Software Requirements Specification for SubLiMat

Uriel Garcilazo Cruz

February 5, 2025

Contents

Revision History

Date	Version	Notes
February 1 2025	1.0	Document Creation

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Throughout this document SI (Système International d'Unités) is NOT employed as the unit system. Instead of the basic units from SI, several units are described under the symbols section below.

symbol	unit	SI

1.2 Table of Symbols

Throughout this document the standard Human Genome Variation Society (HGVS) nomenclature is employed as the unit system. Additional units that are unique tu this document are prefixed with *.

symbol	unit	HGVS
*qa	alignment quality	fundamental unit of alignment quality
bp	base pair	fundamental unit of genetic sequence length
Kb	kilobase unit	one thousand base pairs

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made to be consistent with the bioinformatics literature and with the existing internationally recognized standard for the description of DNA, RNA and protein reading frames. The symbols are listed in alphabetical order.

symbol	unit	description
F	–	Comparative alignment between two sequences
g	qa	penalty associated with a gap in the alignment of two sequences
N_A	bp	Adenine nitrogenous base, element of the purine family of nucleotides with an amine group in Carbon 6 of its pyrimidine ring
N_C	bp	Cytosine nitrogenous base, element of the pyrimidine family of nucleotides with a no methylated carbons making part of its pyrimidine ring

N_G	bp	Guanine nitrogenous base, element of the purine family of nucleotides with an amine group on Carbon 2 and a carbonyl group on Carbon 6 of its pyrimidine ring
N_T	bp	Tymine nitrogenous base, element of the pyrimidine family of nucleotides with a methyl group in Carbon 5 of its pyrimidine ring
Q_{AB}	qa	A collection of base pairs representing a genetic sequence to be compared with another sequence SEQ_A
S	qa	Substitution matrix used to score the alignment of two sequences
SEQ_A	Kb	A collection of base pairs representing a genetic sequence to be compared with another sequence SEQ_B
SEQ_B	Kb	A collection of base pairs representing a genetic sequence to be compared with another sequence SEQ_A
SNP	bp	single nucleotide polymorphism; variation in a single base pair in DNA sequence
T_I	qa	A transition occurrying between nucleotides of the same nitrogenous base families
T_V	qa	A transition occurrying between nucleotides of different nitrogenous base families

1.3 Abbreviations and Acronyms

symbol	**description**
HGVS	Human Genome Variation Society
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
SubLiMat	Substitution Matrix benchmarking with pairwise alignment
TM	Theoretical Model

2 Introduction

Substitution matrices are critical assumptions that greatly impact studies in the area of comparative biology, yet, benchmarking these matrices is a laborious task.

The following section contains an overview of the Software Requirements Specification (SRS) for a substitution matrix benchmark tool via pairwise alignment. The program is referred to as SubLiMat. The purpose of this section is to characterize the purpose, scope of Requirements, characteristics of Intended Reader, and Organization of the SRS document.

2.1 Purpose of Document

The purpose of this document is to provide a detailed and standardized characterization of the elements, theoretical and operational, that surround the SubLiMat software. Such elements include goals, assumptions, and theoretical and instanced models that describe the scientific basis of the software. Moreover, the document is intended to be used as a guide to detail the unique characteristics of the software to improve on its verifiability and correctness.

2.2 Scope of Requirements

The scope of the requirements for the SubLiMat software includes the evaluation of moderate-sized genetic sequences with similar dimensions.

2.3 Characteristics of Intended Reader

The intended readers of this documentation should have a general understanding of genetics, equivalent or higher to a highschool level. Although not necessary, the document may benefit from a reader who possesses a basic understanding of comparative biology equivalent to first year university level or higher.

2.4 Organization of Document

The structure of this document follows the standard template for an SRS document. As presented by Jegatheesan & Smith, 2019 in their SRS example for this section:

> The organization of this document follows the template for an SRS for scientific computing software proposed by Koothoor (2013), Smith and Lai (2005), Smith et al. (2007), and Smith and Koothoor (2016). The presentation follows the standard pattern of presenting goals, theories, definitions, and assumptions. ...
>
> The goal statements are refined to the theoretical models and the theoretical models to the instance models.

3 General System Description

This section provides general information about the system. It identifies the interfaces between the system and its environment, describes the user characteristics and lists the system constraints.

3.1 System Context

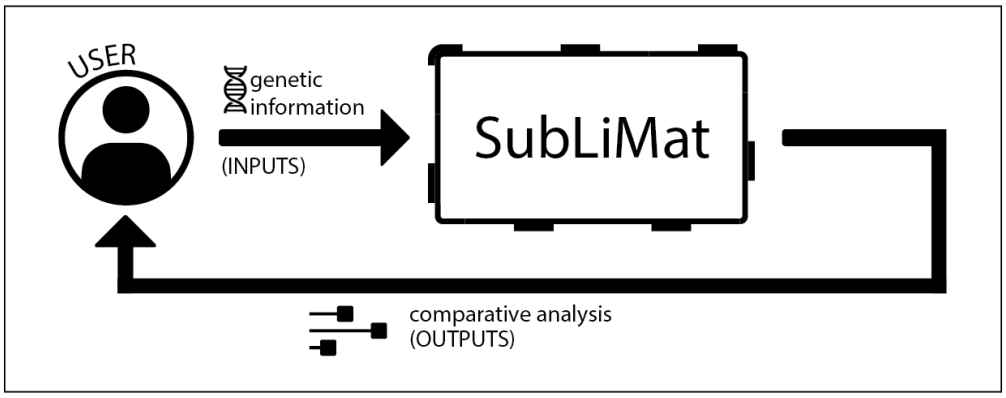


Figure 1: System Context

- User Responsibilities:
  - Provide genetic sequences of DNA
  - Provide meaningful genetic sequences presumed to share a common ancestor
  - Provide genetic sequences of similar dimensions
- SubLiMat Responsibilities:
  - Detect data type mismatch, such as a string of characters instead of a floating point number
  - Determine if there exist any base pairs in the genetic sequences that are not part of the standard genetic code nomenclature for DNA
  - Calculate the alignment quality between two genetic sequences to produce outputs

3.2 User Characteristics

The end user of SubLiMat should have a general understanding of genetics and be familiar with the use of genetic sequences to make hypotheses of common ancestry.

3.3 System Constraints

An alphabet of 4 letters, standardized to the genetic DNA nomenclature, should be used to represent the genetic sequences.

4 Specific System Description

4.1 Problem Description

SubLiMat is intended to solve the uncertainties associated with the influence of substitution matrices on the alignment quality of genetic sequences.

4.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

- Substitution matrix: A square matrix that summarizes the rewards or penalties of moving from one base pair to another and expressed in units of alignment quality (qa)
- Pairwise alignment: A pairwise alignment is the process of aligning two genetic sequences to identify regions of similarity
- Genetic sequence: A genetic sequence is a string of characters that represent the nucleotides of a DNA molecule and expressed in units of base pairs (bp)
- Alignment quality: A measure of the quality of the alignment between two genetic sequences and express in units of alignment quality (qa)
- gap: A gap is a space in the alignment of two genetic sequences that represents a deletion or insertion of a base pair, and penalized with units of alignment quality (qa)

4.1.2 Physical System Description

The physical system of SubLiMat, as shown in Figure 2, includes the following elements:

PS1: sequence SEQ_A and sequence SEQ_B

PS2: Pairwise comparison matrix F between the two sequences

4.1.3 Goal Statements

Given two genetic sequences of size n, the goal statements are:

GS1: Generate scores ranking the quality of the alignment between two genetic sequences across multiple substitution matrices

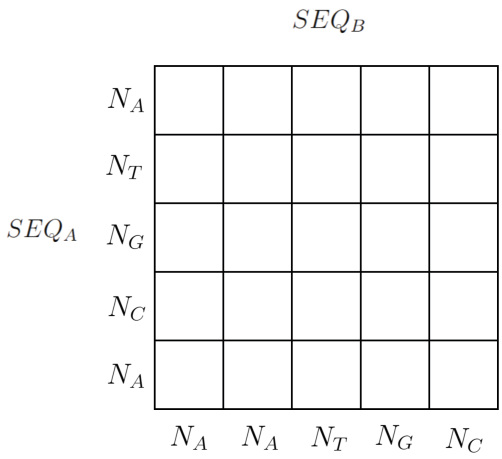
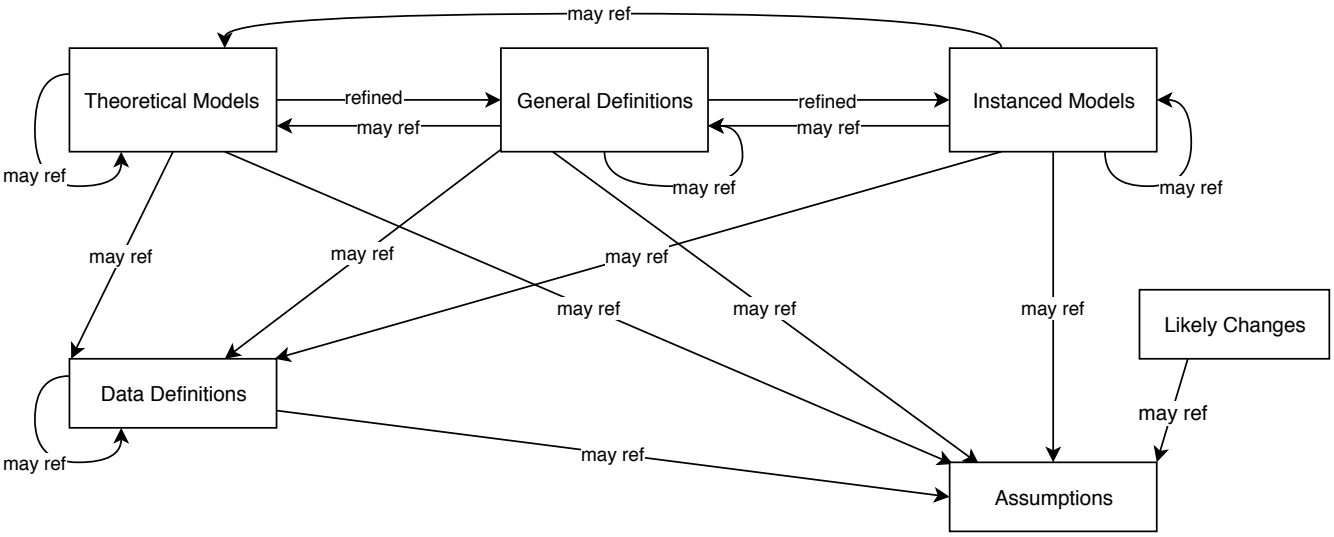


Figure 2: Pairwise comparison of genetic sequences

4.2 Solution Characteristics Specification



The instance models that govern SubLiMat are presented in Subsection 4.2.5. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

4.2.1 Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square

brackets refer to the theoretical model [TM], general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

A1: DNA-sequences-only: The only type of genetic sequences that will be considered are DNA sequences

A2: Two-sequences-only: The only valid number of genetic sequences to be compared is two

4.2.2 Theoretical Models

This section focuses on the general equations and laws that SubLiMat is based on.

RefName: **TM:DPO**

Label: Dynamic Programming Optimization

Equation: $F_{ij} = \max(F_{i-1,j-1} + S(SEQ_{A_i}, SEQ_{B_j}), F_{i,j-1} + g, F_{i-1,j} + g)$

Description: The above equation gives the optimal alignment score between two genetic sequences SEQ_{A_i} and SEQ_{B_j}, given in base pair (bp) units, where $F_{i,j}$ is the alignment score at position i in genetic sequence SEQ_A and position j in genetic sequence SEQ_B (in qa units), g is the gap penalty associated with a deletion or insertion (given in qa units), and S is the substitution matrix (given in qa units).

Notes: None.

Source: Needleman-Wunsch Algorithm, Needleman and Wunsch (1970)

Ref. By: –

Preconditions for TM:DPO: None

Derivation for TM:DPO: Not Applicable

4.2.3 General Definitions

This section collects the laws and equations that will be used in building the instance models.

4.2.4 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given.

Number	DD1
Label	**Comparative alignment matrix**
Symbol	F
Units	qa
Equation	$F_{ij} = \max(F_{i-1,j-1} + S(SEQ_{A_i}, SEQ_{B_j}), F_{i,j-1} + g, F_{i-1,j} + g)$
Description	Comparative matrix of the alignment between two sequences, encoding the positional quality of each possible combination of the base pairs that conform such alignment.
Sources	–
Ref. By	IM1

Number	DD2
Label	**Set of substitution matrices**
Symbol	$\mathbb{S}$
Units	–
Equation	$S_k \in \{S_1, S_2, ..., S_n\}$
Description	Set of substitution matrices that will be used to calculate the alignment quality between two genetic sequences.
Sources	–
Ref. By	IM1

4.2.5 Instance Models

This section transforms the problem defined in Section 4.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 4.2.4 to replace the abstract symbols in the models identified in Sections 4.2.2 and 4.2.3.

The goal GS1 is met by IM1.

Number	IM1
Label	O
Input	SEQ_A,SEQ_B,S_k,g
Output	O_{AB}
Description	$O_{AB} = \forall S_k \in \mathbb{S} : F^k_{i,j} = \max(F^k_{i-1,j-1} + S_k(SEQ_A, SEQ_B), F^k_{i,j-1} + g, F^k_{i-1,j} + g)$ SEQ_A and SEQ_B are biological genetic sequences with bp units S_k is a substitution matrix element of $\mathbb{S}$ g is the gap penalty given in qa units F is comparison matrix between sequences, with each cell given in qa units
Sources	–
Ref. By	–

4.2.6 Input Data Constraints

Table 2 shows the data constraints on the input output variables. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. The column for software constraints restricts the range of inputs to reasonable values. The software constraints will be helpful in the design stage for picking suitable algorithms. The constraints are conservative, to give the user of the model the flexibility to experiment with unusual situations. The column of typical values is intended to provide a feel for a common scenario. The uncertainty column provides an estimate of the confidence with which the physical quantities can be measured. This information would be part of the input if one were performing an uncertainty quantification exercise.

(*) The vector O_{AB} is presented as a named collection of scores

5 Requirements

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

5.1 Functional Requirements

R1: Input SEQ_A, SEQ_B as strings of base pair units (bp), substitution matrix $S \in \mathbb{R}^{n \times n}$, and gap penalty $g \in \mathbb{R}_{<0}$.

Table 2: Input Variables

Var	Physical Constraints	Software Constraints	Typical Value	Uncertainty
SEQ_A	$\|seq_B\| \geq 1$	$\|seq_A\| \approx \|seq_B\|$	1 kb	40%
SEQ_B	$\|seq_A\| \geq 1$	$\|seq_B\| \approx \|seq_A\|$	1 kb	40%
S_k	$S \in \mathbb{R}^{n \times n}, n \geq 4$	$S \in \mathbb{R}^{n \times n}, n \geq 0$	$S \in \mathbb{R}^{4 \times 4}$	0%
F	$F \in \sum \|seq_i\| \times \|seq_j\|$	$\|seq_i\|, \|seq_j\| \geq 1$	$\approx 1kb^2$	20%
g	$g \in \mathbb{R}_{\leq 0}$	–	-2	10%
O_{AB}	$O_{AB} \in \mathbb{R}^{m \times n}, m, n \geq 1$	–	$*\vec{v} = [0, -2, -12]$	0%

R2: Use the inputs stated in IM1 to build a comparative matrix F^k for each substitution matrix S_k in $\mathbb{S}$.

R3: Calculate optimal alignment scores using dynamic programming recursion IM1.

R4: Verify that:

- Input sequences contain only valid nucleotides (A,T,C,G)
- Sequences meet minimum length requirement $|seq_i|, |seq_j| \geq 1$
- Gap penalty is negative $g < 0$
- Substitution matrices are square $n \times n$

R5: Output:

- Aligned sequences with gap insertions
- Alignment scores for each S_k
- Ranking of substitution matrices by alignment quality

5.2 Nonfunctional Requirements

NFR1: **Accuracy** The alignment quality scores produced by SubLiMat shall meet the precision requirements needed for comparative biology research.

NFR2: **Usability** Users with knowledge of genetics and comparative biology, as described in Section 3.2, should be able to successfully use the software with minimal training. The interface shall accept standard sequence formats and provide clear visualization of alignments.

NFR3: **Maintainability** The effort required to modify or extend SubLiMat with new substitution matrices (e.g. protein matrices) should be less than 5% of the original development time, and no more than 40% of the original development time for new alignment algorithms (e.g. heuristics).

NFR4: **Portability** SubLiMat shall run on Linux, Windows 10+, and MacOS 13+ operating systems.

NFR5: **Performance** SubLiMat shall complete alignment calculations for sequences of length n in $O(n^2)$ time complexity.

5.3 Rationale

The rationale behind the assumption A1 relies on the unique benchmarking properties of the set $\mathbb{S}$ that contains only DNA substitution matrices. This improves on the user experience and improves the modularity in the software, enhancing maintainability and portability, which are key nonfunctional requirements.

The second rationale that justifies assumption A2 is the nature of the Needleman-Wunsch algorithm, which guarantees optimal alignment in 2D matrices.

6 Likely Changes

LC1: The software may be extended to include protein sequences, which will require expanding the set of substitution matrices $\mathbb{S}$ to include protein matrices.

7 Unlikely Changes

LC2: The dimensionality of matrix F shall remain 2D, as the Needleman-Wunsch algorithm is designed to optimize global alignment scores.

8 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an "X" may have to be modified as well. Table 5 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 6 shows the dependencies of instance models, requirements, and data constraints on each other. Table 4 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

	A1	A2	TM4.2.2	DD1	IM1
A1	–				
A2		–			
TM4.2.2	X	X	–		
DD1	X	X		–	
IM1	X	X	X		–

Table 4: Traceability Matrix Showing the Connections Between Assumptions and Other Items

	TM4.2.2	DD1	DD2	IM1
TM4.2.2	–	X		X
DD1		–		
DD2			–	
IM1	X	X		–

Table 5: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM1	R1	R2	R3	R4	R5
IM1	–	X		X		X
R1	X	–	X			
R2		X	–			
R3	X			–		X
R4		X		X	–	
R5	X			X		–

Table 6: Traceability Matrix Showing the Connections Between Requirements and Instance Models

The purpose of the traceability graphs is also to provide easy references on what has to be additionally modified if a certain component is changed. The arrows in the graphs represent dependencies. The component at the tail of an arrow is depended on by the component at the head of that arrow. Therefore, if a component is changed, the components that it points to should also be changed. Figure 3 shows the dependencies of theoretical models, general definitions, data definitions, instance models, likely changes, and assumptions on each other. Figure 4 shows the dependencies of instance models, requirements, and data constraints on each other.

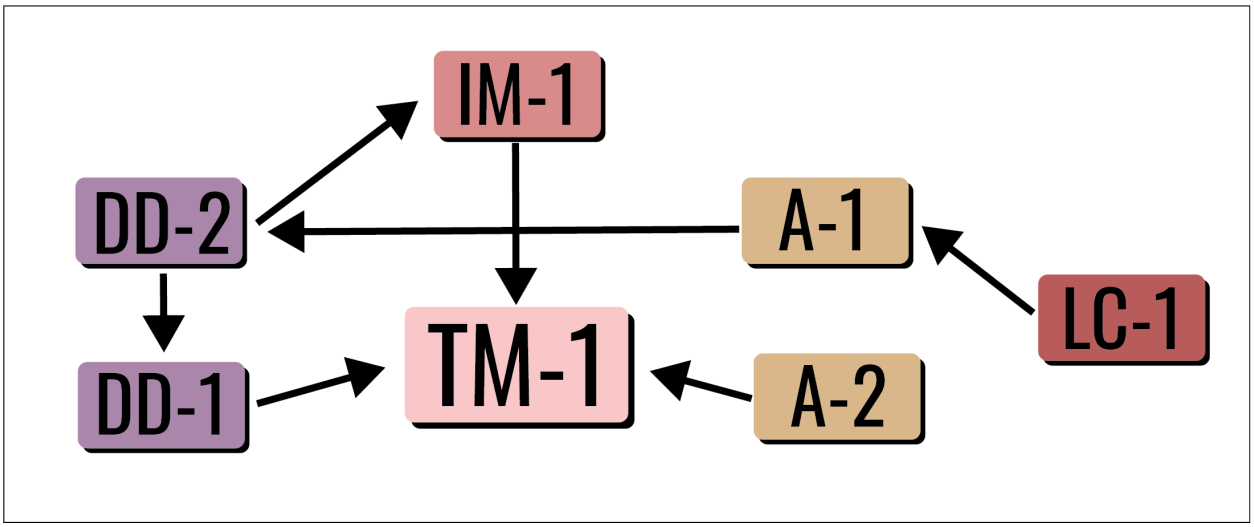


Figure 3: Traceability Matrix Showing the Connections Between Items of Different Sections

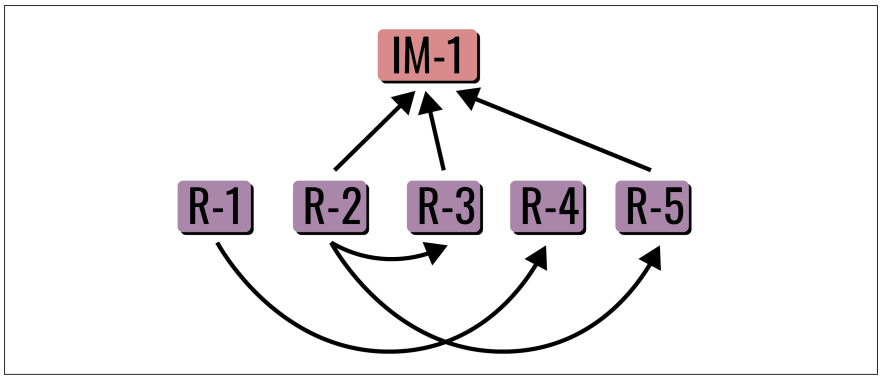


Figure 4: Traceability Matrix Showing the Connections Between Requirements, Instance Models, and Data Constraints

9 Values of Auxiliary Constants

Symbol	Description	Value	Unit
BS	Baseline substitution matrix $s \in \mathbb{S}$	$\begin{bmatrix} 0 & -3 & -1 & -3 \\ -3 & 0 & -3 & -1 \\ -1 & -3 & 0 & -3 \\ -3 & -1 & -3 & 0 \end{bmatrix}$	qa
JC	Jukes Cantor substitution matrix $s \in \mathbb{S}$	$\begin{bmatrix} 1.0 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & 1.0 & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 1.0 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 1.0 \end{bmatrix}$	qa
$K80$	Kimura 1980 substitution matrix $s \in \mathbb{S}$	$\begin{bmatrix} 1.0 & -2.0 & -1.0 & -2.0 \\ -2.0 & 1.0 & -2.0 & -1.0 \\ -1.0 & -2.0 & 1.0 & -2.0 \\ -2.0 & -1.0 & -2.0 & 1.0 \end{bmatrix}$	qa
$HKY85$	Hasegawa-Kishino-Yano 1985 matrix $s \in \mathbb{S}$	$\begin{bmatrix} 1.0 & -2.5 & -1.0 & -2.5 \\ -2.5 & 1.0 & -2.5 & -1.0 \\ -1.0 & -2.5 & 1.0 & -2.5 \\ -2.5 & -1.0 & -2.5 & 1.0 \end{bmatrix}$	qa
$TN93$	Tamura-Nei 1993 substitution matrix $s \in \mathbb{S}$	$\begin{bmatrix} 1.0 & -2.5 & -1.0 & -2.5 \\ -2.5 & 1.0 & -2.5 & -1.5 \\ -1.0 & -2.5 & 1.0 & -2.5 \\ -2.5 & -1.5 & -2.5 & 1.0 \end{bmatrix}$	qa

Table 7: Values of Auxiliary Constants

References

Nirmitha Koothoor. *A Document Driven Approach to Certifying Scientific Computing Software*. Ph.d. thesis, McMaster University, Hamilton, ON, Canada, 2013.

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. ISSN 0022-2836. doi: https://doi.org/10.1016/0022-2836(70)90057-4. URL https://www.sciencedirect.com/science/article/pii/0022283670900574.

W. Spencer Smith and Nirmitha Koothoor. A document-driven method for certifying scientific computing software for use in nuclear safety analysis. *Nuclear Engineering and Technology*, 48(2), April 2016.

W. Spencer Smith and Lei Lai. A new requirements template for scientific computing. In P. J. Agerfalk, N. Kraiem, and J. Ralyte, editors, *Proceedings of the First International*

Workshop on Situational Requirements Engineering Processes - Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05, pages 107–121, Paris, France, 2005. In conjunction with 13th IEEE International Requirements Engineering Conference.

W. Spencer Smith, Lei Lai, and Ridha Khedri. Requirements analysis for engineering computation: A systematic approach for improving software reliability. *Reliable Computing, Special Issue on Reliable Engineering Computation*, 13(1):83–107, February 2007.