SimFluVar: an analytical tool for calculating the codon variation patterns of influenza virus

Insung Ahn

January 2014

ISBN 978-89-294-0470-3

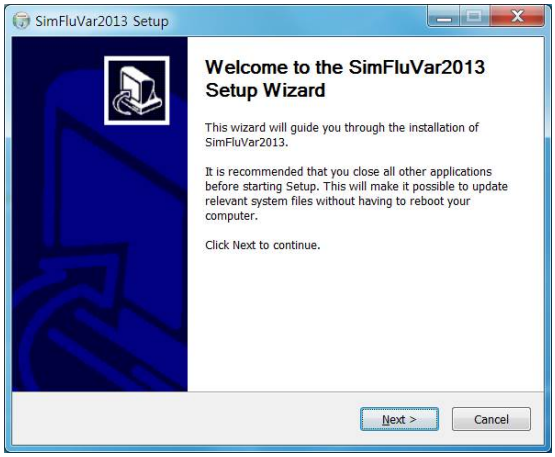
Index

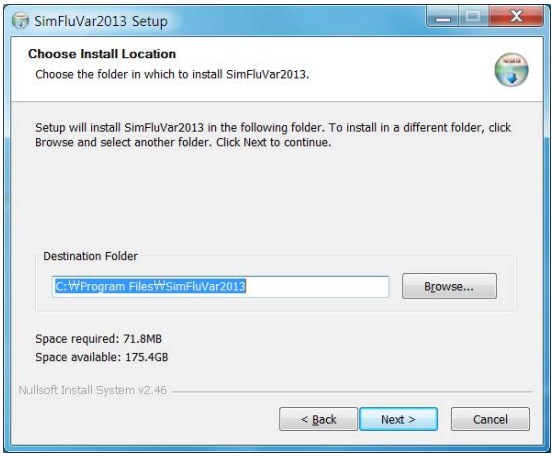
Introduction

SimFluVar is a program in the SimFlu (**Sim**ulation tool for in**flu**enza virus) series; its modification parameters, in 61x61 formations, are calculated based on year-to-year modification patterns extracted from empirically discovered influenza virus' gene information. This program's output is codon-level computation of the modification patterns with time on influenza A virus classified by year; the program also provides analysis results on change in base creation within Wobble base position of each codon.

1. Installation

Setup can be done automatically by simply executing the distributed setup program. First, ensure that Java Runtime Environment (JRE) is already installed on the computer; if it is not, please visit the JAVA distribution page (http://www.oracle.com/technetwork/java/javase/downloads/index.html) to install the latest JRE. After downloading it on the PC, run the program by double-clicking on the 'SimFluVar2013_xxxx.setupexe' file. Upon entering all required information about the setup, you will be taken to the next screen, where the final setup will take place.

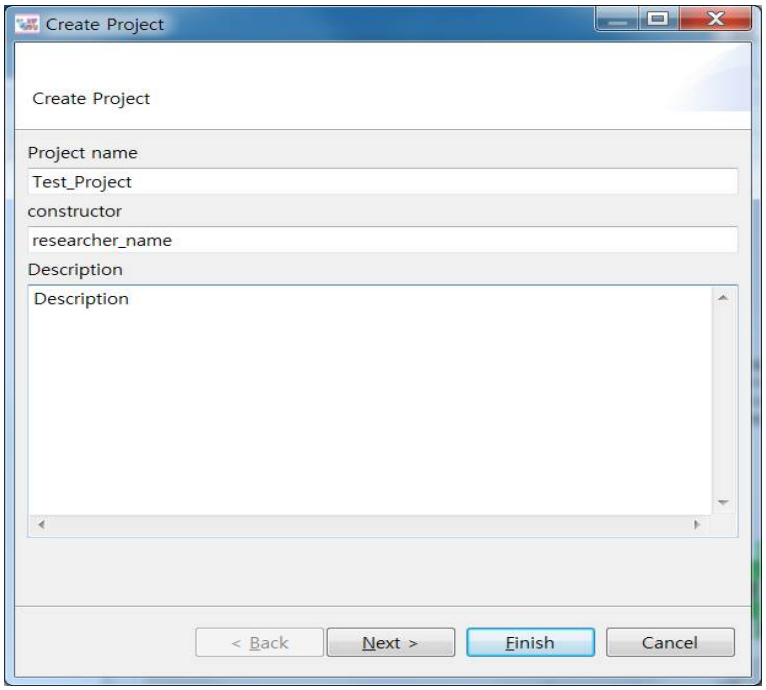




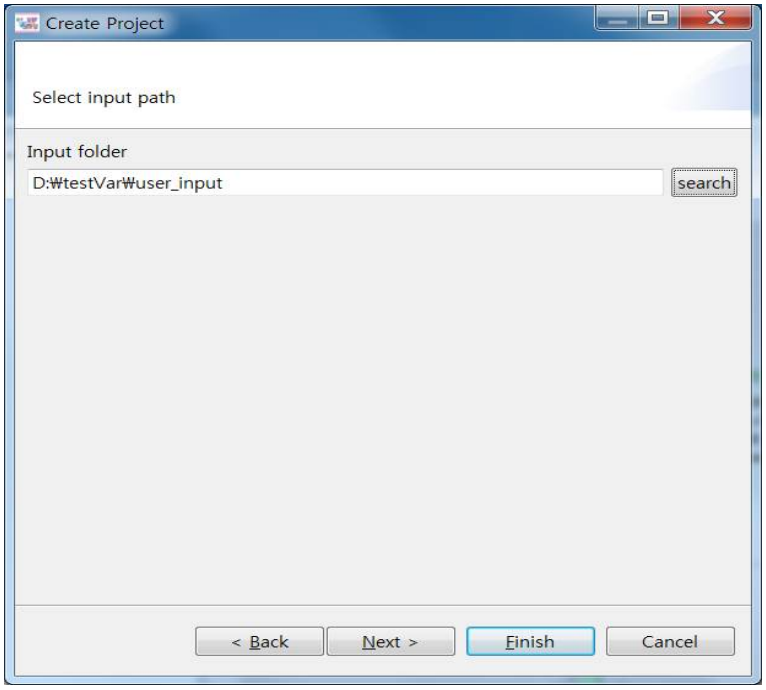
2. Project Creation and Execution

2.1 Executing Project Wizard

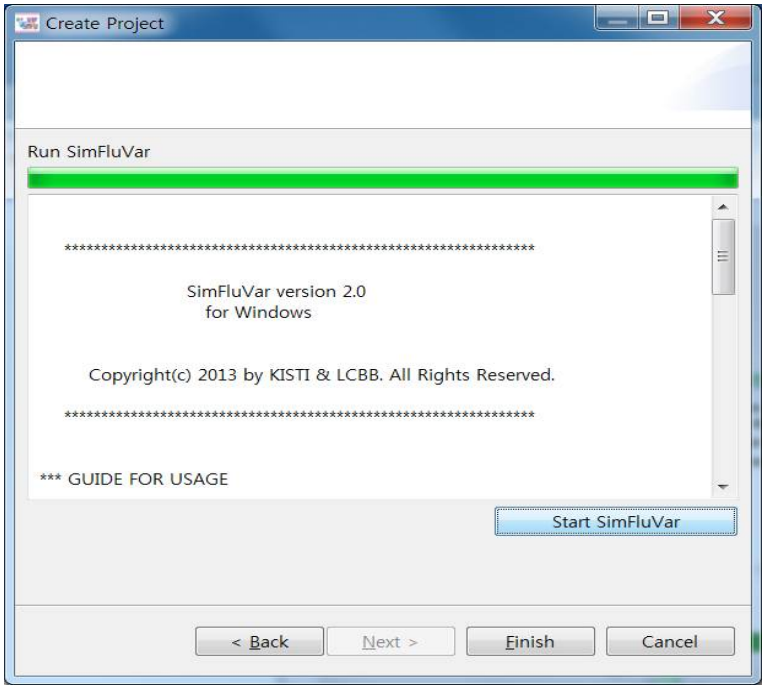
When selecting ‘Project-Crete Project’ from the menu, Project Creation Wizard shown below gets executed.



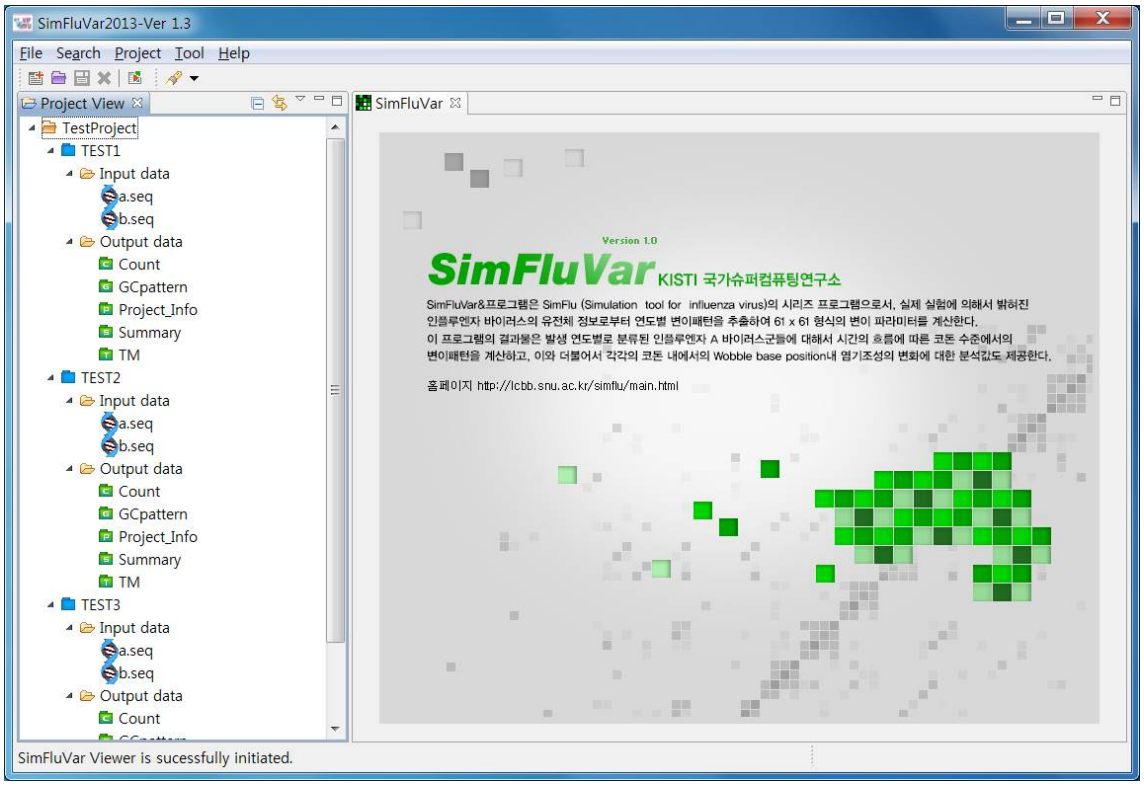
In the above form, enter project name, author, and description and press the Next button to generate output similar to below; it should be noted that having a gap within the project name or leaving any of the slots for author and description blank, the project creation does not get processed at the final stage.



Select the folder containing the input file from above and press the Next button to receive an output file as shown below. Input data need to make up a pair of files (2 total); refer to section, “Program Introduction” for more information.



Selecting ‘Start SimFluVar’ button in the above window will start SimFluVar analysis, and results are displayed in the window upon completion. Next, press the ‘Finish’ button to complete the project creation.

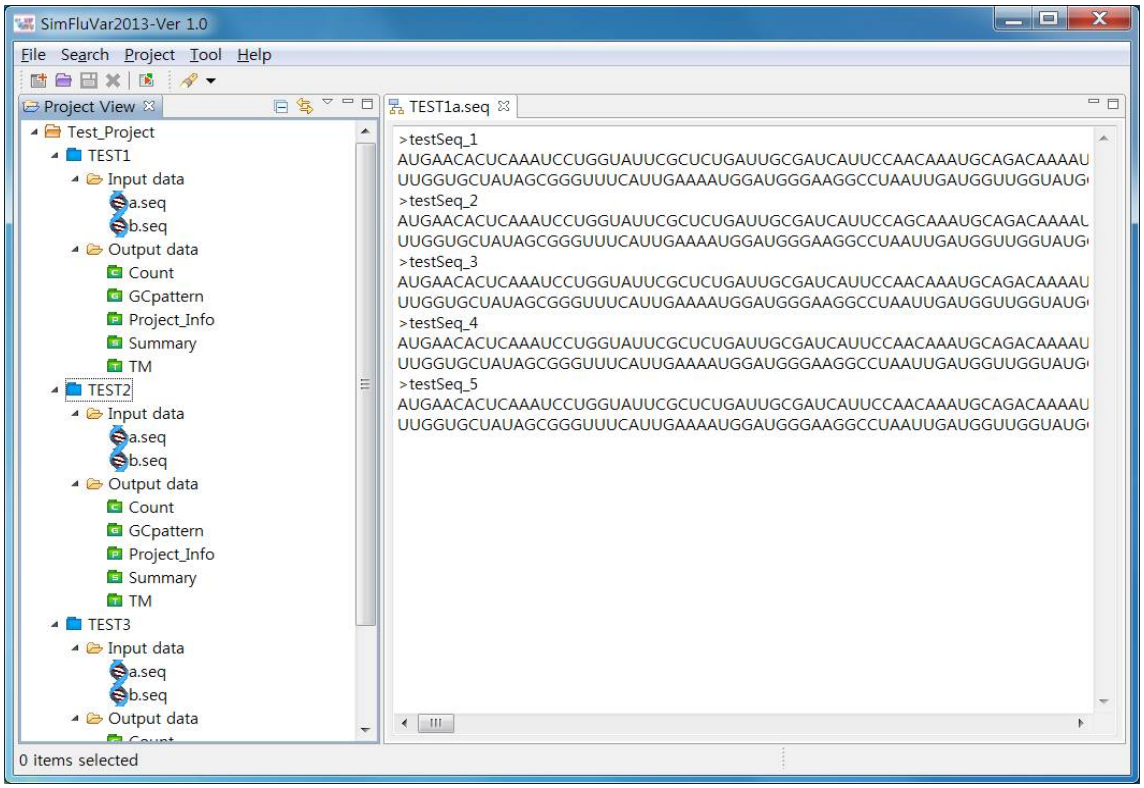


In the ‘Project View’ section on the left hand side of the project screen will emerge a project-related tree, which is mainly divided into ‘Input Data’ and ‘Output Data’. Input Data consists of the input file, while the Output data folder contains the results of Counting matrix, GCPattern, Project info, Summary, and TM (Transition Matrix).

3. Checking Results

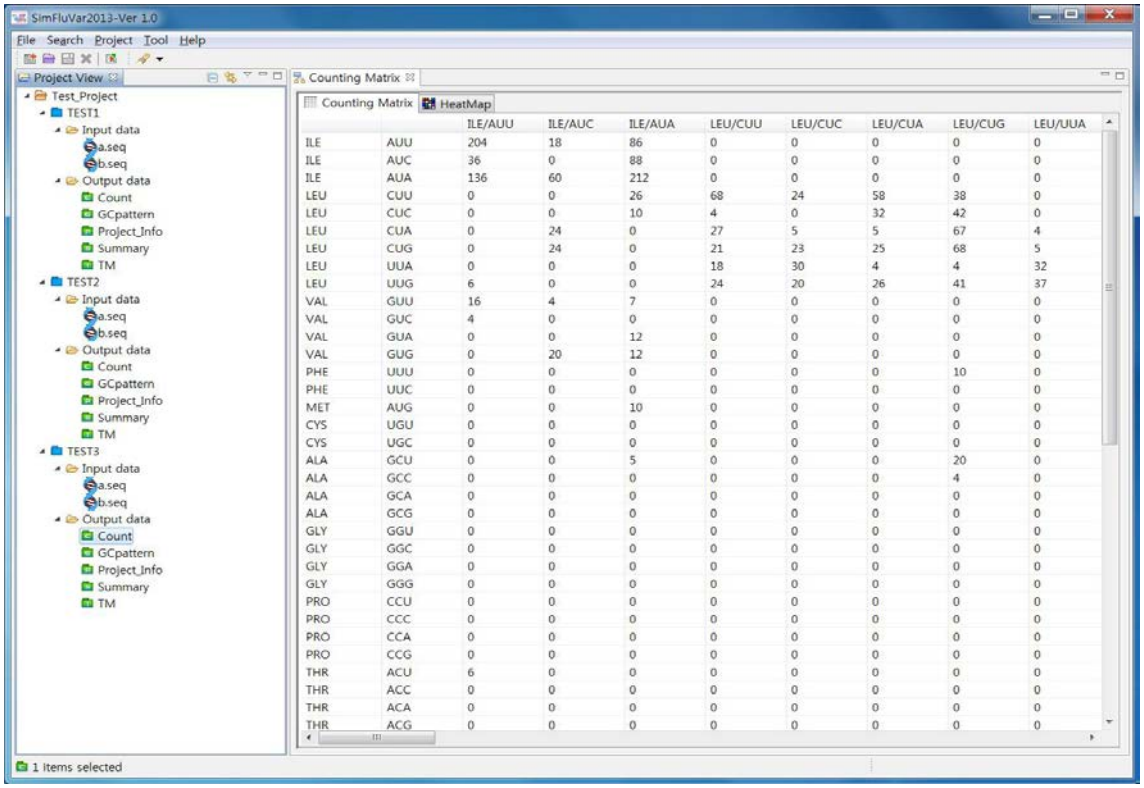
3.1 Input File Output Window

Upon creation of a new project, a node is generated on the tree on the left hand side that indicates input files and output files, where the Input Data folder stores input files. Double-clicking on the DNA-shaped icon will produce a window illustrating input sequence as follows.



3.2 Counting matrix

The ‘Count’ folder among the 5 folders within Output Data folder outputs Counting matrix data. Double-clicking on this folder will generate a screen as shown below; codon in each row is from the initial year, while codon in each column is from the final year.

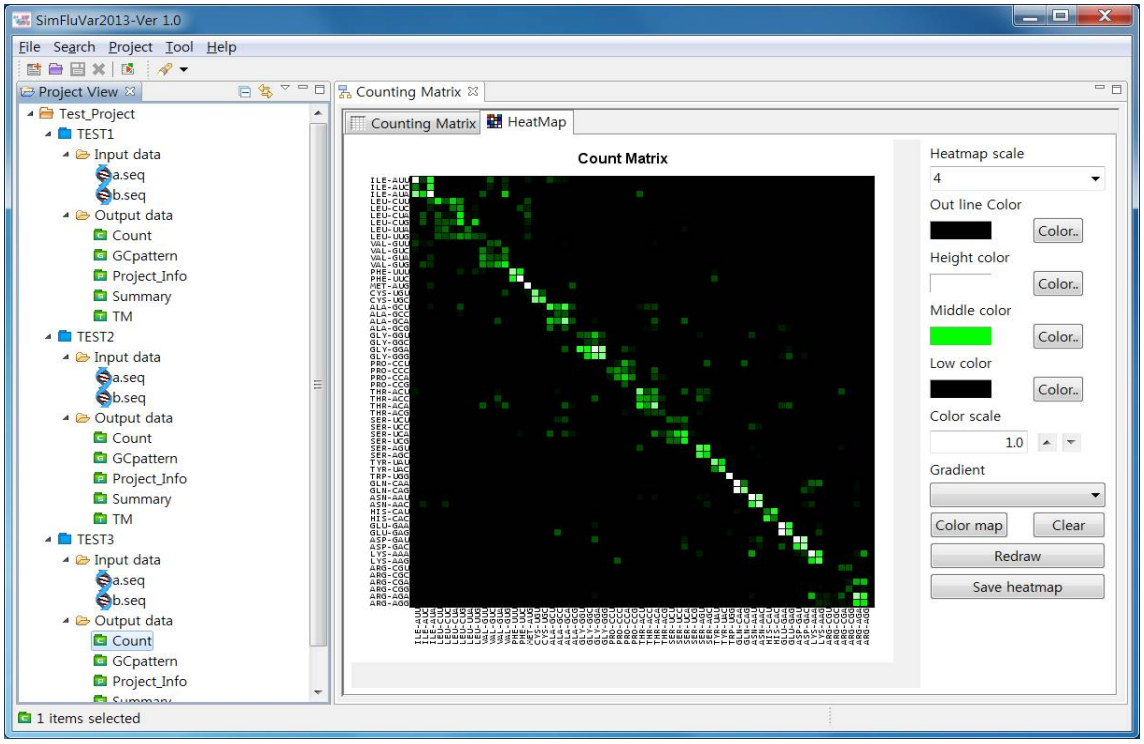


Counting Matrix window consists mainly of 2 tabs: a tab that outputs Counting matrix table in table format, and a tab that outputs graphs in ‘Heat map’ format. Selecting Heat map tab will generate output as follows.

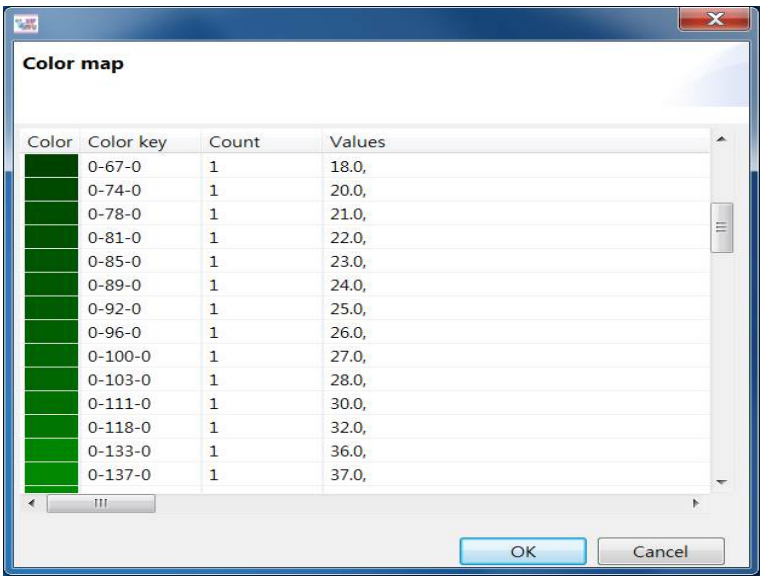
Heat map view at the top outputs matrix-format data in graph format; utilizing the control box on the right hand side enables users to change shape and color of the Heat map.

Using ‘Heat map scale’ function allows users to enlarge or reduce the Heat Map; a high value representing enlarging of the image, while a low value signifies reducing. Select an appropriate value and press the Redraw button to apply the function.

Outline Color function assigns colors to the outline of the little rectangle blocks in Heat map view; High, Middle, Low colors determine how colors are displayed within the Heat map matrix. Basically, the colors in Heat map are determined going from High to Middle to Low, and by this value by pressing the Color button on the right hand side, you can change the color. After the change has been made, press the Redraw button to apply.



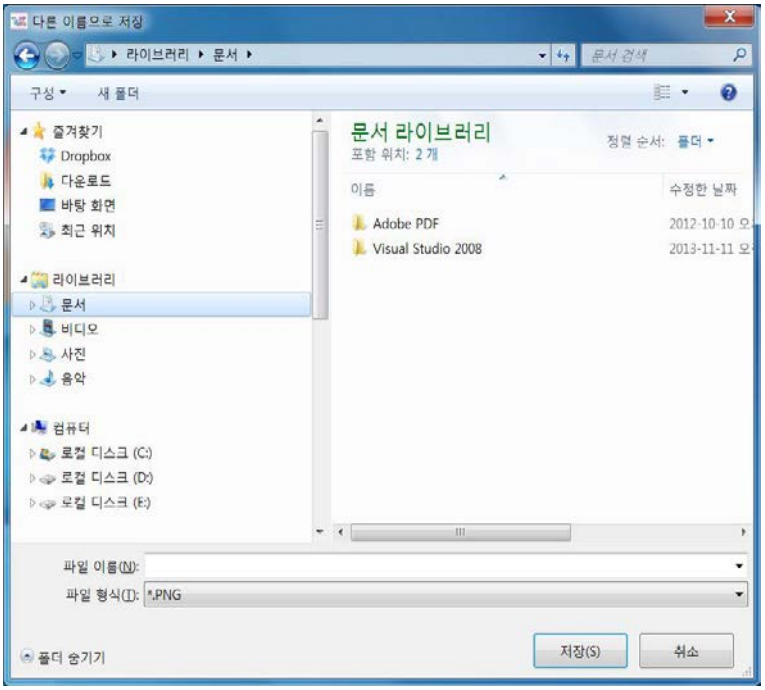
Color scale sets up the scale of color distribution. The closer this value is to 0, the clearer the distinction is between blocks; therefore, to effectively view the subtle distinction between blocks, you can lower this value. Color scale should be set with a value between 0 and 1 and does not support values outside of this range. After the desired scale has been selected, press the Redraw button to apply. Gradient function is grouped combination of preferred values of High, Middle, Low group; select an appropriate combination and press the Redraw button to apply. Pressing the Color map button will produce a dialogue box as shown below.



In the Color map window, ‘Color item’ represents the color of block being output; the ‘Color key’ signifies the RGB value of the Color; ‘Count’ represents number of current values matched with the Color; ‘Values’ signifies actual value of the applicable cell.

In Color map window, designated Color can be changed to different values; in other words, double-clicking on the color box at the top of the Color column, a dialogue will appear that enables users to select the colors. Once new colors have been designating in this box, the selected colors will be changed to the designated ones, which will be reflected in Heat map accordingly. Pressing the Clear button to the right of the Color map button will erase all random selections of colors.

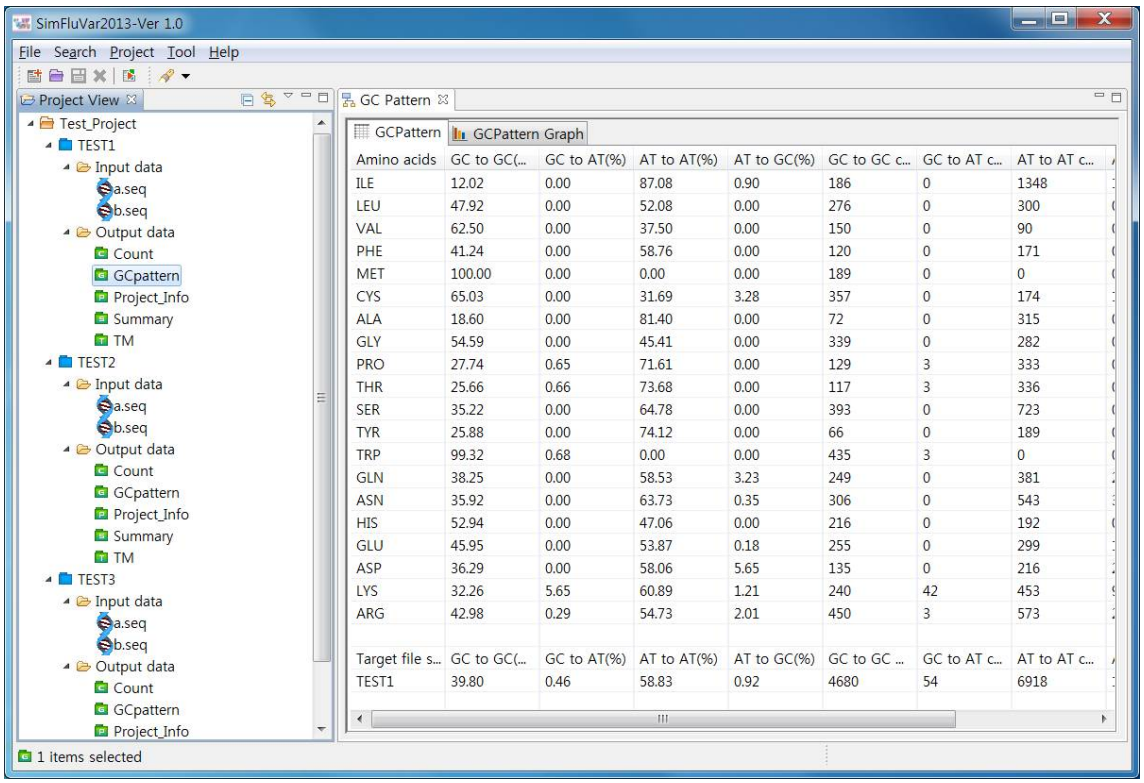
Lastly, save heat map is a function allows users to convert the heat mat matrix on the screen into a specific image file and store it. Pressing the button will produce a dialogue box shown below.



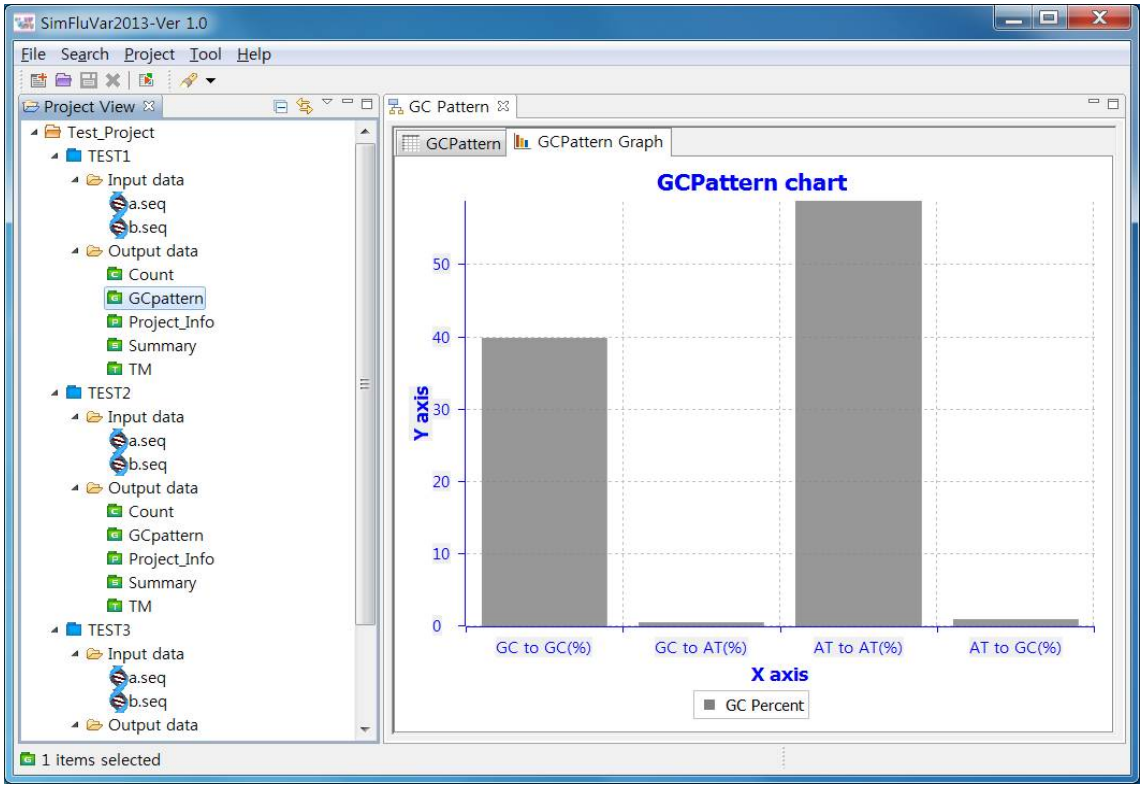
On this screen, you can designate the file format and the file name you wish to use; then, press the Save button to save the designated file; in the current version, 10 file formats are supported including PNG, JPG, BMP, GIF, and PDF.

3.3 GC(%) Pattern matrix

Selecting GCPattern will produce output as shown below. GC(%) Pattern calculates the change in base composition at each codon's wobble codon position and presents it by amino acids encrypted by individual codons. All values are indicated in percentage against whole base, and data counted for proposed percentage will be provided at the end of the percentage. Furthermore, the 'Target file summary' at the bottom is a value calculated holistically instead of dividing initial year's base composition value by amino acids; researchers may obtain and use desired types of results/values.

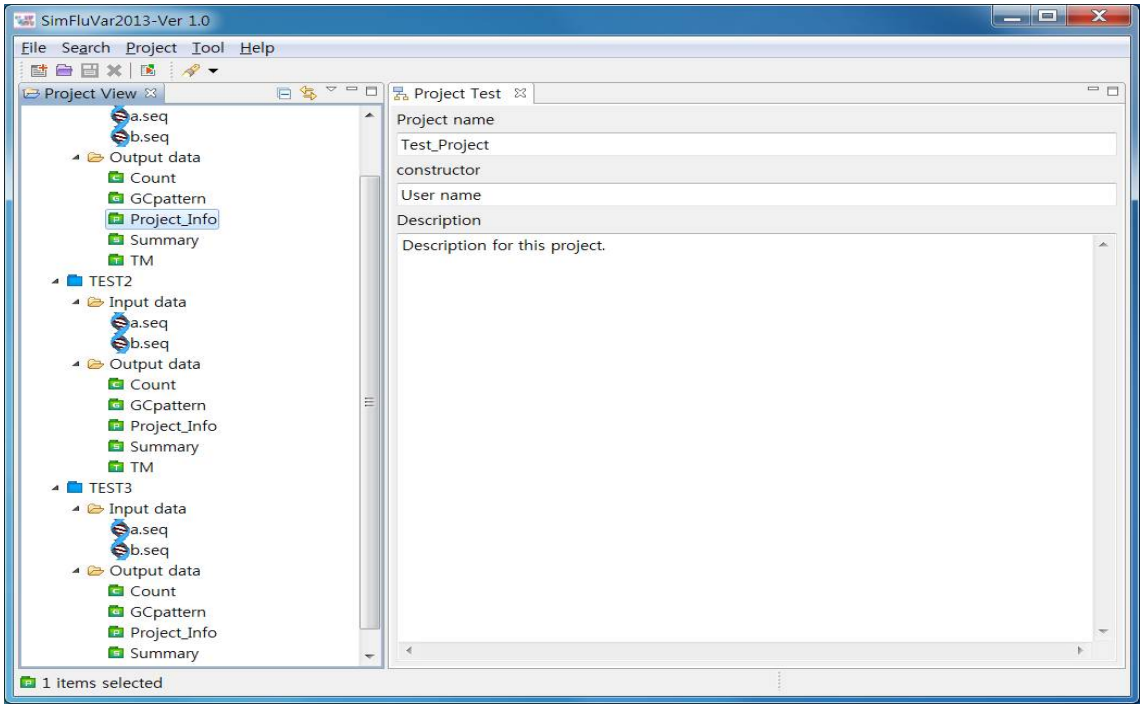


GCPattern view is divided into two views: one that outputs GCPattern matrix in table format and one that outputs the matrix Chart format. The above screenshot is an example of an output in table-format. Selecting GCPattern Graph tab will generate a screen as shown below, allowing users to grasp the overall composition (GC in %) at a glance.



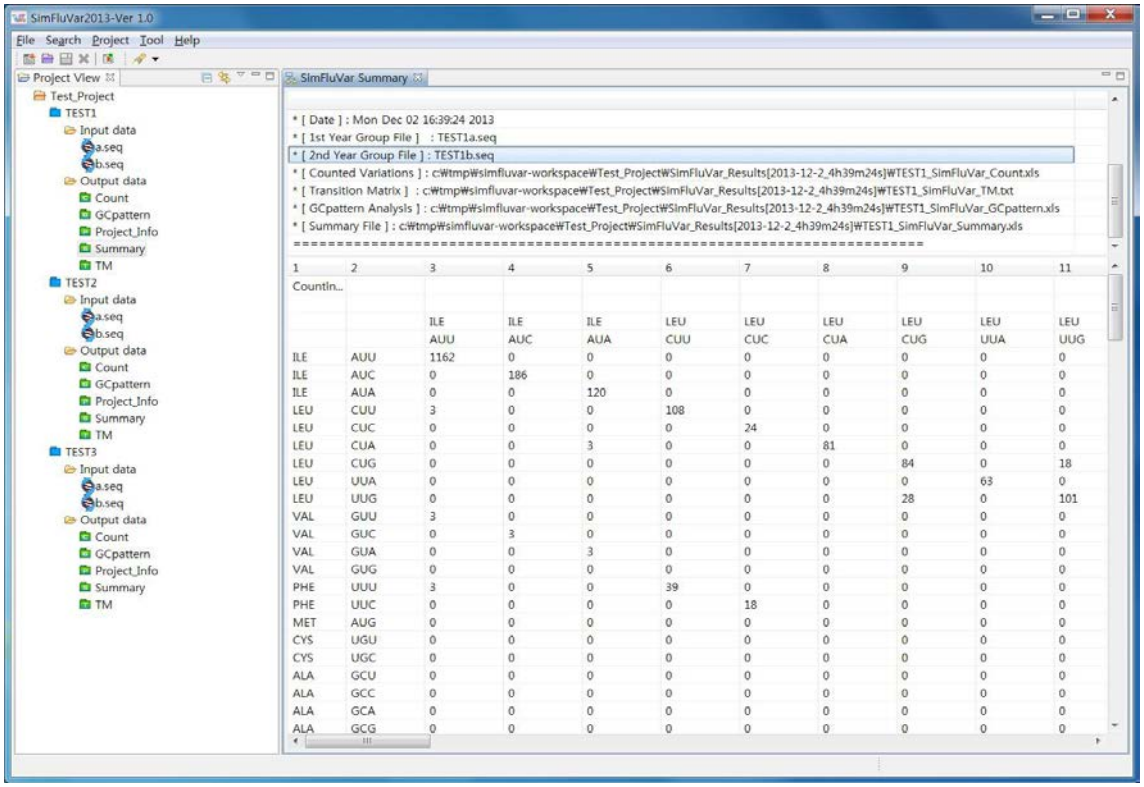
3.4 Project Information (Project_Info)

Project Info view outputs information about the project, double-clicking on Project info node will output the results as follows.



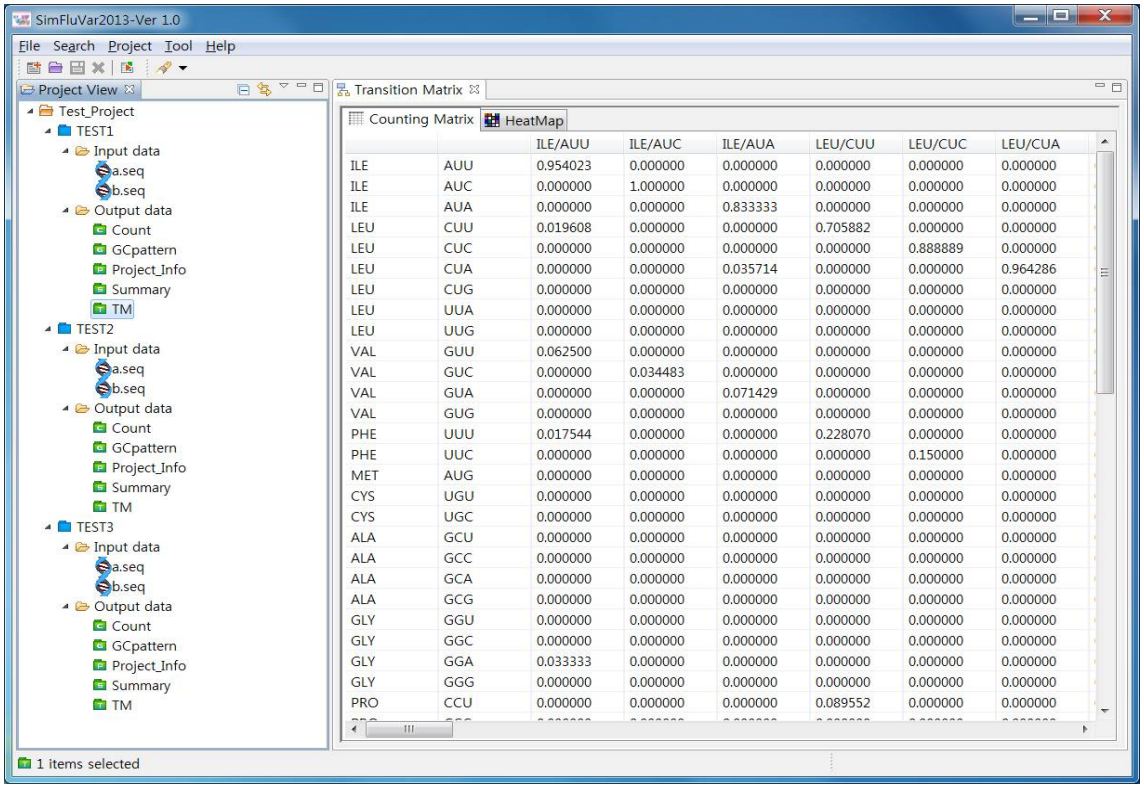
3.5 Summary information

Outputting the Summary node will generate a Summary view as shown below. Summary View outputs the overall results of SimFluVar.



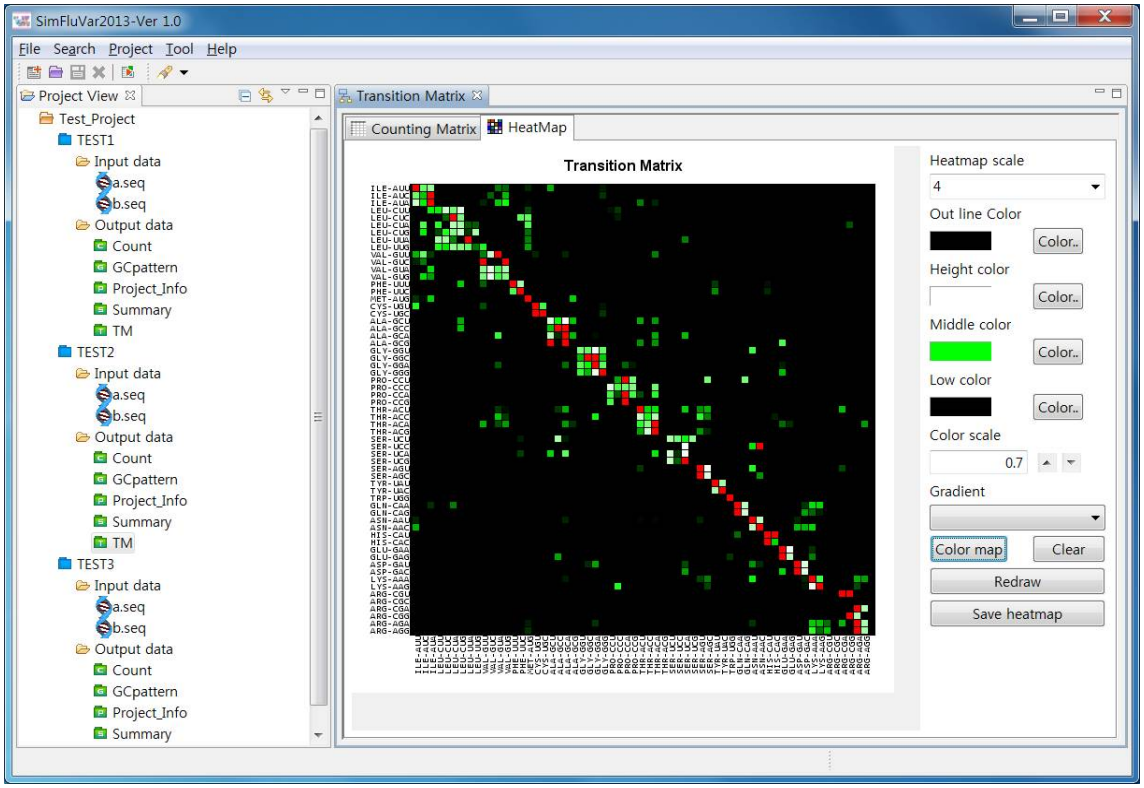
3.6 Transition matrix view (TM)

Double-click on the TM node to output Transition Matrix (TM) view as shown below. TM is a conversion of previous modification value within Counting Matrix to Markov model's Transition matrix format; the roles of matrix's rows and columns are the same as those in Count matrix. The values output from this transition matrix can be utilized as user parameters in the SimFlu-based computer simulation process.



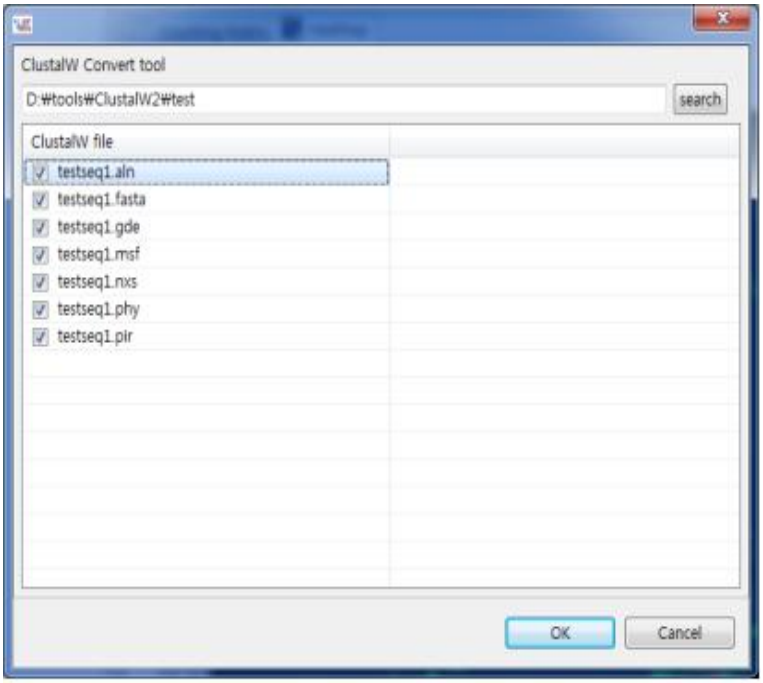
		ILE/AUU	ILE/AUC	ILE/AUA	LEU/CUU	LEU/CUC	LEU/CUA
ILE	AUU	0.954023	0.000000	0.000000	0.000000	0.000000	0.000000
ILE	AUC	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000
ILE	AUA	0.000000	0.000000	0.833333	0.000000	0.000000	0.000000
LEU	CUU	0.019608	0.000000	0.000000	0.705882	0.000000	0.000000
LEU	CUC	0.000000	0.000000	0.000000	0.000000	0.888889	0.000000
LEU	CUA	0.000000	0.000000	0.035714	0.000000	0.000000	0.964286
LEU	CUG	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
LEU	UUA	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
LEU	UUG	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
VAL	GUU	0.062500	0.000000	0.000000	0.000000	0.000000	0.000000
VAL	GUC	0.000000	0.034483	0.000000	0.000000	0.000000	0.000000
VAL	GUA	0.000000	0.000000	0.071429	0.000000	0.000000	0.000000
VAL	GUG	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
PHE	UUU	0.017544	0.000000	0.000000	0.228070	0.000000	0.000000
PHE	UUC	0.000000	0.000000	0.000000	0.000000	0.150000	0.000000
MET	AUG	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
CYS	UGU	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
CYS	UGC	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
ALA	GCU	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
ALA	GCC	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
ALA	GCA	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
ALA	GCG	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
GLY	GGU	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
GLY	GGC	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
GLY	GGA	0.033333	0.000000	0.000000	0.000000	0.000000	0.000000
GLY	GGG	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
PRO	CCU	0.000000	0.000000	0.000000	0.089552	0.000000	0.000000

Similar to Count matrix view, TM view also consists of two views; one outputs the matrix in table format, while the other outputs it in Heat map format. Selecting Heat map tab will generate an output as shown as below, and the basic operation is the same as Heat map as described in Count matrix view section.



4. ClustalW Convert Tool

Within the SimFluVar program, there is a conversion tool that changes ClustalW's result files to its own input file. Selecting 'Tool-ClustalW to SimFluvar' in the menu produces a window as shown below. Next, press the Search button to select the folder containing ClustalW's output file; then, generate all output files created in ClustalW as shown below. From there, select the desired file and press the 'OK'(확인) button to convert the selected file to SimFluVar's input file.



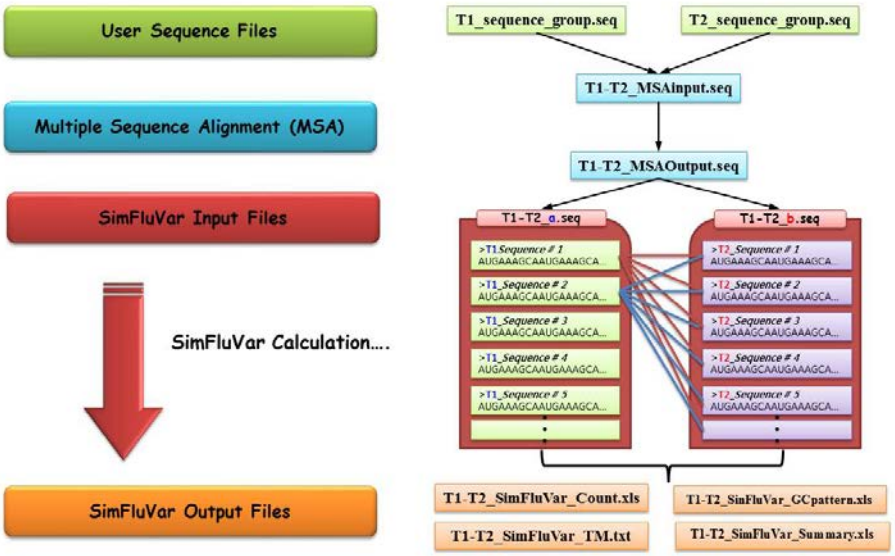
5. Saving/Reading Projects

Click on the project name from the project tree of SimFluVar program and select ‘File>Save Project’ menu to save the selected project as the designated file, which gets stored in ‘*.svf’ format. This file can be read in project format when you select File-Load Project menu.

The 4 types of files produced from completing all these steps are generated automatically with a particular suffix added to the input file name. First, ‘*_SimFluVar_Summary.xls’ file allows users to view the analysis process by providing basic information as the following: count produced from SimFluVar program’s anaylsis phases; all input files used in execution of SimFluVar program and their route; total execution time; transition matrix values. SimFluVar program. Moreover, in order to grasp the change in base composition of wobble codon position within the codons in question, respective analysis data and tabularized results will be provided. Also, ‘*_SimFluVar_Count.xls’, ‘*_SimFluVar_GCpattern.xls’ and ‘*_SimFluVar_TM.txt’ files are provided separately from data provided in the Summary file; among these, ‘*_ SimFluVar _TM.txt’ file may be used through SimFlu program as a user parameter file in computer simulation of influenza virus without requiring extra work, thus promoting the efficiency of linking with SimFlu program. All results files get saved in the applicable project name folder within the same folder SimFluVar program is installed, “simfluvar_workspace” folder.

6. Analytical Process

SimFluVar program can be utilized for computer simulation or to analyze gene modification patterns that exist among time-series influenza virus outbreak groups. Detailed gene modification computation process is as follows.



First, to activate SimFluVar program, ‘multiple sequence alignment’ (MSA) needs to be executed between sequences from 2 subject years. For the example illustrated above, sequences for time ‘T1’ and time ‘T2’ were named ‘T1_sequence_group.seq’ and ‘T2_sequence_group.seq’, respectively.

The two files collected by the user implement MSA using other MSA programs such as ClustalW; for the above example, a file combining sequences of T1 and T2, ‘T1-T2_MSAinput.seq’, was generated and used as input file for MSA. Subsequently, the result file (‘T1-T2_MSAOutput.seq’) containing ‘gap’ from completed sequences must be in FASTA file format, and there are no special restrictions on sequences’ title annotations.

Following the above steps, and if the MSA process for T1 and T2 groups has been successfully executed, it can be construed that the preparation required for activating SimFluVar program is completed.

To start SimFluVar, the result file containing gap generated from MSA needs to be divided into two separate files, each containing individual time, T1 and T2; it should be noted that as SimFluVar automatically searches and registers files with ‘*.seq’ extension in the [user_input] folder within the user-designated folder, the user must

create 2 files using the file extension.

For the above example, in which the file was separated into ‘T1-T2_a.seq’ and ‘T1-T2_b.seq’, the suffix ‘a’ and ‘b’ before the file extension should be noted. When searching input files, SimFluVar program looks for *.seq format files; it then recognizes files with the same file name before the extension as a pair. In other words, the two files returned from searching a file name ‘T1-T2_’ become a pair, and the initial outbreak year and the final outbreak year are distinguished by this suffix; the suffix ‘a’ represents the initial time group (time T1 in the example), while the suffix ‘b’ represents the final time group (time T2 in the example).

Once the input pair gets recognized and a sequential context is established, SimFluVar grasps modification patterns of each codon location by comparing all sequences of T1 time slot with all sequences of T2 time slot at the corresponding codon level. In other words, as illustrated in the above diagram, the sequences of T1 time slot (T1_Sequence #1, #2, …, #*n*, where *n* represents total the number of sequences in T1 time slot) and the sequences of T2 time slot (T2_Sequence #1, #2, …, #*p*, where *p* represents total the number of sequences in T2 time slot) make up a Counting Matrix by measuring the modifications of codons at each codon location; at this time, as the sequences of T1 time slot and those of T2 time slot are individually compared, resulting in n x p instances of comparison. Subsequently, Counting Matrix generated from outbreaks in consecutive years is converted to Markov model’s Transition Matrix (TM).