSA algorithms in CSA-X. Whenever CSA-X uses ProbCons, it is referred to as CSA-PC, and for using T-Coffee, it is called CSA-TCOF. The considered programs for accuracy measurement in this study are: CSA-PC, CSA-TCOF, RE-MuSiC, T-Coffee and ProbCons. Average (AVG) and standard deviation (SD) of Q score and TC score (described in Section 5.3) for these programs are presented in Table 6.1. Among these programs CSA-PC, CSA-TCOF and RE-MuSiC are provided with the regular expression constraints. Section 5.2 describes the process of obtaining regular expression constraints in detail. Two types of constraints are used — constraints obtained from the most conserved regions (MC) and constraints from the least conserved regions (LC) of the correct alignments. The list of constraints used with CSA-X and RE-MuSiC are in Appendix A, and in Appendix B respectively. However, T-Coffee and ProbCons are run without any constraint on the working database, and their results are also included in Table 6.1. For assessing the accuracy of the proposed CSA-X modular program, two types of comparisons are designed:

- CSA-X versus other regular expression constrained MSA program (RE-MuSiC),
- CSA-X versus the underlying MSA program used in CSA-X.

6.1.1 Comparison of CSA-X with regular expression constrained MSA program (RE-MuSiC)

It is observed from Table 6.1, for 200 datasets in the working database that CSA-PC and CSA-TCOF both achieve higher accuracy compared to RE-MuSiC, for using both Q score and TC score. From Table 6.2 it is apparent that on average for Q score, CSA-PC achieves approximately 17.9% and CSA-TCOF achieves almost 17.4% higher score compared to RE-MuSiC when using constraints obtained from the least conserved (LC) region of the correct alignments respectively. However, for the constraints obtained from the most conserved (MC) region of the correct alignments, CSA-PC and CSA-TCOF achieve 17.6% and 16.8% higher score respectively. While for TC score, CSA-PC and CSA-TCOF shows even higher results. For the most conserved region regular expression constraints set, CSA-PC achieves 21.7% and CSA-TCOF achieves 20.6%

		MC	LC	
		AVG	AVG	AVG
Q	CSA-PC	**0.868** (**0.118**)	**0.881** (**0.116**)	-
	CSA-TCOF	0.860 (0.131)	0.876 (0.124)	-
	RE-MuSiC	0.691 (0.197)	0.702 (0.220)	-
	ProbCons	-	-	0.854 (0.153)
	T-Coffee	-	-	0.846 (0.166)
TC	CSA-PC	**0.713**(**0.222**)	**0.730** (**0.244**)	-
	CSA-TCOF	0.702 (0.231)	0.718 (0.244)	-
	RE-MuSiC	0.496 (0.256)	0.487 (0.299)	-
	ProbCons	-	-	0.693
	T-Coffee	-	-	0.680

Table 6.1: The table above lists the average and standard deviation (in parentheses) of Q score and TC score for the working database. MC and LC represent the use of regular expression constraints identified from the correct alignments of the most conserved region and the least conserved region of the benchmark datasets, and '-' represents a score not computed. The last column is computed without the use of regular expressions. The entries that are bold represent the highest value for each type of score and regular expression.

	RE-MuSiC			
	Q		TC	
	MC	LC	MC	LC
CSA-PC	0.176	0.179	0.217	0.243
CSA-TCOF	0.168	0.174	0.206	0.231

Table 6.2: Average Q score and TC score are higher for CSA-PC and CSA-TCOF in comparison to RE-MuSiC. MC and LC represent the use of regular expression constraints identified from the correct alignments of the most conserved region and the least conserved region of the benchmark datasets. Values in this table represent the difference between average Q score or TC score for the mentioned programs.

	CSA-PC vs. ProbCons		CSA-TCOF vs. T-Coffee	
	MC	LC	MC	LC
Q	0.014	0.026	0.014	0.030
TC	0.021	0.038	0.022	0.038

Table 6.3: Average Q score and TC score are higher for CSA-PC and CSA-TCOF in comparison to ProbCons and T-Coffee. MC and LC represent the use of regular expression constraints identified from the correct alignments of the most conserved region and the least conserved region of the benchmark datasets. Values in this table represent the difference between average Q score or TC score for the mentioned programs.

	Ratio for Q score	Ratio for TC score
CSA-PC (LC) >RE-MuSiC	$\frac{195}{200} = 97.5\%$	$\frac{185}{200} = 92.5\%$
CSA-PC (LC) = RE-MuSiC	$\frac{2}{200} = 1\%$	$\frac{5}{200} = 2.5\%$
CSA-PC (MC) >RE-MuSiC	$\frac{192}{200} = 96\%$	$\frac{186}{200} = 93\%$
CSA-PC (MC) = RE-MuSiC	$\frac{4}{200} = 2\%$	$\frac{9}{200} = 4.5\%$
CSA-TCOF (LC) >RE-MuSiC	$\frac{194}{200} = 97\%$	$\frac{184}{200} = 92\%$
CSA-TCOF (LC) = RE-MuSiC	$\frac{4}{200} = 2\%$	$\frac{7}{200} = 3.5\%$
CSA-TCOF (MC) >RE-MuSiC	$\frac{191}{200} = 95.5\%$	$\frac{180}{200} = 90\%$
CSA-TCOF (MC) = RE-MuSiC	$\frac{4}{200} = 2\%$	$\frac{15}{200} = 7.5\%$

Table 6.4: Percentage of datasets for which CSA-X performs higher or equal to RE-MuSiC. Each entry in the second column gives the number of datasets where CSA-X compares to RE-MuSiC as indicated in the first column divided by the total number of datasets (200). CSA-X (LC/MC) indicates that CSA-X is either provided with the LC or MC constraints set, and X indicates the name of the underlying MSA program.

higher results compared to RE-MuSiC, and the score rises to 21.7% and 20.6% respectively for CSA-PC and CSA-TCOF for the LC constraints set.

Out of 200 working datasets for Q score, CSA-PC and CSA-TCOF with LC constraints perform higher for 195 and 194 datasets respectively compared to RE-MuSiC, and CSA-PC and RE-MuSiC perform equally well for 2 datasets, and CSA-TCOF and RE-MuSiC perform equally for 4 datasets. So in fact, CSA-PC and CSA-TCOF perform higher in 98% and 97% of the cases respectively for the working database. In addition, for TC score, CSA-PC and CSA-TCOF with LC constraints set achieves higher score in total for 185 and 184 datasets, and an equal score for 5 and 7 datasets. CSA-PC and CSA-TCOF with MC constraints set achieves a higher score for Q score for 192 and 191 datasets respectively, and for TC score they achieve higher score for 186 and 180 datasets respectively compared to RE-MuSiC. The complete results are provided in Table 6.4. In addition, if all the datasets in BAliBASE 3.0 are considered, instead of just the working datasets, and we define CSA-X as performing better than RE-MuSiC for instances where RE-MuSiC is giving erroneous results, then CSA-PC (LC) gives a higher score for 381 datasets out of 386 datasets, and CSA-TCOF (LC) gives a higher score for 380 datasets out of 386 datasets.

6.1.2 Comparison of CSA-X with the underlying MSA programs used

Furthermore, from Table 6.1 CSA-PC and CSA-TCOF score higher overall than standalone ProbCons and T-Coffee run without any constraints. From Table 6.3 for MC constraints, CSA-PC and CSA-TCOF both shows 1.4% higher Q score and more than 2% higher TC score compared to ProbCons and T-Coffee. Where as, using LC constraints, CSA-PC and CSA-TCOF achieves 2.6% and 2.9% higher Q score and 3.78% and 3.76% higher TC score respectively. According to Thompson et al. [53] the BAliBASE sum-of-pairs score (similar to Q score) increases if a program succeeds in aligning sequences relative to the reference alignment dataset; this means the higher the Q score is, the better the program is at generating accurate alignments, while TC score tests how efficiently the program is aligning all the sequences. This is a more stringent criteria of measurement as a column score can become zero if a single sequence is misaligned [29]. The complete results (Q score and TC score separated for each individual dataset) of these considered programs can be found in the appendix C.

6.1.3 Result analysis on separate groups of the working database

Results of these programs on different groups in the working database show which programs are best suited for aligning sequences with N/C-terminal extensions, sequences with large internal insertions, and sequences with distant members of the family. Figures 6.1, 6.2, 6.3 and 6.4 demonstrate accuracy scores (Q and TC) for the considered programs. Table 6.5 and 6.6 indicates names of the programs that achieves a higher average Q score and TC score respectively. In these tables, rows labelled as MC/LC indicates that CSA-X was either provided with MC/LC regular expression constraints set. In contrast, the row "Recommended" holds the names of recommended programs for constrained alignment on different groups. Based on these results, it is concluded that CSA-X clearly outperforms overall RE-MuSiC. RE-MuSiC only provides higher results in terms of TC score for the RV20 subgroup provided with MC regular expression set. It is worth noticing that the difference between CSA-PC and RE-MuSiC is 17.3%, and for CSA-TCOF and RE-MuSiC is 18.6%. However, Both CSA-PC and CSA-TCOF achieves higher results for this subgroup when used with LC regular expression set; specifically, CSA-PC and CSA-TCOF achieves almost 39% higher results compared to RE-MuSiC.

As a whole, CSA-X achieves higher accuracy compared to RE-MuSiC and the rest of the programs. Moreover, it is observed that CSA-TCOF achieves slightly higher average compared to CSA-PC for subgroup RV40 and RV50, which contains datasets having sequences with N/C-terminal extensions and large internal insertions, while CSA-PC achieves higher score for rest of the subgroups compared to CSA-TCOF. In comparison to standalone ProbCons with CSA-PC, ProbCons achieves higher accuracy in terms of Q and TC score only for short sequences of subgroups RV12 and RV20. However, CSA-PC has higher accuracy compared to ProbCons for the rest of the subgroups of datasets with long and short truncated sequences. If T-Coffee is analyzed against CSA-TCOF, CSA-TCOF obtains good accuracy for Q score in most of the subgroups, with the exception of subgroup RV40 with MC constraints and RV50 with short sequences while used with LC constraints. T-Coffee achieves minimal .2% higher average for Q score in the case of RV40 and 2% higher for RV50 in comparison to CSA-TCOF. For TC score, T-Coffee obtains 7% higher average compared to CSA-TCOF for RV50 subgroup used with LC constraints only for short truncated sequences.

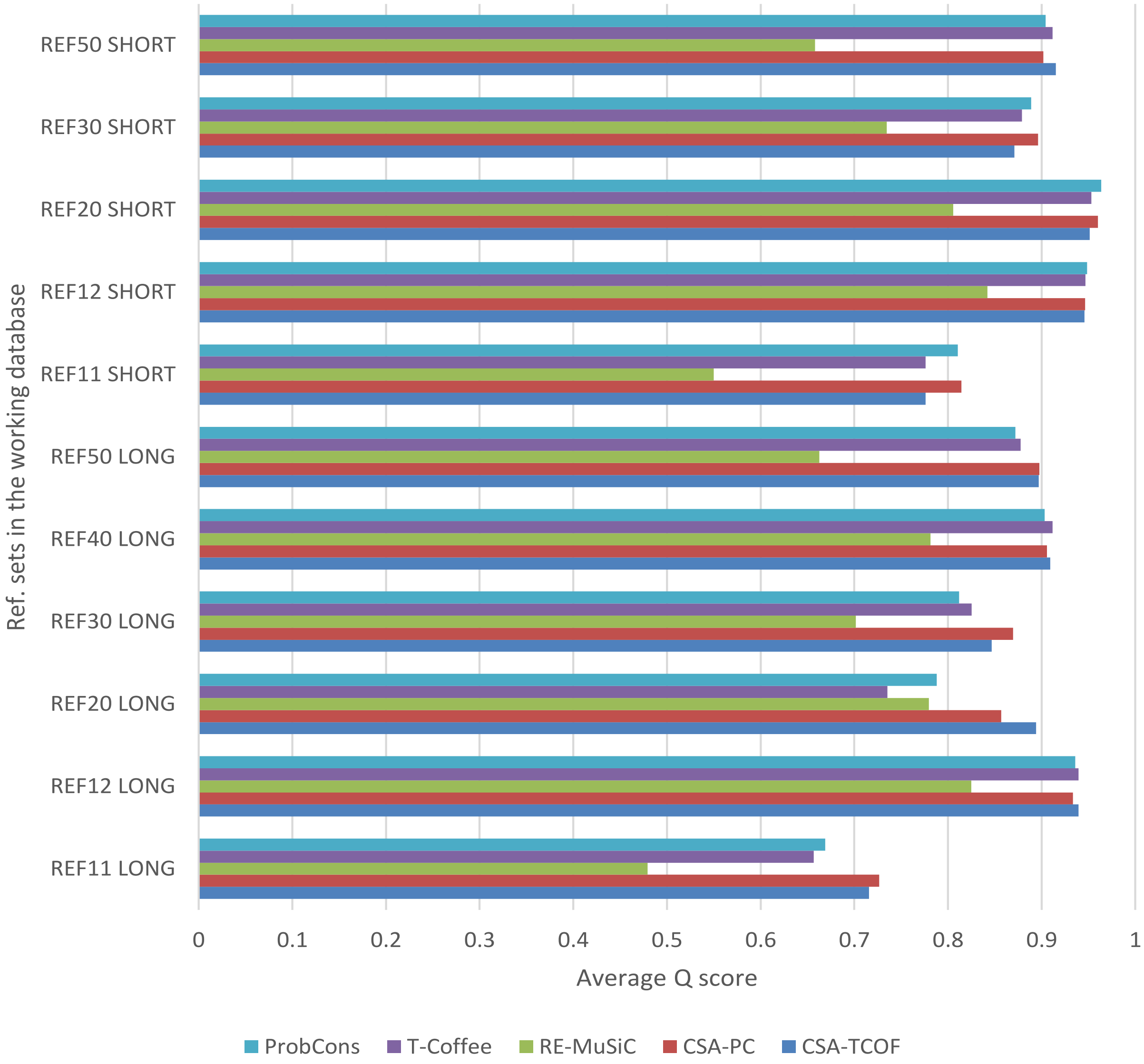


Figure 6.1: Average Q score on various subgroups for each program. CSA-TCOF and CSA-PC uses most conserved regular expression set.

		RV11	**RV12**	**RV20**	**RV30**	**RV40**	**RV50**
Long	MC	CSA-PC	CSA-TCOF	CSA-TCOF	CSA-PC	T-Coffee	CSA-PC
	LC	CSA-PC	CSA-PC	CSA-TCOF	CSA-PC	CSA-TC	CSA-TCOF
Short	MC	CSA-PC	ProbCons	ProbCons	CSA-PC		CSA-TCOF
	LC	CSA-PC	CSA-PC	CSA-TCOF	CSA-PC		T-Coffee
Recommended		CSA-PC	CSA-PC	CSA-TCOF	CSA-PC	CSA-TC/T-Coffee	CSA-TCOF

Table 6.5: Programs obtaining higher Q score in each subgroups based on results from Figure 6.1 and Figure 6.2. "Recommended" holds the names of recommended programs for constrained alignment on different groups.

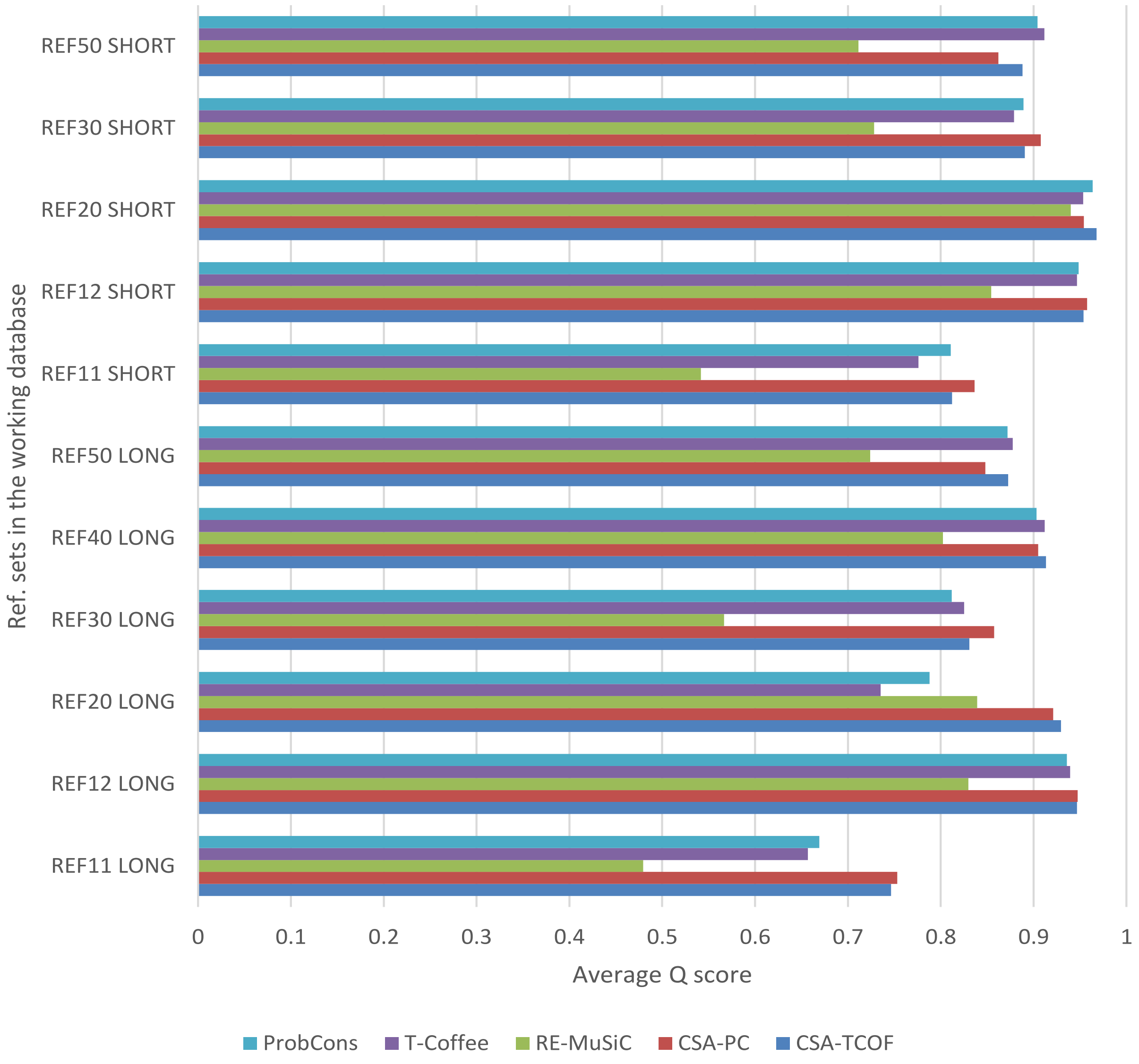


Figure 6.2: Average Q score on various subgroups for each program. CSA-TCOF and CSA-PC uses least conserved regular expression set.

		RV11	RV12	RV20	RV30	RV40	RV50
Long	MC	CSA-PC	CSA-TCOF	RE-MuSiC	CSA-PC	CSA-PC	CSA-PC
	LC	CSA-PC	CSA-PC	CSA-TCOF	CSA-PC	CSA-TC	T-Coffee
Short	MC	CSA-PC	ProbCons	ProbCons	CSA-PC		CSA-TCOF
	LC	CSA-PC	CSA-PC	CSA-TCOF	CSA-PC		CSA-TCOF
Recommended		CSA-PC	CSA-PC	CSA-TCOF	CSA-PC	CSA-PC/CSA-TC	CSA-TCOF

Table 6.6: Programs obtaining higher TC score in each subgroups based on results from Figure 6.3 and Figure 6.4. "Recommended" holds the names of recommended programs for constrained alignment on different groups.

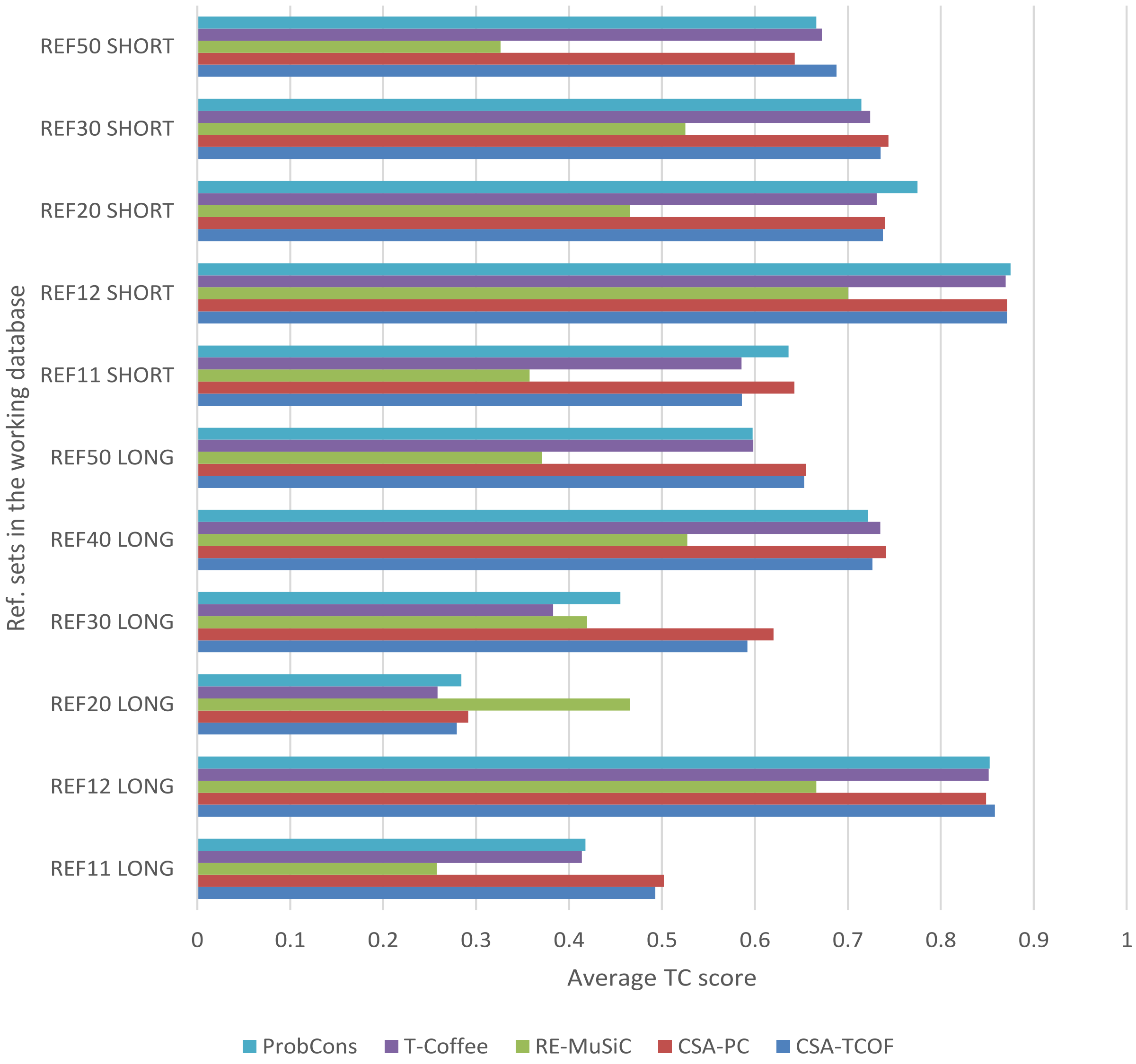


Figure 6.3: Average TC score on various subgroups for each program. CSA-TCOF and CSA-PC uses most conserved regular expression set.

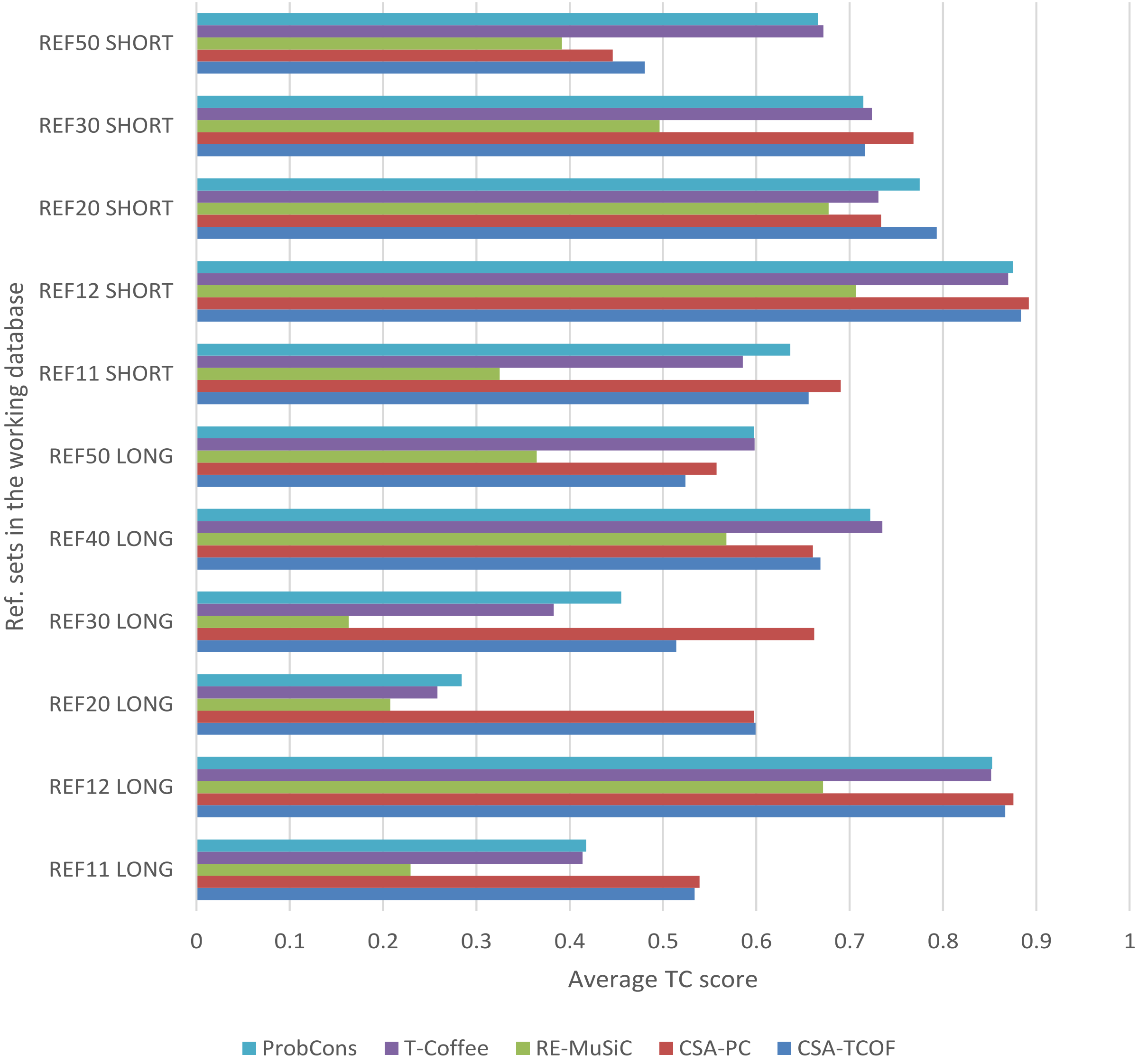


Figure 6.4: Average TC score on various subgroups for each program. CSA-TCOF and CSA-PC uses least conserved regular expression set.

Constraints Used	Compared Programs	Scores	P-value/ Significant
MC	CSA-TCOF vs. RE-MuSiC	Q	<2.2e-16 (yes)
		TC	2.89E-15 (yes)
	CSA-PC vs. RE-MuSiC	Q	<2.2e-16 (yes)
		TC	<2.2e-16 (yes)
LC	CSA-TCOF vs. RE-MuSiC	Q	<2.2e-16 (yes)
		TC	6.66E-16 (yes)
	CSA-PC vs. RE-MuSiC	Q	<2.2e-16 (yes)
		TC	<2.2e-16 (yes)

Table 6.7: Results of the Wilcoxon rank-sum test between CSA-TCOF against RE-MuSiC and CSA-PC against RE-MuSiC.

6.2 Statistical Analysis Results

For 200 (N = 200) working datasets, statistical significance tests are conducted between the different programs considered in this study. The Wilcoxon rank-sum test is carried out between CSA-TCOF and RE-MuSiC and between CSA-PC and RE-MuSiC. As the outcome of CSA-TCOF does not depend upon RE-MuSiC, hence the Wilcoxon rank-sum test is chosen. Table 6.7 shows the results of this test. The null hypothesis is that no significant difference between the two samples is considered. If the test rejects the null hypothesis then it means that there is a significance difference between the two samples. For this test, a 5% significance level is used, which means that if the p-value is less than 0.05 then the null hypothesis is rejected. In Table 6.7, all the p-values are less than 0.05. Hence, the null hypothesis is rejected, and it is determined that the results of CSA-PC and CSA-TCOF are significantly different compared to RE-MuSiC, and are not by chance.

Since the outcome of CSA-TCOF and CSA-PC depend upon T-Coffee and ProbCons respectively, so the Wilcoxon signed-rank test is selected to test if there is significance difference between these programs. The null hypothesis is that there is no significant difference between the samples. For this case, it is also considered that if the p-value is less than 0.05, then null hypothesis is rejected. Table 6.8 indicates the outcome of this test. From the table, it is observed that the results are not significantly different for CSA-TCOF and T-Coffee if CSA-TCOF uses the most conserved (MC) regular expression constraints set. Likewise, CSA-PC with MC constraints set, is not significantly different compared to ProbCons in terms of Q score. This is because ProbCons and T-Coffee both are able to successfully align the most conserved region without the explicit constraints. But the situation changes if CSA-TCOF and CSA-PC uses the least conserved (LC) regular expression constraints set, and from Table 6.8, it is observed that there is significant difference in the results of CSA-TCOF and CSA-PC with T-Coffee and ProbCons respectively if they are supplied with LC constraints set. It is most likely that ProbCons and T-Coffee are unable to align the least conserved

Compared Programs	**Constraints with CSA-X**	**Score**	**P/ Significant**
CSA-TCOF vs. T-Coffee	MC	Q	0.6529 (no)
		TC	0.1579 (no)
	LC	Q	8.18E-10 (yes)
		TC	3.09E-08 (yes)
CSA-PC vs. ProbCons	MC	Q	0.0911 (no)
		TC	0.0201 (yes)
	LC	Q	2.50E-10 (yes)
		TC	1.10E-08 (yes)

Table 6.8: Results of the Wilcoxon signed-rank test between CSA-TCOF against T-Coffee and CSA-PC against ProbCons.

regions properly, so whenever CSA is receiving these regions as constraints, it is aligning them properly and achieving higher accuracy scores.

For comparing RE-MuSiC to CSA-X, using the same regular expressions (whether created from the most or least conserved regions) gives significantly better results to CSA-X. This implies that when an alignment program with regular expression constraints is desired, CSA-X is the preferred program. Also, it is tested whether regular expression constraints with CSA-X is able to perform better than the standalone program used in CSA-X. Although, constraints chosen from the most or least conserved region are not necessarily realistic in terms of regular expression constraints chosen by either an expert user, or created from a database of additional information, using constraints created from the correct alignment does have the advantage of capturing some piece of information the user may know to be true, in a situation where a standalone alignment program is not giving ideal results. And indeed, constraints chosen from the most conserved region do not seem to help versus not using any constraint, however constraints chosen from the least conserved region do help versus not using any constraints.

Chapter 7

Conclusion and Future Directions

The constrained multiple sequence alignment program, CSA-X, allows the user to specify regular expression constraints for the multiple sequence alignment, and the resulting alignment enforces that specific sections matching the regular expression gets aligned. This can improve the accuracy and biological significance of the generated alignments, as functional and structural information regarding the sequences can be expressed using regular expression syntax. Since most of the MSA programs proposed in the literature are fully automated, and allows to only adjust a limited number of parameters, CSA-X offers a unique way to produce alignments according to the constraints stipulated by the user. In addition, the constraints are mentioned in CSA-X in accordance to the hash augmented regular expression syntax, which is an extension to the standard Perl regular expression syntax, is more robust, and supports the use of different quantification operators such as Kleene star and Kleene plus that are not supported by other constrained multiple sequence alignment program such as RE-MuSiC. The use of regular expression constraints enables CSA-X to be used with motifs recorded in the PROSITE database with a simple format conversion.

Research conducted by Papadopoulus and Agarwala [39] showed that the use of motifs from PROSITE and profiles from CDD (Conserved Domain Database) can improve the alignment accuracy of the multiple sequences. Preliminary work for this research showed that CSA-X was outperforming RE-MuSiC when using constraints from PROSITE. And there are some examples where using the regular expressions from PROSITE with CSA-X can improve an alignment versus using the standalone MSA program. But overall, using a standalone MSA program was better than using the PROSITE regular expression constraints with CSA-X, implying that it is not always desirable to use these types of regular expressions as constraints. However, a more systematic study of using constraints from PROSITE and other sources is left as future work.

In this research work, it is shown that if good regular expression constraints are chosen to be used with CSA-X, then the accuracy of the generated alignments increases. Based on the average accuracy scores from the benchmarking analysis and the statistical significance testing, it is shown that CSA-X framework with ProbCons and T-Coffee (known as CSA-PC and CSA-TCOF respectively) generates more accurate alignments compared to RE-MuSiC — the only other implemented CMSA algorithm that uses regular expression constraints. Specifically, on average, CSA-X used with constraints identified from the least conserved regions of the correct alignments, achieves results that are 17.65% more for Q score, and 23.7% more for TC score compared to RE-MuSiC. In fact, out of 200 datasets, using the least conserved constraints set, CSA-X with ProbCons (CSA-PC) and CSA-X with T-Coffee (CSA-TCOF) both achieve higher score for 97.5% and 97% of the cases for Q score, and 92.5% and 92% of the cases for TC score, respectively. Also if all the datasets in BAliBASE 3.0 are considered, instead of just the working datasets, for the least conserved constraints set, CSA-PC gives a higher score for 98.7% of the cases, and CSA-TCOF gives a higher score for 98.4% of the cases for Q score. Furthermore, it is also shown that if good regular expression constraints are chosen from the least conserved portion of the correct alignments, then the results of CSA-X is significantly better than the underlying MSA program. Finally, CSA-X is a modularized tool, and it allows the user to change the underlying multiple sequence alignment program if more efficient programs become available, or a specialized program is required.

However, there are future directions for CSA-X. At present, CSA-X supports only a single regular expression constraint as an input, and this should be extended to support multiple constraints in the future. Though multiple regular expressions can be combined to a single regular expression using the concatenation operator and quantifiers, this is less intuitive than allowing multiple regular expressions. In addition, CSA-X handles multiple regular expression matches by considering all possible alignments of regular expression matched segments. Although this functionality is ideal, the current implementation increases the computational time exponentially as the number of repeated matches increases. This will not be too onerous if there are not many matches (for example, if the regular expression is long enough, there will not likely be very many matches), but is not efficient in general. A good heuristic filtering stage to identify the proper combination of regular expression matched segments can also be designed; this could possibly not compromise the alignment accuracy but reduce the computational time significantly. Restricting CSA-X to specify constraints with only regular expressions can be considered as a limitation. More generalized frameworks that accept constraints in various formats such as regular expressions, context free grammars, and anchor-points can be a significant improvement for the constrained multiple sequence alignment framework. Finally, although the time taken for CSA-X in the verification was comparable to RE-MuSiC overall, a more thorough examination of time complexity is left as future work.

References

[1] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[2] Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

[3] Fabrice Armougom, Sebastien Moretti, Olivier Poirot, Stephane Audic, Pierre Dumas, Basile Schaeli, Vladimir Keduas, and Cedric Notredame. Expresso: automatic incorporation of structural information in multiple sequence alignments using 3D-Coffee. *Nucleic Acids Research*, 34(suppl 2):W604–W608, 2006.

[4] Abdullah N. Arslan. Multiple sequence alignment containing a sequence of regular expressions. In *Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB05)*, pages 1–7. IEEE, 2005.

[5] Abdullah N. Arslan. Regular expression constrained sequence alignment. *Journal of Discrete Algorithms*, 5(4):647–661, 2007.

[6] Abdullah N. Arslan. Sequence alignment guided by common motifs described by context free grammars. *Biotechnology and Bioinformatics Symposium (BIOT)*, 2007.

[7] Humberto Carrillo and David Lipman. The multiple sequence alignment problem in biology. *SIAM Journal on Applied Mathematics*, 48(5):1073–1082, 1988.

[8] Yun-Sheng Chung, Wei-Hsun Lee, Chuan Yi Tang, and Chin Lung Lu. RE-MuSiC: A tool for multiple sequence alignment with regular expression constraints. *Nucleic Acids Research*, 35(suppl 2):W639–W644, 2007.

[9] Yun-Sheng Chung, Chin Lung Lu, and Chuan Yi Tang. Efficient algorithms for regular expression constrained sequence alignment. In *Combinatorial Pattern Matching*, pages 389–400. Springer, 2006.

[10] S Clancy. DNA transcription. *Nature Education*, 1(1):41, 2008.

[11] Suzanne Clancy and William Brown. Translation: DNA to mRNA to protein. *Nature Education*, 2008.

[12] J-P Comet and Jacques Henry. Pairwise sequence alignment using a PROSITE pattern-derived similarity score. *Computers & Chemistry*, 26(5):421–436, 2002.

[13] John S Cottrell and U London. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20(18):3551–3567, 1999.

[14] M. O. Dayhoff and R. M. Schwartz. Chapter 22: A model of evolutionary change in proteins. In *Atlas of Protein Sequence and Structure*. National Biomedical Research Foundation, 1978.

[15] Chuong B Do, Mahathi SP Mahabhashyam, Michael Brudno, and Serafim Batzoglou. ProbCons: probabilistic consistency-based multiple sequence alignment. *Genome Research*, 15(2):330–340, 2005.

[16] Zhihua Du and Feng Lin. Pattern-constrained multiple polypeptide sequence alignment. *Computational Biology and Chemistry*, 29(4):303–307, 2005.

[17] Robert C Edgar. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5):1792–1797, 2004.

[18] Da-Fei Feng and Russell F Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution*, 25(4):351–360, 1987.

[19] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937.

[20] Angelika Görg, Walter Weiss, and Michael J Dunn. Current two-dimensional electrophoresis technology for proteomics. *Proteomics*, 4(12):3665–3685, 2004.

[21] Osamu Gotoh. Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. *Journal of Molecular Biology*, 264(4):823–838, 1996.

[22] Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.

[23] Steven Henikoff, Jorja G. Henikoff, and Shmuel Pietrokovski. Blocks+: a non-redundant database of protein alignment blocks derived from multiple compilations. *Bioinformatics*, 15(6):471–479, 1999.

[24] Liisa Holm and Chris Sander. Touring protein fold space with Dali/FSSP. *Nucleic Acids Research*, 26(1):316–319, 1998.

[25] Nicolas Hulo, Amos Bairoch, Virginie Bulliard, Lorenzo Cerutti, Edouard De Castro, Petra S Langendijk-Genevaux, Marco Pagni, and Christian JA Sigrist. The PROSITE database. *Nucleic Acids Research*, 34(suppl 1):D227–D230, 2006.

[26] Neil C Jones and Pavel Pevzner. *An Introduction to Bioinformatics Algorithms*. MIT press, 2004.

[27] Kazutaka Katoh, Kazuharu Misawa, Kei-ichi Kuma, and Takashi Miyata. MAFFT: a novel method for rapid multiple sequence alignment based on fast fourier transform. *Nucleic Acids Research*, 30(14):3059–3066, 2002.

[28] Sudhir Kumar and Alan Filipski. Multiple sequence alignment: in pursuit of homologous DNA positions. *Genome Research*, 17(2):127–135, 2007.

[29] Timo Lassmann and Erik LL Sonnhammer. Kalign — an accurate and fast multiple sequence alignment algorithm. *BMC Bioinformatics*, 6(1):298, 2005.

[30] Eva K Lee and Kapil Gupta. Algorithms for genomics analysis. In *Handbook of Optimization in Medicine*, pages 1–33. Springer, 2009.

[31] Arthur M Lesk and Cyrus Chothia. How different amino acid sequences determine similar protein structures: the structure and evolutionary dynamics of the globins. *Journal of Molecular Biology*, 136(3):225–270, 1980.

[32] Aron Marchler-Bauer, Shennan Lu, John B Anderson, Faridesh Chitsaz, Myra K. Derbyshire, Jessica H. Fong, Reneta C. Geer, Noreen R. Gonzalez, Praveen F Cherukuri, Carol DeWeese-Scott, Lewis Y Geer, Marc Gwadz, Siqian He, David I Hurwitz, John D Jackson, Zhaoxi Ke, Christopher J. Lanczycki, Fu Lu, Gabriele H. Marchler, Mikhail Mullokandov, Marina V. Omelchenko, Cynthia L. Robertson, James S. Song, Narmada Thanki, Roxanne A. Yamashita, Dachun Zhang, Naigong Zhang, Chanjuan Zheng, and Stephen H. Bryant. CDD: a conserved domain database for protein classification. *Nucleic Acids Research*, 33(suppl 1):D192–D196, 2005.

[33] Elaine R Mardis. Next-generation DNA sequencing methods. *Annual Review of Genomics and Human Genetics*, 9:387–402, 2008.

[34] Marc A Martí-Renom, Ashley C Stuart, András Fiser, Roberto Sánchez, Francisco Melo, and Andrej Šali. Comparative protein structure modeling of genes and genomes. *Annual Review of Biophysics and Biomolecular Structure*, 29(1):291–325, 2000.

[35] Burkhard Morgenstern, Nadine Werner, Sonja J Prohaska, Rasmus Steinkamp, Isabelle Schneider, Amarendran R Subramanian, Peter F Stadler, and Jan Weyer-Menkhoff. Multiple sequence alignment with user-defined constraints at GOBICS. *Bioinformatics*, 21(7):1271–1273, 2005.

[36] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[37] Cédric Notredame, Desmond G Higgins, and Jaap Heringa. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology*, 302(1):205–217, 2000.

[38] Fabiano Sviatopolk-Mirsky Pais, Patrícia de Ruy, Guilherme Oliveira, and Roney Coimbra. Assessing the efficiency of multiple sequence alignment programs. *Algorithms for Molecular Biology*, 9(4), 2014.

[39] Jason S Papadopoulos and Richa Agarwala. COBALT: constraint-based alignment tool for multiple protein sequences. *Bioinformatics*, 23(9):1073–1079, 2007.

[40] Jong Park, Kevin Karplus, Christian Barrett, Richard Hughey, David Haussler, Tim Hubbard, and Cyrus Chothia. Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *Journal of Molecular Biology*, 284(4):1201–1210, 1998.

[41] Jonathan Pevsner. *Bioinformatics and Functional Genomics*. John Wiley & Sons, 2005.

[42] L Pray. DNA replication and causes of mutation. *Nature Education*, 1(1):214, 2008.

[43] Frederick Sanger. The free amino groups of insulin. *Biochemical Journal*, 39(5):507, 1945.

[44] Ilya N Shindyalov and Philip E Bourne. Protein structure alignment by incremental combinatorial extension (ce) of the optimal path. *Protein Engineering*, 11(9):739–747, 1998.

[45] Victor A Simossis and Jaap Heringa. PRALINE: a multiple sequence alignment toolbox that integrates homology-extended and secondary structure information. *Nucleic Acids Research*, 33(suppl 2):W289–W294, 2005.

[46] Temple F Smith and Michael S Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

[47] Qiaojuan Jane Su, Lin Lu, Serge Saxonov, and Douglas L Brutlag. eBLOCKs: enumerating conserved protein blocks to achieve maximal sensitivity and specificity. *Nucleic Acids Research*, 33(suppl 1):D178–D182, 2005.

[48] Amarendran R Subramanian, Jan Weyer-Menkhoff, Michael Kaufmann, and Burkhard Morgenstern. DIALIGN-T: an improved algorithm for segment-based multiple sequence alignment. *BMC Bioinformatics*, 6(1):66, 2005.

[49] Chuan Yi Tang, Chin Lung Lu, Margaret Dah-Tsyr Chang, Yin-Te Tsai, Yuh-Ju Sun, Kun-Mao Chao, Jia-Ming Chang, Yu-Han Chiou, Chia-Mao Wu, Hao-Teng Chang, and Wei-I Chou. Constrained multiple sequence alignment tool development and its application to RNase family alignment. *Journal of Bioinformatics and Computational Biology*, 1(02):267–287, 2003.

[50] Yin Te Tsai, Yen Pin Huang, Ching Ta Yu, and Chin Lung Lu. MuSiC: a tool for multiple sequence alignment with constraints. *Bioinformatics*, 20(14):2309–2311, 2004.

[51] Julie D Thompson, Desmond G Higgins, and Toby J 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.

[52] Julie D Thompson, Patrice Koehl, Raymond Ripp, and Olivier Poch. BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. *Proteins: Structure, Function, and Bioinformatics*, 61(1):127–136, 2005.

[53] Julie D Thompson, Frédéric Plewniak, and Olivie Poch. A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Research*, 27(13):2682–2690, 1999.

[54] Julie D. Thompson, Frédéric Plewniak, and Olivier Poch. BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, 15(1):87–88, 1999.

[55] Marc M Triola and Mario F Triola. *Biostatistics for the Biological and Health Sciences*. Pearson Addison-Wesley Boston, 2006.

Appendix A

Constraints used with CSA-X

The evaluation of CSA-X in Chapter 6 is tested with a set of regular expressions. These are obtained from the most conserved regions and the least conserved regions of the correct alignments of the working database (adapted from the BAliBASE 3.0 benchmark database). A Perl script is used to identify the most conserved and the least conserved regions of the alignments, and later these regions are converted into the hash-augmented regular expressions in a column-by-column fashion. In order to generate the regular expression sets, the script is provided with the length of the hash-augmented regular expression constraints (number of '#' symbols in the constraints) and the number of allowable gaps to exist in the identified most conserved or least conserved regions. Having this information, the Perl script generates the hash-augmented regular expression sets from the most conserved regions and the least conserved regions of the alignments. The list of hash-augmented regular expression constraints used with CSA-X for conducting the benchmark studies are as follows:

BB11001	([WY])#([KH])#([TMEA])#([ML])#([STP])#([AER])#([KAE])#([EK])#([KQ])#([GWMA])#([KP])#([FY])
BB11002	([QERILTK])#([GK])#([WYDV])#([VF])#([P])#([SRGEA])#([NTRSM])#([YLKFIH])#([ILVAT])#([TGEKRQ])#([PLYIREK])#([VYIDSL])
BB11003	([AP])#([VL])#([AK])#([AVS])#([G])#([NC])#([ATP])#([VI])#([IL])#([LAV])#([KR])#([PG])
BB11004	([IWQ])#([ELV])#([EA])#([NP])#([LVI])#([DQK])#([LV])#([FVA])#([IV])#([VMI])#([TNP])#([MIL])
BB11005	([VLMFI])#([LFIVYA])#([VLIAS])#([TCENPH])#([NTSPVA])#([PITC])#([VHCA]?)#([SNQLVH])#([NGSR])#([PASNE])#([LTCGS])#([G])
BB11006	([NQSA])#([STMNP])#([KSATQLC])#([FCVAIT]?)#([TSHR])#([FMIL])#([SVTA])#([IVL])#([P])#([YWF])#([LIVNT])#([SGAN])
BB11007	([HR])#([MFLAR])#([C])#([LMIVP])#([G])#([QREAS])#([HPQGAD])#([LFI])#([ATG])#([KRLI])#([LRHME])#([EQVH])
BB11008	([DP])#([G])#([TL])#([F])#([LI])#([VIL])#([R])#([DEF])#([AS])#([SQE])#([TRS])#([KNS])
BB11009	([SAD])#([C])#([E]?)#([RKE])#([ASE])#([GC])#([SAET])#([C])#([STG])#([ST])#([C])#([AHRK])
BB11010	([GSN])#([D])#([ACS])#([E])#([LVI])#([VIL])#([LA])#([RALN])#([LI])#([LYF])#([EQTA])#([RES])
BB11011	([LVCG])#([PQAR])#([G])#([MND])#([CSXA])#([G])#([GSR])#([AGPS])#([LIV])#([VFM])#([SNCAD])#([SGN])
BB11012	([LIA])#([YWH])#([FA])#([NK])#([GA])#([QRS])#([W])#([KTV])#([TN])#([PK])#([F])#([PED])
BB11013	([F]?)#([NSVM])#([RS])#([WRLQ])#([AQHGR])#([ES])#([IML])#([AQSVK])#([KRAN])#([LYKAE])#([LI])#([PKSQG])
BB11014	([ED])#([LVI])#([E])#([SNR])#([IHAV])#([L])#([LVGM])#([QKATG])#([HA])#([PM])#([NYAKG])#([IV])
BB11015	([VI])#([VI])#([TSG])#([N])#([P])#([VA])#([DN])#([VT])#([MINT])#([VTA])#([QAY])#([LI])
BB11016	([TSQN])#([SITAD])#([MVEHCRI])#([HPKS])#([NDG])#([VI])#([WYFH])#([AVL])#([IVLA])#([G])#([D])#([VLA])
BB11017	([TV])#([ATS])#([E])#([VA])#([ALS])#([R])#([FY])#([RXK])#([YQF])#([I])#([QE])#([NRQ])
BB11018	([NGD])#([IMVLX])#([KYRGNS])#([VLIA])#([IMYVL])#([IVLMQGY])#([D])#([FVAWL])#([AVIP])#([PIYFL])#([NDSGH])#([H])
BB11019	([NPGAQ])#([MVILASF])#([IVPGA])#([LAICPD])#([G])#([HNTMI]?)#([EDG])#([AGIVF])#([VASTLI])#([GF])#([EITRV])#([V])
BB11020	([FWY])#([FIVL])#([VCAGD])#([GT])#([NQDE])#([SRKGNE])#([MFIVL])#([TGS])#([LIYM])#([AVI])#([D])#([LAFVYC])
BB11021	([PGQ])#([H])#([N])#([VIW])#([HEV])#([FTL])#([VID])#([AKIN])#([GP])#([VMAG])#([LAGD])#([GPAD])
BB11022	([QK])#([LR]?)#([GK])#([LY])#([NSTP])#([QLN])#([ASTK])#([EANM])#([LVI])#([ASG])#([QRE])#([KQAR])
BB11023	([HQAKE])#([WF])#([KAET])#([QK])#([EQDK])#([TKY])#([P]?)#([G])#([DVHI])#([NDAKT])#([V])#([VTIE])
BB11024	([P])#([LQI])#([RLK])#([MLY])#([ANY])#([EQYG])#([FWNY])#([GA])#([SNP])#([CVMA])#([HVF])#([R])
BB11025	([KFSD])#([NTK])#([IV])#([QKL])#([SEK])#([LVY])#([EAWT])#([VLRE])#([IKVS])#([GLM])#([KA])#([G]?)
BB11026	([ILD])#([EHVL])#([NDE])#([VCA])#([KLAI])#([AMRELT])#([KSERAD])#([IAQN])#([QL]?)#([DKQGRN])#([KVLIQ])#([ERTDN])
BB11027	([IML])#([FV])#([VLIM])#([LISEV])#([G])#([GPAVL])#([PSHDE])#([GRA])#([ATVR])#([G])#([KR])#([GSRT])
BB11028	([ADSG])#([DAN])#([SDATKQE])#([GADH])#([NKTEISR])#([YR])#([KTVSLFW])#([LCV])#([KVEISTAQ])#([VAGI])#([KVSRTEY])#([NCADG])
BB11029	([KST])#([EDA])#([KAG])#([DQ])#([RIA])#([NAKE])#([DNA])#([LV])#([IAV])#([TAD])#([YW])#([LYI])
BB11030	([SRQKTGEA])#([VCILRPA])#([DLE])#([GVAILST])#([LVI])#([VLIF])#([NSHC])#([NSVGFYM])#([ADIG])#([GSIAE])#([ISFGTLQE])#([SFALMWGRKP])
BB11031	([AXIL])#([QAKSIR])#([KQSNEDW])#([D]?)#([MWHSLKA])#([GD])#([LFAYMI])#([KND])#([IFTVHA])#([AVIF])#([R])#([LIVAT])
BB11032	([FILVM])#([IWGVL])#([LMI])#([VLI])#([AGS])#([GT])#([G])#([ITV])#([GA])#([ILFV])#([TAGSP])#([PY])
BB11033	([QKDMT]?)#([PMLVSTY])#([VQRGILS])#([LTVIFM])#([VLFR])#([YIDVNE])#([FYILV])#([WGSVY])#([ARTE])#([STPD])#([WGTV])#([CTV])
BB11034	([TASN])#([NSTRID])#([VECSHI])#([KP]?)#([GHND])#([IV])#([YFWH])#([AVL])#([VILA])#([G])#([D])#([VILA])
BB11035	([EAQG])#([EDQ])#([LIRA])#([KAE])#([AND])#([LV])#([AV])#([DA])#([YW])#([MYL])#([SAM])#([KSTE])
BB11036	([GAV])#([VIL])#([STDE])#([LCQSM])#([A])#([ALI])#([SALW])#([RED])#([AL])#([ATVLQ])#([ADG])#([AKRQ]?)
BB11037	([RL])#([LV])#([WF])#([WC])#([TSAYC])#([VAIC])#([YF])#([MCYSL])#([FVTL])#([ED])#([RVK])#([MLF])
BB11038	([VIA])#([YH])#([RT])#([D])#([IL])#([KR])#([PAS])#([DASEK])#([N])#([FILV])#([LA])#([IVLM])
BB12001	([P])#([LV])#([IVAL])#([ILAM])#([W])#([LFT])#([NTQ])#([G])#([G])#([P])#([G])#([CG])
BB12002	([LMV])#([G])#([PVTQ])#([AIVGH])#([LM])#([GNH])#([PKR])#([RGV])#([G])#([KL])#([FM])#([P])
BB12003	([NLMFVS])#([ASGK])#([C])#([YW])#([C])#([YNEQI])#([KAGDY])#([LAV])#([PE])#([DEKN])#([HNDS])#([VAKE])
BB12004	([AV])#([Y])#([E])#([P])#([VLIES])#([WS])#([A])#([I])#([GN])#([DK]?)#([TKAS])#([GSD])
BB12005	([LINV])#([D])#([MTVAI])#([WY])#([E])#([H])#([AS])#([FY])#([YH])#([LVMY])#([QDR])#([YF])
BB12006	([LM])#([VLI])#([SA])#([Y])#([AT])#([P])#([P])#([G])#([AMG])#([DEG])#([PMK])#([P])
BB12007	([C])#([E])#([Y])#([AMSGV])#([H])#([A])#([M])#([G])#([N])#([SG])#([LVPN])#([G])
BB12008	([ASC])#([FLYVMI])#([GACLM])#([LMFAIV])#([TS])#([E])#([PRA])#([NGQDA])#([ASV])#([G])#([TS])#([DN])
BB12009	([K])#([WTQ])#([F])#([NDS])#([SNTGR])#([EAQKS])#([KSQ])#([G])#([FKH])#([G])#([FL])#([I])
BB12010	([LV])#([HN])#([Y])#([SG])#([LQT])#([QSGA])#([LVCA])#([FY])#([EG])#([G])#([MLI])#([KQR])
BB12011	([Y])#([TSI])#([DTEAN])#([YH])#([ACS])#([VTI])#([RKQNEHST])#([WYTV])#([Y])#([NQDT])#([TVADRK])#([G])
BB12012	([D])#([VI])#([VAT])#([AG])#([H])#([E])#([ILV])#([TS])#([H])#([G])#([VIF])#([T])
BB12013	([VSI])#([W])#([NI])#([YF])#([HQ])#([CVT])#([WS])#([NVT])#([ED])#([ASGCV])#([W])#([MF])
BB12014	([RQV])#([IV])#([QKR])#([VINT])#([W])#([FVY])#([QSI])#([N])#([RHK])#([R])#([AMYRCT])#([RK])
BB12015	([G])#([WFL])#([ETD])#([EKTQAIV])#([GAH])#([VLI])#([AQLIEFV])#([QLDKNGT])#([MARKLF])#([SPEQKA])#([VAIEK])#([G])
BB12016	([VA])#([L])#([LV])#([D])#([T])#([G])#([AV])#([D])#([TDR])#([ST])#([VI])#([LVI])
BB12017	([QEYK])#([PF])#([YFIVL])#([VCQL])#([TN])#([LM])#([FYHS])#([H])#([WFY])#([DPE])#([VLTM])#([P])
BB12018	([G])#([G])#([L])#([G])#([R])#([L])#([A])#([AS])#([C])#([F])#([LI])#([D])
BB12019	([P])#([P])#([E])#([P])#([NS])#([G])#([YI])#([L])#([H])#([I])#([G])#([H])
BB12020	([G])#([STE])#([VR])#([VI])#([TRDE])#([Y])#([SKE])#([C])#([NSKR])#([SGKP])#([GT])#([YF])

BB12021	([LETS])#([VI])#([SNKVE])#([AVEP])#([NEKDA])#([G])#([W])#([C])#([TASLQ])#([ASV])#([WY])#([VTA])
BB12022	([N])#([P])#([D])#([GRN])#([DER])#([ELRV])#([ERQS])#([G]?)#([AP])#([W])#([C])#([YF])
BB12023	([S])#([C])#([H])#([T])#([G])#([LVI])#([RGN])#([RK])#([TSN])#([AV])#([G])#([WY])
BB12024	([HR])#([I])#([GE])#([I])#([D])#([VI])#([N])#([S])#([IL])#([VRI])#([SP])#([VIT])
BB12025	([VA])#([TS])#([L])#([GTV])#([C])#([LT])#([VIA])#([KTR])#([GD])#([YF])#([FY])#([P])
BB12026	([FVILM])#([ILV])#([G])#([IV])#([N])#([SNTA])#([R])#([DNS])#([L])#([EADKCRHG])#([TRDK])#([LF])
BB12027	([AGTS])#([IVLK])#([AMLV])#([LQVGIM])#([D])#([TL])#([KQPN])#([G])#([PALI])#([EK])#([ILVM])#([R])
BB12028	([YF])#([WI])#([LI])#([VAI])#([AKR])#([N])#([SF])#([W])#([NGT])#([AKSPET])#([DSQPG])#([W])
BB12029	([FL])#([QRKFV])#([PL])#([VSATDG])#([YH])#([FNY])#([P])#([FY])#([VT])#([ES])#([P])#([GS])
BB12030	([WY])#([TNK])#([RV])#([L])#([P])#([Q])#([G])#([FWM])#([KVAT])#([NGLC])#([S])#([P])
BB12031	([D])#([Y])#([ST])#([Q])#([I])#([E])#([LM])#([RVA])#([VIL])#([LM])#([AS])#([H])
BB12032	([GQE])#([KQEG])#([NDTKH])#([LISNYV])#([C])#([YF])#([KLRT])#([MKNEYR])#([FASIRTW])#([MQRWTH])#([VCRTM])#([ADHFVEI])
BB12033	([G])#([T])#([S])#([AMS])#([AS])#([ASTC])#([P])#([LHG])#([AV])#([AS])#([G])#([VILA])
BB12034	([R])#([S])#([CN])#([D])#([VMI])#([FALP])#([LV])#([GA])#([LNH])#([PHN])#([FI])#([N])
BB12036	([MLSV])#([G])#([LPAG])#([P])#([GA])#([AS])#([G])#([K])#([G])#([T])#([QV])#([ASC])
BB12037	([WF])#([VLI])#([HKEQLA])#([DRESQK])#([NH])#([IAV])#([QHEGAVSK])#([FANVS])#([F])#([G])#([G])#([DNE])
BB12038	([WYFV])#([TRI])#([YF])#([PRLASENH])#([G])#([S])#([L])#([T])#([TS])#([P])#([PT])#([LC])
BB12039	([A]?)#([VADSTEC])#([C])#([EKVSI])#([PDEQGNT])#([ELVYMTNQ])#([C])#([P])#([NTSVQAMI])#([GVEDASN])#([ASCV])#([ILF])
BB12040	([H])#([KR])#([KEL])#([ETI])#([H])#([DE])#([GKISL])#([F])#([IV])#([NQKR])#([ADTKR])#([LAV])
BB12041	([VIAE])#([KAES])#([C])#([DQVN])#([DQTKSA])#([C])#([H])#([H])#([DTLKMV])#([PVWLF])#([GPDEV])#([DGAN])
BB12042	([G])#([IV])#([IVM])#([LI])#([T])#([APG])#([SA])#([H])#([N])#([P])#([GP])#([GED])
BB12044	([E])#([YFH])#([VT])#([S])#([AV])#([N])#([P])#([TN])#([GK])#([PDFYE])#([MLI])#([HN])
BB20001	([PKI])#([RK])#([GRPSK])#([KPIRA]?)#([MLPVIT])#([SETN])#([SAG])#([YHF])#([AMNFL])#([FLNYVQ])#([FWTY])#([VLKSMF])
BB20002	([GNRDKSA]?)#([HSQKRNED])#([ETIVLQKR])#([GLK])#([YCFWELIV])#([AIFV])#([PG])#([SAWYRET])#([STNVDR])#([YFPLIK])#([LVSIA])#([VATQNERGK])
BB20020	([QT])#([SMTCV])#([TIVA])#([SA])#([E])#([AV])#([ASL])#([R])#([YF])#([KQNRT])#([FYPQ])#([I])
BB30006	([DNHEPS])#([G])#([TKSAHNLVD])#([FY])#([LM])#([VILA])#([R])#([DAEQPKF])#([ARSC])#([SDEKNART])#([TNSER])#([KNSAVPHT])
BB30017	([AGQLENH])#([FW])#([ETAKS])#([QRKES])#([EADQK])#([TKN])#([G])#([G]?)#([IHVDQ])#([KADTEN])#([VL])#([TKVIR])
BB30027	([EAQTK]?)#([EDQNK])#([LMIARQV])#([KEHAN])#([AVQNDL])#([LMIV])#([AVI])#([DTEA])#([YHWF])#([MIFLY])#([SGAKVE])#([KTSENQ])
BB40003	([P])#([VIL])#([IVL])#([LI])#([W])#([L])#([NT])#([G])#([G])#([P])#([G])#([C])
BB40005	([R])#([GACKRN])#([FM])#([THQ])#([QML])#([DPN])#([D])#([AMG])#([H])#([IT])#([FIL])#([CVA])
BB40006	([LVMIF])#([VFYLI])#([VLIF])#([NCTIG])#([STN])#([P])#([NQSGH])#([N])#([P])#([TLSI])#([GA])#([ALQKTVR])
BB40007	([HQ])#([V])#([Q])#([CIV])#([N])#([ASPV])#([STN])#([KQP])#([FG])#([HQT])#([QA])#([G])
BB40008	([ASCT])#([FLYM])#([CAGM])#([LMVAQ])#([TS])#([E])#([PALM])#([AGNS])#([SAH])#([G])#([ST])#([DNH])
BB40009	([LITVK])#([GASTP])#([FY])#([GS])#([FYQWHSRNAG])#([G])#([PDIASVQKR])#([HR])#([RAFMLHVG])#([C])#([IPMLHV])#([AG])
BB40010	([K])#([WTQ])#([FY])#([NSD])#([SADTR])#([EDQTKS])#([KSQ])#([GN])#([FYKH])#([G])#([F])#([I])
BB40014	([N])#([A])#([AFY])#([W])#([INDY])#([GN])#([DSQTERG])#([KQA])#([M])#([IVLT])#([YF])#([G])
BB40018	([LVI])#([LI])#([DN])#([T])#([G])#([AV])#([D])#([TDIVKA])#([ST])#([VI])#([LVI])#([TEANSK])
BB40019	([YIF])#([R])#([FILT])#([S])#([IV])#([SA])#([W])#([PST])#([R])#([VIL])#([LFV])#([P])
BB40022	([C])#([R])#([NS])#([P])#([DRG])#([GRANS])#([DERSVQA])#([ELIRAVKG])#([ERGTKQSN])#([GSA]?)#([AP])#([W])
BB40025	([Q])#([SMT])#([TIAV])#([SPAT])#([E])#([AV])#([AS])#([R])#([YF])#([KQNRT])#([FYP])#([I])
BB40032	([L])#([P])#([Q])#([G])#([FWM])#([KAL])#([NG])#([S])#([P])#([TA])#([LI])#([FC])
BB40033	([KAPSTVN])#([EFYWH])#([ERKNQHTA])#([NDAE])#([L])#([P])#([ILQYVM])#([YRLKI])#([LMYFIV])#([TVACSNY])#([ACGEDQ])#([YSLFWIV])
BB40043	([LVFMI])#([WYGALIV])#([MFICL])#([LFIV])#([ASG])#([TG])#([G])#([TSV])#([AG])#([ILFM])#([GATS])#([PY])
BB40045	([VIALEQG])#([KAEDS])#([C])#([DQVGTN])#([DQTKSA])#([C])#([H])#([HTA])#([DTLPKMV])#([PVWLFD])#([GPDNEVI])#([DGANEK])
BB40048	([P])#([P])#([ED])#([PA])#([NS])#([G])#([YI])#([LA])#([H])#([IL])#([G])#([H])
BB50002	([VAITYL])#([GAT])#([RFHLKA])#([PVILG])#([NSDGE])#([VAHSI])#([GI])#([KR])#([STG])#([TS])#([LTAI])#([LFTV])
BB50004	([D])#([VLIY])#([YF])#([IV])#([NSMDI])#([D])#([A])#([FY])#([GA])#([TAV])#([AI])#([H])
BB50005	([C])#([H])#([TS])#([G])#([LIV])#([RGSN])#([RK])#([TSN])#([ADV])#([G])#([WY])#([NK])
BB50010	([P])#([ILQYVMF])#([YRLKI])#([LMYFIVR])#([TVACSNY])#([ACGEDQY])#([YSLFWIVN])#([TSAGCNV])#([PTLNSCQK])#([CAVLTM])#([YFHV])#([R])
BB50013	([GHN])#([NDS])#([TCLAIM])#([H])#([IVL])#([Y])#([MTSLNVKD])#([NR])#([HQ])#([VIFLY])#([EDYN])#([QNLAKG])
BB50016	([IVALCM])#([HY])#([RCT])#([D])#([ILV])#([K])#([GPSA])#([ASKEQH])#([N])#([ICVL])#([LMI])#([IVL])
BBS11001	([WY])#([KH])#([TMEA])#([ML])#([STP])#([AER])#([KAE])#([EK])#([KQ])#([GWMA])#([KP])#([FY])
BBS11002	([QERILTK])#([GK])#([WYDV])#([VF])#([P])#([SRGEA])#([NTRSM])#([YLKFIH])#([ILVAT])#([TGEKRQ])#([PLYIREK])#([VYIDSL])
BBS11003	([AP])#([VL])#([AK])#([AVS])#([G])#([NC])#([ATP])#([VI])#([IL])#([LAV])#([KR])#([PG])
BBS11004	([IWQ])#([ELV])#([EA])#([NP])#([LVI])#([DQK])#([LV])#([FVA])#([IV])#([VMI])#([TNP])#([MIL])
BBS11005	([VLMFI])#([LFIVYA])#([VLIAS])#([TCENPH])#([NTSPVA])#([PITC])#([VHCA]?)#([SNQLVH])#([NGSR])#([PASNE])#([LTCGS])#([G])
BBS11006	([NQSA])#([STMNP])#([KSATQLC])#([FCVAIT]?)#([TSHR])#([FMIL])#([SVTA])#([IVL])#([P])#([YWF])#([LIVNT])#([SGAN])
BBS11007	([HR])#([MFLAR])#([C])#([LMIVP])#([G])#([QREAS])#([HPQGAD])#([LFI])#([ATG])#([KRLI])#([LRHME])#([EQVH])

BBS11008	([DP])#([G])#([TL])#([F])#([LI])#([VIL])#([R])#([DEF])#([AS])#([SQE])#([TRS])#([KNS])
BBS11009	([SAD])#([C])#([E]?)#([RKE])#([ASE])#([GC])#([SAET])#([C])#([STG])#([ST])#([C])#([AHRK])
BBS11010	([GSN])#([D])#([ACS])#([E])#([LVI])#([VIL])#([LA])#([RALN])#([LI])#([LYF])#([EQTA])#([RES])
BBS11011	([LVCG])#([PQAR])#([G])#([MND])#([CSXA])#([G])#([GSR])#([AGPS])#([LIV])#([VFM])#([SNCAD])#([SGN])
BBS11012	([LIA])#([YWH])#([FA])#([NK])#([GA])#([QRS])#([W])#([KTV])#([TN])#([PK])#([F])#([PED])
BBS11013	([F]?)#([NSVM])#([RS])#([WRLQ])#([AQHGR])#([ES])#([IML])#([AQSVK])#([KRAN])#([LYKAE])#([LI])#([PKSQG])
BBS11014	([ED])#([LVI])#([E])#([SNR])#([IHAV])#([L])#([LVGM])#([QKATG])#([HA])#([PM])#([NYAKG])#([IV])
BBS11015	([VI])#([VI])#([TSG])#([N])#([P])#([VA])#([DN])#([VT])#([MINT])#([VTA])#([QAY])#([LI])
BBS11016	([TSQN])#([SITAD])#([MVEHCRI])#([HPKS])#([NDG])#([VI])#([WYFH])#([AVL])#([IVLA])#([G])#([D])#([VLA])
BBS11017	([TV])#([ATS])#([E])#([VA])#([ALS])#([R])#([FY])#([RXK])#([YQF])#([I])#([QE])#([NRQ])
BBS11018	([NGD])#([IMVLX])#([KYRGNS])#([VLIA])#([IMYVL])#([IVLMQGY])#([D])#([FVAWL])#([AVIP])#([PIYFL])#([NDSGH])#([H])
BBS11019	([NPGAQ])#([MVILASF])#([IVPGA])#([LAICPD])#([G])#([HNTMI]?)#([EDG])#([AGIVF])#([VASTLI])#([GF])#([EITRV])#([V])
BBS11020	([FWY])#([FIVL])#([VCAGD])#([GT])#([NQDE])#([SRKGNE])#([MFIVL])#([TGS])#([LIYM])#([AVI])#([D])#([LAFVYC])
BBS11021	([PGQ])#([H])#([N])#([VIW])#([HEV])#([FTL])#([VID])#([AKIN])#([GP])#([VMAG])#([LAGD])#([GPAD])
BBS11022	([QK])#([LR]?)#([GK])#([LY])#([NSTP])#([QLN])#([ASTK])#([EANM])#([LVI])#([ASG])#([QRE])#([KQAR])
BBS11023	([HQAKE])#([WF])#([KAET])#([QK])#([EQDK])#([TKY])#([P]?)#([G])#([DVHI])#([NDAKT])#([V])#([VTIE])
BBS11024	([P])#([LQI])#([RLK])#([MLY])#([ANY])#([EQYG])#([FWNY])#([GA])#([SNP])#([CVMA])#([HVF])#([R])
BBS11025	([KFSD])#([NTK])#([IV])#([QKL])#([SEK])#([LVY])#([EAWT])#([VLRE])#([IKVS])#([GLM])#([KA])#([G]?)
BBS11026	([ILD])#([EHVL])#([NDE])#([VCA])#([KLAI])#([AMRELT])#([KSERAD])#([IAQN])#([QL]?)#([DKQGRN])#([KVLIQ])#([ERTDN])
BBS11027	([IML])#([FV])#([VLIM])#([LISEV])#([G])#([GPAVL])#([PSHDE])#([GRA])#([ATVR])#([G])#([KR])#([GSRT])
BBS11028	([ADSG])#([DAN])#([SDATKQE])#([GADH])#([NKTEISR])#([YR])#([KTVSLFW])#([LCV])#([KVEISTAQ])#([VAGI])#([KVSRTEY])#([NCADG])
BBS11029	([KST])#([EDA])#([KAG])#([DQ])#([RIA])#([NAKE])#([DNA])#([LV])#([IAV])#([TAD])#([YW])#([LYI])
BBS11030	([SRQKTGEA])#([VCILRPA])#([DLE])#([GVAILST])#([LVI])#([VLIF])#([NSHC])#([NSVGFYM])#([ADIG])#([GSIAE])#([ISFGTLQE])#([SFALMWGRKP])
BBS11031	([AXIL])#([QAKSIR])#([KQSNEDW])#([D]?)#([MWHSLKA])#([GD])#([LFAYMI])#([KND])#([IFTVHA])#([AVIF])#([R])#([LIVAT])
BBS11032	([FILVM])#([IWGVL])#([LMI])#([VLI])#([AGS])#([GT])#([G])#([ITV])#([GA])#([ILFV])#([TAGSP])#([PY])
BBS11033	([PMLVSTY])#([VQRGILS])#([LTVIFM])#([VLFR])#([YIDVNE])#([FYILV])#([WGSVY])#([ARTE])#([STPD])#([WGTV])#([CTV])#([G]?)
BBS11034	([TASN])#([NSTRID])#([VECSHI])#([KP]?)#([GHND])#([IV])#([YFWH])#([AVL])#([VILA])#([G])#([D])#([VILA])
BBS11035	([EAQG])#([EDQ])#([LIRA])#([KAE])#([AND])#([LV])#([AV])#([DA])#([YW])#([MYL])#([SAM])#([KSTE])
BBS11036	([GAV])#([VIL])#([STDE])#([LCQSM])#([A])#([ALI])#([SALW])#([RED])#([AL])#([ATVLQ])#([ADG])#([AKRQ]?)
BBS11037	([RL])#([LV])#([WF])#([WC])#([TSAYC])#([VAIC])#([YF])#([MCYSL])#([FVTL])#([ED])#([RVK])#([MLF])
BBS11038	([VIA])#([YH])#([RT])#([D])#([IL])#([KR])#([PAS])#([DASEK])#([N])#([FILV])#([LA])#([IVLM])
BBS12001	([P])#([LV])#([IVAL])#([ILAM])#([W])#([LFT])#([NTQ])#([G])#([G])#([P])#([G])#([CG])
BBS12002	([LMV])#([G])#([PVTQ])#([AIVGH])#([LM])#([GNH])#([PKR])#([RGV])#([G])#([KL])#([FM])#([P])
BBS12003	([NLMFVS])#([ASGK])#([C])#([YW])#([C])#([YNEQI])#([KAGDY])#([LAV])#([PE])#([DEKN])#([HNDS])#([VAKE])
BBS12004	([AV])#([Y])#([E])#([P])#([VLIES])#([WS])#([A])#([I])#([GN])#([DK]?)#([TKAS])#([GSD])
BBS12005	([LINV])#([D])#([MTVAI])#([WY])#([E])#([H])#([AS])#([FY])#([YH])#([LVMY])#([QDR])#([YF])
BBS12006	([LM])#([VLI])#([SA])#([Y])#([AT])#([P])#([P])#([G])#([AMG])#([DEG])#([PMK])#([P])
BBS12007	([C])#([E])#([Y])#([AMSGV])#([H])#([A])#([M])#([G])#([N])#([SG])#([LVPN])#([G])
BBS12008	([ASC])#([FLYVMI])#([GACLM])#([LMFAIV])#([TS])#([E])#([PRA])#([NGQDA])#([ASV])#([G])#([TS])#([DN])
BBS12009	([K])#([WTQ])#([F])#([NDS])#([SNTGR])#([EAQKS])#([KSQ])#([G])#([FKH])#([G])#([FL])#([I])
BBS12010	([LV])#([HN])#([Y])#([SG])#([LQT])#([QSGA])#([LVCA])#([FY])#([EG])#([G])#([MLI])#([KQR])
BBS12011	([Y])#([TSI])#([DTEAN])#([YH])#([ACS])#([VTI])#([RKQNEHST])#([WYTV])#([Y])#([NQDT])#([TVADRK])#([G])
BBS12012	([D])#([VI])#([VAT])#([AG])#([H])#([E])#([ILV])#([TS])#([H])#([G])#([VIF])#([T])
BBS12013	([VSI])#([W])#([NI])#([YF])#([HQ])#([CVT])#([WS])#([NVT])#([ED])#([ASGCV])#([W])#([MF])
BBS12014	([RQV])#([IV])#([QKR])#([VINT])#([W])#([FVY])#([QSI])#([N])#([RHK])#([R])#([AMYRCT])#([RK])
BBS12015	([G])#([WFL])#([ETD])#([EKTQAIV])#([GAH])#([VLI])#([AQLIEFV])#([QLDKNGT])#([MARKLF])#([SPEQKA])#([VAIEK])#([G])
BBS12016	([VA])#([L])#([LV])#([D])#([T])#([G])#([AV])#([D])#([TDR])#([ST])#([VI])#([LVI])
BBS12017	([QEYK])#([PF])#([YFIVL])#([VCQL])#([TN])#([LM])#([FYHS])#([H])#([WFY])#([DPE])#([VLTM])#([P])
BBS12018	([G])#([G])#([L])#([G])#([R])#([L])#([A])#([AS])#([C])#([F])#([LI])#([D])
BBS12019	([P])#([P])#([E])#([P])#([NS])#([G])#([YI])#([L])#([H])#([I])#([G])#([H])
BBS12020	([G])#([STE])#([VR])#([VI])#([TRDE])#([Y])#([SKE])#([C])#([NSKR])#([SGKP])#([GT])#([YF])
BBS12021	([LETS])#([VI])#([SNKVE])#([AVEP])#([NEKDA])#([G])#([W])#([C])#([TASLQ])#([ASV])#([WY])#([VTA])
BBS12022	([N])#([P])#([D])#([GRN])#([DER])#([ELRV])#([ERQS])#([G]?)#([AP])#([W])#([C])#([YF])
BBS12023	([S])#([C])#([H])#([T])#([G])#([LVI])#([RGN])#([RK])#([TSN])#([AV])#([G])#([WY])
BBS12024	([HR])#([I])#([GE])#([I])#([D])#([VI])#([N])#([S])#([IL])#([VRI])#([SP])#([VIT])
BBS12025	([VA])#([TS])#([L])#([GTV])#([C])#([LT])#([VIA])#([KTR])#([GD])#([YF])#([FY])#([P])
BBS12026	([FVILM])#([ILV])#([G])#([IV])#([N])#([SNTA])#([R])#([DNS])#([L])#([EADKCRHG])#([TRDK])#([LF])
BBS12027	([AGTS])#([IVLK])#([AMLV])#([LQVGIM])#([D])#([TL])#([KQPN])#([G])#([PALI])#([EK])#([ILVM])#([R])

BBS12028	([YF])#([WI])#([LI])#([VAI])#([AKR])#([N])#([SF])#([W])#([NGT])#([AKSPET])#([DSQPG])#([W])
BBS12029	([FL])#([QRKFV])#([PL])#([VSATDG])#([YH])#([FNY])#([P])#([FY])#([VT])#([ES])#([P])#([GS])
BBS12030	([WY])#([TNK])#([RV])#([L])#([P])#([Q])#([G])#([FWM])#([KVAT])#([NGLC])#([S])#([P])
BBS12031	([D])#([Y])#([ST])#([Q])#([I])#([E])#([LM])#([RVA])#([VIL])#([LM])#([AS])#([H])
BBS12032	([GQE])#([KQEG])#([NDTKH])#([LISNYV])#([C])#([YF])#([KLRT])#([MKNEYR])#([FASIRTW])#([MQRWTH])#([VCRTM])#([ADHFVEI])
BBS12033	([G])#([T])#([S])#([AMS])#([AS])#([ASTC])#([P])#([LHG])#([AV])#([AS])#([G])#([VILA])
BBS12034	([R])#([S])#([CN])#([D])#([VMI])#([FALP])#([LV])#([GA])#([LNH])#([PHN])#([FI])#([N])
BBS12036	([MLSV])#([G])#([LPAG])#([P])#([GA])#([AS])#([G])#([K])#([G])#([T])#([QV])#([ASC])
BBS12037	([WF])#([VLI])#([HKEQLA])#([DRESQK])#([NH])#([IAV])#([QHEGAVSK])#([FANVS])#([F])#([G])#([G])#([DNE])
BBS12038	([WYFV])#([TRI])#([YF])#([PRLASENH])#([G])#([S])#([L])#([T])#([TS])#([P])#([PT])#([LC])
BBS12039	([A]?)#([VADSTEC])#([C])#([EKVSI])#([PDEQGNT])#([ELVYMTNQ])#([C])#([P])#([NTSVQAMI])#([GVEDASN])#([ASCV])#([ILF])
BBS12040	([H])#([KR])#([KEL])#([ETI])#([H])#([DE])#([GKISL])#([F])#([IV])#([NQKR])#([ADTKR])#([LAV])
BBS12041	([VIAE])#([KAES])#([C])#([DQVN])#([DQTKSA])#([C])#([H])#([H])#([DTLKMV])#([PVWLF])#([GPDEV])#([DGAN])
BBS12042	([G])#([IV])#([IVM])#([LI])#([T])#([APG])#([SA])#([H])#([N])#([P])#([GP])#([GED])
BBS12044	([E])#([YFH])#([VT])#([S])#([AV])#([N])#([P])#([TN])#([GK])#([PDFYE])#([MLI])#([HN])
BBS20001	([PKI])#([RK])#([GRPSK])#([KPIRA]?)#([MLPVIT])#([SETN])#([SAG])#([YHF])#([AMNFL])#([FLNYVQ])#([FWTY])#([VLKSMF])
BBS20002	([TKNQPECGYLRS])#([LITKFAEYVM])#([VFYAIR])#([IVRKY])#([AVY])#([LVRMY])#([YFGWQR])#([DKPNA])#([YFCS])#([QEDMVHKTAR])#([TASPGQD])#([NRMQGVEDTP]?)
BBS20020	([QT])#([SMTCV])#([TIVA])#([SA])#([E])#([AV])#([ASL])#([R])#([YF])#([KQNRT])#([FYPQ])#([I])
BBS30006	([DNHEPS])#([G])#([TKSAHNLVD])#([FY])#([LM])#([VILA])#([R])#([DAEQPKF])#([ARSC])#([SDEKNART])#([TNSER])#([KNSAVPHT])
BBS30017	([AGQLENH])#([FW])#([ETAKS])#([QRKES])#([EADQK])#([TKN])#([G])#([G]?)#([IHVDQ])#([KADTEN])#([VL])#([TKVIR])
BBS30027	([EAQTK]?)#([EDQNK])#([LMIARQV])#([KEHAN])#([AVQNDL])#([LMIV])#([AVI])#([DTEA])#([YHWF])#([MIFLY])#([SGAKVE])#([KTSENQ])
BBS50002	([VAITYL])#([GAT])#([RFHLKA])#([PVILG])#([NSDGE])#([VAHSI])#([GI])#([KR])#([STG])#([TS])#([LTAI])#([LFTV])
BBS50005	([C])#([H])#([TS])#([G])#([LIV])#([RGSN])#([RK])#([TSN])#([ADV])#([G])#([WY])#([NK])
BBS50010	([P])#([ILQYVMF])#([YRLKI])#([LMYFIVR])#([TVACSNY])#([ACGEDQY])#([YSLFWIVN])#([TSAGCNV])#([PTLNSCQK])#([CAVLTM])#([YFHV])#([R])
BBS50013	([GHN])#([NDS])#([TCLAIM])#([H])#([IVL])#([Y])#([MTSLNVKD])#([NR])#([HQ])#([VIFLY])#([EDYN])#([QNLAKG])
BBS50016	([IVALCM])#([HY])#([RCT])#([D])#([ILV])#([K])#([GPSA])#([ASKEQH])#([N])#([ICVL])#([LMI])#([IVL])

Table A.1: Most conserved (MC) constraints list for the working database.

```
BB11001    ([EKNL])#([MYL])#([KPA])#([TNRG])#([YFW])#([IKRS])#([PYEA])#([PRD])#([KPHD])#([GRPN])#([ERDY])#([KLG]?)
BB11002    ([KGQMI]?)#([NGYFSE])#([LVQFKMPI])#([FIYRVA])#([VYRQ])#([AYV])#([LYQIT])#([YWRHQKS])#([DA])#([FYS])#([VEKRTM])#([APK]?)
BB11003    ([DERK])#([TYLA])#([TYFC])#([IAE])#([KRDT])#([EAV])#([LAVI])#([PLNG])#([DRSL])#([WYGK])#([ARSD])#([EYP]?)
BB11004    ([RQVY])#([SINM])#([YFM])#([INK])#([EN]?)#([HQVL])#([KGLW])#([MLDV])#([QKA])#([GSTV])#([NYSH])#([PRT])
BB11005    ([TPSE]?)#([KSQAGDPERV])#([NAEPDGQ])#([YWHLTFGIR])#([IQLVYF])#([AQDESVKH])#([EAQRYNM])#([NTILVAXRDW])#([HWVRANDTI])#([KHAIDVQEG])#([RWKYIMAFNQ])#([LIEGTYVA])
BB11006    ([LETAVY]?)#([KAIDEQL])#([VEFNTR])#([FITDVYR])#([NDFAGHS])#([PTEVQG])#([PMGSEVA])#([RILGVTN]?)#([NPAYQ])#([QFGNLIHV])#([LDTPQR])#([PLSERQ])
BB11007    ([AVTPGHS]?)#([RAFGDTL])#([TDGPHKVR])#([VFEIDSL])#([IPDLEA])#([LDMKGHEW])#([PLYNGV])#([QENLIHSPY])#([GSPLF])#([YDSLQETK])#([ASNLIED])#([DFLNR])
BB11008    ([EQK])#([LD])#([IVTH])#([NEA])#([HFL])#([YHG])#([RQSK])#([NLFY])#([ENAY])#([SRP])#([LGKR])#([ARG]?)
BB11009    ([GDV]?)#([NGT])#([VART])#([EVFS])#([FLI])#([QEC])#([CAI])#([PNA])#([DSQA])#([DGN])#([VER])#([YTS])
BB11010    ([TVRD])#([FRTG])#([DKRL])#([GLWI])#([LTF])#([NAGQ])#([ELD])#([MHFL])#([SMAN])#([PYTD])#([VDHL])#([LCGI])
BB11011    ([QNVS])#([EYTK])#([MVEG])#([FKSA])#([QRWLT])#([NIA])#([IKY])#([DNQGT])#([KETA])#([KITP])#([INAVE])#([ESLG]?)
BB11012    ([GAR]?)#([PVRT])#([WTNA])#([NDFP])#([KSGQ])#([DPYK])#([EDTN])#([IL])#([SARK])#([TS])#([TRVA])#([DTNS])
BB11013    ([NKVR])#([HLYMG])#([WSL])#([NIKT])#([SRDK])#([TRPFL])#([MLVGK])#([RLSA])#([RAYV])#([KAGRP])#([VGT])#([VSRE]?)
BB11014    ([VYSL]?)#([RNSYH])#([FLAVS])#([SKAL])#([HVRTFA])#([AFSTM])#([RFVS])#([DEKTL])#([PNVYQ])#([ISCVD])#([FLWG])#([GNDLT]?)
BB11015    ([TQ]?)#([LAG])#([KHGE])#([KAQ])#([VEWG])#([LTV])#([IVT])#([CVME])#([SGC])#([TNIA])#([LRPY])#([LRSV])
BB11016    ([IADKMPHT])#([VGLAPR])#([DRNEHSP])#([RNSGFYL])#([LGVIDYR])#([TPRKAVEM])#([THYES])#([GMKAQR])#([VITQM]?)#([AMGYRDTH])#([ANEDLK])#([LFKGDPH])
BB11017    ([DHIP])#([LSTA])#([L])#([LTKD])#([TS])#([FSRA])#([MYI])#([ELGT])#([ADXT])#([VLI])#([NMYF])#([KSGY])
BB11018    ([MYKDTVAHRI])#([YNDISEQKA])#([GMWIYNLTKSD])#([LIGVTW])#([KETSRHN])#([ASGQHLFKR])#([MVWYDRGAT])#([LAGKMYIDV])#([ESFDNLTVI])#([GDMVYTAS])#([SCPFVNRK])#([EVSHGDP]?)
BB11019    ([KLTM]?)#([GSYDVKRTF])#([RKCTMEP])#([LDNSKEGP])#([RSDHYTLAEG])#([AVYFNLTRI])#([EPVRNL])#([MKIAQE])#([LATISEQ])#([RVLMINAS])#([DAESQLW])#([MDIAWLGQ])
BB11020    ([EWVFMLI])#([ILWYQF])#([TNEVLFSA])#([EYNADLKQ])#([TIQYLMAF])#([LAVFGNS])#([QTMDSAG])#([DELYH]?)#([ITKYLA]?)#([FLRVQCKP])#([NHLVPRIM])#([DKQASFI]?)
BB11021    ([VIW])#([HEV])#([FTL])#([VID])#([AKIN])#([GP])#([VMAG])#([LAGD])#([GPAD])#([EDPV])#([AG])#([AT]?)
BB11022    ([DET]?)#([SWQP])#([IHFR])#([SRAE])#([SANR])#([RDE])#([VFI])#([KIL])#([SAVK])#([KGRL])#([RLI])#([IKA])
BB11023    ([EAKLQ])#([IAQEK])#([AFILVG])#([AKVF])#([KRAEN])#([NYATKM])#([FGRIDN])#([YFAGKL])#([RTVEYPG])#([PILYW])#([RKPTGN])#([DQLPA]?)
BB11024    ([CALT])#([GCH])#([TGS])#([VSGC])#([QHGS])#([LYRA])#([DLY])#([FGNL])#([SEGD])#([LNW]?)#([PFLQ])#([SAL])
BB11025    ([KDLE])#([KTDR])#([IM])#([VLK])#([QGAD])#([KFRT])#([KAL])#([LIER])#([ASRI])#([GAET])#([DYK])#([AKL]?)
BB11026    ([DMTER])#([TSLY])#([ILD])#([EHVL])#([NDE])#([VCA])#([KLAI])#([AMRELT])#([KSERAD])#([IAQN])#([QL]?)#([DKQGRN])
BB11027    ([MISR]?)#([DPMLE])#([QDAR])#([AGWTR])#([IEVLCA])#([SKRM])#([FLSVQDI])#([EDVLPAF])#([RQKDVN])#([DMATVLF])#([ILAG])#([VKPELD])
BB11028    ([NKP]?)#([RLPISKQV])#([KDVPQLERS])#([IMVSGHTYL])#([KDASEWLT])#([IVAG])#([KVATSPQW])#([AELIVQ])#([GA])#([FSANEKQD])#([TASRDP])#([HARKVIT])
BB11029    ([IGWF])#([PQG])#([GAQE])#([TPI])#([KVP])#([M])#([AQP])#([FGPA])#([GQNY])#([GAVY])#([LTNR])#([AV]?)
BB11030    ([EAYISNLP]?)#([SMAPLDGF])#([VEDYFLHR])#([ETAKHDISQN])#([RADNQEK])#([FTYWVCIAP])#([RADSENVL])#([KDGILRNSYEP])#([VLAIGMTYF])#([VFQIWLMY])#([EGRHKADS])#([ISAVQLTN])
BB11031    ([GLWNEITQ]?)#([LQGMDVH])#([YEANPHTRQG])#([DHSGIWKEQL])#([VYTWISFDRNA])#([LIWVMT])#([TQNADKR])#([KWSTCNPDF])#([YIHNAQE])#([WKLAYITVN])#([NLEAGQDW])#([RHAYWKF])
BB11032    ([DKRQE])#([TGDYMAI])#([VITAQ])#([RDQCHA])#([DQLSK])#([MILPAWK])#([TMLNFY])#([GVKLDC])#([HSENTQ])#([WLTRQE])#([PARVGE])#([SAQGN]?)
BB11033    ([PDRSK])#([LFMQESAY])#([IAFLVTY])#([NEHITDFA])#([LKSERC])#([ALFID])#([ASYLQGP])#([NEIDAGKQ])#([TEKGDV])#([YRNFKLQ])#([SDTHRPK])#([KEXGFA]?)
BB11034    ([CSRYL])#([VLPAN])#([NHGPSQ])#([VNML])#([GAQSE])#([CLKVA])#([VIYDEL])#([PLFG])#([KSHEMWG])#([KHGAFDL])#([VALKHSRG])#([MLFVGDE]?)
BB11035    ([TQKG]?)#([NGME])#([AQTPI])#([VAFGR])#([KTGP])#([KALI])#([YLAM])#([SNKT])#([DEGA])#([EAQG])#([EDQ])#([LIRA])
BB11036    ([SEHF]?)#([ETGWL])#([RLMPIEA])#([LDMTWYG])#([ALIWQDS])#([KMVLD])#([LQKAGTI])#([NAVQHE])#([QEDRP])#([LQMVYIT])#([LVNET])#([RPKNT])
BB11037    ([LSVWY])#([LAIY])#([FLAHE])#([HMTQF])#([FPK])#([AKPI])#([SPLT])#([IPLE])#([QIKSD])#([LDNR])#([KSVFL])#([RLDV]?)
BB11038    ([QLVK])#([GSAENK])#([LVHSIT])#([FLAP])#([SDYWFK])#([KDFILPQ])#([VFATKRL])#([LYQADS])#([ETYNLSQ])#([RAHSLDQ])#([LTDSGWE])#([NEPMQG]?)
BB12001    ([ENVYTGDQRK])#([NTQLSEKDA])#([IANDFKE])#([LQICPVM])#([NLQESTI])#([LAEGSYKDNI])#([LPVIFTSAE])#([LYAIEQS])#([SQRTND])#([YRISVTLAG])#([TVSIAMD])#([RGWSN])
BB12002    ([VILSFY])#([KNRQ])#([TSLN])#([KRD])#([DK])#([QKRVN])#([PRLG])#([QYTCI])#([VFMLI])#([QHG])#([VTA])#([FLRGCT])
BB12003    ([C])#([YNEQI])#([KAGDY])#([LAV])#([PE])#([DEKN])#([HNDS])#([VAKE])#([RGKTLPD])#([TIVWLS])#([KISAYWV])#([GVDSYP]?)
BB12004    ([LFYW])#([VTIRLCA])#([QNEAS])#([VSTEKIQGRAN])#([FLVAIT])#([NKVACQSIT])#([ENAGWPQL])#([HLAQFNVR])#([TDKILSVQ]?)#([IFVPRSKQYE])#([SDPAELGNQ])#([PALINTFK]?)
BB12005    ([IYP]?)#([NREKVAIF])#([KEDGAS])#([LIA])#([EKRPLAS])#([KVTLRGP])#([DETPAF])#([LEVAIF])#([ATRFLNI])#([FNIHL])#([NAT])#([LYGAT])
BB12006    ([MGRP])#([YG])#([RVI])#([VPIE])#([RYPA])#([LVS])#([STN])#([DTV])#([KQA])#([PTDV])#([HDAK])#([TSL]?)
BB12007    ([WVASD])#([QFPINSV])#([HQPSDIVA])#([QLVDAS])#([GPFRVL])#([KGDVPY])#([TFEPRYQ]?)#([LNGVI])#([FSMKL])#([IWRVFTE])#([SVCGLT])#([RYTDV])
BB12008    ([VSGFC])#([AGTLISVY])#([ILTRPGHAM])#([TSAVHLP])#([LYVFTGA])#([SGIMLFC])#([ASTIVE])#([THISALV])#([VSHQNP])#([SNGQP])#([LMQSADT])#([IALG]?)
BB12009    ([EDC])#([IVRP])#([VLEI])#([EHVNP])#([GSDP])#([NDLK])#([RKSN])#([Q]?)#([GK])#([PSRF])#([QHPS])#([AK])
BB12010    ([WYCTK]?)#([NDKSAF])#([ARQL]?)#([DTP])#([KGYI])#([GKEYIA])#([WFY])#([GHDSF])#([QDEN])#([PLGD])#([RKQCE])#([ILFM])
BB12011    ([AELFPN]?)#([GVQRMNP])#([TSGLFVYI])#([VWIKEQDRT])#([DKFNCTRI])#([STHIGD])#([LITENKFD])#([DETVSFP])#([VQSEFYG])#([ILGMFE])#([PENTIS])#([PGNTS]?)
BB12012    ([DHKC])#([TVYD])#([KQTA])#([DSYN])#([ILGS])#([NMSK])#([IVD])#([NSYI])#([SHGD])#([IHPL])#([DPLE])#([GRIA])
BB12013    ([VLFIA]?)#([TLKSHV])#([RSGKT])#([PANQLEG])#([MTLQRKA])#([KQTSIF])#([KIVSYR])#([MHDFRLV])#([FVLHST])#([N]?)#([RGPQ])#([EATVSND])
BB12014    ([IRATSG])#([YRPKLAS])#([TREGMD])#([RLK])#([EIAL])#([EDHNQK])#([LIVF])#([AMVS])#([QNSKE])#([RAMNKS])#([TLIC])#([NSHG])
BB12015    ([TVDMEQNP]?)#([FKMLSRVHA])#([PVEIA])#([KGDSAEQ])#([RKINSQA])#([GSKEN])#([QDVCSTK])#([TKAVER])#([CVIA])#([VTKLNSHG])#([VILCM])#([HEDSWR])
BB12016    ([M]?)#([TDKI])#([PQDKE])#([VIL])#([TLEVG])#([IV])#([KEQI])#([KIVYN])#([KCVR])#([GED])#([RHKG])#([HKRS])
BB12017    ([SFGKIYD])#([MLNTPV])#([RVEMFNL])#([YDKER])#([LWIGC])#([VFLA])#([RANEK])#([KEDH])#([RQDIN])#([LGNH])#([PAKWIL])#([KTEAN]?)
BB12018    ([DKAP])#([RK]?)#([IDE])#([PKA])#([EVIR])#([LKF])#([RDPE])#([QALE])#([IV])#([ILDK])#([EKFA])#([QEIF])
BB12019    ([LESK]?)#([VME])#([IVSEF])#([KLVIH])#([QKVST])#([GNVD])#([FSA])#([AIFY])#([IAG]?)#([EDV])#([PRSL])#([SGALN])
BB12020    ([R]?)#([TIDG])#([IVPE])#([SPLI])#([NGR])#([GN])#([YQG])#([LKAQ])#([SIV])#([SRPD])#([GAV])#([FSNP])
```

BB12021	([LH])#([EKDGHN])#([Y])#([RNVI])#([HQNAEK])#([DE])#([AT])#([STAN])#([SKDAE])#([VSA])#([QESDA])#([RGKD]?)
BB12022	([AQSP])#([LHQY])#([SGTN])#([KYST])#([DIHAL])#([QPASK])#([DSHY])#([FKP]?)#([NFRH])#([PSG])#([ANDRE])#([VKAWG])
BB12023	([FLN]?)#([SVKE])#([RQPTK])#([VLI])#([PAH])#([PLTIH])#([RNQDV])#([IVTNA])#([D]?)#([SHYQN])#([GTEI])#([LMAP])
BB12024	([EP])#([SN]?)#([HS]?)#([GE])#([RAEQ])#([DIK])#([YCS])#([ITH])#([LIV])#([SA])#([HEAQ])#([VITN])
BB12025	([SHT]?)#([SNGQ])#([THN])#([KYTI])#([VTS])#([DQEA])#([KE])#([KSA])#([ILW])#([VSG])#([PLR])#([RA]?)
BB12026	([LVRITA])#([KGENSRAWQDP])#([DEAQYSVRGN])#([VIKRAS])#([VAKEQRD])#([QRKSLITEVDAF])#([LKDEQWVRMI])#([SRIVLET])#([LADQKEIS])#([RSEATNLKI])#([REILSGDMVQA])#([VPYSKAREQ]?)
BB12027	([LVHADNQK])#([TDQESYK])#([IVLMA])#([RKQ])#([EAKQRSD])#([VLFIY])#([LCAIV])#([GEARTKSI])#([EKGNDARL])#([QAHRFSCKY])#([GNASK])#([AHRN]?)
BB12028	([EVHSY]?)#([AEPGRTF])#([GKSIE])#([YSERKG])#([SNKGH])#([TRCVYPH])#([SILHTYV])#([YRLAG])#([KQYAWF])#([ECYTRKS])#([DSAVTI])#([KPSTGD])
BB12029	([STMKANDE])#([LRIAV])#([PSKDEAF])#([GAVPT])#([AYLKPSN])#([SPDREY])#([LIVHYT])#([FEVDTYPKL])#([SLDAPGI])#([GPYAVS])#([GTSHKVAR])#([LRITKPV])
BB12030	([EIGH]?)#([LGKDQ])#([DQKG])#([CHVL])#([QRKLE])#([QTEA])#([GKICA])#([TIVFG])#([REKQD])#([AEDQ])#([LIV])#([LRAKI])
BB12031	([GALENQFD])#([LEANDK])#([LASETQF])#([ECAFLHN])#([SDVGQKRA])#([PCVGIMENA])#([KDQETNFA])#([AFVIESG])#([LSETNDV])#([EAFDSQNR])#([EIVHKQN])#([AEQKLSPV])
BB12032	([KQEG])#([NDTKH])#([LISNYV])#([C])#([YF])#([KLRT])#([MKNEYR])#([FASIRTW])#([MQRWTH])#([VCRTM])#([ADHFVEI])#([AKPHTV]?)
BB12033	([VMQLT])#([QATPR])#([YHIEVDA])#([LEPGIQ])#([SVQETI])#([IVTLFKR])#([LQAYDR])#([STDAV])#([ASKQICL])#([VKILSD])#([GPSIKR])#([KN]?)
BB12034	([LD]?)#([GKQ])#([DSY])#([TND])#([NDS])#([IVLA])#([AND])#([YDV])#([LF])#([HNER])#([EMA])#([NMKL])
BB12036	([APNSK])#([DSHKQ])#([IV])#([RLYDECQ])#([EANTD])#([LAFH])#([LFI])#([GSQKDR])#([GKTSD])#([LAKIMS])#([ATLGK])#([RSPEF]?)
BB12037	([STQFDRNGP]?)#([KADETVWLQ])#([ILVFDNTGA])#([SPTFIELND])#([REVIKPLHM])#([EDSTALP])#([DKNEASPQ])#([FYIHGKNV])#([MLYVIRFE])#([SEKDQTLY])#([GILNRAMFDW])#([VMLKYS])
BB12038	([KQTLAER])#([YVDFR])#([DSVGCKN])#([PAKGHRDV])#([SKGTN])#([LSH])#([KTGPSQD])#([PRAQIGK])#([LFIDA])#([SEFKLN])#([VFLGM])#([VLQT]?)
BB12039	([GDE]?)#([DGE])#([EGPDTHKY])#([TNIHQV])#([YFSIEAV])#([VEYIKT])#([IPKVAF])#([EDSHNRK])#([PAIE])#([SADE])#([LKSVADE])#([C])
BB12040	([NAKE])#([GLTIM])#([ILA])#([FN])#([HADK])#([LIG])#([ADYH])#([IANED])#([DNLA])#([DRLQ])#([NSKQ])#([SEN]?)
BB12041	([DIVL]?)#([VPAIDX])#([PEQSV])#([ASVPD])#([DVPCM])#([GTPAEK])#([AMDGVH])#([KSVM])#([IPVK])#([DKVIAT])#([FQEDKA])#([IFVHPA])
BB12042	([FY])#([GVP])#([RANS])#([NPLD])#([FSRI])#([FYES])#([TNK])#([RIP])#([YLPE])#([DQCI])#([YAEN])#([EDI]?)
BB12044	([LHAIVSKE])#([DTVGLRAI])#([LVHQTISK])#([KIQALESNV])#([DKALGHSIQ])#([KGVQDFTMEA])#([DVGQLPS])#([GLTDEPIK])#([IENQVTHA])#([SELANDPM])#([PITAKG])#([DLETPQWV])
BB20001	([EDARKTM])#([RKADIQ])#([EATNL])#([MVLIY])#([KQRDSP])#([TESNADG])#([YFW])#([IEKVNSR])#([PARKNV])#([PNKEQGR])#([KGYMLQND])#([GQEAPKN]?)
BB20002	([LFMVI])#([RKTQECHW])#([CKPQAER])#([DGSRN])#([EDATQP])#([ERKHIQLTVCA])#([YLFIVMK])#([YQHEFTRALMD])#([LINV])#([LIVSTFW])#([DNESGRHK])#([STNDEYKVPRG]?)
BB20020	([NQSDKTVL]?)#([ALTNDGYSPEV])#([NPREQTK])#([STEAIN])#([ALQKIVTE])#([AP])#([SRAGIKE])#([ASTF])#([LFIM])#([MIL])#([VIR])#([LCIVAF])
BB30006	([NKRDSYLACQPVT])#([NVIPDRQLE])#([KLYQTISF])#([LHTSQY])#([IYFCLAS])#([KRIHLV])#([IVLEQ])#([FDNLSQKVP])#([HKTRQNSEDY])#([RDATESPN])#([DKSGQEN])#([TDANHPKGQE]?)
BB30017	([DAEGQ]?)#([PNDKS])#([GYSLETVF])#([AYHGSN])#([LFEKGAPIV])#([VPKIMLNS])#([STIVFLA])#([YIVERKWFQS])#([SQYMVFPG])#([GQPDT])#([AGVSNT])#([AGVLYFS])
BB30027	([GKLISRTD])#([APEY])#([EDASQGTNP])#([EAYSNDTV])#([LEI])#([YFVLIKMSDT])#([KDAQPES])#([A]?)#([KAFIQY])#([MLKIRV])#([KISLTNDAEP])#([GNAYDH])
BB40003	([NSLTDEAK])#([AGILKF])#([QLCFVSM])#([LMSNIED])#([AEGSNI])#([PIVAT])#([YEMLQGSF])#([QITKSDE])#([RNKILHGTSE])#([TIVAL])#([GNWA])#([N]?)
BB40005	([SQHD]?)#([RMENDGQSY])#([GKDSRVEN])#([KRGPMQHA])#([DRKSENQA])#([LRQKTFSH])#([GLYVREQKS])#([SAEQKGTMCYV])#([MITLKS])#([DARTNS])#([VYIPMAL])#([NGEQSTD])
BB40006	([RSIAKQFGEP]?)#([AVPNDE])#([FWAYEDLH])#([VQXLIRFNKY])#([EQRDANISHL])#([MAESLQYGNK])#([ATRLNIPMY])#([RWVHGIFYP])#([EHAIKGSQTD])#([AWQYRKLTEF])#([YLIFMV])#([RIKLEN])
BB40007	([PAFSCI]?)#([LPVA])#([NLPRT])#([FALRQG])#([AFTSVY])#([ASEQD]?)#([SAMTPLN])#([ETGQ])#([SGLV])#([STHD])#([PTNQGV])#([EQYISDLT])
BB40008	([VENR]?)#([DYTKVMLRN])#([SFVAPRM])#([RTQAMSID])#([KTEQWLF])#([ICRYNTMAWFQ])#([DYTH])#([QHRKTGED])#([ENKSLYTGD])#([GNETSY])#([KTIEMLYN])#([IYFVLM])
BB40009	([EDVPGQHKIL]?)#([PWIHVGMN])#([PSYAQLFNHG])#([APSEDL])#([EMTAVPWHGIL])#([FLMYAWIEKH])#([ASLKCDEFYV])#([RKEAWVSTMGD])#([LHMESTRKF])#([RDQLNGS])#([ANEDKLP])#([NTSEGADKLIY])
BB40010	([RKDSGN])#([SQ]?)#([GK])#([PRKAF])#([QATSPY])#([AKS])#([AGTV])#([NKVDIE])#([VIDNEL])#([TQVLRY])#([KSITRL])#([ELGAKST])
BB40014	([LFTN]?)#([QPRDNS])#([AGFSNV])#([TYSKAD])#([LCITN])#([IMWVPYNF])#([TQARSYL])#([NDAT])#([EKATGSD])#([DGYVT])#([EKNQRA])#([NVQPETS])
BB40018	([THNQ]?)#([RHKQGTSN])#([HKRCSG])#([IAVLQKH])#([KIRFE])#([TGARKI])#([RTNP])#([MVILYF])#([LMCIV])#([VTLI])#([AGLFP])#([DPEYN]?)
BB40019	([PMRKAF]?)#([LGVIPESHY])#([GLARW])#([PSTK])#([MVDKPTS])#([ATYPMFS])#([ADVFGNE])#([SIPNATW])#([SGRPAFD])#([WKS])#([LI]?)#([CPIDYKTQ])
BB40022	([ATSMQKHD])#([DNGSER])#([QPLREDNSG])#([PNESKTRGFA])#([GKVPLIR])#([DRLTYIS]?)#([FWYVRIS])#([EQDRKG])#([YLVE])#([C])#([DNMSER])#([LIV]?)
BB40025	([SEAIV]?)#([ADPTSYER])#([TAREPDSNQ])#([EGSLAD])#([AQVE])#([AIESKNYQLM])#([KTRSNE])#([YHNVIGTA])#([VLF])#([FC])#([KTLAP])#([DGNSE])
BB40032	([LGPAQ])#([ALIPV])#([VASQG])#([RNPKQH])#([QCR]?)#([QVIAR])#([APEI])#([PTK])#([LVIP])#([IVPKTH])#([IVWN])#([PEKAI])
BB40033	([AGIERKNLQY])#([LITV])#([IVLEMQFY])#([NRDKAVEQS])#([FVEIWLK])#([MTAWVLFYI])#([LMREKDN])#([DNKESALR])#([LKFEIQMRA])#([HLMNYQFIT])#([TIAEDKQ])#([KHRSEIFALN])
BB40043	([TVNPRSIED]?)#([GEALCIMT])#([KRNDVETSLF])#([VILGC])#([TLIVEAKS])#([KSMDVRI])#([VIKRQTE])#([QHKETASV])#([NHKVSILR])#([WVMIELAP])#([TNKSAPE]?)#([HKSDIL]?)
BB40045	([DIAVLNY]?)#([VPALIDX])#([PEQSAV])#([ASVPD])#([DVPACM])#([GTPAEDKIM])#([AMDGVH])#([KSPVMTL])#([IPKVL])#([DKVAITM])#([FQEPDKAGL])#([IFVAHPTS])
BB40048	([PEIQDV]?)#([SQPKTRL])#([LSAKDNCE])#([KLAMIV])#([DAKGSLTN])#([ANDQTMGK])#([VALNTKDMC])#([AEPNFIK])#([GAPISVLK]?)#([KETQYDG])#([AKSDLITQH])#([FGRAKI])
BB50002	([EDVLRN]?)#([KFEAPSVD])#([AETSIHPYK]?)#([DRLEKI])#([LEGNAFTK])#([LVMQDRK])#([PKSELQDA])#([HPEANIVYFL])#([LEVFYQRX])#([QLNKRMTEAS])#([FYGEQTMKI])#([LSHKITVAFEC])
BB50004	([D]?)#([FCLYV])#([DEI])#([KEARI])#([ASDNK])#([GKFLMN])#([ALSIVN])#([EKPDG])#([IFLED])#([VACY])#([PRGKQNE])#([KSEARN])
BB50005	([KSTI])#([N]?)#([VARIT])#([ENASGDTF])#([RKCQDTI])#([LIVQ])#([KRAVYL])#([QEKRSTGN])#([VLKIMFE])#([LVI])#([LEVAQNDR])#([HGDNEVK])
BB50010	([AGIKNLQEYD])#([LITV])#([IVLEQFY])#([NRDEQSAK])#([FVEIWLKQ])#([MTAWLFYI])#([LMRKDNT])#([DNKLRSEG])#([LKFEQMRAIV])#([HLMQFITY])#([TIAKDE])#([KHRFEALNQ])
BB50013	([SDT])#([TNLS])#([NDFSE])#([VIL])#([NKARQSD])#([DYFVKI])#([LI])#([RNPKLQE])#([LMGKDQENA])#([IMENHYLK])#([QGTSNHRY])#([HNG]?)
BB50016	([TMIEGDL]?)#([KPQLEIGRMFW])#([ILRSPTMQVDE]?)#([AKTRIVPENF])#([TVNGEHQLFIS])#([PIRLNTHE])#([KGQHRPADIV])#([NPDHSKRY])#([WYLFV])#([LRTKEVHQMP])#([KVIPLRAD])#([GLIVEAQT])
BBS11001	([REQ])#([EKNL])#([MYL])#([KPA])#([TNRG])#([YFW])#([IKRS])#([PYEA])#([PRD])#([KPHD])#([GRPN])#([ERDY])
BBS11002	([TKHWLA])#([KELARP])#([G])#([EDPQ])#([KCIAVL])#([LMKVI])#([RTLYE])#([VIL])#([LINWF])#([GHKPS])#([YRGFK])#([NESQL]?)
BBS11003	([DERK])#([TYLA])#([TYFC])#([IAE])#([KRDT])#([EAV])#([LAVI])#([PLNG])#([DRSL])#([WYGK])#([ARSD])#([EYP]?)
BBS11004	([RQVY])#([SINM])#([YFM])#([INK])#([EN]?)#([HQVL])#([KGLW])#([MLDV])#([QKA])#([GSTV])#([NYSH])#([PRT])
BBS11005	([TPSE]?)#([KSQAGDPERV])#([NAEPDGQ])#([YWHLTFGIR])#([IQLVYF])#([AQDESVKH])#([EAQRYNM])#([NTILVAXRDW])#([HWVRANDTI])#([KHAIDVQEG])#([RWKYIMAFNQ])#([LIEGTYVA])
BBS11006	([LETAVY]?)#([KAIDEQL])#([VEFNTR])#([FITDVYR])#([NDFAGHS])#([PTEVQG])#([PMGSEVA])#([RILGVTN]?)#([NPAYQ])#([QFGNLIHV])#([LDTPQR])#([PLSERQ])
BBS11007	([RAFGDTL])#([TDGPHKVR])#([VFEIDSL])#([IPDLEA])#([LDMKGHEW])#([PLYNGV])#([QENLIHSPY])#([GSPLF])#([YDSLQETK])#([ASNLIED])#([DFLNR])#([DHSTARKGQ])

BBS11008	([EQK])#([LD])#([IVTH])#([NEA])#([HFL])#([YHG])#([RQSK])#([NLFY])#([ENAY])#([SRP])#([LGKR])#([ARG]?)
BBS11009	([GDV]?)#([NGT])#([VART])#([EVFS])#([FLI])#([QEC])#([CAI])#([PNA])#([DSQA])#([DGN])#([VER])#([YTS])
BBS11010	([N])#([RHA])#([DQTH])#([EAF])#([L])#([LRI])#([SAGK])#([VELK])#([LYA])#([PGF])#([ADVE])#([GWE]?)
BBS11011	([SQYD])#([VTHR])#([RSFCI])#([WACSG])#([VTGLH])#([MGA])#([NVSFL])#([ALN])#([LIVM])#([GNVE])#([VKSAG])#([K]?)
BBS11012	([GAR]?)#([PVRT])#([WTNA])#([NDFP])#([KSGQ])#([DPYK])#([EDTN])#([IL])#([SARK])#([TS])#([TRVA])#([DTNS])
BBS11013	([KIRST])#([NKVR])#([HLYMG])#([WSL])#([NIKT])#([SRDK])#([TRPFL])#([MLVGK])#([RLSA])#([RAYV])#([KAGRP])#([VGT])
BBS11014	([VYSL]?)#([RNSYH])#([FLAVS])#([SKAL])#([HVRTFA])#([AFSTM])#([RFVS])#([DEKTL])#([PNVYQ])#([ISCVD])#([FLWG])#([GNDLT]?)
BBS11015	([TQ]?)#([LAG])#([KHGE])#([KAQ])#([VEWG])#([LTV])#([IVT])#([CVME])#([SGC])#([TNIA])#([LRPY])#([LRSV])
BBS11016	([IADKMPHT])#([VGLAPR])#([DRNEHSP])#([RNSGFYL])#([LGVIDYR])#([TPRKAVEM])#([THYES])#([GMKAQR])#([VITQM]?)#([AMGYRDTH])#([ANEDLK])#([LFKGDPH])
BBS11017	([DHIP])#([LSTA])#([L])#([LTKD])#([TS])#([FSRA])#([MYI])#([ELGT])#([ADXT])#([VLI])#([NMYF])#([KSGY])
BBS11018	([MYKDTVAHRI])#([YNDISEQKA])#([GMWIYNLTKSD])#([LIGVTW])#([KETSRHN])#([ASGQHLFKR])#([MVWYDRGAT])#([LAGKMYIDV])#([ESFDNLTVI])#([GDMVYTAS])#([SCPFVNRK])#([EVSHGDP]?)
BBS11019	([KLTM]?)#([GSYDVKRTF])#([RKCTMEP])#([LDNSKEGP])#([RSDHYTLAEG])#([AVYFNLTRI])#([EPVRNL])#([MKIAQE])#([LATISEQ])#([RVLMINAS])#([DAESQLW])#([MDIAWLGQ])
BBS11020	([EWVFMLI])#([ILWYQF])#([TNEVLFSA])#([EYNADLKQ])#([TIQYLMAF])#([LAVFGNS])#([QTMDSAG])#([DELYH]?)#([ITKYLA]?)#([FLRVQCKP])#([NHLVPRIM])#([DKQASFI]?)
BBS11021	([VIW])#([HEV])#([FTL])#([VID])#([AKIN])#([GP])#([VMAG])#([LAGD])#([GPAD])#([EDPV])#([AG])#([AT]?)
BBS11022	([IHFR])#([SRAE])#([SANR])#([RDE])#([VFI])#([KIL])#([SAVK])#([KGRL])#([RLI])#([IKA])#([QK])#([LR]?)
BBS11023	([QAVG])#([EAKLQ])#([IAQEK])#([AFILVG])#([AKVF])#([KRAEN])#([NYATKM])#([FGRIDN])#([YFAGKL])#([RTVEYPG])#([PILYW])#([RKPTGN])
BBS11024	([CALT])#([GCH])#([TGS])#([VSGC])#([QHGS])#([LYRA])#([DLY])#([FGNL])#([SEGD])#([LNW]?)#([PFLQ])#([SAL])
BBS11025	([TK]?)#([RWEA])#([IFL])#([KDLE])#([KTDR])#([IM])#([VLK])#([QGAD])#([KFRT])#([KAL])#([LIER])#([ASRI])
BBS11026	([DMTER])#([TSLY])#([ILD])#([EHVL])#([NDE])#([VCA])#([KLAI])#([AMRELT])#([KSERAD])#([IAQN])#([QL]?)#([DKQGRN])
BBS11027	([MISR]?)#([DPMLE])#([QDAR])#([AGWTR])#([IEVLCA])#([SKRM])#([FLSVQDI])#([EDVLPAF])#([RQKDVN])#([DMATVLF])#([ILAG])#([VKPELD])
BBS11028	([NKP]?)#([RLPISKQV])#([KDVPQLERS])#([IMVSGHTYL])#([KDASEWLT])#([IVAG])#([KVATSPQW])#([AELIVQ])#([GA])#([FSANEKQD])#([TASRDP])#([HARKVIT])
BBS11029	([IGWF])#([PQG])#([GAQE])#([TPI])#([KVP])#([M])#([AQP])#([FGPA])#([GQNY])#([GAVY])#([LTNR])#([AV]?)
BBS11030	([EAYISNLP]?)#([SMAPLDGF])#([VEDYFLHR])#([ETAKHDISQN])#([RADNQEK])#([FTYWVCIAP])#([RADSENVL])#([KDGILRNSYEP])#([VLAIGMTYF])#([VFQIWLMY])#([EGRHKADS])#([ISAVQLTN])
BBS11031	([GLWNEITQ]?)#([LQGMDVH])#([YEANPHTRQG])#([DHSGIWKEQL])#([VYTWISFDRNA])#([LIWVMT])#([TQNADKR])#([KWSTCNPDF])#([YIHNAQE])#([WKLAYITVN])#([NLEAGQDW])#([RHAYWKF])
BBS11032	([DKRQE])#([TGDYMAI])#([VITAQ])#([RDQCHA])#([DQLSK])#([MILPAWK])#([TMLNFY])#([GVKLDC])#([HSENTQ])#([WLTRQE])#([PARVGE])#([SAQGN]?)
BBS11033	([PDRSK])#([LFMQESAY])#([IAFLVTY])#([NEHITDFA])#([LKSERC])#([ALFID])#([ASYLQGP])#([NEIDAGKQ])#([TEKGDV])#([YRNFKLQ])#([SDTHRPK])#([KEXGFA]?)
BBS11034	([CSRYL])#([VLPAN])#([NHGPSQ])#([VNML])#([GAQSE])#([CLKVA])#([VIYDEL])#([PLFG])#([KSHEMWG])#([KHGAFDL])#([VALKHSRG])#([MLFVGDE]?)
BBS11035	([TQKG]?)#([NGME])#([AQTPI])#([VAFGR])#([KTGP])#([KALI])#([YLAM])#([SNKT])#([DEGA])#([EAQG])#([EDQ])#([LIRA])
BBS11036	([KSDTRG])#([VALI])#([YQLKEDR])#([AVCT])#([RYTFVI])#([SRPEVIA])#([VWGEF])#([YQLARDN])#([DITWGLV])#([SPGHT])#([RMFLDT])#([DYHSRIA]?)
BBS11037	([LSVWY])#([LAIY])#([FLAHE])#([HMTQF])#([FPK])#([AKPI])#([SPLT])#([IPLE])#([QIKSD])#([LDNR])#([KSVFL])#([RLDV]?)
BBS11038	([NE]?)#([NKVQRY])#([LNHRA])#([ARDNLG])#([FPSG])#([DESRCA])#([AEKQL])#([TRGE])#([PIARL])#([DTMGL])#([YFALQ])#([DEARGQHL])
BBS12001	([ENVYTGDQRK])#([NTQLSEKDA])#([IANDFKE])#([LQICPVM])#([NLQESTI])#([LAEGSYKDNI])#([LPVIFTSAE])#([LYAIEQS])#([SQRTND])#([YRISVTLAG])#([TVSIAMD])#([RGWSN])
BBS12002	([VILSFY])#([KNRQ])#([TSLN])#([KRD])#([DK])#([QKRVN])#([PRLG])#([QYTCI])#([VFMLI])#([QHG])#([VTA])#([FLRGCT])
BBS12003	([C])#([YNEQI])#([KAGDY])#([LAV])#([PE])#([DEKN])#([HNDS])#([VAKE])#([RGKTLPD])#([TIVWLS])#([KISAYWV])#([GVDSYP]?)
BBS12004	([LFYW])#([VTIRLCA])#([QNEAS])#([VSTEKIQGRAN])#([FLVAIT])#([NKVACQSIT])#([ENAGWPQL])#([HLAQFNVR])#([TDKILSVQ]?)#([IFVPRSKQYE])#([SDPAELGNQ])#([PALINTFK]?)
BBS12005	([IYP]?)#([NREKVAIF])#([KEDGAS])#([LIA])#([EKRPLAS])#([KVTLRGP])#([DETPAF])#([LEVAIF])#([ATRFLNI])#([FNIHL])#([NAT])#([LYGAT])
BBS12006	([MGRP])#([YG])#([RVI])#([VPIE])#([RYPA])#([LVS])#([STN])#([DTV])#([KQA])#([PTDV])#([HDAK])#([TSL]?)
BBS12007	([WVASD])#([QFPINSV])#([HQPSDIVA])#([QLVDAS])#([GPFRVL])#([KGDVPY])#([TFEPRYQ]?)#([LNGVI])#([FSMKL])#([IWRVFTE])#([SVCGLT])#([RYTDV])
BBS12008	([VSGFC])#([AGTLISVY])#([ILTRPGHAM])#([TSAVHLP])#([LYVFTGA])#([SGIMLFC])#([ASTIVE])#([THISALV])#([VSHQNP])#([SNGQP])#([LMQSADT])#([IALG]?)
BBS12009	([EDC])#([IVRP])#([VLEI])#([EHVNP])#([GSDP])#([NDLK])#([RKSN])#([Q]?)#([GK])#([PSRF])#([QHPS])#([AK])
BBS12010	([GHDSF])#([QDEN])#([PLGD])#([RKQCE])#([ILFM])#([QVT])#([PTQR])#([FYDSW])#([QGSPE])#([NPTFD])#([LIA])#([TEHDSK])
BBS12011	([AELFPN]?)#([GVQRMNP])#([TSGLFVYI])#([VWIKEQDRT])#([DKFNCTRI])#([STHIGD])#([LITENKFD])#([DETVSFP])#([VQSEFYG])#([ILGMFE])#([PENTIS])#([PGNTS]?)
BBS12012	([DHKC])#([TVYD])#([KQTA])#([DSYN])#([ILGS])#([NMSK])#([IVD])#([NSYI])#([SHGD])#([IHPL])#([DPLE])#([GRIA])
BBS12013	([VLFIA]?)#([TLKSHV])#([RSGKT])#([PANQLEG])#([MTLQRKA])#([KQTSIF])#([KIVSYR])#([MHDFRLV])#([FVLHST])#([N]?)#([RGPQ])#([EATVSND])
BBS12014	([IRATSG])#([YRPKLAS])#([TREGMD])#([RLK])#([EIAL])#([EDHNQK])#([LIVF])#([AMVS])#([QNSKE])#([RAMNKS])#([TLIC])#([NSHG])
BBS12015	([GSLVIQA]?)#([HVDPGAR])#([PGFYDL])#([GNAP])#([IPHKG])#([IEPM])#([PGLA])#([PA])#([HNG])#([AEKQS])#([TDEK])#([LV])
BBS12016	([M]?)#([TDKI])#([PQDKE])#([VIL])#([TLEVG])#([IV])#([KEQI])#([KIVYN])#([KCVR])#([GED])#([RHKG])#([HKRS])
BBS12017	([SFGKIYD])#([MLNTPV])#([RVEMFNL])#([YDKER])#([LWIGC])#([VFLA])#([RANEK])#([KEDH])#([RQDIN])#([LGNH])#([PAKWIL])#([KTEAN]?)
BBS12018	([DKAP])#([RK]?)#([IDE])#([PKA])#([EVIR])#([LKF])#([RDPE])#([QALE])#([IV])#([ILDK])#([EKFA])#([QEIF])
BBS12019	([LESK]?)#([VME])#([IVSEF])#([KLVIH])#([QKVST])#([GNVD])#([FSA])#([AIFY])#([IAG]?)#([EDV])#([PRSL])#([SGALN])
BBS12020	([R]?)#([TIDG])#([IVPE])#([SPLI])#([NGR])#([GN])#([YQG])#([LKAQ])#([SIV])#([SRPD])#([GAV])#([FSNP])
BBS12021	([LH])#([EKDGHN])#([Y])#([RNVI])#([HQNAEK])#([DE])#([AT])#([STAN])#([SKDAE])#([VSA])#([QESDA])#([RGKD]?)
BBS12022	([AQSP])#([LHQY])#([SGTN])#([KYST])#([DIHAL])#([QPASK])#([DSHY])#([FKP]?)#([NFRH])#([PSG])#([ANDRE])#([VKAWG])
BBS12023	([FLN]?)#([SVKE])#([RQPTK])#([VLI])#([PAH])#([PLTIH])#([RNQDV])#([IVTNA])#([D]?)#([SHYQN])#([GTEI])#([LMAP])
BBS12024	([EP])#([SN]?)#([HS]?)#([GE])#([RAEQ])#([DIK])#([YCS])#([ITH])#([LIV])#([SA])#([HEAQ])#([VITN])
BBS12025	([TS]?)#([VKE])#([SALD])#([AKGL])#([AGQR])#([KQPN])#([TPKV])#([TRS])#([PES])#([P])#([SQK])#([V])
BBS12026	([LVRITA])#([KGENSRAWQDP])#([DEAQYSVRGN])#([VIKRAS])#([VAKEQRD])#([QRKSLITEVDAF])#([LKDEQWVRMI])#([SRIVLET])#([LADQKEIS])#([RSEATNLKI])#([REILSGDMVQA])#([VPYSKAREQ]?)
BBS12027	([LVHADNQK])#([TDQESYK])#([IVLMA])#([RKQ])#([EAKQRSD])#([VLFIY])#([LCAIV])#([GEARTKSI])#([EKGNDARL])#([QAHRFSCKY])#([GNASK])#([AHRN]?)

BBS12028	([EVHSY]?)#([AEPGRTF])#([GKSIE])#([YSERKG])#([SNKGH])#([TRCVYPH])#([SILHTYV])#([YRLAG])#([KQYAWF])#([ECYTRKS])#([DSAVTI])#([KPSTGD])
BBS12029	([STMKANDE])#([LRIAV])#([PSKDEAF])#([GAVPT])#([AYLKPSN])#([SPDREY])#([LIVHYT])#([FEVDTYPKL])#([SLDAPGI])#([GPYAVS])#([GTSHKVAR])#([LRITKPV])
BBS12030	([EIGH]?)#([LGKDQ])#([DQKG])#([CHVL])#([QRKLE])#([QTEA])#([GKICA])#([TIVFG])#([REKQD])#([AEDQ])#([LIV])#([LRAKI])
BBS12031	([GALENQFD])#([LEANDK])#([LASETQF])#([ECAFLHN])#([SDVGQKRA])#([PCVGIMENA])#([KDQETNFA])#([AFVIESG])#([LSETNDV])#([EAFDSQNR])#([EIVHKQN])#([AEQKLSPV])
BBS12032	([KQEG])#([NDTKH])#([LISNYV])#([C])#([YF])#([KLRT])#([MKNEYR])#([FASIRTW])#([MQRWTH])#([VCRTM])#([ADHFVEI])#([AKPHTV]?)
BBS12033	([VMQLT])#([QATPR])#([YHIEVDA])#([LEPGIQ])#([SVQETI])#([IVTLFKR])#([LQAYDR])#([STDAV])#([ASKQICL])#([VKILSD])#([GPSIKR])#([KN]?)
BBS12034	([LD]?)#([GKQ])#([DSY])#([TND])#([NDS])#([IVLA])#([AND])#([YDV])#([LF])#([HNER])#([EMA])#([NMKL])
BBS12036	([GDS]?)#([YIVK])#([LHIV])#([RSEVK])#([NATKIR])#([IFV])#([NDSQPR])#([GAC])#([ESTDN])#([QARG]?)#([DTES])#([MPTV])
BBS12037	([STQFDRNGP]?)#([KADETVWLQ])#([ILVFDNTGA])#([SPTFIELND])#([REVIKPLHM])#([EDSTALP])#([DKNEASPQ])#([FYIHGKNV])#([MLYVIRFE])#([SEKDQTLY])#([GILNRAMFDW])#([VMLKYS])
BBS12038	([KQTLAER])#([YVDFR])#([DSVGCKN])#([PAKGHRDV])#([SKGTN])#([LSH])#([KTGPSQD])#([PRAQIGK])#([LFIDA])#([SEFKLN])#([VFLGM])#([VLQT]?)
BBS12039	([GDE]?)#([DGE])#([EGPDTHKY])#([TNIHQV])#([YFSIEAV])#([VEYIKT])#([IPKVAF])#([EDSHNRK])#([PAIE])#([SADE])#([LKSVADE])#([C])
BBS12040	([NAKE])#([GLTIM])#([ILA])#([FN])#([HADK])#([LIG])#([ADYH])#([IANED])#([DNLA])#([DRLQ])#([NSKQ])#([SEN]?)
BBS12041	([VPAIDX])#([PEQSV])#([ASVPD])#([DVPCM])#([GTPAEK])#([AMDGVH])#([KSVM])#([IPVK])#([DKVIAT])#([FQEDKA])#([IFVHPA])#([AEKLF])
BBS12042	([FY])#([GVP])#([RANS])#([NPLD])#([FSRI])#([FYES])#([TNK])#([RIP])#([YLPE])#([DQCI])#([YAEN])#([EDI]?)
BBS12044	([LHAIVSKE])#([DTVGLRAI])#([LVHQTISK])#([KIQALESNV])#([DKALGHSIQ])#([KGVQDFTMEA])#([DVGQLPS])#([GLTDEPIK])#([IENQVTHA])#([SELANDPM])#([PITAKG])#([DLETPQWV])
BBS20001	([VIFAWKM]?)#([NKRISAQE])#([FVINS])#([STAG])#([EQAD])#([FVIL])#([SALTGN])#([KRSQ])#([KRDMEATVI])#([CGILA])#([SGA])#([EAKDR])
BBS20002	([LFMVI])#([RKTQECHW])#([CKPQAER])#([DGSRN])#([EDATQP])#([ERKHIQLTVCA])#([YLFIVMK])#([YQHEFTRALMD])#([LINV])#([LIVSTFW])#([DNESGRHK])#([STNDEYKVPRG]?)
BBS20020	([NQSDKTVL]?)#([ALTNDGYSPEV])#([NPREQTK])#([STEAIN])#([ALQKIVTE])#([AP])#([SRAGIKE])#([ASTF])#([LFIM])#([MIL])#([VIR])#([LCIVAF])
BBS30006	([NKRDSYLACQPVT])#([NVIPDRQLE])#([KLYQTISF])#([LHTSQY])#([IYFCLAS])#([KRIHLV])#([IVLEQ])#([FDNLSQKVP])#([HKTRQNSEDY])#([RDATESPN])#([DKSGQEN])#([TDANHPKGQE]?)
BBS30017	([DAEGQ]?)#([PNDKS])#([GYSLETVF])#([AYHGSN])#([LFEKGAPIV])#([VPKIMLNS])#([STIVFLA])#([YIVERKWFQS])#([SQYMVFPG])#([GQPDT])#([AGVSNT])#([AGVLYFS])
BBS30027	([GKLISRTD])#([APEY])#([EDASQGTNP])#([EAYSNDTV])#([LEI])#([YFVLIKMSDT])#([KDAQPES])#([A]?)#([KAFIQY])#([MLKIRV])#([KISLTNDAEP])#([GNAYDH])
BBS50002	([EDVLRN]?)#([KFEAPSVD])#([AETSIHPYK]?)#([DRLEKI])#([LEGNAFTK])#([LVMQDRK])#([PKSELQDA])#([HPEANIVYFL])#([LEVFYQRX])#([QLNKRMTEAS])#([FYGEQTMKI])#([LSHKITVAFEC])
BBS50005	([KSTI])#([N]?)#([VARIT])#([ENASGDTF])#([RKCQDTI])#([LIVQ])#([KRAVYL])#([QEKRSTGN])#([VLKIMFE])#([LVI])#([LEVAQNDR])#([HGDNEVK])
BBS50010	([AGIKNLQEYD])#([LITV])#([IVLEQFY])#([NRDEQSAK])#([FVEIWLKQ])#([MTAWLFYI])#([LMRKDNT])#([DNKLRSEG])#([LKFEQMRAIV])#([HLMQFITY])#([TIAKDE])#([KHRFEALNQ])
BBS50013	([SDT])#([TNLS])#([NDFSE])#([VIL])#([NKARQSD])#([DYFVKI])#([LI])#([RNPKLQE])#([LMGKDQENA])#([IMENHYLK])#([QGTSNHRY])#([HNG]?)
BBS50016	([TMIEGDL]?)#([KPQLEIGRMFW])#([ILRSPTMQVDE]?)#([AKTRIVPENF])#([TVNGEHQLFIS])#([PIRLNTHE])#([KGQHRPADIV])#([NPDHSKRY])#([WYLFV])#([LRTKEVHQMP])#([KVIPLRAD])#([GLIVEAQT])

Table A.2: Least conserved (LC) constraints list for the working database.

Appendix B

Constraints used with RE-MuSiC

For conducting the benchmarking studies, CSA-X and RE-MuSIC are provided with similar constraints. As the format of specifying constraints in those programs are different, equivalent constraint sets are derived for them using a Perl script. The list of constraints used with RE-MuSiC are as follows:

BB11001	[WY]-[KH]-[TMEA]-[ML]-[STP]-[AER]-[KAE]-[EK]-[KQ]-[GWMA]-[KP]-[FY]
BB11002	[QERILTK]-[GK]-[WYDV]-[VF]-[P]-[SRGEA]-[NTRSM]-[YLKFIH]-[ILVAT]-[TGEKRQ]-[PLYIREK]-[VYIDSL]
BB11003	[AP]-[VL]-[AK]-[AVS]-[G]-[NC]-[ATP]-[VI]-[IL]-[LAV]-[KR]-[PG]
BB11004	[IWQ]-[ELV]-[EA]-[NP]-[LVI]-[DQK]-[LV]-[FVA]-[IV]-[VMI]-[TNP]-[MIL]
BB11005	[VLMFI]-[LFIVYA]-[VLIAS]-[TCENPH]-[NTSPVA]-[PITC]-[VHCA](0,1)-[SNQLVH]-[NGSR]-[PASNE]-[LTCGS]-[G]
BB11006	[NQSA]-[STMNP]-[KSATQLC]-[FCVAIT](0,1)-[TSHR]-[FMIL]-[SVTA]-[IVL]-[P]-[YWF]-[LIVNT]-[SGAN]
BB11007	[HR]-[MFLAR]-[C]-[LMIVP]-[G]-[QREAS]-[HPQGAD]-[LFI]-[ATG]-[KRLI]-[LRHME]-[EQVH]
BB11008	[DP]-[G]-[TL]-[F]-[LI]-[VIL]-[R]-[DEF]-[AS]-[SQE]-[TRS]-[KNS]
BB11009	[SAD]-[C]-[E](0,1)-[RKE]-[ASE]-[GC]-[SAET]-[C]-[STG]-[ST]-[C]-[AHRK]
BB11010	[GSN]-[D]-[ACS]-[E]-[LVI]-[VIL]-[LA]-[RALN]-[LI]-[LYF]-[EQTA]-[RES]
BB11011	[LVCG]-[PQAR]-[G]-[MND]-[CSXA]-[G]-[GSR]-[AGPS]-[LIV]-[VFM]-[SNCAD]-[SGN]
BB11012	[LIA]-[YWH]-[FA]-[NK]-[GA]-[QRS]-[W]-[KTV]-[TN]-[PK]-[F]-[PED]
BB11013	[F](0,1)-[NSVM]-[RS]-[WRLQ]-[AQHGR]-[ES]-[IML]-[AQSVK]-[KRAN]-[LYKAE]-[LI]-[PKSQG]
BB11014	[ED]-[LVI]-[E]-[SNR]-[IHAV]-[L]-[LVGM]-[QKATG]-[HA]-[PM]-[NYAKG]-[IV]
BB11015	[VI]-[VI]-[TSG]-[N]-[P]-[VA]-[DN]-[VT]-[MINT]-[VTA]-[QAY]-[LI]
BB11016	[TSQN]-[SITAD]-[MVEHCRI]-[HPKS]-[NDG]-[VI]-[WYFH]-[AVL]-[IVLA]-[G]-[D]-[VLA]
BB11017	[TV]-[ATS]-[E]-[VA]-[ALS]-[R]-[FY]-[RXK]-[YQF]-[I]-[QE]-[NRQ]
BB11018	[NGD]-[IMVLX]-[KYRGNS]-[VLIA]-[IMYVL]-[IVLMQGY]-[D]-[FVAWL]-[AVIP]-[PIYFL]-[NDSGH]-[H]
BB11019	[NPGAQ]-[MVILASF]-[IVPGA]-[LAICPD]-[G]-[HNTMI](0,1)-[EDG]-[AGIVF]-[VASTLI]-[GF]-[EITRV]-[V]
BB11020	[FWY]-[FIVL]-[VCAGD]-[GT]-[NQDE]-[SRKGNE]-[MFIVL]-[TGS]-[LIYM]-[AVI]-[D]-[LAFVYC]
BB11021	[PGQ]-[H]-[N]-[VIW]-[HEV]-[FTL]-[VID]-[AKIN]-[GP]-[VMAG]-[LAGD]-[GPAD]
BB11022	[QK]-[LR](0,1)-[GK]-[LY]-[NSTP]-[QLN]-[ASTK]-[EANM]-[LVI]-[ASG]-[QRE]-[KQAR]
BB11023	[HQAKE]-[WF]-[KAET]-[QK]-[EQDK]-[TKY]-[P](0,1)-[G]-[DVHI]-[NDAKT]-[V]-[VTIE]
BB11024	[P]-[LQI]-[RLK]-[MLY]-[ANY]-[EQYG]-[FWNY]-[GA]-[SNP]-[CVMA]-[HVF]-[R]
BB11025	[KFSD]-[NTK]-[IV]-[QKL]-[SEK]-[LVY]-[EAWT]-[VLRE]-[IKVS]-[GLM]-[KA]-[G](0,1)
BB11026	[ILD]-[EHVL]-[NDE]-[VCA]-[KLAI]-[AMRELT]-[KSERAD]-[IAQN]-[QL](0,1)-[DKQGRN]-[KVLIQ]-[ERTDN]
BB11027	[IML]-[FV]-[VLIM]-[LISEV]-[G]-[GPAVL]-[PSHDE]-[GRA]-[ATVR]-[G]-[KR]-[GSRT]
BB11028	[ADSG]-[DAN]-[SDATKQE]-[GADH]-[NKTEISR]-[YR]-[KTVSLFW]-[LCV]-[KVEISTAQ]-[VAGI]-[KVSRTEY]-[NCADG]
BB11029	[KST]-[EDA]-[KAG]-[DQ]-[RIA]-[NAKE]-[DNA]-[LV]-[IAV]-[TAD]-[YW]-[LYI]
BB11030	[SRQKTGEA]-[VCILRPA]-[DLE]-[GVAILST]-[LVI]-[VLIF]-[NSHC]-[NSVGFYM]-[ADIG]-[GSIAE]-[ISFGTLQE]-[SFALMWGRKP]
BB11031	[AXIL]-[QAKSIR]-[KQSNEDW]-[D](0,1)-[MWHSLKA]-[GD]-[LFAYMI]-[KND]-[IFTVHA]-[AVIF]-[R]-[LIVAT]
BB11032	[FILVM]-[IWGVL]-[LMI]-[VLI]-[AGS]-[GT]-[G]-[ITV]-[GA]-[ILFV]-[TAGSP]-[PY]
BB11033	[QKDMT](0,1)-[PMLVSTY]-[VQRGILS]-[LTVIFM]-[VLFR]-[YIDVNE]-[FYILV]-[WGSVY]-[ARTE]-[STPD]-[WGTV]-[CTV]
BB11034	[TASN]-[NSTRID]-[VECSHI]-[KP](0,1)-[GHND]-[IV]-[YFWH]-[AVL]-[VILA]-[G]-[D]-[VILA]
BB11035	[EAQG]-[EDQ]-[LIRA]-[KAE]-[AND]-[LV]-[AV]-[DA]-[YW]-[MYL]-[SAM]-[KSTE]
BB11036	[GAV]-[VIL]-[STDE]-[LCQSM]-[A]-[ALI]-[SALW]-[RED]-[AL]-[ATVLQ]-[ADG]-[AKRQ](0,1)
BB11037	[RL]-[LV]-[WF]-[WC]-[TSAYC]-[VAIC]-[YF]-[MCYSL]-[FVTL]-[ED]-[RVK]-[MLF]
BB11038	[VIA]-[YH]-[RT]-[D]-[IL]-[KR]-[PAS]-[DASEK]-[N]-[FILV]-[LA]-[IVLM]
BB12001	[P]-[LV]-[IVAL]-[ILAM]-[W]-[LFT]-[NTQ]-[G]-[G]-[P]-[G]-[CG]
BB12002	[LMV]-[G]-[PVTQ]-[AIVGH]-[LM]-[GNH]-[PKR]-[RGV]-[G]-[KL]-[FM]-[P]
BB12003	[NLMFVS]-[ASGK]-[C]-[YW]-[C]-[YNEQI]-[KAGDY]-[LAV]-[PE]-[DEKN]-[HNDS]-[VAKE]
BB12004	[AV]-[Y]-[E]-[P]-[VLIES]-[WS]-[A]-[I]-[GN]-[DK](0,1)-[TKAS]-[GSD]
BB12005	[LINV]-[D]-[MTVAI]-[WY]-[E]-[H]-[AS]-[FY]-[YH]-[LVMY]-[QDR]-[YF]
BB12006	[LM]-[VLI]-[SA]-[Y]-[AT]-[P]-[P]-[G]-[AMG]-[DEG]-[PMK]-[P]
BB12007	[C]-[E]-[Y]-[AMSGV]-[H]-[A]-[M]-[G]-[N]-[SG]-[LVPN]-[G]
BB12008	[ASC]-[FLYVMI]-[GACLM]-[LMFAIV]-[TS]-[E]-[PRA]-[NGQDA]-[ASV]-[G]-[TS]-[DN]
BB12009	[K]-[WTQ]-[F]-[NDS]-[SNTGR]-[EAQKS]-[KSQ]-[G]-[FKH]-[G]-[FL]-[I]
BB12010	[LV]-[HN]-[Y]-[SG]-[LQT]-[QSGA]-[LVCA]-[FY]-[EG]-[G]-[MLI]-[KQR]
BB12011	[Y]-[TSI]-[DTEAN]-[YH]-[ACS]-[VTI]-[RKQNEHST]-[WYTV]-[Y]-[NQDT]-[TVADRK]-[G]
BB12012	[D]-[VI]-[VAT]-[AG]-[H]-[E]-[ILV]-[TS]-[H]-[G]-[VIF]-[T]
BB12013	[VSI]-[W]-[NI]-[YF]-[HQ]-[CVT]-[WS]-[NVT]-[ED]-[ASGCV]-[W]-[MF]
BB12014	[RQV]-[IV]-[QKR]-[VINT]-[W]-[FVY]-[QSI]-[N]-[RHK]-[R]-[AMYRCT]-[RK]
BB12015	[G]-[WFL]-[ETD]-[EKTQAIV]-[GAH]-[VLI]-[AQLIEFV]-[QLDKNGT]-[MARKLF]-[SPEQKA]-[VAIEK]-[G]
BB12016	[VA]-[L]-[LV]-[D]-[T]-[G]-[AV]-[D]-[TDR]-[ST]-[VI]-[LVI]
BB12017	[QEYK]-[PF]-[YFIVL]-[VCQL]-[TN]-[LM]-[FYHS]-[H]-[WFY]-[DPE]-[VLTM]-[P]
BB12018	[G]-[G]-[L]-[G]-[R]-[L]-[A]-[AS]-[C]-[F]-[LI]-[D]
BB12019	[P]-[P]-[E]-[P]-[NS]-[G]-[YI]-[L]-[H]-[I]-[G]-[H]
BB12020	[G]-[STE]-[VR]-[VI]-[TRDE]-[Y]-[SKE]-[C]-[NSKR]-[SGKP]-[GT]-[YF]

BB12021	[LETS]-[VI]-[SNKVE]-[AVEP]-[NEKDA]-[G]-[W]-[C]-[TASLQ]-[ASV]-[WY]-[VTA]
BB12022	[N]-[P]-[D]-[GRN]-[DER]-[ELRV]-[ERQS]-[G](0,1)-[AP]-[W]-[C]-[YF]
BB12023	[S]-[C]-[H]-[T]-[G]-[LVI]-[RGN]-[RK]-[TSN]-[AV]-[G]-[WY]
BB12024	[HR]-[I]-[GE]-[I]-[D]-[VI]-[N]-[S]-[IL]-[VRI]-[SP]-[VIT]
BB12025	[VA]-[TS]-[L]-[GTV]-[C]-[LT]-[VIA]-[KTR]-[GD]-[YF]-[FY]-[P]
BB12026	[FVILM]-[ILV]-[G]-[IV]-[N]-[SNTA]-[R]-[DNS]-[L]-[EADKCRHG]-[TRDK]-[LF]
BB12027	[AGTS]-[IVLK]-[AMLV]-[LQVGIM]-[D]-[TL]-[KQPN]-[G]-[PALI]-[EK]-[ILVM]-[R]
BB12028	[YF]-[WI]-[LI]-[VAI]-[AKR]-[N]-[SF]-[W]-[NGT]-[AKSPET]-[DSQPG]-[W]
BB12029	[FL]-[QRKFV]-[PL]-[VSATDG]-[YH]-[FNY]-[P]-[FY]-[VT]-[ES]-[P]-[GS]
BB12030	[WY]-[TNK]-[RV]-[L]-[P]-[Q]-[G]-[FWM]-[KVAT]-[NGLC]-[S]-[P]
BB12031	[D]-[Y]-[ST]-[Q]-[I]-[E]-[LM]-[RVA]-[VIL]-[LM]-[AS]-[H]
BB12032	[GQE]-[KQEG]-[NDTKH]-[LISNYV]-[C]-[YF]-[KLRT]-[MKNEYR]-[FASIRTW]-[MQRWTH]-[VCRTM]-[ADHFVEI]
BB12033	[G]-[T]-[S]-[AMS]-[AS]-[ASTC]-[P]-[LHG]-[AV]-[AS]-[G]-[VILA]
BB12034	[R]-[S]-[CN]-[D]-[VMI]-[FALP]-[LV]-[GA]-[LNH]-[PHN]-[FI]-[N]
BB12036	[MLSV]-[G]-[LPAG]-[P]-[GA]-[AS]-[G]-[K]-[G]-[T]-[QV]-[ASC]
BB12037	[WF]-[VLI]-[HKEQLA]-[DRESQK]-[NH]-[IAV]-[QHEGAVSK]-[FANVS]-[F]-[G]-[G]-[DNE]
BB12038	[WYFV]-[TRI]-[YF]-[PRLASENH]-[G]-[S]-[L]-[T]-[TS]-[P]-[PT]-[LC]
BB12039	[A](0,1)-[VADSTEC]-[C]-[EKVSI]-[PDEQGNT]-[ELVYMTNQ]-[C]-[P]-[NTSVQAMI]-[GVEDASN]-[ASCV]-[ILF]
BB12040	[H]-[KR]-[KEL]-[ETI]-[H]-[DE]-[GKISL]-[F]-[IV]-[NQKR]-[ADTKR]-[LAV]
BB12041	[VIAE]-[KAES]-[C]-[DQVN]-[DQTKSA]-[C]-[H]-[H]-[DTLKMV]-[PVWLF]-[GPDEV]-[DGAN]
BB12042	[G]-[IV]-[IVM]-[LI]-[T]-[APG]-[SA]-[H]-[N]-[P]-[GP]-[GED]
BB12044	[E]-[YFH]-[VT]-[S]-[AV]-[N]-[P]-[TN]-[GK]-[PDFYE]-[MLI]-[HN]
BB20001	[PKI]-[RK]-[GRPSK]-[KPIRA](0,1)-[MLPVIT]-[SETN]-[SAG]-[YHF]-[AMNFL]-[FLNYVQ]-[FWTY]-[VLKSMF]
BB20002	[GNRDKSA](0,1)-[HSQKRNED]-[ETIVLQKR]-[GLK]-[YCFWELIV]-[AIFV]-[PG]-[SAWYRET]-[STNVDR]-[YFPLIK]-[LVSIA]-[VATQNERGK]
BB20020	[QT]-[SMTCV]-[TIVA]-[SA]-[E]-[AV]-[ASL]-[R]-[YF]-[KQNRT]-[FYPQ]-[I]
BB30006	[DNHEPS]-[G]-[TKSAHNLVD]-[FY]-[LM]-[VILA]-[R]-[DAEQPKF]-[ARSC]-[SDEKNART]-[TNSER]-[KNSAVPHT]
BB30017	[AGQLENH]-[FW]-[ETAKS]-[QRKES]-[EADQK]-[TKN]-[G]-[G](0,1)-[IHVDQ]-[KADTEN]-[VL]-[TKVIR]
BB30027	[EAQTK](0,1)-[EDQNK]-[LMIARQV]-[KEHAN]-[AVQNDL]-[LMIV]-[AVI]-[DTEA]-[YHWF]-[MIFLY]-[SGAKVE]-[KTSENQ]
BB40003	[P]-[VIL]-[IVL]-[LI]-[W]-[L]-[NT]-[G]-[G]-[P]-[G]-[C]
BB40005	[R]-[GACKRN]-[FM]-[THQ]-[QML]-[DPN]-[D]-[AMG]-[H]-[IT]-[FIL]-[CVA]
BB40006	[LVMIF]-[VFYLI]-[VLIF]-[NCTIG]-[STN]-[P]-[NQSGH]-[N]-[P]-[TLSI]-[GA]-[ALQKTVR]
BB40007	[HQ]-[V]-[Q]-[CIV]-[N]-[ASPV]-[STN]-[KQP]-[FG]-[HQT]-[QA]-[G]
BB40008	[ASCT]-[FLYM]-[CAGM]-[LMVAQ]-[TS]-[E]-[PALM]-[AGNS]-[SAH]-[G]-[ST]-[DNH]
BB40009	[LITVK]-[GASTP]-[FY]-[GS]-[FYQWHSRNAG]-[G]-[PDIASVQKR]-[HR]-[RAFMLHVG]-[C]-[IPMLHV]-[AG]
BB40010	[K]-[WTQ]-[FY]-[NSD]-[SADTR]-[EDQTKS]-[KSQ]-[GN]-[FYKH]-[G]-[F]-[I]
BB40014	[N]-[A]-[AFY]-[W]-[INDY]-[GN]-[DSQTERG]-[KQA]-[M]-[IVLT]-[YF]-[G]
BB40018	[LVI]-[LI]-[DN]-[T]-[G]-[AV]-[D]-[TDIVKA]-[ST]-[VI]-[LVI]-[TEANSK]
BB40019	[YIF]-[R]-[FILT]-[S]-[IV]-[SA]-[W]-[PST]-[R]-[VIL]-[LFV]-[P]
BB40022	[C]-[R]-[NS]-[P]-[DRG]-[GRANS]-[DERSVQA]-[ELIRAVKG]-[ERGTKQSN]-[GSA](0,1)-[AP]-[W]
BB40025	[Q]-[SMT]-[TIAV]-[SPAT]-[E]-[AV]-[AS]-[R]-[YF]-[KQNRT]-[FYP]-[I]
BB40032	[L]-[P]-[Q]-[G]-[FWM]-[KAL]-[NG]-[S]-[P]-[TA]-[LI]-[FC]
BB40033	[KAPSTVN]-[EFYWH]-[ERKNQHTA]-[NDAE]-[L]-[P]-[ILQYVM]-[YRLKI]-[LMYFIV]-[TVACSNY]-[ACGEDQ]-[YSLFWIV]
BB40043	[LVFMI]-[WYGALIV]-[MFICL]-[LFIV]-[ASG]-[TG]-[G]-[TSV]-[AG]-[ILFM]-[GATS]-[PY]
BB40045	[VIALEQG]-[KAEDS]-[C]-[DQVGTN]-[DQTKSA]-[C]-[H]-[HTA]-[DTLPKMV]-[PVWLFD]-[GPDNEVI]-[DGANEK]
BB40048	[P]-[P]-[ED]-[PA]-[NS]-[G]-[YI]-[LA]-[H]-[IL]-[G]-[H]
BB50002	[VAITYL]-[GAT]-[RFHLKA]-[PVILG]-[NSDGE]-[VAHSI]-[GI]-[KR]-[STG]-[TS]-[LTAI]-[LFTV]
BB50004	[D]-[VLIY]-[YF]-[IV]-[NSMDI]-[D]-[A]-[FY]-[GA]-[TAV]-[AI]-[H]
BB50005	[C]-[H]-[TS]-[G]-[LIV]-[RGSN]-[RK]-[TSN]-[ADV]-[G]-[WY]-[NK]
BB50010	[P]-[ILQYVMF]-[YRLKI]-[LMYFIVR]-[TVACSNY]-[ACGEDQY]-[YSLFWIVN]-[TSAGCNV]-[PTLNSCQK]-[CAVLTM]-[YFHV]-[R]
BB50013	[GHN]-[NDS]-[TCLAIM]-[H]-[IVL]-[Y]-[MTSLNVKD]-[NR]-[HQ]-[VIFLY]-[EDYN]-[QNLAKG]
BB50016	[IVALCM]-[HY]-[RCT]-[D]-[ILV]-[K]-[GPSA]-[ASKEQH]-[N]-[ICVL]-[LMI]-[IVL]
BBS11001	[WY]-[KH]-[TMEA]-[ML]-[STP]-[AER]-[KAE]-[EK]-[KQ]-[GWMA]-[KP]-[FY]
BBS11002	[QERILTK]-[GK]-[WYDV]-[VF]-[P]-[SRGEA]-[NTRSM]-[YLKFIH]-[ILVAT]-[TGEKRQ]-[PLYIREK]-[VYIDSL]
BBS11003	[AP]-[VL]-[AK]-[AVS]-[G]-[NC]-[ATP]-[VI]-[IL]-[LAV]-[KR]-[PG]
BBS11004	[IWQ]-[ELV]-[EA]-[NP]-[LVI]-[DQK]-[LV]-[FVA]-[IV]-[VMI]-[TNP]-[MIL]
BBS11005	[VLMFI]-[LFIVYA]-[VLIAS]-[TCENPH]-[NTSPVA]-[PITC]-[VHCA](0,1)-[SNQLVH]-[NGSR]-[PASNE]-[LTCGS]-[G]
BBS11006	[NQSA]-[STMNP]-[KSATQLC]-[FCVAIT](0,1)-[TSHR]-[FMIL]-[SVTA]-[IVL]-[P]-[YWF]-[LIVNT]-[SGAN]
BBS11007	[HR]-[MFLAR]-[C]-[LMIVP]-[G]-[QREAS]-[HPQGAD]-[LFI]-[ATG]-[KRLI]-[LRHME]-[EQVH]

BBS11008	[DP]-[G]-[TL]-[F]-[LI]-[VIL]-[R]-[DEF]-[AS]-[SQE]-[TRS]-[KNS]
BBS11009	[SAD]-[C]-[E](0,1)-[RKE]-[ASE]-[GC]-[SAET]-[C]-[STG]-[ST]-[C]-[AHRK]
BBS11010	[GSN]-[D]-[ACS]-[E]-[LVI]-[VIL]-[LA]-[RALN]-[LI]-[LYF]-[EQTA]-[RES]
BBS11011	[LVCG]-[PQAR]-[G]-[MND]-[CSXA]-[G]-[GSR]-[AGPS]-[LIV]-[VFM]-[SNCAD]-[SGN]
BBS11012	[LIA]-[YWH]-[FA]-[NK]-[GA]-[QRS]-[W]-[KTV]-[TN]-[PK]-[F]-[PED]
BBS11013	[F](0,1)-[NSVM]-[RS]-[WRLQ]-[AQHGR]-[ES]-[IML]-[AQSVK]-[KRAN]-[LYKAE]-[LI]-[PKSQG]
BBS11014	[ED]-[LVI]-[E]-[SNR]-[IHAV]-[L]-[LVGM]-[QKATG]-[HA]-[PM]-[NYAKG]-[IV]
BBS11015	[VI]-[VI]-[TSG]-[N]-[P]-[VA]-[DN]-[VT]-[MINT]-[VTA]-[QAY]-[LI]
BBS11016	[TSQN]-[SITAD]-[MVEHCRI]-[HPKS]-[NDG]-[VI]-[WYFH]-[AVL]-[IVLA]-[G]-[D]-[VLA]
BBS11017	[TV]-[ATS]-[E]-[VA]-[ALS]-[R]-[FY]-[RXK]-[YQF]-[I]-[QE]-[NRQ]
BBS11018	[NGD]-[IMVLX]-[KYRGNS]-[VLIA]-[IMYVL]-[IVLMQGY]-[D]-[FVAWL]-[AVIP]-[PIYFL]-[NDSGH]-[H]
BBS11019	[NPGAQ]-[MVILASF]-[IVPGA]-[LAICPD]-[G]-[HNTMI](0,1)-[EDG]-[AGIVF]-[VASTLI]-[GF]-[EITRV]-[V]
BBS11020	[FWY]-[FIVL]-[VCAGD]-[GT]-[NQDE]-[SRKGNE]-[MFIVL]-[TGS]-[LIYM]-[AVI]-[D]-[LAFVYC]
BBS11021	[PGQ]-[H]-[N]-[VIW]-[HEV]-[FTL]-[VID]-[AKIN]-[GP]-[VMAG]-[LAGD]-[GPAD]
BBS11022	[QK]-[LR](0,1)-[GK]-[LY]-[NSTP]-[QLN]-[ASTK]-[EANM]-[LVI]-[ASG]-[QRE]-[KQAR]
BBS11023	[HQAKE]-[WF]-[KAET]-[QK]-[EQDK]-[TKY]-[P](0,1)-[G]-[DVHI]-[NDAKT]-[V]-[VTIE]
BBS11024	[P]-[LQI]-[RLK]-[MLY]-[ANY]-[EQYG]-[FWNY]-[GA]-[SNP]-[CVMA]-[HVF]-[R]
BBS11025	[KFSD]-[NTK]-[IV]-[QKL]-[SEK]-[LVY]-[EAWT]-[VLRE]-[IKVS]-[GLM]-[KA]-[G](0,1)
BBS11026	[ILD]-[EHVL]-[NDE]-[VCA]-[KLAI]-[AMRELT]-[KSERAD]-[IAQN]-[QL](0,1)-[DKQGRN]-[KVLIQ]-[ERTDN]
BBS11027	[IML]-[FV]-[VLIM]-[LISEV]-[G]-[GPAVL]-[PSHDE]-[GRA]-[ATVR]-[G]-[KR]-[GSRT]
BBS11028	[ADSG]-[DAN]-[SDATKQE]-[GADH]-[NKTEISR]-[YR]-[KTVSLFW]-[LCV]-[KVEISTAQ]-[VAGI]-[KVSRTEY]-[NCADG]
BBS11029	[KST]-[EDA]-[KAG]-[DQ]-[RIA]-[NAKE]-[DNA]-[LV]-[IAV]-[TAD]-[YW]-[LYI]
BBS11030	[SRQKTGEA]-[VCILRPA]-[DLE]-[GVAILST]-[LVI]-[VLIF]-[NSHC]-[NSVGFYM]-[ADIG]-[GSIAE]-[ISFGTLQE]-[SFALMWGRKP]
BBS11031	[AXIL]-[QAKSIR]-[KQSNEDW]-[D](0,1)-[MWHSLKA]-[GD]-[LFAYMI]-[KND]-[IFTVHA]-[AVIF]-[R]-[LIVAT]
BBS11032	[FILVM]-[IWGVL]-[LMI]-[VLI]-[AGS]-[GT]-[G]-[ITV]-[GA]-[ILFV]-[TAGSP]-[PY]
BBS11033	[PMLVSTY]-[VQRGILS]-[LTVIFM]-[VLFR]-[YIDVNE]-[FYILV]-[WGSVY]-[ARTE]-[STPD]-[WGTV]-[CTV]-[G](0,1)
BBS11034	[TASN]-[NSTRID]-[VECSHI]-[KP](0,1)-[GHND]-[IV]-[YFWH]-[AVL]-[VILA]-[G]-[D]-[VILA]
BBS11035	[EAQG]-[EDQ]-[LIRA]-[KAE]-[AND]-[LV]-[AV]-[DA]-[YW]-[MYL]-[SAM]-[KSTE]
BBS11036	[GAV]-[VIL]-[STDE]-[LCQSM]-[A]-[ALI]-[SALW]-[RED]-[AL]-[ATVLQ]-[ADG]-[AKRQ](0,1)
BBS11037	[RL]-[LV]-[WF]-[WC]-[TSAYC]-[VAIC]-[YF]-[MCYSL]-[FVTL]-[ED]-[RVK]-[MLF]
BBS11038	[VIA]-[YH]-[RT]-[D]-[IL]-[KR]-[PAS]-[DASEK]-[N]-[FILV]-[LA]-[IVLM]
BBS12001	[P]-[LV]-[IVAL]-[ILAM]-[W]-[LFT]-[NTQ]-[G]-[G]-[P]-[G]-[CG]
BBS12002	[LMV]-[G]-[PVTQ]-[AIVGH]-[LM]-[GNH]-[PKR]-[RGV]-[G]-[KL]-[FM]-[P]
BBS12003	[NLMFVS]-[ASGK]-[C]-[YW]-[C]-[YNEQI]-[KAGDY]-[LAV]-[PE]-[DEKN]-[HNDS]-[VAKE]
BBS12004	[AV]-[Y]-[E]-[P]-[VLIES]-[WS]-[A]-[I]-[GN]-[DK](0,1)-[TKAS]-[GSD]
BBS12005	[LINV]-[D]-[MTVAI]-[WY]-[E]-[H]-[AS]-[FY]-[YH]-[LVMY]-[QDR]-[YF]
BBS12006	[LM]-[VLI]-[SA]-[Y]-[AT]-[P]-[P]-[G]-[AMG]-[DEG]-[PMK]-[P]
BBS12007	[C]-[E]-[Y]-[AMSGV]-[H]-[A]-[M]-[G]-[N]-[SG]-[LVPN]-[G]
BBS12008	[ASC]-[FLYVMI]-[GACLM]-[LMFAIV]-[TS]-[E]-[PRA]-[NGQDA]-[ASV]-[G]-[TS]-[DN]
BBS12009	[K]-[WTQ]-[F]-[NDS]-[SNTGR]-[EAQKS]-[KSQ]-[G]-[FKH]-[G]-[FL]-[I]
BBS12010	[LV]-[HN]-[Y]-[SG]-[LQT]-[QSGA]-[LVCA]-[FY]-[EG]-[G]-[MLI]-[KQR]
BBS12011	[Y]-[TSI]-[DTEAN]-[YH]-[ACS]-[VTI]-[RKQNEHST]-[WYTV]-[Y]-[NQDT]-[TVADRK]-[G]
BBS12012	[D]-[VI]-[VAT]-[AG]-[H]-[E]-[ILV]-[TS]-[H]-[G]-[VIF]-[T]
BBS12013	[VSI]-[W]-[NI]-[YF]-[HQ]-[CVT]-[WS]-[NVT]-[ED]-[ASGCV]-[W]-[MF]
BBS12014	[RQV]-[IV]-[QKR]-[VINT]-[W]-[FVY]-[QSI]-[N]-[RHK]-[R]-[AMYRCT]-[RK]
BBS12015	[G]-[WFL]-[ETD]-[EKTQAIV]-[GAH]-[VLI]-[AQLIEFV]-[QLDKNGT]-[MARKLF]-[SPEQKA]-[VAIEK]-[G]
BBS12016	[VA]-[L]-[LV]-[D]-[T]-[G]-[AV]-[D]-[TDR]-[ST]-[VI]-[LVI]
BBS12017	[QEYK]-[PF]-[YFIVL]-[VCQL]-[TN]-[LM]-[FYHS]-[H]-[WFY]-[DPE]-[VLTM]-[P]
BBS12018	[G]-[G]-[L]-[G]-[R]-[L]-[A]-[AS]-[C]-[F]-[LI]-[D]
BBS12019	[P]-[P]-[E]-[P]-[NS]-[G]-[YI]-[L]-[H]-[I]-[G]-[H]
BBS12020	[G]-[STE]-[VR]-[VI]-[TRDE]-[Y]-[SKE]-[C]-[NSKR]-[SGKP]-[GT]-[YF]
BBS12021	[LETS]-[VI]-[SNKVE]-[AVEP]-[NEKDA]-[G]-[W]-[C]-[TASLQ]-[ASV]-[WY]-[VTA]
BBS12022	[N]-[P]-[D]-[GRN]-[DER]-[ELRV]-[ERQS]-[G](0,1)-[AP]-[W]-[C]-[YF]
BBS12023	[S]-[C]-[H]-[T]-[G]-[LVI]-[RGN]-[RK]-[TSN]-[AV]-[G]-[WY]
BBS12024	[HR]-[I]-[GE]-[I]-[D]-[VI]-[N]-[S]-[IL]-[VRI]-[SP]-[VIT]
BBS12025	[VA]-[TS]-[L]-[GTV]-[C]-[LT]-[VIA]-[KTR]-[GD]-[YF]-[FY]-[P]
BBS12026	[FVILM]-[ILV]-[G]-[IV]-[N]-[SNTA]-[R]-[DNS]-[L]-[EADKCRHG]-[TRDK]-[LF]
BBS12027	[AGTS]-[IVLK]-[AMLV]-[LQVGIM]-[D]-[TL]-[KQPN]-[G]-[PALI]-[EK]-[ILVM]-[R]

BBS12028	[YF]-[WI]-[LI]-[VAI]-[AKR]-[N]-[SF]-[W]-[NGT]-[AKSPET]-[DSQPG]-[W]
BBS12029	[FL]-[QRKFV]-[PL]-[VSATDG]-[YH]-[FNY]-[P]-[FY]-[VT]-[ES]-[P]-[GS]
BBS12030	[WY]-[TNK]-[RV]-[L]-[P]-[Q]-[G]-[FWM]-[KVAT]-[NGLC]-[S]-[P]
BBS12031	[D]-[Y]-[ST]-[Q]-[I]-[E]-[LM]-[RVA]-[VIL]-[LM]-[AS]-[H]
BBS12032	[GQE]-[KQEG]-[NDTKH]-[LISNYV]-[C]-[YF]-[KLRT]-[MKNEYR]-[FASIRTW]-[MQRWTH]-[VCRTM]-[ADHFVEI]
BBS12033	[G]-[T]-[S]-[AMS]-[AS]-[ASTC]-[P]-[LHG]-[AV]-[AS]-[G]-[VILA]
BBS12034	[R]-[S]-[CN]-[D]-[VMI]-[FALP]-[LV]-[GA]-[LNH]-[PHN]-[FI]-[N]
BBS12036	[MLSV]-[G]-[LPAG]-[P]-[GA]-[AS]-[G]-[K]-[G]-[T]-[QV]-[ASC]
BBS12037	[WF]-[VLI]-[HKEQLA]-[DRESQK]-[NH]-[IAV]-[QHEGAVSK]-[FANVS]-[F]-[G]-[G]-[DNE]
BBS12038	[WYFV]-[TRI]-[YF]-[PRLASENH]-[G]-[S]-[L]-[T]-[TS]-[P]-[PT]-[LC]
BBS12039	[A](0,1)-[VADSTEC]-[C]-[EKVSI]-[PDEQGNT]-[ELVYMTNQ]-[C]-[P]-[NTSVQAMI]-[GVEDASN]-[ASCV]-[ILF]
BBS12040	[H]-[KR]-[KEL]-[ETI]-[H]-[DE]-[GKISL]-[F]-[IV]-[NQKR]-[ADTKR]-[LAV]
BBS12041	[VIAE]-[KAES]-[C]-[DQVN]-[DQTKSA]-[C]-[H]-[H]-[DTLKMV]-[PVWLF]-[GPDEV]-[DGAN]
BBS12042	[G]-[IV]-[IVM]-[LI]-[T]-[APG]-[SA]-[H]-[N]-[P]-[GP]-[GED]
BBS12044	[E]-[YFH]-[VT]-[S]-[AV]-[N]-[P]-[TN]-[GK]-[PDFYE]-[MLI]-[HN]
BBS20001	[PKI]-[RK]-[GRPSK]-[KPIRA](0,1)-[MLPVIT]-[SETN]-[SAG]-[YHF]-[AMNFL]-[FLNYVQ]-[FWTY]-[VLKSMF]
BBS20002	[TKNQPECGYLRS]-[LITKFAEYVM]-[VFYAIR]-[IVRKY]-[AVY]-[LVRMY]-[YFGWQR]-[DKPNA]-[YFCS]-[QEDMVHKTAR]-[TASPGQD]-[NRMQGVEDTP](0,1)
BBS20020	[QT]-[SMTCV]-[TIVA]-[SA]-[E]-[AV]-[ASL]-[R]-[YF]-[KQNRT]-[FYPQ]-[I]
BBS30006	[DNHEPS]-[G]-[TKSAHNLVD]-[FY]-[LM]-[VILA]-[R]-[DAEQPKF]-[ARSC]-[SDEKNART]-[TNSER]-[KNSAVPHT]
BBS30017	[AGQLENH]-[FW]-[ETAKS]-[QRKES]-[EADQK]-[TKN]-[G]-[G](0,1)-[IHVDQ]-[KADTEN]-[VL]-[TKVIR]
BBS30027	[EAQTK](0,1)-[EDQNK]-[LMIARQV]-[KEHAN]-[AVQNDL]-[LMIV]-[AVI]-[DTEA]-[YHWF]-[MIFLY]-[SGAKVE]-[KTSENQ]
BBS50002	[VAITYL]-[GAT]-[RFHLKA]-[PVILG]-[NSDGE]-[VAHSI]-[GI]-[KR]-[STG]-[TS]-[LTAI]-[LFTV]
BBS50005	[C]-[H]-[TS]-[G]-[LIV]-[RGSN]-[RK]-[TSN]-[ADV]-[G]-[WY]-[NK]
BBS50010	[P]-[ILQYVMF]-[YRLKI]-[LMYFIVR]-[TVACSNY]-[ACGEDQY]-[YSLFWIVN]-[TSAGCNV]-[PTLNSCQK]-[CAVLTM]-[YFHV]-[R]
BBS50013	[GHN]-[NDS]-[TCLAIM]-[H]-[IVL]-[Y]-[MTSLNVKD]-[NR]-[HQ]-[VIFLY]-[EDYN]-[QNLAKG]
BBS50016	[IVALCM]-[HY]-[RCT]-[D]-[ILV]-[K]-[GPSA]-[ASKEQH]-[N]-[ICVL]-[LMI]-[IVL]

Table B.1: Most conserved (MC) constraints list for the working database used with RE-MuSiC.

BB11001	[EKNL]-[MYL]-[KPA]-[TNRG]-[YFW]-[IKRS]-[PYEA]-[PRD]-[KPHD]-[GRPN]-[ERDY]-[KLG](0,1)
BB11002	[KGQMI](0,1)-[NGYFSE]-[LVQFKMPI]-[FIYRVA]-[VYRQ]-[AYV]-[LYQIT]-[YWRHQKS]-[DA]-[FYS]-[VEKRTM]-[APK](0,1)
BB11003	[DERK]-[TYLA]-[TYFC]-[IAE]-[KRDT]-[EAV]-[LAVI]-[PLNG]-[DRSL]-[WYGK]-[ARSD]-[EYP](0,1)
BB11004	[RQVY]-[SINM]-[YFM]-[INK]-[EN](0,1)-[HQVL]-[KGLW]-[MLDV]-[QKA]-[GSTV]-[NYSH]-[PRT]
BB11005	[TPSE](0,1)-[KSQAGDPERV]-[NAEPDGQ]-[YWHLTFGIR]-[IQLVYF]-[AQDESVKH]-[EAQRYNM]-[NTILVAXRDW]-[HWVRANDTI]-[KHAIDVQEG]-[RWKYIMAFNQ]-[LIEGTYVA]
BB11006	[LETAVY](0,1)-[KAIDEQL]-[VEFNTR]-[FITDVYR]-[NDFAGHS]-[PTEVQG]-[PMGSEVA]-[RILGVTN](0,1)-[NPAYQ]-[QFGNLIHV]-[LDTPQR]-[PLSERQ]
BB11007	[AVTPGHS](0,1)-[RAFGDTL]-[TDGPHKVR]-[VFEIDSL]-[IPDLEA]-[LDMKGHEW]-[PLYNGV]-[QENLIHSPY]-[GSPLF]-[YDSLQETK]-[ASNLIED]-[DFLNR]
BB11008	[EQK]-[LD]-[IVTH]-[NEA]-[HFL]-[YHG]-[RQSK]-[NLFY]-[ENAY]-[SRP]-[LGKR]-[ARG](0,1)
BB11009	[GDV](0,1)-[NGT]-[VART]-[EVFS]-[FLI]-[QEC]-[CAI]-[PNA]-[DSQA]-[DGN]-[VER]-[YTS]
BB11010	[TVRD]-[FRTG]-[DKRL]-[GLWI]-[LTF]-[NAGQ]-[ELD]-[MHFL]-[SMAN]-[PYTD]-[VDHL]-[LCGI]
BB11011	[QNVS]-[EYTK]-[MVEG]-[FKSA]-[QRWLT]-[NIA]-[IKY]-[DNQGT]-[KETA]-[KITP]-[INAVE]-[ESLG](0,1)
BB11012	[GAR](0,1)-[PVRT]-[WTNA]-[NDFP]-[KSGQ]-[DPYK]-[EDTN]-[IL]-[SARK]-[TS]-[TRVA]-[DTNS]
BB11013	[NKVR]-[HLYMG]-[WSL]-[NIKT]-[SRDK]-[TRPFL]-[MLVGK]-[RLSA]-[RAYV]-[KAGRP]-[VGT]-[VSRE](0,1)
BB11014	[VYSL](0,1)-[RNSYH]-[FLAVS]-[SKAL]-[HVRTFA]-[AFSTM]-[RFVS]-[DEKTL]-[PNVYQ]-[ISCVD]-[FLWG]-[GNDLT](0,1)
BB11015	[TQ](0,1)-[LAG]-[KHGE]-[KAQ]-[VEWG]-[LTV]-[IVT]-[CVME]-[SGC]-[TNIA]-[LRPY]-[LRSV]
BB11016	[IADKMPHT]-[VGLAPR]-[DRNEHSP]-[RNSGFYL]-[LGVIDYR]-[TPRKAVEM]-[THYES]-[GMKAQR]-[VITQM](0,1)-[AMGYRDTH]-[ANEDLK]-[LFKGDPH]
BB11017	[DHIP]-[LSTA]-[L]-[LTKD]-[TS]-[FSRA]-[MYI]-[ELGT]-[ADXT]-[VLI]-[NMYF]-[KSGY]
BB11018	[MYKDTVAHRI]-[YNDISEQKA]-[GMWIYNLTKSD]-[LIGVTW]-[KETSRHN]-[ASGQHLFKR]-[MVWYDRGAT]-[LAGKMYIDV]-[ESFDNLTVI]-[GDMVYTAS]-[SCPFVNRK]-[EVSHGDP](0,1)
BB11019	[KLTM](0,1)-[GSYDVKRTF]-[RKCTMEP]-[LDNSKEGP]-[RSDHYTLAEG]-[AVYFNLTRI]-[EPVRNL]-[MKIAQE]-[LATISEQ]-[RVLMINAS]-[DAESQLW]-[MDIAWLGQ]
BB11020	[EWVFMLI]-[ILWYQF]-[TNEVLFSA]-[EYNADLKQ]-[TIQYLMAF]-[LAVFGNS]-[QTMDSAG]-[DELYH](0,1)-[ITKYLA](0,1)-[FLRVQCKP]-[NHLVPRIM]-[DKQASFI](0,1)
BB11021	[VIW]-[HEV]-[FTL]-[VID]-[AKIN]-[GP]-[VMAG]-[LAGD]-[GPAD]-[EDPV]-[AG]-[AT](0,1)
BB11022	[DET](0,1)-[SWQP]-[IHFR]-[SRAE]-[SANR]-[RDE]-[VFI]-[KIL]-[SAVK]-[KGRL]-[RLI]-[IKA]
BB11023	[EAKLQ]-[IAQEK]-[AFILVG]-[AKVF]-[KRAEN]-[NYATKM]-[FGRIDN]-[YFAGKL]-[RTVEYPG]-[PILYW]-[RKPTGN]-[DQLPA](0,1)
BB11024	[CALT]-[GCH]-[TGS]-[VSGC]-[QHGS]-[LYRA]-[DLY]-[FGNL]-[SEGD]-[LNW](0,1)-[PFLQ]-[SAL]
BB11025	[KDLE]-[KTDR]-[IM]-[VLK]-[QGAD]-[KFRT]-[KAL]-[LIER]-[ASRI]-[GAET]-[DYK]-[AKL](0,1)
BB11026	[DMTER]-[TSLY]-[ILD]-[EHVL]-[NDE]-[VCA]-[KLAI]-[AMRELT]-[KSERAD]-[IAQN]-[QL](0,1)-[DKQGRN]
BB11027	[MISR](0,1)-[DPMLE]-[QDAR]-[AGWTR]-[IEVLCA]-[SKRM]-[FLSVQDI]-[EDVLPAF]-[RQKDVN]-[DMATVLF]-[ILAG]-[VKPELD]
BB11028	[NKP](0,1)-[RLPISKQV]-[KDVPQLERS]-[IMVSGHTYL]-[KDASEWLT]-[IVAG]-[KVATSPQW]-[AELIVQ]-[GA]-[FSANEKQD]-[TASRDP]-[HARKVIT]
BB11029	[IGWF]-[PQG]-[GAQE]-[TPI]-[KVP]-[M]-[AQP]-[FGPA]-[GQNY]-[GAVY]-[LTNR]-[AV](0,1)
BB11030	[EAYISNLP](0,1)-[SMAPLDGF]-[VEDYFLHR]-[ETAKHDISQN]-[RADNQEK]-[FTYWVCIAP]-[RADSENVL]-[KDGILRNSYEP]-[VLAIGMTYF]-[VFQIWLMY]-[EGRHKADS]-[ISAVQLTN]
BB11031	[GLWNEITQ](0,1)-[LQGMDVH]-[YEANPHTRQG]-[DHSGIWKEQL]-[VYTWISFDRNA]-[LIWVMT]-[TQNADKR]-[KWSTCNPDF]-[YIHNAQE]-[WKLAYITVN]-[NLEAGQDW]-[RHAYWKF]
BB11032	[DKRQE]-[TGDYMAI]-[VITAQ]-[RDQCHA]-[DQLSK]-[MILPAWK]-[TMLNFY]-[GVKLDC]-[HSENTQ]-[WLTRQE]-[PARVGE]-[SAQGN](0,1)
BB11033	[PDRSK]-[LFMQESAY]-[IAFLVTY]-[NEHITDFA]-[LKSERC]-[ALFID]-[ASYLQGP]-[NEIDAGKQ]-[TEKGDV]-[YRNFKLQ]-[SDTHRPK]-[KEXGFA](0,1)
BB11034	[CSRYL]-[VLPAN]-[NHGPSQ]-[VNML]-[GAQSE]-[CLKVA]-[VIYDEL]-[PLFG]-[KSHEMWG]-[KHGAFDL]-[VALKHSRG]-[MLFVGDE](0,1)
BB11035	[TQKG](0,1)-[NGME]-[AQTPI]-[VAFGR]-[KTGP]-[KALI]-[YLAM]-[SNKT]-[DEGA]-[EAQG]-[EDQ]-[LIRA]
BB11036	[SEHF](0,1)-[ETGWL]-[RLMPIEA]-[LDMTWYG]-[ALIWQDS]-[KMVLD]-[LQKAGTI]-[NAVQHE]-[QEDRP]-[LQMVYIT]-[LVNET]-[RPKNT]
BB11037	[LSVWY]-[LAIY]-[FLAHE]-[HMTQF]-[FPK]-[AKPI]-[SPLT]-[IPLE]-[QIKSD]-[LDNR]-[KSVFL]-[RLDV](0,1)
BB11038	[QLVK]-[GSAENK]-[LVHSIT]-[FLAP]-[SDYWFK]-[KDFILPQ]-[VFATKRL]-[LYQADS]-[ETYNLSQ]-[RAHSLDQ]-[LTDSGWE]-[NEPMQG](0,1)
BB12001	[ENVYTGDQRK]-[NTQLSEKDA]-[IANDFKE]-[LQICPVM]-[NLQESTI]-[LAEGSYKDNI]-[LPVIFTSAE]-[LYAIEQS]-[SQRTND]-[YRISVTLAG]-[TVSIAMD]-[RGWSN]
BB12002	[VILSFY]-[KNRQ]-[TSLN]-[KRD]-[DK]-[QKRVN]-[PRLG]-[QYTCI]-[VFMLI]-[QHG]-[VTA]-[FLRGCT]
BB12003	[C]-[YNEQI]-[KAGDY]-[LAV]-[PE]-[DEKN]-[HNDS]-[VAKE]-[RGKTLPD]-[TIVWLS]-[KISAYWV]-[GVDSYP](0,1)
BB12004	[LFYW]-[VTIRLCA]-[QNEAS]-[VSTEKIQGRAN]-[FLVAIT]-[NKVACQSIT]-[ENAGWPQL]-[HLAQFNVR]-[TDKILSVQ](0,1)-[IFVPRSKQYE]-[SDPAELGNQ]-[PALINTFK](0,1)
BB12005	[IYP](0,1)-[NREKVAIF]-[KEDGAS]-[LIA]-[EKRPLAS]-[KVTLRGP]-[DETPAF]-[LEVAIF]-[ATRFLNI]-[FNIHL]-[NAT]-[LYGAT]
BB12006	[MGRP]-[YG]-[RVI]-[VPIE]-[RYPA]-[LVS]-[STN]-[DTV]-[KQA]-[PTDV]-[HDAK]-[TSL](0,1)
BB12007	[WVASD]-[QFPINSV]-[HQPSDIVA]-[QLVDAS]-[GPFRVL]-[KGDVPY]-[TFEPRYQ](0,1)-[LNGVI]-[FSMKL]-[IWRVFTE]-[SVCGLT]-[RYTDV]
BB12008	[VSGFC]-[AGTLISVY]-[ILTRPGHAM]-[TSAVHLP]-[LYVFTGA]-[SGIMLFC]-[ASTIVE]-[THISALV]-[VSHQNP]-[SNGQP]-[LMQSADT]-[IALG](0,1)
BB12009	[EDC]-[IVRP]-[VLEI]-[EHVNP]-[GSDP]-[NDLK]-[RKSN]-[Q](0,1)-[GK]-[PSRF]-[QHPS]-[AK]
BB12010	[WYCTK](0,1)-[NDKSAF]-[ARQL](0,1)-[DTP]-[KGYI]-[GKEYIA]-[WFY]-[GHDSF]-[QDEN]-[PLGD]-[RKQCE]-[ILFM]
BB12011	[AELFPN](0,1)-[GVQRMNP]-[TSGLFVYI]-[VWIKEQDRT]-[DKFNCTRI]-[STHIGD]-[LITENKFD]-[DETVSFP]-[VQSEFYG]-[ILGMFE]-[PENTIS]-[PGNTS](0,1)
BB12012	[DHKC]-[TVYD]-[KQTA]-[DSYN]-[ILGS]-[NMSK]-[IVD]-[NSYI]-[SHGD]-[IHPL]-[DPLE]-[GRIA]
BB12013	[VLFIA](0,1)-[TLKSHV]-[RSGKT]-[PANQLEG]-[MTLQRKA]-[KQTSIF]-[KIVSYR]-[MHDFRLV]-[FVLHST]-[N](0,1)-[RGPQ]-[EATVSND]
BB12014	[IRATSG]-[YRPKLAS]-[TREGMD]-[RLK]-[EIAL]-[EDHNQK]-[LIVF]-[AMVS]-[QNSKE]-[RAMNKS]-[TLIC]-[NSHG]
BB12015	[TVDMEQNP](0,1)-[FKMLSRVHA]-[PVEIA]-[KGDSAEQ]-[RKINSQA]-[GSKEN]-[QDVCSTK]-[TKAVER]-[CVIA]-[VTKLNSHG]-[VILCM]-[HEDSWR]
BB12016	[M](0,1)-[TDKI]-[PQDKE]-[VIL]-[TLEVG]-[IV]-[KEQI]-[KIVYN]-[KCVR]-[GED]-[RHKG]-[HKRS]
BB12017	[SFGKIYD]-[MLNTPV]-[RVEMFNL]-[YDKER]-[LWIGC]-[VFLA]-[RANEK]-[KEDH]-[RQDIN]-[LGNH]-[PAKWIL]-[KTEAN](0,1)
BB12018	[DKAP]-[RK](0,1)-[IDE]-[PKA]-[EVIR]-[LKF]-[RDPE]-[QALE]-[IV]-[ILDK]-[EKFA]-[QEIF]
BB12019	[LESK](0,1)-[VME]-[IVSEF]-[KLVIH]-[QKVST]-[GNVD]-[FSA]-[AIFY]-[IAG](0,1)-[EDV]-[PRSL]-[SGALN]
BB12020	[R](0,1)-[TIDG]-[IVPE]-[SPLI]-[NGR]-[GN]-[YQG]-[LKAQ]-[SIV]-[SRPD]-[GAV]-[FSNP]

BB12021	[LH]-[EKDGHN]-[Y]-[RNVI]-[HQNAEK]-[DE]-[AT]-[STAN]-[SKDAE]-[VSA]-[QESDA]-[RGKD](0,1)
BB12022	[AQSP]-[LHQY]-[SGTN]-[KYST]-[DIHAL]-[QPASK]-[DSHY]-[FKP](0,1)-[NFRH]-[PSG]-[ANDRE]-[VKAWG]
BB12023	[FLN](0,1)-[SVKE]-[RQPTK]-[VLI]-[PAH]-[PLTIH]-[RNQDV]-[IVTNA]-[D](0,1)-[SHYQN]-[GTEI]-[LMAP]
BB12024	[EP]-[SN](0,1)-[HS](0,1)-[GE]-[RAEQ]-[DIK]-[YCS]-[ITH]-[LIV]-[SA]-[HEAQ]-[VITN]
BB12025	[SHT](0,1)-[SNGQ]-[THN]-[KYTI]-[VTS]-[DQEA]-[KE]-[KSA]-[ILW]-[VSG]-[PLR]-[RA](0,1)
BB12026	[LVRITA]-[KGENSRAWQDP]-[DEAQYSVRGN]-[VIKRAS]-[VAKEQRD]-[QRKSLITEVDAF]-[LKDEQWVRMI]-[SRIVLET]-[LADQKEIS]-[RSEATNLKI]-[REILSGDMVQA]-[VPYSKAREQ](0,1)
BB12027	[LVHADNQK]-[TDQESYK]-[IVLMA]-[RKQ]-[EAKQRSD]-[VLFIY]-[LCAIV]-[GEARTKSI]-[EKGNDARL]-[QAHRFSCKY]-[GNASK]-[AHRN](0,1)
BB12028	[EVHSY](0,1)-[AEPGRTF]-[GKSIE]-[YSERKG]-[SNKGH]-[TRCVYPH]-[SILHTYV]-[YRLAG]-[KQYAWF]-[ECYTRKS]-[DSAVTI]-[KPSTGD]
BB12029	[STMKANDE]-[LRIAV]-[PSKDEAF]-[GAVPT]-[AYLKPSN]-[SPDREY]-[LIVHYT]-[FEVDTYPKL]-[SLDAPGI]-[GPYAVS]-[GTSHKVAR]-[LRITKPV]
BB12030	[EIGH](0,1)-[LGKDQ]-[DQKG]-[CHVL]-[QRKLE]-[QTEA]-[GKICA]-[TIVFG]-[REKQD]-[AEDQ]-[LIV]-[LRAKI]
BB12031	[GALENQFD]-[LEANDK]-[LASETQF]-[ECAFLHN]-[SDVGQKRA]-[PCVGIMENA]-[KDQETNFA]-[AFVIESG]-[LSETNDV]-[EAFDSQNR]-[EIVHKQN]-[AEQKLSPV]
BB12032	[KQEG]-[NDTKH]-[LISNYV]-[C]-[YF]-[KLRT]-[MKNEYR]-[FASIRTW]-[MQRWTH]-[VCRTM]-[ADHFVEI]-[AKPHTV](0,1)
BB12033	[VMQLT]-[QATPR]-[YHIEVDA]-[LEPGIQ]-[SVQETI]-[IVTLFKR]-[LQAYDR]-[STDAV]-[ASKQICL]-[VKILSD]-[GPSIKR]-[KN](0,1)
BB12034	[LD](0,1)-[GKQ]-[DSY]-[TND]-[NDS]-[IVLA]-[AND]-[YDV]-[LF]-[HNER]-[EMA]-[NMKL]
BB12036	[APNSK]-[DSHKQ]-[IV]-[RLYDECQ]-[EANTD]-[LAFH]-[LFI]-[GSQKDR]-[GKTSD]-[LAKIMS]-[ATLGK]-[RSPEF](0,1)
BB12037	[STQFDRNGP](0,1)-[KADETVWLQ]-[ILVFDNTGA]-[SPTFIELND]-[REVIKPLHM]-[EDSTALP]-[DKNEASPQ]-[FYIHGKNV]-[MLYVIRFE]-[SEKDQTLY]-[GILNRAMFDW]-[VMLKYS]
BB12038	[KQTLAER]-[YVDFR]-[DSVGCKN]-[PAKGHRDV]-[SKGTN]-[LSH]-[KTGPSQD]-[PRAQIGK]-[LFIDA]-[SEFKLN]-[VFLGM]-[VLQT](0,1)
BB12039	[GDE](0,1)-[DGE]-[EGPDTHKY]-[TNIHQV]-[YFSIEAV]-[VEYIKT]-[IPKVAF]-[EDSHNRK]-[PAIE]-[SADE]-[LKSVADE]-[C]
BB12040	[NAKE]-[GLTIM]-[ILA]-[FN]-[HADK]-[LIG]-[ADYH]-[IANED]-[DNLA]-[DRLQ]-[NSKQ]-[SEN](0,1)
BB12041	[DIVL](0,1)-[VPAIDX]-[PEQSV]-[ASVPD]-[DVPCM]-[GTPAEK]-[AMDGVH]-[KSVM]-[IPVK]-[DKVIAT]-[FQEDKA]-[IFVHPA]
BB12042	[FY]-[GVP]-[RANS]-[NPLD]-[FSRI]-[FYES]-[TNK]-[RIP]-[YLPE]-[DQCI]-[YAEN]-[EDI](0,1)
BB12044	[LHAIVSKE]-[DTVGLRAI]-[LVHQTISK]-[KIQALESNV]-[DKALGHSIQ]-[KGVQDFTMEA]-[DVGQLPS]-[GLTDEPIK]-[IENQVTHA]-[SELANDPM]-[PITAKG]-[DLETPQWV]
BB20001	[EDARKTM]-[RKADIQ]-[EATNL]-[MVLIY]-[KQRDSP]-[TESNADG]-[YFW]-[IEKVNSR]-[PARKNV]-[PNKEQGR]-[KGYMLQND]-[GQEAPKN](0,1)
BB20002	[LFMVI]-[RKTQECHW]-[CKPQAER]-[DGSRN]-[EDATQP]-[ERKHIQLTVCA]-[YLFIVMK]-[YQHEFTRALMD]-[LINV]-[LIVSTFW]-[DNESGRHK]-[STNDEYKVPRG](0,1)
BB20020	[NQSDKTVL](0,1)-[ALTNDGYSPEV]-[NPREQTK]-[STEAIN]-[ALQKIVTE]-[AP]-[SRAGIKE]-[ASTF]-[LFIM]-[MIL]-[VIR]-[LCIVAF]
BB30006	[NKRDSYLACQPVT]-[NVIPDRQLE]-[KLYQTISF]-[LHTSQY]-[IYFCLAS]-[KRIHLV]-[IVLEQ]-[FDNLSQKVP]-[HKTRQNSEDY]-[RDATESPN]-[DKSGQEN]-[TDANHPKGQE](0,1)
BB30017	[DAEGQ](0,1)-[PNDKS]-[GYSLETVF]-[AYHGSN]-[LFEKGAPIV]-[VPKIMLNS]-[STIVFLA]-[YIVERKWFQS]-[SQYMVFPG]-[GQPDT]-[AGVSNT]-[AGVLYFS]
BB30027	[GKLISRTD]-[APEY]-[EDASQGTNP]-[EAYSNDTV]-[LEI]-[YFVLIKMSDT]-[KDAQPES]-[A](0,1)-[KAFIQY]-[MLKIRV]-[KISLTNDAEP]-[GNAYDH]
BB40003	[NSLTDEAK]-[AGILKF]-[QLCFVSM]-[LMSNIED]-[AEGSNI]-[PIVAT]-[YEMLQGSF]-[QITKSDE]-[RNKILHGTSE]-[TIVAL]-[GNWA]-[N](0,1)
BB40005	[SQHD](0,1)-[RMENDGQSY]-[GKDSRVEN]-[KRGPMQHA]-[DRKSENQA]-[LRQKTFSH]-[GLYVREQKS]-[SAEQKGTMCYV]-[MITLKS]-[DARTNS]-[VYIPMAL]-[NGEQSTD]
BB40006	[RSIAKQFGEP](0,1)-[AVPNDE]-[FWAYEDLH]-[VQXLIRFNKY]-[EQRDANISHL]-[MAESLQYGNK]-[ATRLNIPMY]-[RWVHGIFYP]-[EHAIKGSQTD]-[AWQYRKLTEF]-[YLIFMV]-[RIKLEN]
BB40007	[PAFSCI](0,1)-[LPVA]-[NLPRT]-[FALRQG]-[AFTSVY]-[ASEQD](0,1)-[SAMTPLN]-[ETGQ]-[SGLV]-[STHD]-[PTNQGV]-[EQYISDLT]
BB40008	[VENR](0,1)-[DYTKVMLRN]-[SFVAPRM]-[RTQAMSID]-[KTEQWLF]-[ICRYNTMAWFQ]-[DYTH]-[QHRKTGED]-[ENKSLYTGD]-[GNETSY]-[KTIEMLYN]-[IYFVLM]
BB40009	[EDVPGQHKIL](0,1)-[PWIHVGMN]-[PSYAQLFNHG]-[APSEDL]-[EMTAVPWHGIL]-[FLMYAWIEKH]-[ASLKCDEFYV]-[RKEAWVSTMGD]-[LHMESTRKF]-[RDQLNGS]-[ANEDKLP]-[NTSEGADKLIY]
BB40010	[RKDSGN]-[SQ](0,1)-[GK]-[PRKAF]-[QATSPY]-[AKS]-[AGTV]-[NKVDIE]-[VIDNEL]-[TQVLRY]-[KSITRL]-[ELGAKST]
BB40014	[LFTN](0,1)-[QPRDNS]-[AGFSNV]-[TYSKAD]-[LCITN]-[IMWVPYNF]-[TQARSYL]-[NDAT]-[EKATGSD]-[DGYVT]-[EKNQRA]-[NVQPETS]
BB40018	[THNQ](0,1)-[RHKQGTSN]-[HKRCSG]-[IAVLQKH]-[KIRFE]-[TGARKI]-[RTNP]-[MVILYF]-[LMCIV]-[VTLI]-[AGLFP]-[DPEYN](0,1)
BB40019	[PMRKAF](0,1)-[LGVIPESHY]-[GLARW]-[PSTK]-[MVDKPTS]-[ATYPMFS]-[ADVFGNE]-[SIPNATW]-[SGRPAFD]-[WKS]-[LI](0,1)-[CPIDYKTQ]
BB40022	[ATSMQKHD]-[DNGSER]-[QPLREDNSG]-[PNESKTRGFA]-[GKVPLIR]-[DRLTYIS](0,1)-[FWYVRIS]-[EQDRKG]-[YLVE]-[C]-[DNMSER]-[LIV](0,1)
BB40025	[SEAIV](0,1)-[ADPTSYER]-[TAREPDSNQ]-[EGSLAD]-[AQVE]-[AIESKNYQLM]-[KTRSNE]-[YHNVIGTA]-[VLF]-[FC]-[KTLAP]-[DGNSE]
BB40032	[LGPAQ]-[ALIPV]-[VASQG]-[RNPKQH]-[QCR](0,1)-[QVIAR]-[APEI]-[PTK]-[LVIP]-[IVPKTH]-[IVWN]-[PEKAI]
BB40033	[AGIERKNLQY]-[LITV]-[IVLEMQFY]-[NRDKAVEQS]-[FVEIWLK]-[MTAWVLFYI]-[LMREKDN]-[DNKESALR]-[LKFEIQMRA]-[HLMNYQFIT]-[TIAEDKQ]-[KHRSEIFALN]
BB40043	[TVNPRSIED](0,1)-[GEALCIMT]-[KRNDVETSLF]-[VILGC]-[TLIVEAKS]-[KSMDVRI]-[VIKRQTE]-[QHKETASV]-[NHKVSILR]-[WVMIELAP]-[TNKSAPE](0,1)-[HKSDIL](0,1)
BB40045	[DIAVLNY](0,1)-[VPALIDX]-[PEQSAV]-[ASVPD]-[DVPACM]-[GTPAEDKIM]-[AMDGVH]-[KSPVMTL]-[IPKVL]-[DKVAITM]-[FQEPDKAGL]-[IFVAHPTS]
BB40048	[PEIQDV](0,1)-[SQPKTRL]-[LSAKDNCE]-[KLAMIV]-[DAKGSLTN]-[ANDQTMGK]-[VALNTKDMC]-[AEPNFIK]-[GAPISVLK](0,1)-[KETQYDG]-[AKSDLITQH]-[FGRAKI]
BB50002	[EDVLRN](0,1)-[KFEAPSVD]-[AETSIHPYK](0,1)-[DRLEKI]-[LEGNAFTK]-[LVMQDRK]-[PKSELQDA]-[HPEANIVYFL]-[LEVFYQRX]-[QLNKRMTEAS]-[FYGEQTMKI]-[LSHKITVAFEC]
BB50004	[D](0,1)-[FCLYV]-[DEI]-[KEARI]-[ASDNK]-[GKFLMN]-[ALSIVN]-[EKPDG]-[IFLED]-[VACY]-[PRGKQNE]-[KSEARN]
BB50005	[KSTI]-[N](0,1)-[VARIT]-[ENASGDTF]-[RKCQDTI]-[LIVQ]-[KRAVYL]-[QEKRSTGN]-[VLKIMFE]-[LVI]-[LEVAQNDR]-[HGDNEVK]
BB50010	[AGIKNLQEYD]-[LITV]-[IVLEQFY]-[NRDEQSAK]-[FVEIWLKQ]-[MTAWLFYI]-[LMRKDNT]-[DNKLRSEG]-[LKFEQMRAIV]-[HLMQFITY]-[TIAKDE]-[KHRFEALNQ]
BB50013	[SDT]-[TNLS]-[NDFSE]-[VIL]-[NKARQSD]-[DYFVKI]-[LI]-[RNPKLQE]-[LMGKDQENA]-[IMENHYLK]-[QGTSNHRY]-[HNG](0,1)
BB50016	[TMIEGDL](0,1)-[KPQLEIGRMFW]-[ILRSPTMQVDE](0,1)-[AKTRIVPENF]-[TVNGEHQLFIS]-[PIRLNTHE]-[KGQHRPADIV]-[NPDHSKRY]-[WYLFV]-[LRTKEVHQMP]-[KVIPLRAD]-[GLIVEAQT]
BBS11001	[REQ]-[EKNL]-[MYL]-[KPA]-[TNRG]-[YFW]-[IKRS]-[PYEA]-[PRD]-[KPHD]-[GRPN]-[ERDY]
BBS11002	[TKHWLA]-[KELARP]-[G]-[EDPQ]-[KCIAVL]-[LMKVI]-[RTLYE]-[VIL]-[LINWF]-[GHKPS]-[YRGFK]-[NESQL](0,1)
BBS11003	[DERK]-[TYLA]-[TYFC]-[IAE]-[KRDT]-[EAV]-[LAVI]-[PLNG]-[DRSL]-[WYGK]-[ARSD]-[EYP](0,1)
BBS11004	[RQVY]-[SINM]-[YFM]-[INK]-[EN](0,1)-[HQVL]-[KGLW]-[MLDV]-[QKA]-[GSTV]-[NYSH]-[PRT]
BBS11005	[TPSE](0,1)-[KSQAGDPERV]-[NAEPDGQ]-[YWHLTFGIR]-[IQLVYF]-[AQDESVKH]-[EAQRYNM]-[NTILVAXRDW]-[HWVRANDTI]-[KHAIDVQEG]-[RWKYIMAFNQ]-[LIEGTYVA]
BBS11006	[LETAVY](0,1)-[KAIDEQL]-[VEFNTR]-[FITDVYR]-[NDFAGHS]-[PTEVQG]-[PMGSEVA]-[RILGVTN](0,1)-[NPAYQ]-[QFGNLIHV]-[LDTPQR]-[PLSERQ]
BBS11007	[RAFGDTL]-[TDGPHKVR]-[VFEIDSL]-[IPDLEA]-[LDMKGHEW]-[PLYNGV]-[QENLIHSPY]-[GSPLF]-[YDSLQETK]-[ASNLIED]-[DFLNR]-[DHSTARKGQ]

BBS11008	[EQK]-[LD]-[IVTH]-[NEA]-[HFL]-[YHG]-[RQSK]-[NLFY]-[ENAY]-[SRP]-[LGKR]-[ARG](0,1)
BBS11009	[GDV](0,1)-[NGT]-[VART]-[EVFS]-[FLI]-[QEC]-[CAI]-[PNA]-[DSQA]-[DGN]-[VER]-[YTS]
BBS11010	[N]-[RHA]-[DQTH]-[EAF]-[L]-[LRI]-[SAGK]-[VELK]-[LYA]-[PGF]-[ADVE]-[GWE](0,1)
BBS11011	[SQYD]-[VTHR]-[RSFCI]-[WACSG]-[VTGLH]-[MGA]-[NVSFL]-[ALN]-[LIVM]-[GNVE]-[VKSAG]-[K](0,1)
BBS11012	[GAR](0,1)-[PVRT]-[WTNA]-[NDFP]-[KSGQ]-[DPYK]-[EDTN]-[IL]-[SARK]-[TS]-[TRVA]-[DTNS]
BBS11013	[KIRST]-[NKVR]-[HLYMG]-[WSL]-[NIKT]-[SRDK]-[TRPFL]-[MLVGK]-[RLSA]-[RAYV]-[KAGRP]-[VGT]
BBS11014	[VYSL](0,1)-[RNSYH]-[FLAVS]-[SKAL]-[HVRTFA]-[AFSTM]-[RFVS]-[DEKTL]-[PNVYQ]-[ISCVD]-[FLWG]-[GNDLT](0,1)
BBS11015	[TQ](0,1)-[LAG]-[KHGE]-[KAQ]-[VEWG]-[LTV]-[IVT]-[CVME]-[SGC]-[TNIA]-[LRPY]-[LRSV]
BBS11016	[IADKMPHT]-[VGLAPR]-[DRNEHSP]-[RNSGFYL]-[LGVIDYR]-[TPRKAVEM]-[THYES]-[GMKAQR]-[VITQM](0,1)-[AMGYRDTH]-[ANEDLK]-[LFKGDPH]
BBS11017	[DHIP]-[LSTA]-[L]-[LTKD]-[TS]-[FSRA]-[MYI]-[ELGT]-[ADXT]-[VLI]-[NMYF]-[KSGY]
BBS11018	[MYKDTVAHRI]-[YNDISEQKA]-[GMWIYNLTKSD]-[LIGVTW]-[KETSRHN]-[ASGQHLFKR]-[MVWYDRGAT]-[LAGKMYIDV]-[ESFDNLTVI]-[GDMVYTAS]-[SCPFVNRK]-[EVSHGDP](0,1)
BBS11019	[KLTM](0,1)-[GSYDVKRTF]-[RKCTMEP]-[LDNSKEGP]-[RSDHYTLAEG]-[AVYFNLTRI]-[EPVRNL]-[MKIAQE]-[LATISEQ]-[RVLMINAS]-[DAESQLW]-[MDIAWLGQ]
BBS11020	[EWVFMLI]-[ILWYQF]-[TNEVLFSA]-[EYNADLKQ]-[TIQYLMAF]-[LAVFGNS]-[QTMDSAG]-[DELYH](0,1)-[ITKYLA](0,1)-[FLRVQCKP]-[NHLVPRIM]-[DKQASFI](0,1)
BBS11021	[VIW]-[HEV]-[FTL]-[VID]-[AKIN]-[GP]-[VMAG]-[LAGD]-[GPAD]-[EDPV]-[AG]-[AT](0,1)
BBS11022	[IHFR]-[SRAE]-[SANR]-[RDE]-[VFI]-[KIL]-[SAVK]-[KGRL]-[RLI]-[IKA]-[QK]-[LR](0,1)
BBS11023	[QAVG]-[EAKLQ]-[IAQEK]-[AFILVG]-[AKVF]-[KRAEN]-[NYATKM]-[FGRIDN]-[YFAGKL]-[RTVEYPG]-[PILYW]-[RKPTGN]
BBS11024	[CALT]-[GCH]-[TGS]-[VSGC]-[QHGS]-[LYRA]-[DLY]-[FGNL]-[SEGD]-[LNW](0,1)-[PFLQ]-[SAL]
BBS11025	[TK](0,1)-[RWEA]-[IFL]-[KDLE]-[KTDR]-[IM]-[VLK]-[QGAD]-[KFRT]-[KAL]-[LIER]-[ASRI]
BBS11026	[DMTER]-[TSLY]-[ILD]-[EHVL]-[NDE]-[VCA]-[KLAI]-[AMRELT]-[KSERAD]-[IAQN]-[QL](0,1)-[DKQGRN]
BBS11027	[MISR](0,1)-[DPMLE]-[QDAR]-[AGWTR]-[IEVLCA]-[SKRM]-[FLSVQDI]-[EDVLPAF]-[RQKDVN]-[DMATVLF]-[ILAG]-[VKPELD]
BBS11028	[NKP](0,1)-[RLPISKQV]-[KDVPQLERS]-[IMVSGHTYL]-[KDASEWLT]-[IVAG]-[KVATSPQW]-[AELIVQ]-[GA]-[FSANEKQD]-[TASRDP]-[HARKVIT]
BBS11029	[IGWF]-[PQG]-[GAQE]-[TPI]-[KVP]-[M]-[AQP]-[FGPA]-[GQNY]-[GAVY]-[LTNR]-[AV](0,1)
BBS11030	[EAYISNLP](0,1)-[SMAPLDGF]-[VEDYFLHR]-[ETAKHDISQN]-[RADNQEK]-[FTYWVCIAP]-[RADSENVL]-[KDGILRNSYEP]-[VLAIGMTYF]-[VFQIWLMY]-[EGRHKADS]-[ISAVQLTN]
BBS11031	[GLWNEITQ](0,1)-[LQGMDVH]-[YEANPHTRQG]-[DHSGIWKEQL]-[VYTWISFDRNA]-[LIWVMT]-[TQNADKR]-[KWSTCNPDF]-[YIHNAQE]-[WKLAYITVN]-[NLEAGQDW]-[RHAYWKF]
BBS11032	[DKRQE]-[TGDYMAI]-[VITAQ]-[RDQCHA]-[DQLSK]-[MILPAWK]-[TMLNFY]-[GVKLDC]-[HSENTQ]-[WLTRQE]-[PARVGE]-[SAQGN](0,1)
BBS11033	[PDRSK]-[LFMQESAY]-[IAFLVTY]-[NEHITDFA]-[LKSERC]-[ALFID]-[ASYLQGP]-[NEIDAGKQ]-[TEKGDV]-[YRNFKLQ]-[SDTHRPK]-[KEXGFA](0,1)
BBS11034	[CSRYL]-[VLPAN]-[NHGPSQ]-[VNML]-[GAQSE]-[CLKVA]-[VIYDEL]-[PLFG]-[KSHEMWG]-[KHGAFDL]-[VALKHSRG]-[MLFVGDE](0,1)
BBS11035	[TQKG](0,1)-[NGME]-[AQTPI]-[VAFGR]-[KTGP]-[KALI]-[YLAM]-[SNKT]-[DEGA]-[EAQG]-[EDQ]-[LIRA]
BBS11036	[KSDTRG]-[VALI]-[YQLKEDR]-[AVCT]-[RYTFVI]-[SRPEVIA]-[VWGEF]-[YQLARDN]-[DITWGLV]-[SPGHT]-[RMFLDT]-[DYHSRIA](0,1)
BBS11037	[LSVWY]-[LAIY]-[FLAHE]-[HMTQF]-[FPK]-[AKPI]-[SPLT]-[IPLE]-[QIKSD]-[LDNR]-[KSVFL]-[RLDV](0,1)
BBS11038	[NE](0,1)-[NKVQRY]-[LNHRA]-[ARDNLG]-[FPSG]-[DESRCA]-[AEKQL]-[TRGE]-[PIARL]-[DTMGL]-[YFALQ]-[DEARGQHL]
BBS12001	[ENVYTGDQRK]-[NTQLSEKDA]-[IANDFKE]-[LQICPVM]-[NLQESTI]-[LAEGSYKDNI]-[LPVIFTSAE]-[LYAIEQS]-[SQRTND]-[YRISVTLAG]-[TVSIAMD]-[RGWSN]
BBS12002	[VILSFY]-[KNRQ]-[TSLN]-[KRD]-[DK]-[QKRVN]-[PRLG]-[QYTCI]-[VFMLI]-[QHG]-[VTA]-[FLRGCT]
BBS12003	[C]-[YNEQI]-[KAGDY]-[LAV]-[PE]-[DEKN]-[HNDS]-[VAKE]-[RGKTLPD]-[TIVWLS]-[KISAYWV]-[GVDSYP](0,1)
BBS12004	[LFYW]-[VTIRLCA]-[QNEAS]-[VSTEKIQGRAN]-[FLVAIT]-[NKVACQSIT]-[ENAGWPQL]-[HLAQFNVR]-[TDKILSVQ](0,1)-[IFVPRSKQYE]-[SDPAELGNQ]-[PALINTFK](0,1)
BBS12005	[IYP](0,1)-[NREKVAIF]-[KEDGAS]-[LIA]-[EKRPLAS]-[KVTLRGP]-[DETPAF]-[LEVAIF]-[ATRFLNI]-[FNIHL]-[NAT]-[LYGAT]
BBS12006	[MGRP]-[YG]-[RVI]-[VPIE]-[RYPA]-[LVS]-[STN]-[DTV]-[KQA]-[PTDV]-[HDAK]-[TSL](0,1)
BBS12007	[WVASD]-[QFPINSV]-[HQPSDIVA]-[QLVDAS]-[GPFRVL]-[KGDVPY]-[TFEPRYQ](0,1)-[LNGVI]-[FSMKL]-[IWRVFTE]-[SVCGLT]-[RYTDV]
BBS12008	[VSGFC]-[AGTLISVY]-[ILTRPGHAM]-[TSAVHLP]-[LYVFTGA]-[SGIMLFC]-[ASTIVE]-[THISALV]-[VSHQNP]-[SNGQP]-[LMQSADT]-[IALG](0,1)
BBS12009	[EDC]-[IVRP]-[VLEI]-[EHVNP]-[GSDP]-[NDLK]-[RKSN]-[Q](0,1)-[GK]-[PSRF]-[QHPS]-[AK]
BBS12010	[GHDSF]-[QDEN]-[PLGD]-[RKQCE]-[ILFM]-[QVT]-[PTQR]-[FYDSW]-[QGSPE]-[NPTFD]-[LIA]-[TEHDSK]
BBS12011	[AELFPN](0,1)-[GVQRMNP]-[TSGLFVYI]-[VWIKEQDRT]-[DKFNCTRI]-[STHIGD]-[LITENKFD]-[DETVSFP]-[VQSEFYG]-[ILGMFE]-[PENTIS]-[PGNTS](0,1)
BBS12012	[DHKC]-[TVYD]-[KQTA]-[DSYN]-[ILGS]-[NMSK]-[IVD]-[NSYI]-[SHGD]-[IHPL]-[DPLE]-[GRIA]
BBS12013	[VLFIA](0,1)-[TLKSHV]-[RSGKT]-[PANQLEG]-[MTLQRKA]-[KQTSIF]-[KIVSYR]-[MHDFRLV]-[FVLHST]-[N](0,1)-[RGPQ]-[EATVSND]
BBS12014	[IRATSG]-[YRPKLAS]-[TREGMD]-[RLK]-[EIAL]-[EDHNQK]-[LIVF]-[AMVS]-[QNSKE]-[RAMNKS]-[TLIC]-[NSHG]
BBS12015	[GSLVIQA](0,1)-[HVDPGAR]-[PGFYDL]-[GNAP]-[IPHKG]-[IEPM]-[PGLA]-[PA]-[HNG]-[AEKQS]-[TDEK]-[LV]
BBS12016	[M](0,1)-[TDKI]-[PQDKE]-[VIL]-[TLEVG]-[IV]-[KEQI]-[KIVYN]-[KCVR]-[GED]-[RHKG]-[HKRS]
BBS12017	[SFGKIYD]-[MLNTPV]-[RVEMFNL]-[YDKER]-[LWIGC]-[VFLA]-[RANEK]-[KEDH]-[RQDIN]-[LGNH]-[PAKWIL]-[KTEAN](0,1)
BBS12018	[DKAP]-[RK](0,1)-[IDE]-[PKA]-[EVIR]-[LKF]-[RDPE]-[QALE]-[IV]-[ILDK]-[EKFA]-[QEIF]
BBS12019	[LESK](0,1)-[VME]-[IVSEF]-[KLVIH]-[QKVST]-[GNVD]-[FSA]-[AIFY]-[IAG](0,1)-[EDV]-[PRSL]-[SGALN]
BBS12020	[R](0,1)-[TIDG]-[IVPE]-[SPLI]-[NGR]-[GN]-[YQG]-[LKAQ]-[SIV]-[SRPD]-[GAV]-[FSNP]
BBS12021	[LH]-[EKDGHN]-[Y]-[RNVI]-[HQNAEK]-[DE]-[AT]-[STAN]-[SKDAE]-[VSA]-[QESDA]-[RGKD](0,1)
BBS12022	[AQSP]-[LHQY]-[SGTN]-[KYST]-[DIHAL]-[QPASK]-[DSHY]-[FKP](0,1)-[NFRH]-[PSG]-[ANDRE]-[VKAWG]
BBS12023	[FLN](0,1)-[SVKE]-[RQPTK]-[VLI]-[PAH]-[PLTIH]-[RNQDV]-[IVTNA]-[D](0,1)-[SHYQN]-[GTEI]-[LMAP]
BBS12024	[EP]-[SN](0,1)-[HS](0,1)-[GE]-[RAEQ]-[DIK]-[YCS]-[ITH]-[LIV]-[SA]-[HEAQ]-[VITN]
BBS12025	[TS](0,1)-[VKE]-[SALD]-[AKGL]-[AGQR]-[KQPN]-[TPKV]-[TRS]-[PES]-[P]-[SQK]-[V]
BBS12026	[LVRITA]-[KGENSRAWQDP]-[DEAQYSVRGN]-[VIKRAS]-[VAKEQRD]-[QRKSLITEVDAF]-[LKDEQWVRMI]-[SRIVLET]-[LADQKEIS]-[RSEATNLKI]-[REILSGDMVQA]-[VPYSKAREQ](0,1)
BBS12027	[LVHADNQK]-[TDQESYK]-[IVLMA]-[RKQ]-[EAKQRSD]-[VLFIY]-[LCAIV]-[GEARTKSI]-[EKGNDARL]-[QAHRFSCKY]-[GNASK]-[AHRN](0,1)

BBS12028	[EVHSY](0,1)-[AEPGRTF]-[GKSIE]-[YSERKG]-[SNKGH]-[TRCVYPH]-[SILHTYV]-[YRLAG]-[KQYAWF]-[ECYTRKS]-[DSAVTI]-[KPSTGD]
BBS12029	[STMKANDE]-[LRIAV]-[PSKDEAF]-[GAVPT]-[AYLKPSN]-[SPDREY]-[LIVHYT]-[FEVDTYPKL]-[SLDAPGI]-[GPYAVS]-[GTSHKVAR]-[LRITKPV]
BBS12030	[EIGH](0,1)-[LGKDQ]-[DQKG]-[CHVL]-[QRKLE]-[QTEA]-[GKICA]-[TIVFG]-[REKQD]-[AEDQ]-[LIV]-[LRAKI]
BBS12031	[GALENQFD]-[LEANDK]-[LASETQF]-[ECAFLHN]-[SDVGQKRA]-[PCVGIMENA]-[KDQETNFA]-[AFVIESG]-[LSETNDV]-[EAFDSQNR]-[EIVHKQN]-[AEQKLSPV]
BBS12032	[KQEG]-[NDTKH]-[LISNYV]-[C]-[YF]-[KLRT]-[MKNEYR]-[FASIRTW]-[MQRWTH]-[VCRTM]-[ADHFVEI]-[AKPHTV](0,1)
BBS12033	[VMQLT]-[QATPR]-[YHIEVDA]-[LEPGIQ]-[SVQETI]-[IVTLFKR]-[LQAYDR]-[STDAV]-[ASKQICL]-[VKILSD]-[GPSIKR]-[KN](0,1)
BBS12034	[LD](0,1)-[GKQ]-[DSY]-[TND]-[NDS]-[IVLA]-[AND]-[YDV]-[LF]-[HNER]-[EMA]-[NMKL]
BBS12036	[GDS](0,1)-[YIVK]-[LHIV]-[RSEVK]-[NATKIR]-[IFV]-[NDSQPR]-[GAC]-[ESTDN]-[QARG](0,1)-[DTES]-[MPTV]
BBS12037	[STQFDRNGP](0,1)-[KADETVWLQ]-[ILVFDNTGA]-[SPTFIELND]-[REVIKPLHM]-[EDSTALP]-[DKNEASPQ]-[FYIHGKNV]-[MLYVIRFE]-[SEKDQTLY]-[GILNRAMFDW]-[VMLKYS]
BBS12038	[KQTLAER]-[YVDFR]-[DSVGCKN]-[PAKGHRDV]-[SKGTN]-[LSH]-[KTGPSQD]-[PRAQIGK]-[LFIDA]-[SEFKLN]-[VFLGM]-[VLQT](0,1)
BBS12039	[GDE](0,1)-[DGE]-[EGPDTHKY]-[TNIHQV]-[YFSIEAV]-[VEYIKT]-[IPKVAF]-[EDSHNRK]-[PAIE]-[SADE]-[LKSVADE]-[C]
BBS12040	[NAKE]-[GLTIM]-[ILA]-[FN]-[HADK]-[LIG]-[ADYH]-[IANED]-[DNLA]-[DRLQ]-[NSKQ]-[SEN](0,1)
BBS12041	[VPAIDX]-[PEQSV]-[ASVPD]-[DVPCM]-[GTPAEK]-[AMDGVH]-[KSVM]-[IPVK]-[DKVIAT]-[FQEDKA]-[IFVHPA]-[AEKLF]
BBS12042	[FY]-[GVP]-[RANS]-[NPLD]-[FSRI]-[FYES]-[TNK]-[RIP]-[YLPE]-[DQCI]-[YAEN]-[EDI](0,1)
BBS12044	[LHAIVSKE]-[DTVGLRAI]-[LVHQTISK]-[KIQALESNV]-[DKALGHSIQ]-[KGVQDFTMEA]-[DVGQLPS]-[GLTDEPIK]-[IENQVTHA]-[SELANDPM]-[PITAKG]-[DLETPQWV]
BBS20001	[VIFAWKM](0,1)-[NKRISAQE]-[FVINS]-[STAG]-[EQAD]-[FVIL]-[SALTGN]-[KRSQ]-[KRDMEATVI]-[CGILA]-[SGA]-[EAKDR]
BBS20002	[LFMVI]-[RKTQECHW]-[CKPQAER]-[DGSRN]-[EDATQP]-[ERKHIQLTVCA]-[YLFIVMK]-[YQHEFTRALMD]-[LINV]-[LIVSTFW]-[DNESGRHK]-[STNDEYKVPRG](0,1)
BBS20020	[NQSDKTVL](0,1)-[ALTNDGYSPEV]-[NPREQTK]-[STEAIN]-[ALQKIVTE]-[AP]-[SRAGIKE]-[ASTF]-[LFIM]-[MIL]-[VIR]-[LCIVAF]
BBS30006	[NKRDSYLACQPVT]-[NVIPDRQLE]-[KLYQTISF]-[LHTSQY]-[IYFCLAS]-[KRIHLV]-[IVLEQ]-[FDNLSQKVP]-[HKTRQNSEDY]-[RDATESPN]-[DKSGQEN]-[TDANHPKGQE](0,1)
BBS30017	[DAEGQ](0,1)-[PNDKS]-[GYSLETVF]-[AYHGSN]-[LFEKGAPIV]-[VPKIMLNS]-[STIVFLA]-[YIVERKWFQS]-[SQYMVFPG]-[GQPDT]-[AGVSNT]-[AGVLYFS]
BBS30027	[GKLISRTD]-[APEY]-[EDASQGTNP]-[EAYSNDTV]-[LEI]-[YFVLIKMSDT]-[KDAQPES]-[A](0,1)-[KAFIQY]-[MLKIRV]-[KISLTNDAEP]-[GNAYDH]
BBS50002	[EDVLRN](0,1)-[KFEAPSVD]-[AETSIHPYK](0,1)-[DRLEKI]-[LEGNAFTK]-[LVMQDRK]-[PKSELQDA]-[HPEANIVYFL]-[LEVFYQRX]-[QLNKRMTEAS]-[FYGEQTMKI]-[LSHKITVAFEC]
BBS50005	[KSTI]-[N](0,1)-[VARIT]-[ENASGDTF]-[RKCQDTI]-[LIVQ]-[KRAVYL]-[QEKRSTGN]-[VLKIMFE]-[LVI]-[LEVAQNDR]-[HGDNEVK]
BBS50010	[AGIKNLQEYD]-[LITV]-[IVLEQFY]-[NRDEQSAK]-[FVEIWLKQ]-[MTAWLFYI]-[LMRKDNT]-[DNKLRSEG]-[LKFEQMRAIV]-[HLMQFITY]-[TIAKDE]-[KHRFEALNQ]
BBS50013	[SDT]-[TNLS]-[NDFSE]-[VIL]-[NKARQSD]-[DYFVKI]-[LI]-[RNPKLQE]-[LMGKDQENA]-[IMENHYLK]-[QGTSNHRY]-[HNG](0,1)
BBS50016	[TMIEGDL](0,1)-[KPQLEIGRMFW]-[ILRSPTMQVDE](0,1)-[AKTRIVPENF]-[TVNGEHQLFIS]-[PIRLNTHE]-[KGQHRPADIV]-[NPDHSKRY]-[WYLFV]-[LRTKEVHQMP]-[KVIPLRAD]-[GLIVEAQT]

Table B.2: Least conserved (LC) constraints list for the working database used with RE-MuSiC.

Appendix C

Q score and TC score for the working database

This study presents a benchmarking analysis on the programs: CSA-X (CSA-PC and CSA-TCOF), RE-MuSiC, T-Coffee, and ProbCons in terms of Q score and TC score. The complete list of scores for each individual datasets in the working database is as follows:

Datasets	CSA-TCOF				CSA-PC				RE-MuSiC				T-Coffee		ProbCons	
	MC		LC		MC		LC		MC		LC					
	Q	TC	Q	TC	Q	TC	Q	TC	Q	TC	Q	TC	Q	TC	Q	TC
BB11001	0.965	0.93	0.968	0.947	1	1	1	1	0.991	0.982	0.795	0.754	0.965	0.93	1	1
BB11002	0.7	0.353	0.838	0.353	0.7	0.353	0.838	0.353	0.75	0	0.7	0.353	0.611	0	0.611	0
BB11003	0.675	0.502	0.748	0.607	0.714	0.547	0.762	0.632	0.527	0.211	0.634	0.385	0.641	0.494	0.716	0.551
BB11004	0.665	0.473	0.72	0.56	0.678	0.5	0.726	0.576	0.253	0.114	0.38	0.19	0.706	0.554	0.672	0.5
BB11005	0.534	0.216	0.707	0.352	0.52	0.216	0.55	0.227	0.345	0.148	0.272	0.0682	0.568	0.239	0.491	0.205
BB11006	0.584	0.375	0.581	0.396	0.566	0.375	0.671	0.417	0.262	0	0.37	0.146	0.55	0.354	0.567	0.354
BB11007	0.812	0.606	0.874	0.704	0.843	0.676	0.866	0.69	0.707	0.401	0.554	0.359	0.812	0.599	0.843	0.676
BB11008	0.765	0.673	0.782	0.755	0.745	0.653	0.908	0.898	0.531	0.347	0.646	0.592	0.748	0.673	0.765	0.694
BB11009	0.803	0.767	0.764	0.7	0.758	0.7	0.767	0.683	0.631	0.483	0.525	0.25	0.278	0	0.75	0.683
BB11010	0.759	0.615	0.776	0.641	0.784	0.654	0.774	0.641	0.256	0	0.395	0.282	0.784	0.654	0.774	0.641
BB11011	0.52	0.391	0.511	0.37	0.641	0.391	0.535	0.109	0.309	0.196	0.37	0.239	0.509	0.391	0.578	0.217
BB11012	0.949	0.92	0.949	0.92	0.951	0.924	0.95	0.924	0.819	0.693	0.734	0.574	0.951	0.924	0.951	0.924
BB11013	0.424	0.345	0.248	0.138	0.638	0.483	0.586	0.31	0.0241	0	0.445	0.345	0.166	0	0.162	0
BB11014	0.881	0.776	0.883	0.783	0.883	0.78	0.88	0.773	0.68	0.563	0.659	0.545	0.884	0.787	0.883	0.78
BB11015	0.745	0.591	0.818	0.702	0.769	0.636	0.831	0.732	0.58	0.343	0.556	0.313	0.719	0.545	0.769	0.636
BB11016	0.654	0.127	0.806	0.545	0.692	0.209	0.855	0.627	0.495	0.0818	0.513	0.109	0.67	0.218	0.62	0
BB11017	0.892	0.848	0.905	0.868	0.792	0.695	0.831	0.742	0.809	0.728	0.708	0.583	0.898	0.854	0.776	0.662
BB11018	0.886	0.713	0.877	0.624	0.834	0.594	0.792	0.337	0.596	0.366	0.553	0.178	0.9	0.733	0.84	0.644
BB11019	0.793	0.412	0.808	0.433	0.737	0.278	0.782	0.371	0.399	0	0.45	0.175	0.788	0.392	0.751	0.32
BB11020	0.869	0.573	0.872	0.587	0.8	0.413	0.797	0.467	0.432	0	0.478	0.12	0.83	0.52	0.834	0.493
BB11021	0.779	0.694	0.745	0.612	0.769	0.673	0.769	0.673	0.575	0.388	0.429	0.102	0.772	0.673	0.769	0.673
BB11022	0.652	0.424	0.778	0.576	0.697	0.606	0.672	0.576	0.369	0	0.318	0.212	0.096	0	0.162	0
BB11023	0.692	0.5	0.764	0.524	0.744	0.524	0.743	0.549	0.292	0	0.247	0.061	0.695	0.439	0.72	0.5
BB11024	0.519	0.223	0.489	0.194	0.526	0.233	0.574	0.233	0.184	0	0.238	0.068	0.547	0.272	0.521	0.223
BB11025	0.579	0.368	0.518	0.263	0.579	0.368	0.491	0.316	0.246	0	0.368	0.368	0.167	0	0.184	0
BB11026	0.626	0.308	0.626	0.308	0.502	0.308	0.531	0.308	0.289	0	0.201	0	0.26	0	0.392	0
BB11027	0.563	0.3	0.672	0.414	0.628	0.329	0.733	0.529	0.524	0.329	0.346	0.171	0.597	0.329	0.628	0.329
BB11028	0.84	0.2	0.96	0.8	0.88	0.4	1	1	0.95	0.75	0.556	0.2	0.76	0	0.656	0
BB11029	0.52	0.471	0.601	0.569	0.533	0.471	0.644	0.569	0.562	0.49	0.431	0.373	0.507	0.471	0.526	0.471
BB11030	0.856	0.377	0.903	0.58	0.764	0.145	0.858	0.507	0.453	0.174	0.521	0.116	0.858	0.377	0.836	0.29
BB11031	0.733	0.472	0.668	0.104	0.704	0.236	0.646	0	0.285	0	0.325	0.0755	0.729	0.255	0.649	0.0566
BB11032	0.93	0.763	0.933	0.763	0.931	0.737	0.915	0.737	0.541	0.0658	0.53	0.289	0.902	0.737	0.913	0.737
BB11033	0.658	0.278	0.6	0	0.681	0.556	0.62	0.278	0.365	0	0.286	0	0.579	0	0.547	0.333
BB11034	0.686	0.24	0.846	0.56	0.738	0.34	0.88	0.6	0.444	0.15	0.314	0.08	0.704	0.23	0.709	0.24
BB11035	0.624	0.486	0.646	0.541	0.668	0.541	0.632	0.541	0.546	0.351	0.522	0.297	0.67	0.541	0.7	0.541

BB11036	0.639	0.359	0.656	0.37	0.679	0.38	0.641	0.38	0.415	0.174	0.408	0.0978	0.622	0.38	0.633	0.38
BB11037	0.527	0.3	0.634	0.384	0.631	0.389	0.661	0.414	0.246	0	0.428	0.296	0.617	0.399	0.628	0.384
BB11038	0.902	0.756	0.894	0.741	0.908	0.763	0.91	0.763	0.538	0.193	0.629	0.43	0.9	0.756	0.899	0.733
BB12001	0.918	0.811	0.923	0.832	0.923	0.811	0.929	0.832	0.823	0.643	0.809	0.656	0.919	0.803	0.926	0.82
BB12002	0.885	0.708	0.909	0.781	0.898	0.737	0.915	0.788	0.839	0.672	0.829	0.642	0.874	0.679	0.903	0.752
BB12003	0.985	0.973	0.988	0.973	0.969	0.946	0.985	0.973	0.753	0.514	0.847	0.676	0.978	0.946	0.985	0.973
BB12004	0.972	0.855	0.977	0.862	0.974	0.882	0.984	0.895	0.833	0.539	0.806	0.546	0.976	0.882	0.973	0.849
BB12005	0.951	0.832	0.952	0.839	0.956	0.839	0.957	0.847	0.891	0.759	0.881	0.723	0.947	0.825	0.959	0.847
BB12006	0.994	0.988	0.991	0.982	0.991	0.982	0.988	0.976	0.921	0.88	0.921	0.88	0.994	0.988	0.991	0.982
BB12007	0.907	0.797	0.911	0.797	0.913	0.794	0.92	0.801	0.79	0.644	0.787	0.593	0.908	0.794	0.909	0.792
BB12008	0.953	0.845	0.979	0.897	0.958	0.845	0.981	0.904	0.91	0.723	0.861	0.638	0.958	0.852	0.96	0.856
BB12009	0.825	0.594	0.931	0.828	0.825	0.609	0.945	0.922	0.9	0.75	0.925	0.812	0.845	0.625	0.828	0.609
BB12010	0.956	0.869	0.956	0.869	0.962	0.869	0.962	0.869	0.802	0.592	0.857	0.65	0.956	0.869	0.944	0.835
BB12011	0.845	0.554	0.817	0.415	0.852	0.585	0.858	0.571	0.687	0.426	0.589	0.315	0.863	0.577	0.851	0.571
BB12012	0.81	0.689	0.827	0.695	0.812	0.677	0.808	0.677	0.574	0.457	0.607	0.433	0.827	0.701	0.812	0.677
BB12013	0.979	0.934	0.978	0.934	0.976	0.936	0.976	0.934	0.904	0.81	0.925	0.834	0.977	0.934	0.976	0.936
BB12014	0.995	0.977	1	1	0.99	0.977	0.995	0.977	1	1	0.979	0.977	1	1	0.995	0.977
BB12015	0.994	0.967	1	1	0.997	0.983	0.997	0.983	0.958	0.883	0.833	0.6	1	1	0.997	0.983
BB12016	0.9	0.767	0.947	0.877	0.895	0.781	0.942	0.904	0.822	0.685	0.893	0.767	0.897	0.753	0.889	0.767
BB12017	0.977	0.929	0.976	0.925	0.979	0.933	0.979	0.933	0.848	0.679	0.805	0.545	0.976	0.925	0.979	0.933
BB12018	0.955	0.919	0.953	0.922	0.954	0.927	0.959	0.937	0.919	0.885	0.919	0.878	0.95	0.916	0.954	0.927
BB12019	0.93	0.867	0.941	0.887	0.931	0.874	0.936	0.884	0.879	0.786	0.862	0.764	0.933	0.872	0.931	0.874
BB12020	0.966	0.942	0.978	0.957	0.973	0.957	0.978	0.957	0.865	0.797	0.792	0.696	0.973	0.957	0.978	0.957
BB12021	0.853	0.728	0.848	0.741	0.833	0.679	0.843	0.728	0.801	0.63	0.655	0.481	0.861	0.716	0.838	0.679
BB12022	1	1	1	1	0.986	0.98	0.986	0.98	0.941	0.902	0.792	0.588	1	1	0.986	0.98
BB12023	0.944	0.891	0.941	0.883	0.922	0.855	0.944	0.891	0.785	0.625	0.769	0.601	0.943	0.889	0.922	0.855
BB12024	1	1	0.994	0.988	0.994	0.988	0.991	0.982	0.899	0.827	0.866	0.786	0.994	0.988	0.991	0.982
BB12025	0.969	0.969	1	1	0.922	0.922	1	1	0.781	0.672	0.859	0.844	0.953	0.922	0.922	0.922
BB12026	0.954	0.783	0.952	0.783	0.945	0.764	0.958	0.777	0.874	0.529	0.862	0.592	0.949	0.783	0.949	0.777
BB12027	0.981	0.935	0.977	0.911	0.98	0.902	0.98	0.911	0.903	0.707	0.847	0.724	0.977	0.894	0.981	0.911
BB12028	0.967	0.887	0.962	0.879	0.968	0.927	0.962	0.911	0.789	0.492	0.852	0.629	0.963	0.895	0.968	0.927
BB12029	0.995	0.985	0.991	0.974	0.993	0.974	0.99	0.969	0.943	0.8	0.953	0.887	0.997	0.99	0.993	0.979
BB12030	0.98	0.955	0.982	0.949	0.979	0.949	0.983	0.955	0.899	0.78	0.868	0.734	0.981	0.955	0.979	0.949
BB12031	0.881	0.753	0.888	0.762	0.875	0.741	0.893	0.763	0.799	0.648	0.83	0.668	0.878	0.751	0.877	0.739
BB12032	0.969	0.889	0.984	0.926	0.934	0.815	1	1	0.772	0.667	0.758	0.519	0.951	0.778	0.985	0.963
BB12033	0.888	0.789	0.931	0.857	0.901	0.807	0.928	0.839	0.728	0.447	0.655	0.366	0.904	0.826	0.901	0.807
BB12034	0.907	0.834	0.936	0.885	0.916	0.853	0.937	0.885	0.897	0.82	0.903	0.82	0.914	0.848	0.916	0.853
BB12036	0.99	0.975	0.991	0.975	0.995	0.988	0.993	0.981	0.938	0.864	0.925	0.815	0.995	0.988	0.995	0.988
BB12037	0.939	0.784	0.953	0.784	0.94	0.756	0.951	0.788	0.598	0	0.673	0.36	0.941	0.78	0.943	0.788

BB12038	0.952	0.882	0.95	0.891	0.958	0.899	0.951	0.891	0.719	0.521	0.771	0.479	0.953	0.891	0.953	0.891
BB12039	0.953	0.886	0.936	0.8	0.931	0.857	0.941	0.857	0.825	0.657	0.926	0.829	0.937	0.8	0.928	0.829
BB12040	0.981	0.969	0.988	0.969	0.97	0.948	0.992	0.979	0.988	0.969	0.931	0.906	0.996	0.99	0.992	0.979
BB12041	0.949	0.87	0.896	0.717	0.848	0.652	0.848	0.652	0.64	0.413	0.759	0.63	0.912	0.739	0.848	0.652
BB12042	0.795	0.69	0.806	0.672	0.773	0.652	0.801	0.672	0.568	0.438	0.63	0.503	0.789	0.662	0.773	0.652
BB12044	0.919	0.783	0.919	0.785	0.922	0.794	0.923	0.799	0.853	0.679	0.832	0.595	0.918	0.783	0.923	0.799
BB20001	0.837	0	0.995	0.96	0.77	0	0.998	0.98	0.868	0	0.88	0.78	0.596	0	0.538	0
BB20002	0.9	0.0625	0.85	0.0625	0.848	0.0625	0.812	0	0.778	0	0.657	0	0.676	0	0.876	0.0625
BB20020	0.946	0.775	0.944	0.775	0.953	0.812	0.953	0.812	0.872	0.623	0.802	0.616	0.934	0.775	0.95	0.79
BB30006	0.887	0.705	0.814	0.568	0.867	0.705	0.852	0.659	0.74	0.318	0.779	0.568	0.814	0.568	0.816	0.386
BB30017	0.794	0.571	0.829	0.657	0.809	0.61	0.812	0.6	0.49	0.171	0.494	0.19	0.787	0.581	0.796	0.571
BB30027	0.859	0.5	0.849	0.318	0.933	0.545	0.909	0.727	0.47	0	0.832	0.5	0.875	0	0.824	0.409
BB40003	0.937	0.828	0.939	0.83	0.939	0.845	0.941	0.828	0.832	0.67	0.811	0.629	0.94	0.828	0.938	0.839
BB40005	0.927	0.727	0.95	0.776	0.92	0.718	0.954	0.81	0.893	0.649	0.841	0.529	0.928	0.733	0.929	0.756
BB40006	0.826	0.42	0.851	0.491	0.831	0.444	0.857	0.473	0.644	0.337	0.629	0.231	0.831	0.45	0.824	0.414
BB40007	0.834	0.548	0.873	0.596	0.836	0.548	0.868	0.587	0.69	0.279	0.671	0.183	0.836	0.548	0.836	0.548
BB40008	0.922	0.771	0.925	0.784	0.932	0.8	0.942	0.8	0.799	0.498	0.87	0.706	0.927	0.788	0.932	0.8
BB40009	0.91	0.597	0.82	0	0.906	0.611	0.806	0	0.746	0.486	0.734	0.486	0.896	0.59	0.907	0.611
BB40010	0.844	0.557	0.919	0.672	0.869	0.607	0.925	0.705	0.918	0.803	0.893	0.689	0.872	0.59	0.865	0.574
BB40014	0.88	0.646	0.89	0.658	0.874	0.62	0.884	0.665	0.888	0.671	0.809	0.494	0.89	0.658	0.884	0.665
BB40018	0.944	0.883	0.927	0.833	0.93	0.85	0.919	0.85	0.794	0.567	0.795	0.683	0.937	0.817	0.913	0.833
BB40019	0.918	0.606	0.92	0.61	0.98	0.916	0.913	0.59	0.89	0.719	0.803	0.452	0.919	0.606	0.915	0.594
BB40022	1	1	1	1	1	1	0.98	0.963	0.965	0.852	0.837	0.704	1	1	1	1
BB40025	0.946	0.869	0.956	0.897	0.951	0.869	0.957	0.883	0.823	0.552	0.806	0.593	0.946	0.876	0.951	0.869
BB40032	0.969	0.926	0.97	0.932	0.962	0.914	0.97	0.938	0.862	0.741	0.833	0.667	0.97	0.938	0.97	0.938
BB40033	0.959	0.873	0.95	0.778	0.954	0.873	0.845	0	0.608	0.254	0.572	0.127	0.947	0.841	0.949	0.873
BB40043	0.836	0.7	0.918	0.825	0.851	0.725	0.918	0.825	0.777	0.562	0.818	0.65	0.844	0.725	0.856	0.725
BB40045	0.869	0.605	0.883	0.684	0.725	0.474	0.761	0.526	0.572	0.237	0.679	0.474	0.88	0.711	0.748	0.447
BB40048	0.939	0.796	0.838	0	0.94	0.789	0.943	0.791	0.938	0.778	0.885	0.667	0.939	0.796	0.939	0.787
BB50002	0.714	0.31	0.693	0	0.692	0.333	0.768	0.405	0.509	0	0.332	0.167	0.582	0	0.555	0
BB50004	0.975	0.913	0.99	0.963	0.981	0.936	0.981	0.933	0.942	0.822	0.904	0.779	0.983	0.94	0.981	0.936
BB50005	0.976	0.889	0.976	0.893	0.971	0.863	0.976	0.891	0.897	0.693	0.657	0.305	0.976	0.889	0.976	0.887
BB50010	0.88	0.374	0.718	0	0.913	0.596	0.679	0	0.635	0.0909	0.56	0.0606	0.875	0.384	0.886	0.505
BB50013	0.954	0.819	0.972	0.866	0.964	0.858	0.973	0.882	0.886	0.583	0.915	0.772	0.949	0.787	0.964	0.858
BB50016	0.883	0.613	0.887	0.423	0.864	0.344	0.712	0.233	0.476	0	0.609	0.141	0.901	0.589	0.87	0.399
BBS11001	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
BBS11002	0.838	0.353	0.971	0.882	0.838	0.353	1	1	0.956	0.824	0.838	0.353	0.838	0.353	0.838	0.353
BBS11003	0.754	0.628	0.819	0.7	0.804	0.672	0.823	0.704	0.668	0.441	0.644	0.417	0.778	0.64	0.804	0.672
BBS11004	0.735	0.603	0.768	0.641	0.766	0.636	0.764	0.636	0.494	0.391	0.38	0.19	0.692	0.56	0.764	0.636

BBS11005	0.567	0.25	0.702	0.42	0.581	0.33	0.645	0.375	0.325	0.148	0.286	0.0682	0.566	0.307	0.584	0.295
BBS11006	0.59	0.333	0.658	0.417	0.6	0.354	0.673	0.417	0.183	0	0.383	0.146	0.614	0.375	0.606	0.333
BBS11007	0.876	0.746	0.87	0.704	0.87	0.725	0.87	0.725	0.654	0.359	0.533	0.359	0.874	0.739	0.87	0.725
BBS11008	0.833	0.776	0.827	0.776	0.796	0.673	0.969	0.939	0.694	0.612	0.646	0.592	0.847	0.776	0.82	0.714
BBS11009	0.883	0.867	0.867	0.833	0.878	0.867	0.853	0.817	0.681	0.567	0.686	0.517	0.869	0.833	0.861	0.833
BBS11010	0.741	0.615	0.744	0.628	0.803	0.692	0.776	0.654	0.485	0.269	0.551	0.487	0.744	0.628	0.795	0.679
BBS11011	0.643	0.391	0.822	0.652	0.667	0.413	0.696	0.391	0.441	0.239	0.48	0.391	0.726	0.565	0.615	0.283
BBS11012	0.949	0.92	0.947	0.916	0.951	0.924	0.95	0.924	0.799	0.661	0.734	0.574	0.951	0.924	0.951	0.924
BBS11013	0.762	0.655	0.859	0.793	0.807	0.517	0.914	0.828	0.134	0	0.645	0.345	0.666	0.586	0.931	0.897
BBS11014	0.885	0.773	0.888	0.78	0.885	0.78	0.887	0.78	0.692	0.563	0.671	0.487	0.888	0.783	0.887	0.783
BBS11015	0.799	0.692	0.874	0.813	0.801	0.697	0.867	0.798	0.58	0.343	0.599	0.399	0.801	0.692	0.806	0.702
BBS11016	0.804	0.573	0.816	0.555	0.878	0.682	0.866	0.636	0.542	0.245	0.663	0.445	0.816	0.6	0.871	0.655
BBS11017	0.918	0.868	0.932	0.901	0.813	0.722	0.852	0.768	0.809	0.728	0.708	0.583	0.914	0.874	0.799	0.689
BBS11018	0.898	0.713	0.908	0.762	0.878	0.673	0.832	0.446	0.617	0.436	0.652	0.515	0.904	0.772	0.877	0.653
BBS11019	0.82	0.454	0.853	0.505	0.794	0.454	0.852	0.495	0.408	0	0.458	0.175	0.811	0.412	0.821	0.454
BBS11020	0.914	0.733	0.891	0.653	0.856	0.667	0.907	0.773	0.607	0.28	0.535	0.267	0.866	0.533	0.894	0.787
BBS11021	0.908	0.816	0.867	0.735	0.891	0.796	0.891	0.796	0.707	0.551	0.779	0.653	0.908	0.816	0.898	0.796
BBS11022	0.722	0.545	0.687	0.576	0.985	0.97	0.97	0.939	0.399	0	0.626	0.515	0.732	0.545	0.939	0.879
BBS11023	0.74	0.537	0.774	0.585	0.746	0.524	0.801	0.61	0.294	0	0.339	0.061	0.759	0.561	0.765	0.561
BBS11024	0.604	0.369	0.56	0.33	0.68	0.534	0.672	0.524	0.445	0.291	0.356	0.252	0.557	0.291	0.684	0.544
BBS11025	0.544	0.368	0.737	0.684	0.842	0.684	0.737	0.684	0.272	0	0.421	0.368	0.474	0.368	0.737	0.684
BBS11026	0.623	0.308	0.564	0.308	0.802	0.615	0.758	0.615	0.48	0	0.348	0	0.498	0	0.692	0.308
BBS11027	0.619	0.314	0.712	0.514	0.729	0.529	0.81	0.657	0.514	0.329	0.333	0.171	0.643	0.386	0.729	0.529
BBS11028	0.878	0.8	1	1	1	1	1	1	0.906	0.75	0.626	0.55	1	1	1	1
BBS11029	0.634	0.588	0.706	0.686	0.634	0.588	0.745	0.686	0.562	0.49	0.431	0.373	0.608	0.569	0.627	0.588
BBS11030	0.884	0.522	0.93	0.739	0.8	0.464	0.885	0.681	0.65	0.116	0.554	0.116	0.885	0.522	0.833	0.449
BBS11031	0.725	0.481	0.82	0.557	0.769	0.406	0.755	0.377	0.217	0	0.378	0.0755	0.751	0.434	0.724	0.321
BBS11032	0.923	0.737	0.941	0.776	0.947	0.789	0.933	0.789	0.569	0.145	0.604	0.368	0.941	0.776	0.933	0.789
BBS11033	0.721	0.278	0.714	0.278	0.982	0.944	0.964	0.889	0.395	0	0.422	0.278	0.737	0.278	0.964	0.889
BBS11034	0.848	0.59	0.855	0.58	0.843	0.53	0.854	0.54	0.468	0.33	0.592	0.38	0.846	0.55	0.836	0.5
BBS11035	0.738	0.622	0.738	0.676	0.762	0.676	0.727	0.676	0.435	0.351	0.522	0.297	0.773	0.676	0.784	0.676
BBS11036	0.701	0.424	0.715	0.446	0.731	0.446	0.715	0.446	0.482	0.196	0.458	0.272	0.701	0.435	0.709	0.446
BBS11037	0.553	0.246	0.651	0.399	0.633	0.34	0.674	0.419	0.353	0.163	0.323	0.138	0.625	0.374	0.639	0.355
BBS11038	0.914	0.785	0.879	0.726	0.907	0.756	0.895	0.8	0.665	0.526	0.688	0.407	0.889	0.719	0.919	0.8
BBS12001	0.931	0.832	0.936	0.836	0.927	0.807	0.934	0.832	0.811	0.615	0.802	0.623	0.92	0.803	0.929	0.82
BBS12002	0.898	0.715	0.918	0.781	0.9	0.715	0.924	0.788	0.868	0.715	0.856	0.723	0.896	0.715	0.905	0.73
BBS12003	0.98	0.973	0.988	0.973	0.969	0.946	0.985	0.973	0.772	0.514	0.847	0.676	0.973	0.946	0.985	0.973
BBS12004	0.973	0.868	0.977	0.862	0.977	0.888	0.981	0.895	0.85	0.559	0.843	0.539	0.975	0.875	0.98	0.895
BBS12005	0.972	0.927	0.974	0.934	0.98	0.934	0.977	0.927	0.864	0.679	0.873	0.701	0.967	0.912	0.977	0.927

BBS12006	0.994	0.988	0.991	0.982	0.994	0.988	0.991	0.982	0.921	0.88	0.921	0.88	0.994	0.988	0.994	0.988
BBS12007	0.907	0.796	0.91	0.796	0.914	0.797	0.921	0.802	0.788	0.644	0.784	0.59	0.907	0.789	0.91	0.793
BBS12008	0.965	0.86	0.982	0.9	0.965	0.886	0.983	0.915	0.924	0.771	0.869	0.738	0.963	0.852	0.968	0.889
BBS12009	0.845	0.641	0.931	0.828	0.852	0.641	0.981	0.953	0.9	0.75	0.925	0.812	0.852	0.641	0.852	0.641
BBS12010	0.957	0.874	0.957	0.874	0.964	0.879	0.964	0.879	0.802	0.592	0.832	0.636	0.957	0.874	0.964	0.879
BBS12011	0.851	0.562	0.88	0.585	0.853	0.577	0.877	0.574	0.723	0.54	0.69	0.455	0.859	0.58	0.857	0.565
BBS12012	0.805	0.689	0.834	0.701	0.817	0.683	0.812	0.683	0.731	0.543	0.576	0.433	0.838	0.701	0.817	0.683
BBS12013	0.979	0.936	0.978	0.936	0.979	0.938	0.979	0.938	0.914	0.839	0.931	0.836	0.978	0.936	0.979	0.938
BBS12014	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
BBS12015	0.997	0.983	0.995	0.983	0.997	0.983	0.988	0.967	0.928	0.783	0.955	0.867	1	1	0.997	0.983
BBS12016	0.919	0.808	0.974	0.945	0.932	0.849	0.97	0.945	0.849	0.685	0.766	0.575	0.934	0.863	0.932	0.849
BBS12017	0.981	0.937	0.977	0.929	0.98	0.937	0.98	0.937	0.871	0.757	0.821	0.601	0.976	0.925	0.98	0.937
BBS12018	0.959	0.922	0.959	0.924	0.962	0.928	0.966	0.939	0.93	0.881	0.932	0.878	0.955	0.915	0.962	0.928
BBS12019	0.941	0.887	0.947	0.899	0.94	0.887	0.945	0.897	0.888	0.791	0.884	0.791	0.939	0.884	0.94	0.887
BBS12020	0.966	0.942	0.978	0.957	0.973	0.957	0.978	0.957	0.865	0.797	0.792	0.696	0.973	0.957	0.978	0.957
BBS12021	0.991	0.974	0.991	0.974	0.983	0.949	0.991	0.974	0.909	0.821	0.812	0.692	0.983	0.949	0.983	0.949
BBS12022	1	1	1	1	1	1	1	1	0.91	0.843	0.949	0.902	1	1	1	1
BBS12023	0.944	0.891	0.944	0.891	0.922	0.855	0.944	0.891	0.785	0.625	0.759	0.585	0.943	0.889	0.922	0.855
BBS12024	1	1	0.994	0.988	0.997	0.994	0.994	0.988	0.922	0.857	0.908	0.845	0.994	0.988	0.994	0.988
BBS12025	0.935	0.922	0.932	0.922	1	1	1	1	0.81	0.75	0.859	0.844	0.935	0.922	1	1
BBS12026	0.952	0.783	0.966	0.796	0.948	0.771	0.965	0.79	0.886	0.471	0.873	0.548	0.954	0.783	0.954	0.777
BBS12027	0.976	0.911	0.979	0.911	0.977	0.894	0.979	0.911	0.844	0.561	0.844	0.748	0.977	0.894	0.977	0.894
BBS12028	0.961	0.879	0.96	0.871	0.968	0.927	0.963	0.911	0.821	0.581	0.836	0.621	0.963	0.887	0.968	0.927
BBS12029	0.993	0.974	0.991	0.974	0.992	0.974	0.992	0.969	0.944	0.8	0.959	0.908	0.993	0.979	0.992	0.969
BBS12030	0.98	0.955	0.982	0.949	0.981	0.955	0.985	0.96	0.954	0.887	0.894	0.785	0.977	0.944	0.981	0.955
BBS12031	0.88	0.753	0.888	0.762	0.875	0.744	0.888	0.757	0.802	0.647	0.83	0.668	0.878	0.751	0.878	0.741
BBS12032	0.986	0.963	1	1	0.942	0.852	1	1	0.849	0.519	0.771	0.519	0.992	0.963	0.992	0.963
BBS12033	0.888	0.795	0.923	0.857	0.904	0.82	0.931	0.851	0.782	0.571	0.717	0.484	0.901	0.832	0.904	0.82
BBS12034	0.916	0.848	0.945	0.899	0.922	0.862	0.944	0.894	0.897	0.82	0.903	0.82	0.923	0.862	0.922	0.862
BBS12036	0.991	0.981	0.993	0.981	0.995	0.988	0.993	0.981	0.957	0.907	0.925	0.815	0.995	0.988	0.995	0.988
BBS12037	0.945	0.792	0.96	0.8	0.944	0.792	0.951	0.776	0.742	0.524	0.735	0.464	0.947	0.796	0.946	0.8
BBS12038	0.963	0.882	0.968	0.891	0.977	0.95	0.97	0.933	0.839	0.706	0.723	0.479	0.965	0.891	0.977	0.95
BBS12039	0.951	0.857	0.937	0.829	0.946	0.857	0.94	0.857	0.825	0.657	0.931	0.857	0.947	0.829	0.922	0.829
BBS12040	0.985	0.979	0.988	0.969	0.97	0.948	0.992	0.979	0.975	0.969	0.931	0.906	0.996	0.99	0.992	0.979
BBS12041	0.933	0.826	0.907	0.739	0.922	0.761	0.922	0.761	0.821	0.717	0.831	0.739	0.922	0.761	0.922	0.761
BBS12042	0.809	0.714	0.814	0.703	0.788	0.683	0.816	0.703	0.568	0.438	0.641	0.503	0.805	0.69	0.788	0.683
BBS12044	0.919	0.781	0.919	0.785	0.926	0.803	0.925	0.803	0.845	0.679	0.837	0.639	0.919	0.788	0.925	0.803
BBS20001	1	1	0.998	0.98	1	1	0.998	0.98	0.993	0.96	0.898	0.78	0.998	0.98	0.998	0.98
BBS20002	0.908	0.438	0.962	0.625	0.931	0.438	0.914	0.438	0.931	0.312	0.717	0	0.914	0.438	0.944	0.562

BBS20020	0.946	0.775	0.944	0.775	0.95	0.783	0.95	0.783	0.896	0.761	0.802	0.616	0.948	0.775	0.949	0.783
BBS30006	0.894	0.75	0.847	0.636	0.88	0.727	0.868	0.682	0.809	0.591	0.799	0.568	0.857	0.659	0.858	0.659
BBS30017	0.821	0.638	0.864	0.695	0.848	0.686	0.864	0.714	0.496	0.171	0.534	0.19	0.84	0.695	0.848	0.667
BBS30027	0.898	0.818	0.961	0.818	0.961	0.818	0.991	0.909	0.88	0.727	0.871	0.818	0.94	0.818	0.961	0.818
BBS50002	0.829	0.548	0.829	0.524	0.768	0.357	0.752	0.357	0.524	0.357	0.419	0.286	0.825	0.548	0.766	0.357
BBS50005	0.976	0.889	0.976	0.893	0.972	0.863	0.972	0.865	0.912	0.711	0.656	0.305	0.976	0.889	0.977	0.889
BBS50010	0.896	0.495	0.848	0.111	0.907	0.576	0.848	0.131	0.624	0	0.641	0.111	0.902	0.535	0.906	0.586
BBS50013	0.956	0.827	0.974	0.874	0.956	0.835	0.97	0.866	0.925	0.811	0.916	0.685	0.949	0.787	0.956	0.835
BBS50016	0.919	0.681	0.814	0	0.907	0.583	0.769	0.0123	0.572	0.0798	0.658	0.245	0.906	0.601	0.917	0.663

Table C.1: Q score and TC score for different programs in the working database.