Laboratory of Natural Information Processing
DA-IICT Gandhinagar

Biospectrogram

User Manual

BIOSPECTROGRAM

User Manual

Laboratory of Natural Information Processing
DA-IICT, Gandhinagar, Gujarat 382007
http://www.guptalab.org/biospectrogram

Documentation version 1.0

This file last modified on September 28, 2012.

Credits & Team

Principle Investigator: Manish K. Gupta, PhD.
Key Developers: Nilay Chheda* and Naman Turakhia*
Supporting Developers: Ruchin Shah & Jigar Raisinghani
Software Logo: Hiren Kangad

* Key developers contributed equally to the project

Acknowledgments

We thank Deeksha Gupta for useful discussion in the early stage of the project. All the icons are taken from the internet from the link "http://icons.mysitemyway.com/category/yellow-road-sign-icons/" These icons are free to use for personal as well as commercial use "http://icons.mysitemyway.com/terms-of-use/" MATLAB is registered trademark of Mathworks, USA. For FFT java script we thank Silvere Martin-Michiellot and also to Craig A. Lindley whose open source code has been used with modifications.

Table of contents

1 Introduction

Molecular biology has produced a vast amount of digital data in last decade. There is a need to understand the genome of various species and plants. Mathematics, Computers Science and Statistics are playing a major role in understand the data. One of the new branches has emerged called genomic signal processing which employs digital signal processing (DSP) techniques to study the spectral properties of the biological data and answer some biological questions. This is done by converting the biological (DNA or protein) data into numerical data so that a DSP technique can be applied for its analysis. Biospectrgram is open-source software to facilitate this process. It applies 23 well-known encodings on the biological data and also performs 6 transformations. Using the user choice encodings, random encodings and other transformations and filters available in MATLAB one can do tremendous spectral analysis. This document is prepared to give users an overview of bio spectrogram software, utilities available in bio spectrogram and motive behind the development of the software.

Entire software can be well understood by understanding its four major functionalities. (See Figure 1)

1. Data Collector (Fetch/Import)
2. Encode
3. Transformation
4. Export

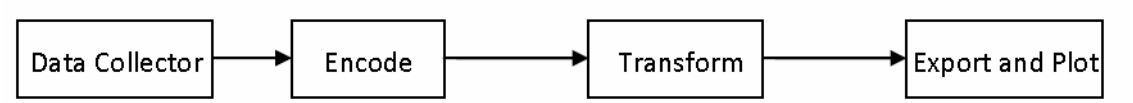

Figure 1: Basic Building Blocks of Biospectrogram

Data Collector fetches the data from National Center for Biotechnology Information (NCBI) server or a user can also import data from his own machine/network. Encoder module provides 20 different encodings on DNA data and 3 encodings on Protein data. User can apply following transformation in Transformation module on the encoded data. The relationship between encodings and transformations are given in Figure 2. The solid arrows represent transformations available in biospectrogram and broken arrow represents the transformation that can be applied by MATLAB by exporting the encoded files from biospectrogram. User can also export the (both encoded and transformed) files to MATLAB files for plotting and further analysis.

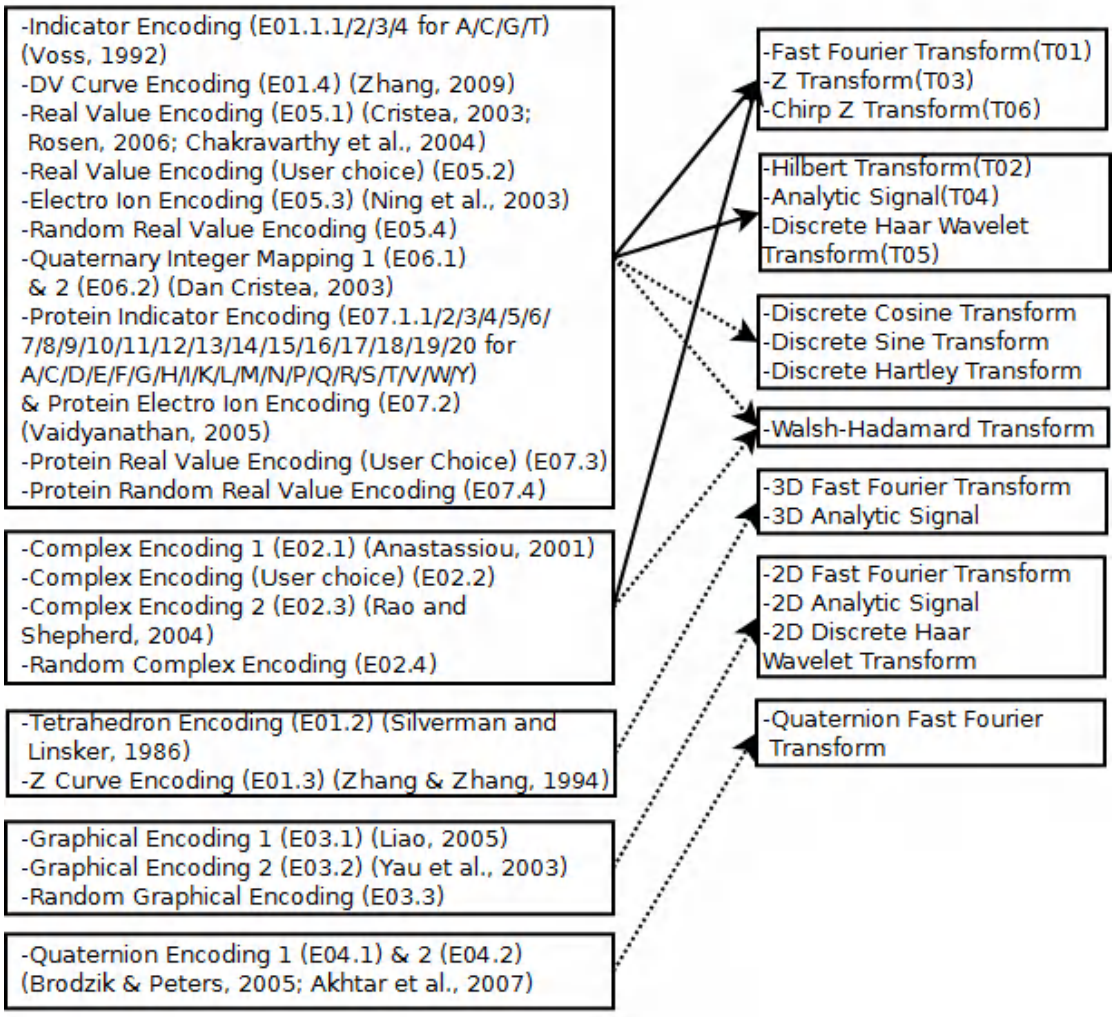

Figure 2: Basic Relationships of Encodings and Transformations

Transformations that have been implemented in Biospectrogram are listed below:

1. Fast Fourier transform (T01)
2. Hilbert Transform (T02)
3. Z transform (T03)
4. Analytic Signal (T04)
5. DHWavelet (T05)
6. Chirp Z Transform (T06)

2 Fetch

"Fetch" forms very fundamental block of the software. We are fetching data from the online database of DNA. Entire database is hosted online by the NCBI. There is a utility provided by NCBI called "entrez" which helps to download DNA or protein files based on some accession number from Nucleotide or Protein databases. This utility works by performing certain queries on those databases. An example of such query is shown below:
http://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=protein&rettype=gb&retmode=text&id

Here, "Eutils" comes under entrez utilities provided by the "NCBI". We are using specifically "Efetch". Main query starts after the "?" in the address. We need to provide some attributes in the query. First attribute we are giving is the "db" which stands for database. We have selected protein database. Then retype stands for retrieval type which is used to specify the file format you want the DNA file in. Here we have given "gb" which stands for genbank file format. Similarly "retmode" stands for retrieval mode which can be given as text or xml.

Very important attribute is the id where we provide the accession number of particular genome. Each and every record of these databases is given unique accession number. Annexure-1 lists some popular genomes with their accession numbers. Figure 3 shows the screen shot of bio spectrogram. When the application starts it shows the most recent fetched file in the upper pane. If you are using for the first time or if there are no files in your history then you will see the blank pane.

Figure 3: Basic Graphical User Interface (GUI) snapshot of Biospectrogram

Figure 4 shows the screenshot of the application when fifth button indicating, "fetch" is pressed. You can see that the button is slightly highlighted. When user presses that button, first an information dialog box opens up describing e-fetch policy of the NCBI database and then a popup appears asking for valid accession number.

Before downloading any sequence from the internet, our application checks user's internet connectivity, validity of accession number entered and already existing file being requested to download.

Once all the validation is done, software starts downloading DNA files. Figure 5 is showing the screenshot of the state when files are being downloaded. There is an indicator that shows how much percentage of files is already downloaded. This way user can keep track of how many percentage of file is already downloaded which is really useful information in case when files are really big in size or the internet connection is very slow.

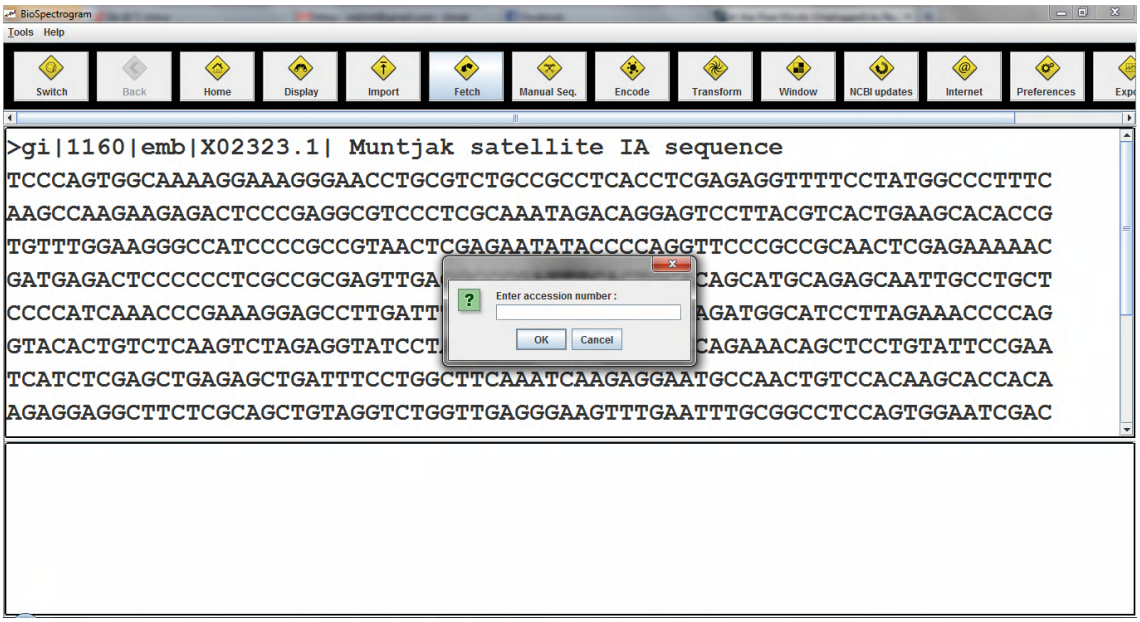

Figure 4: Snapshot of pressing "fetch" button on Biospectrogram

If user press "CANCEL" while files are being downloaded, all downloads will be stopped and incomplete files will be deleted.

Figure 5: Status of Biospectrogram when it is downloading a file from NCBI

On successful download of "fasta" and "genbank" file, software shows the prompt acknowledging that operation has been successful as shown in Figure 6. As soon as download is over, newly downloaded fasta file is shown in the upper pane which acts as an input displaying pane. Additionally, very first button saying "BACK" is also enabled once more than one files are in use after software is started.

Figure 6: Biospectrogram showing the dialog when file from NCBI has been downloaded

There are two more ways to put fasta files in history of bio spectrogram. One of those ways is to import a fasta file from local or network location of a system running our application. Another way is to just enter a sequence directly in an input box.

3 Import a file

Once the button saying "Import" is pressed, a file chooser dialog opens up as shown in Figure 7.

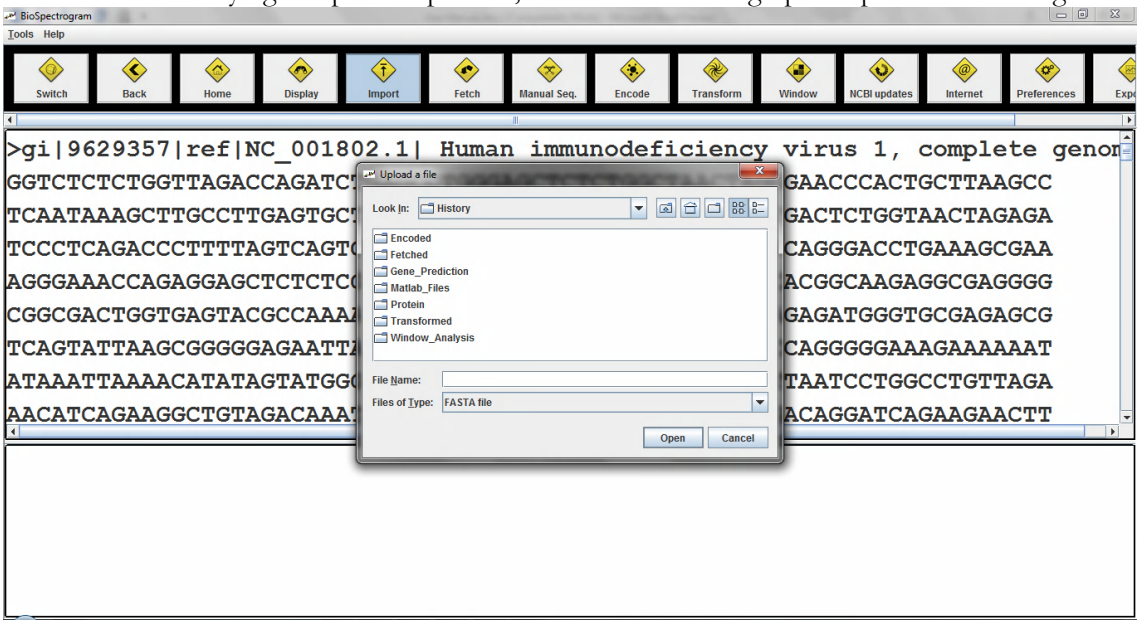

Figure 7: Snapshot showing the dialog after pressing import button

Once user selects a file with valid extensions like .fasta, .fa, .fna, .fsa, or .mpfa, it will be added to the user's fetched history folder and on successful addition of upload application will show a confirmation message as shown in Figure 8. These all formats will be converted to fasta format first before uploading to user's history folder.

Figure 8: Snapshot showing file importing in History folder

4 Manual Sequence

On pressing the button saying "Manual Seq.", a dialog box open up asking user to enter a sequence manually as shown in Figure 9.

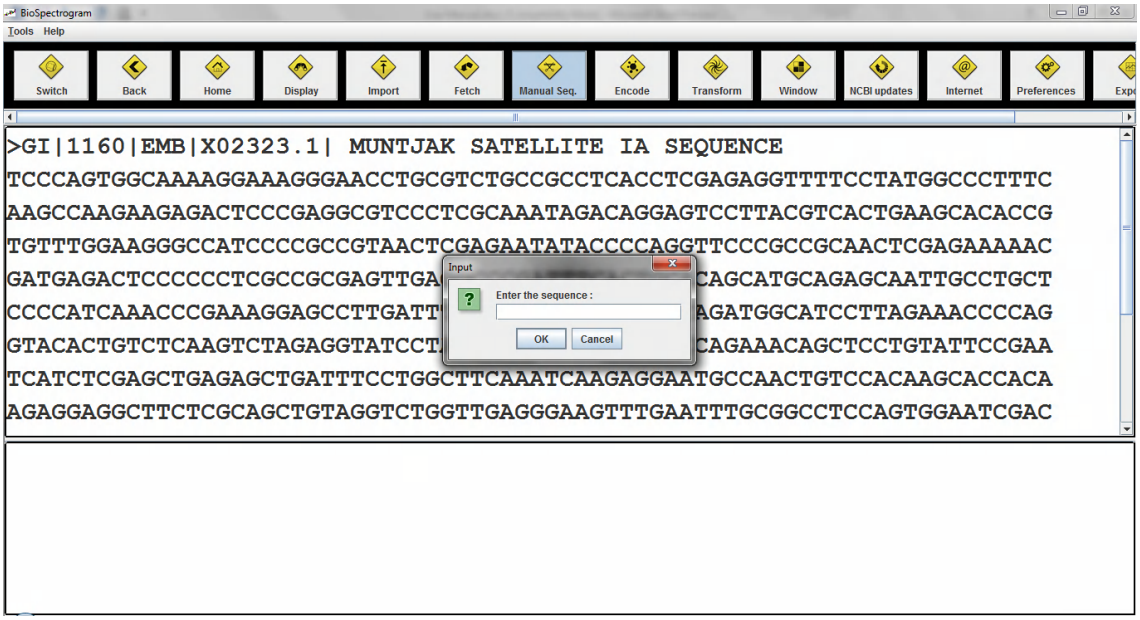

Figure 9: Importing a customized sequnece (copy paste) via Manual sequence Button

User needs to copy a valid sequence (with valid characters only) and paste in the input box and press enter. After pressing OK, user will be asked to enter a name for the sequence entered. On giving valid name, file will be created in the user's history folder and it will be displayed in the upper pane of the application as shown in Figure 10. This feature can also be used to copy & paste a portion of genomic data from any file.

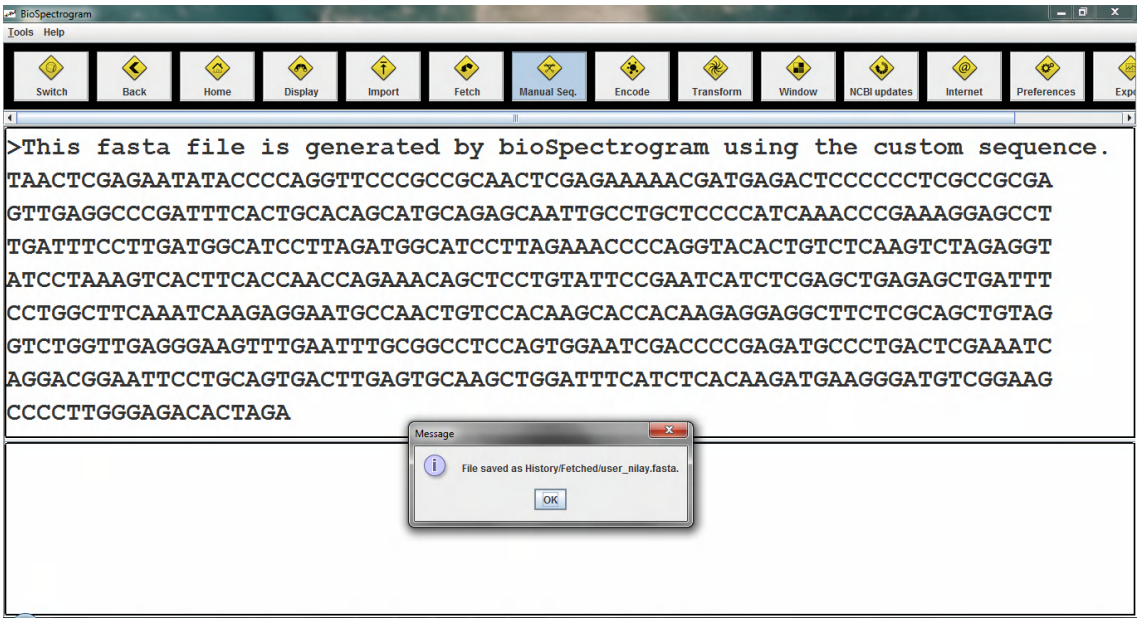

Figure 10: Snapshot showing customized sequence is saved in History folder

This completes our very first phase of fetching. All other modules are dependent on the fetch module because in encodings and transformation user has to have fetched files in his history.

5 Encode

Encoding is used to map the symbols of DNA or protein sequence to the different spaces like real numbers, complex numbers etc. When user clicks on the button saying "Encode", a dialog appears showing two different dropdown menus. User has to select the fetched file from the first dropdown list and encoding scheme from the second dropdown list as shown in the screenshot in Figure 11. Initially no items are selected in any of two lists as shown in the first screenshot. We can see that the third button is slightly highlighted indicating it is pressed. Here we have shown the screen shot of both the dropdown menu. When user download any new sequence from internet the list automatically get updated. User can also set the upper limit of files to be kept in the history. We are showing second dropdown menu that contains all the names of encoding scheme.

Figure 11: Snapshot showing encodings dropdown list

In Figure 12 & 13, we have shown outputs for two different encoding schemes. One of them is "Random Complex Number" Encoding scheme and the other is "Real Value Encoding". Both of these encoding schemes together cover almost all the cases that may arise during any of the other encoding scheme. These encodings are applied on the fasta file with accession number X07654 as shown. Most of the other encoding scheme also follows same methodology. One of the variant of real value encoding requires a user to give certain inputs. These inputs can be entered in the input box which will be generated as and when required.

Figure 12: Snapshot showing random complex encoding chosen

As shown in Figure 12 & 13, random complex encoding generates some random complex numbers by its own and afterwards on clicking "OK", encoding is being done by the application in the back. After successfully encoding all the characters, encoded file is created in the encoded

history folder and written with the relevant content. Finally, encoded output is displayed in the second pane of the application as shown below which is usually used for displaying different types of output at different points of time.

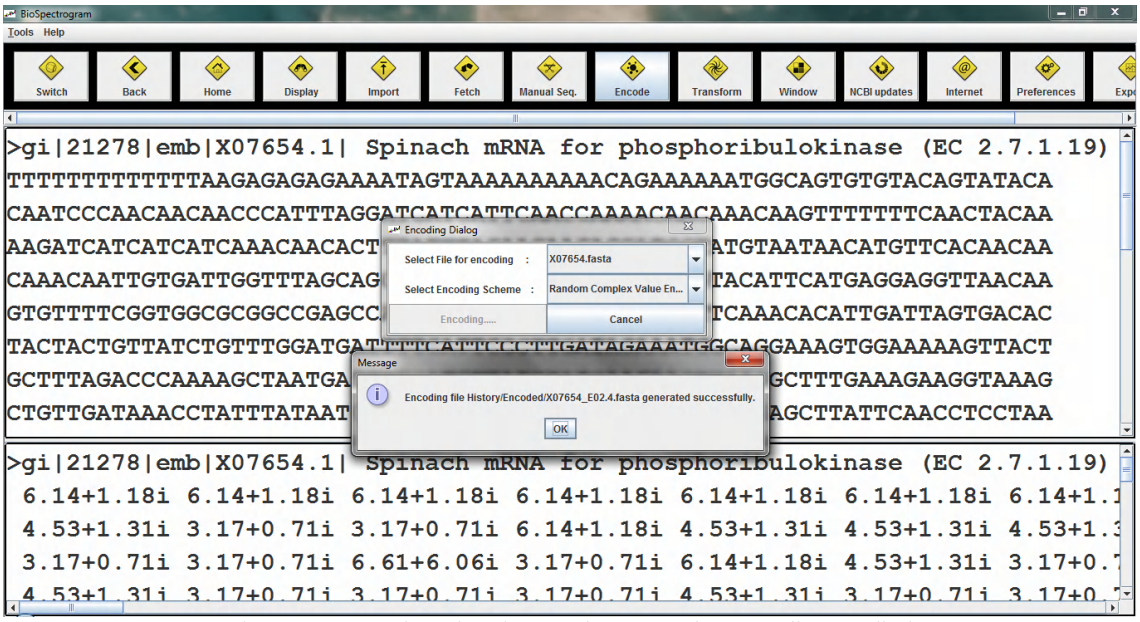

Figure 13: Snapshot showing random complex encoding applied

Now we will walk you through the procedure of real value encoding for fasta file X07654.

Figure 14: Snapshot showing user choice real value encoding

First of all press “encode” button then select one of the fetched files from the first dropdown menu and the encoding scheme from the other dropdown menu. On pressing enter, screen as shown in Figure 14 will appear. It asks whether user wants to give enter real values for encoding. Here user is supposed to enter a character ‘y’ or ‘n’. Input is not case sensitive. After entering ‘y’

and pressing enter, application will ask for four real values for four characters of DNA sequence. One of those dialogs is shown below.

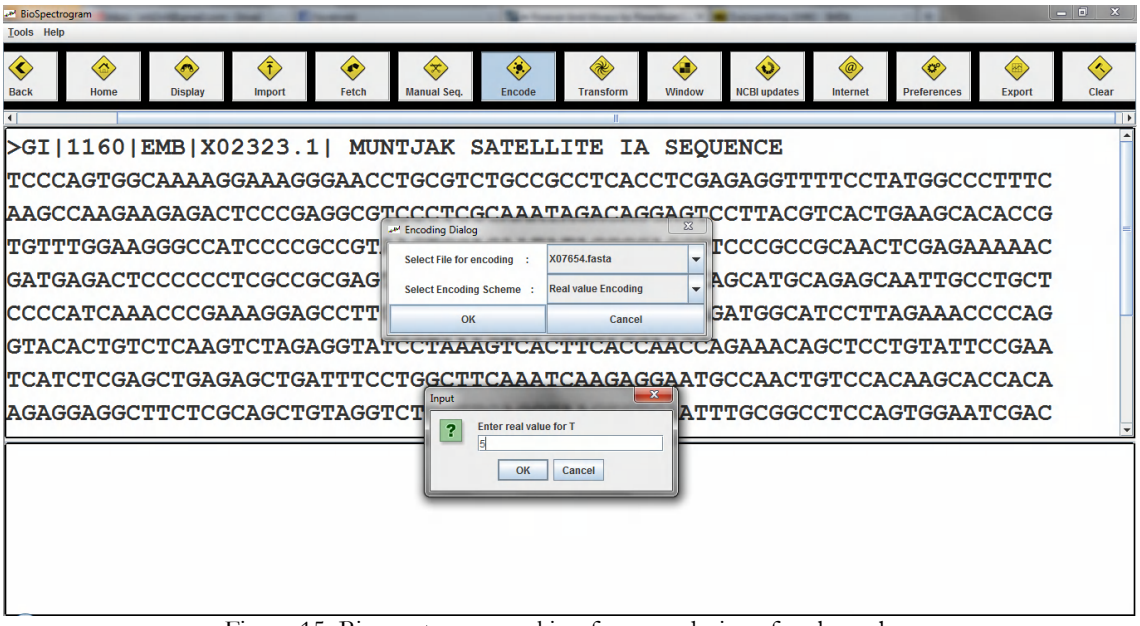

Figure 15: Biospectrogram asking for user choice of real number

Once all the inputs are given, encoding is done using the given input. On successful completion of encoding, a screen as shown in Figure 16 appears with confirmation dialog and encoded file in the output display pane.

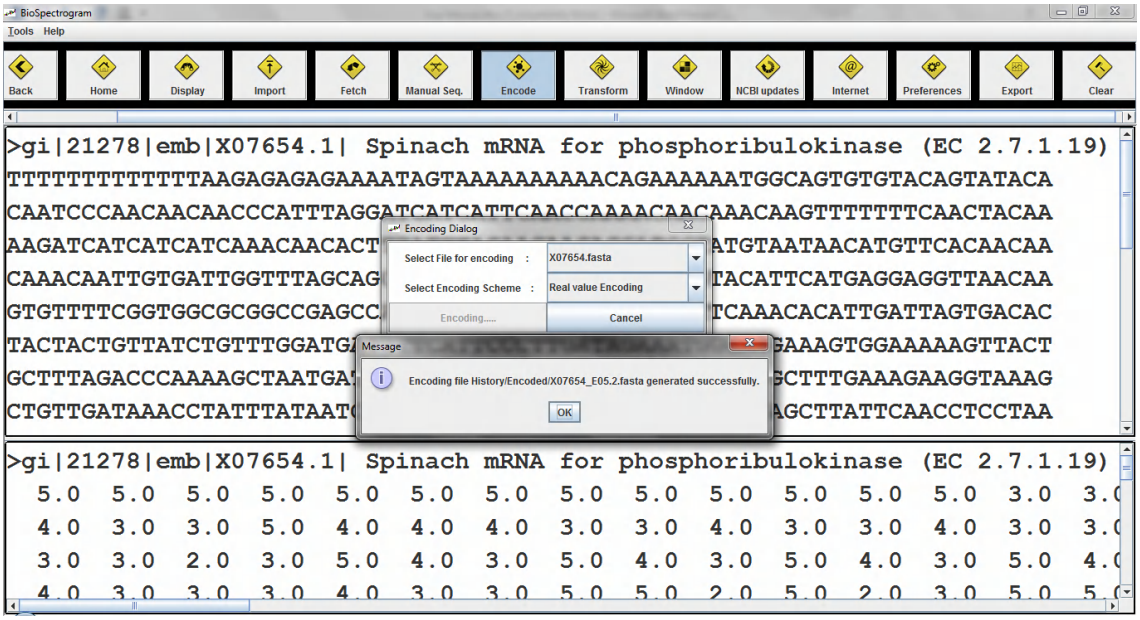

Figure 16: Snapshot showing user choice real value encoding completed

6 Transform

Transformation is the third and a key feature of our application. All the files that are encoded and converted in some number format from symbolic DNA or Protein sequence can be

analyzed by applying different signal processing transformations like Fast Fourier Transform, Hilbert Transform etc.

In Figure 17, we have shown the screenshot of the application when button for transformation is pressed. A dialog box similar to one in the encoding part appears with two dropdown menu. One for selecting encoded file and the other for selecting the transformation scheme. Number of files to be kept in the encoded list, can also be controlled by the user.

Figure 17: Snapshot for transformation button

Now, we will use fetched file with accession number X07654 and encoded file with encoding Random Complex Encoding that will have name, "X07654_E02.4.fasta" (See Figure 2 for corresponding codes). Figure 18 shows the screenshot of the final transformation output. Transformed file is created in transformed directory with proper name as described in the coding conventions for transformations.

Some of the transformation cannot be used with certain encoded files. If user tries to do that, an error message will appear saying that this transformation scheme cannot be applied to specified file (Figure 2 shows about which transformation can be applied to which encoding).

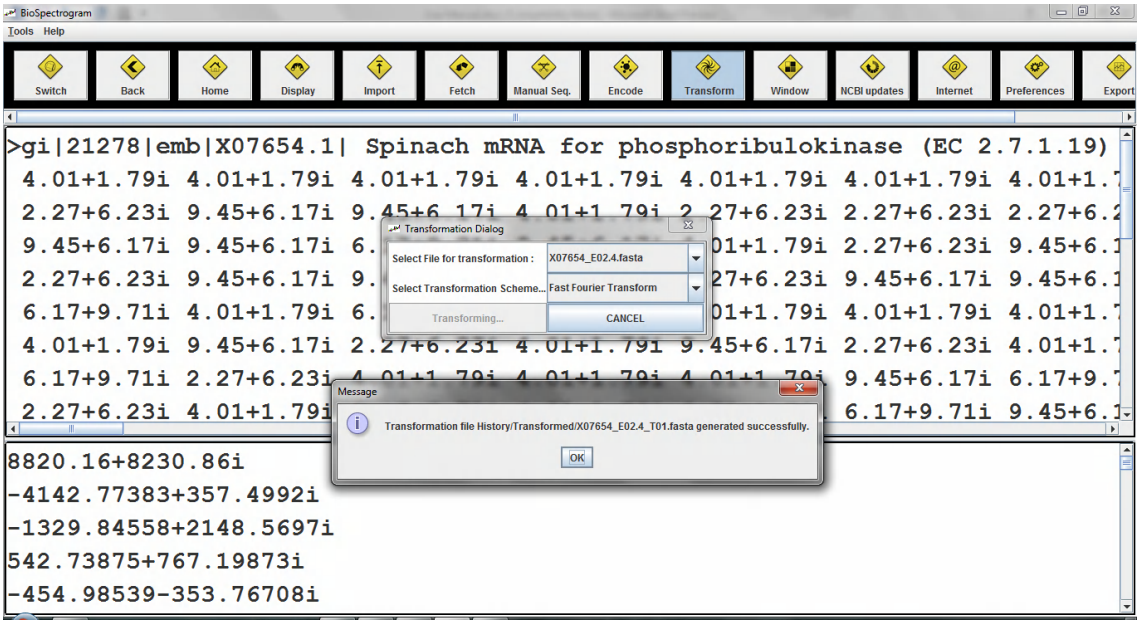

Figure 18: Screenshot of the final transformation output

7 Export to MATLAB

Export to MATLBA is our feature which creates a bridge between our application and MATLAB. This feature is developed in the place of "Plot" functionality. Since we have all the output saved in proper format, we can simply export it to the formats which can be plotted and analyzed in some other tools like MATLAB.

Using this feature, user can create MATLAB script files to plot. As shown in the screenshot in Figure 19, when a user clicks on Export to MATLAB button a dialog pops up showing one dropdown menu. This pop up menu contains list of files which have been created after applying some encoding and transformations. User can choose transformation file from there and click "OK" button. This will create new script file with the name same as transformed file and extension ".m".

All the ".m" files are saved in the directory "/History/Matlab_Files". Same function can be called from Tools Menu as well. User can click on "Tools->Export to Matlab" and get the same dialog box for creating MATLAB script files.

As shown in the Figure 19, MATLAB files can be generated for encoded and transformed files both. By selecting either of them, a simple file selector dialog will appear in which user can simply select any file and click "OK" to create MATLAB script file.

Figure 19: Screenshot of MATLAB export button

After successful creation of file, screenshot as shown in Figure 20 will appear with confirmation dialog. Since MATLAB does not accept file names with "." in the name, generated ".m" file will replace all the dots (.) with underscore (_).

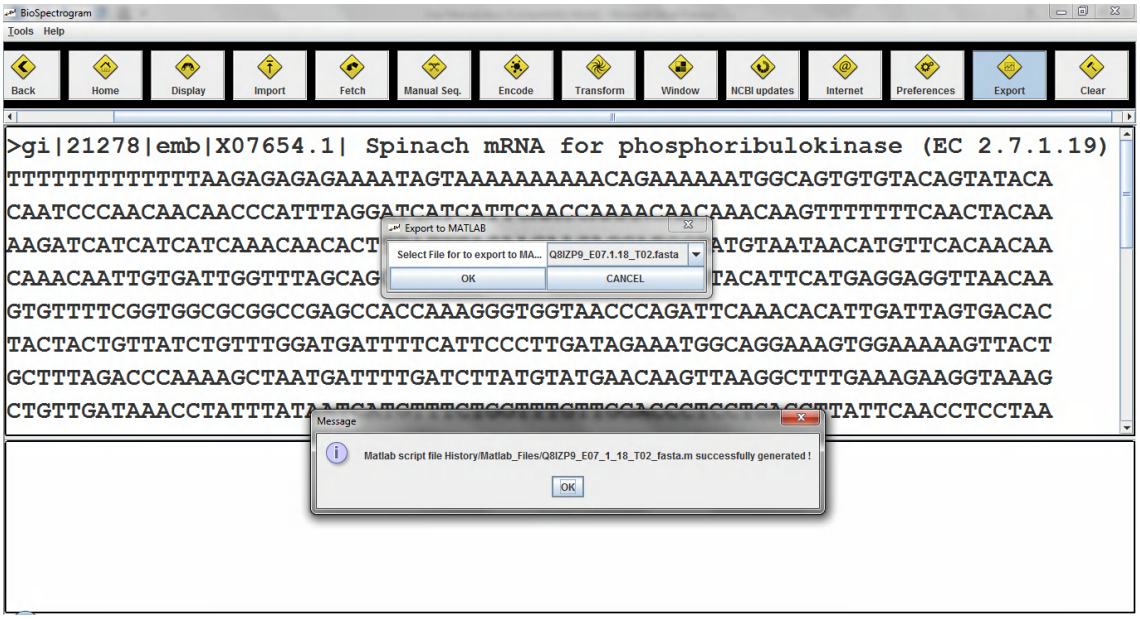

Figure 20: Screenshot showing the generation of the MATLAB file

8 Window Analysis

Window analysis feature provides three schemes for window analysis:

1.) Sliding Window Analysis
2.) Stagnant Window Analysis
3.) C Yin,Yau Gene Prediction

When the user click on Window button, a dialog box appears as shown in the following Figure 21:

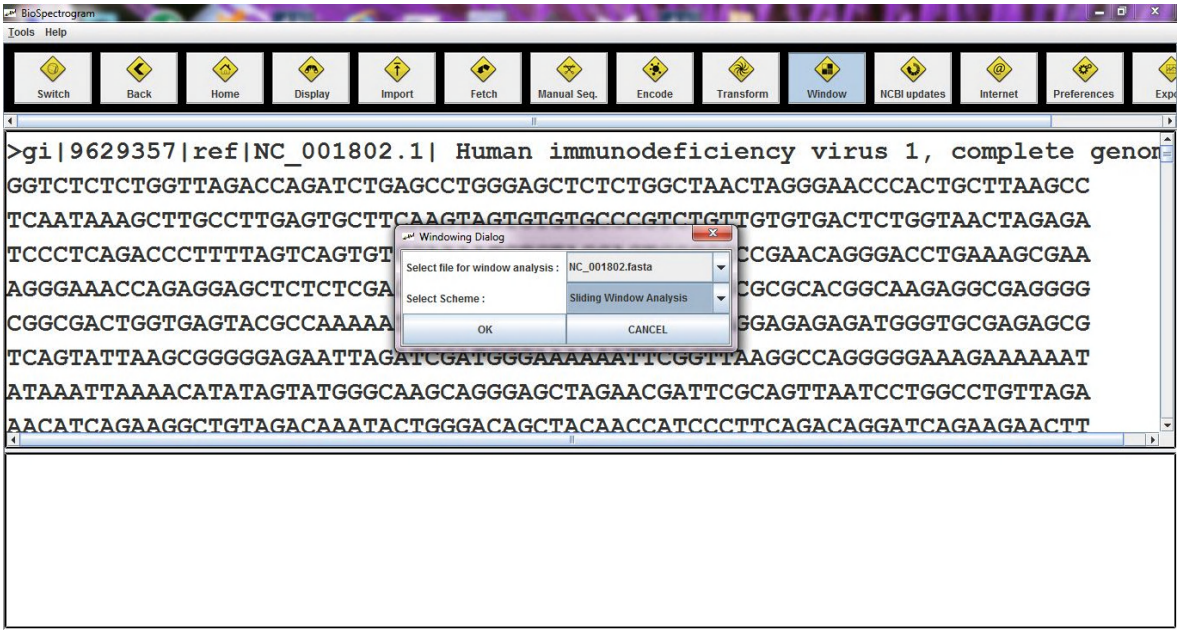

Figure 21: Window analysis dialog box

8.1 Sliding Window Analysis

Select a file for window analysis from the first drop down menu and select a windowing scheme from the second drop down menu. When NC_001802.fasta (HIV-1 genome) is selected from the first dropdown menu, and sliding window analysis is selected from the second drop down menu and OK button is clicked, the a dialog box, shown in Figure 22, appears showing the number of characters in the fasta file.

Clicking on OK button, a new dialog box, shown in Figure 23, appears asking if the user wants to use a forward sliding window or a backward sliding window for analysis. Backward sliding window will start from the end of the DNA sequence and move towards the start of the sequence.

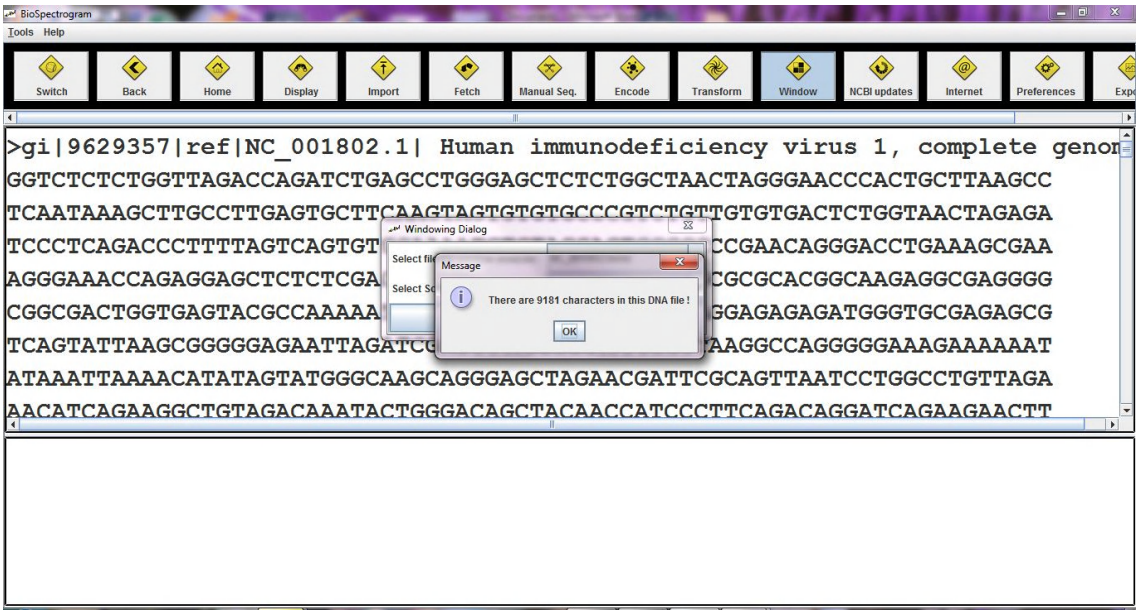

Figure 22: Number of characters in the fasta file selected for window analysis

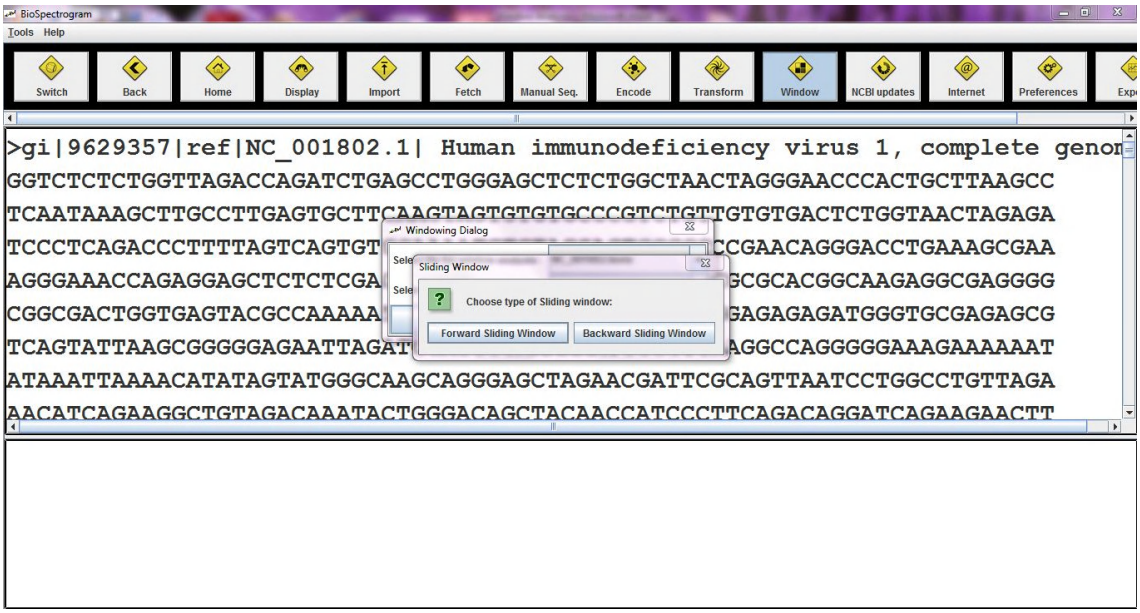

Figure 23: Forward sliding window and backward sliding window option for sliding window analysis

Selecting the forward sliding window option, a new dialog box, which is displayed in Figure 24, appears asking the user to select the encoding scheme and the transformation scheme to be used for window analysis. Selecting the Indicator encoding for A from the first dropdown menu, and Fast Fourier Transform from the second dropdown menu and clicking on OK, a new dialog box appears depicting the number of characters in the file. Clicking on OK, another dialog box appears giving the user an option of choosing a single window size or a range of window sizes. If a range of window sizes is chosen, the user will be asked for the starting window size and ending window size for the range, as shown in Figure 25 and Figure 26. If a single window size is chosen, the user will be asked to enter a window size.

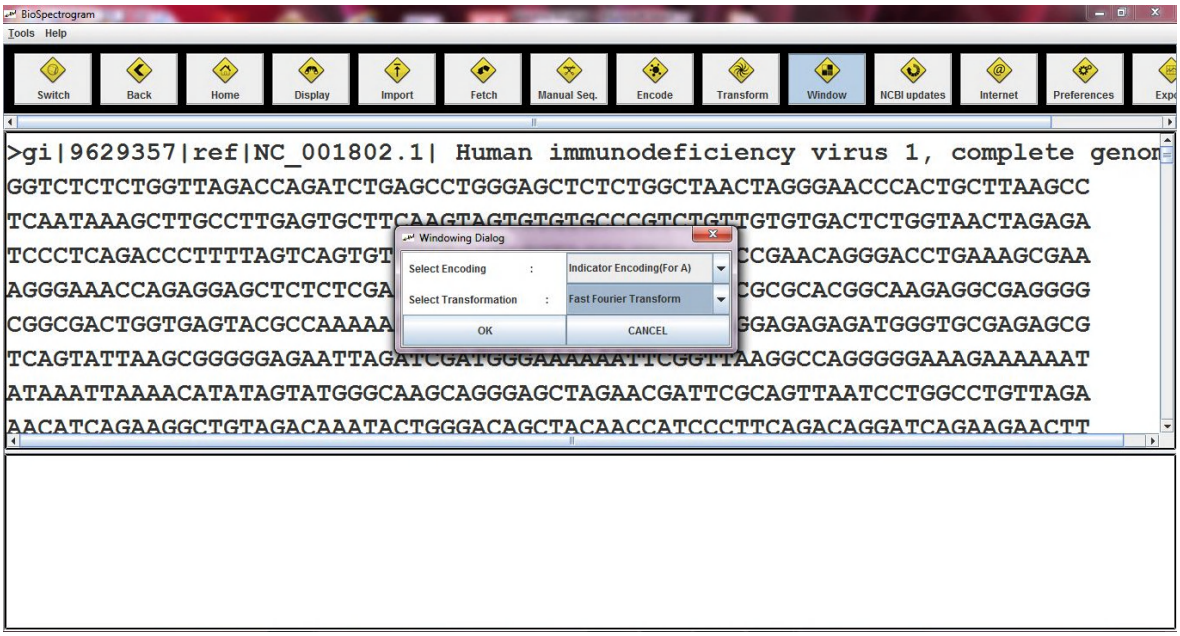

Figure 24: Encoding and Transformation selection dialog box for sliding window analysis

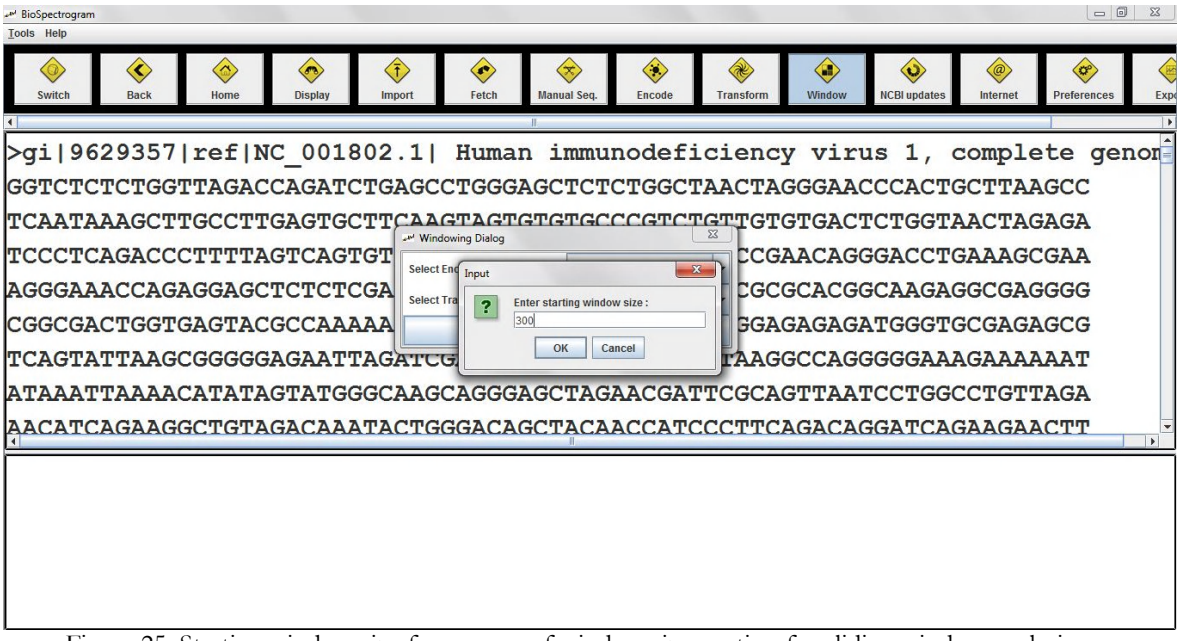

Figure 25: Starting window size for a range of window sizes option for sliding window analysis

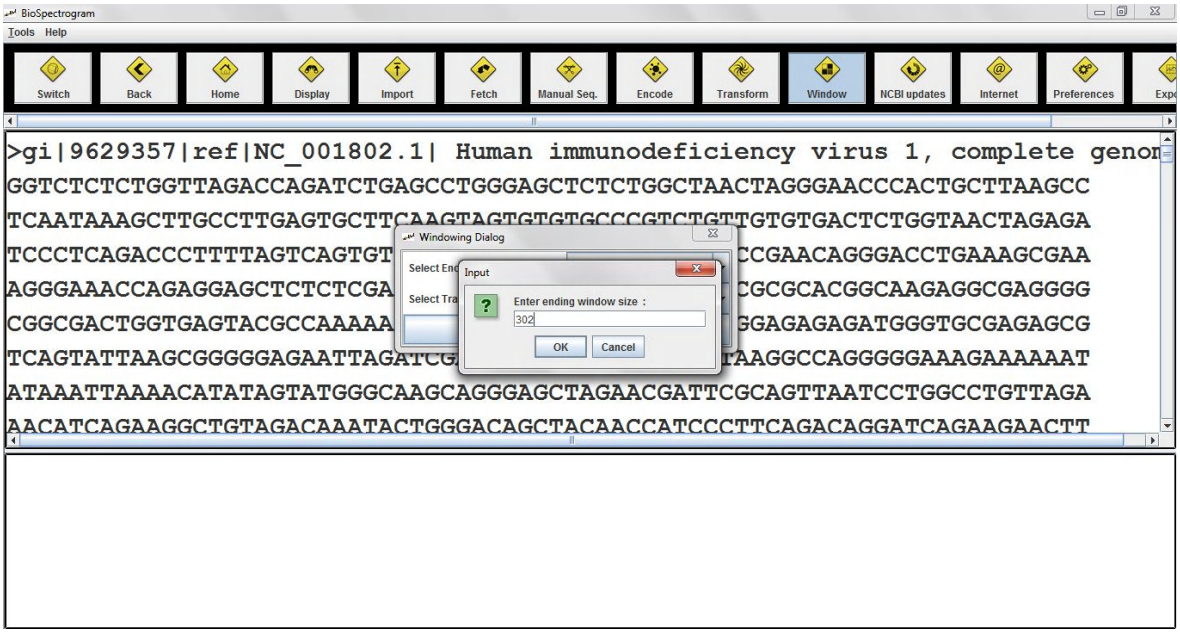

Figure 26: Ending window size for a range of window sizes option for sliding window analysis

When starting window size is entered 300 and ending window size 302, the bio spectrogram starts the sliding window analysis with forward sliding window in the window size range from 300 to 302 using Indicator encoding for A and Fast Fourier Transform.

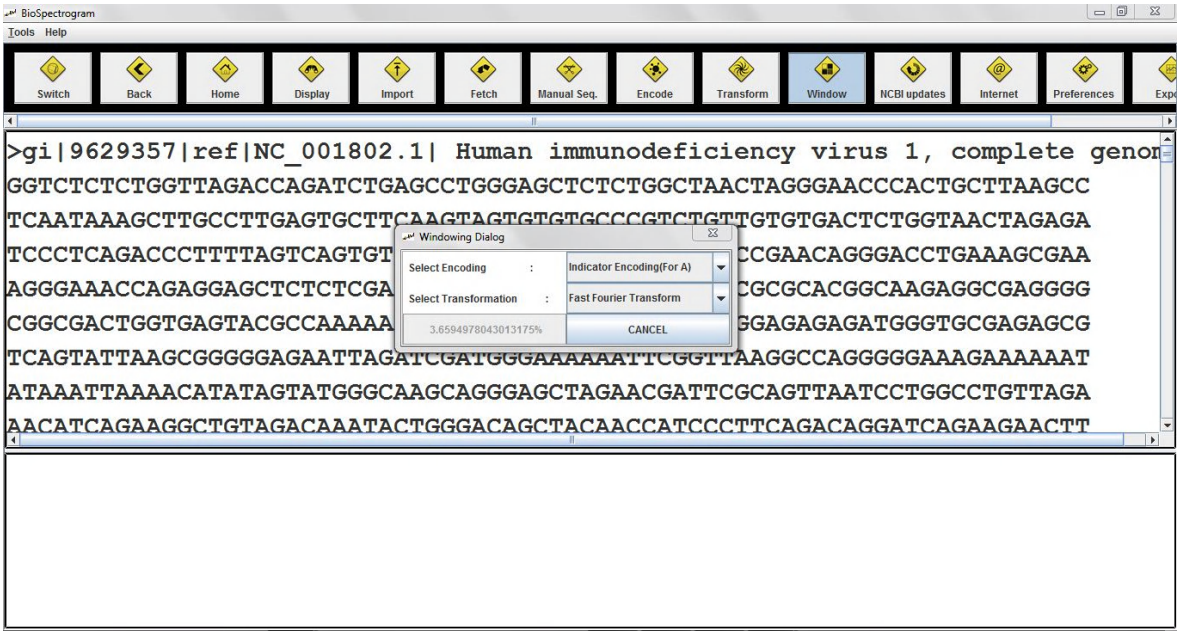

Figure 27: Sliding window analysis showing percentage progress in computation

In this case, NC_001802 DNA sequence has 9181 characters. So, the selected encoding and transformation will be applied to subsequences 1-300, thereafter 2-301,…, 8882-9181, 1-301, 2-302, …, 8881-9181, 1-302, 2-303, …, 8880-9181. The percentage progress in the computation of the encoding and transformation is shown in place of OK button in windowing dialog box, as displayed in Figure 27. The user can cancel the process at any time by clicking on the cancel button in the windowing dialog box. In case of any other encoding or transformation scheme chosen, appropriate dialog boxes will appear asking the user for parameters needed for the

selected encoding and transformation. At the end of the computation of encoding and transformations, a Matlab script will be generated for plotting all the transformation files for each window size in the range of window sizes and a dialog box, displayed in Figure 28, appears displaying information about names and location of the files created in the process, and depicting successful generation of Matlab script files with their location.

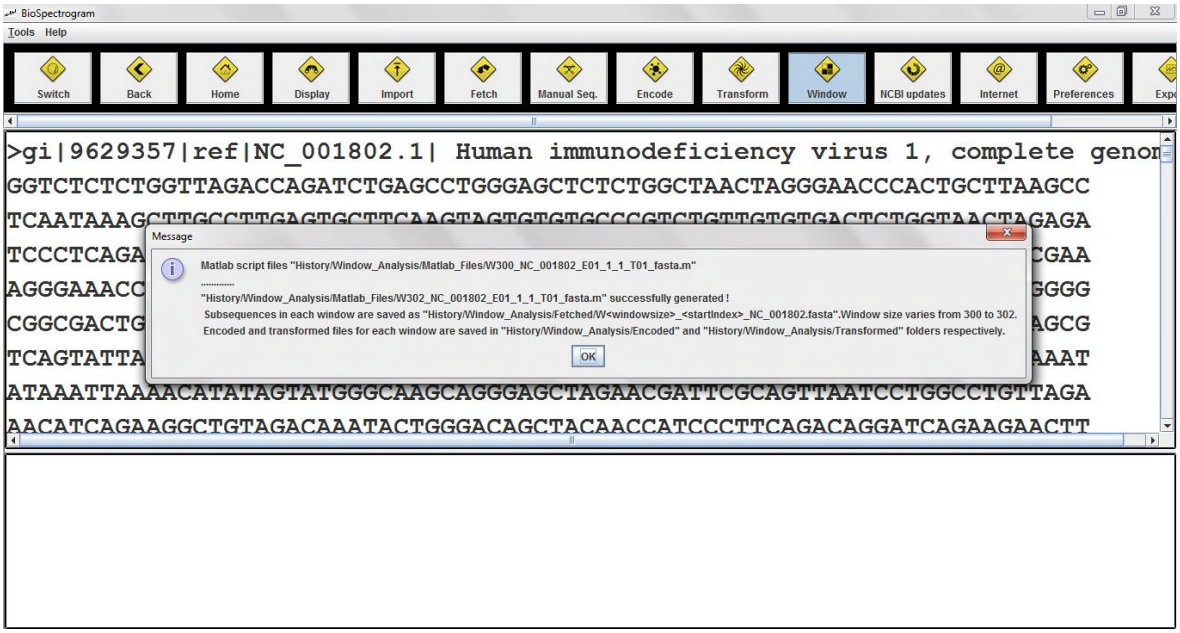

Figure 28: Message indicating successful completion of sliding window analysis

For sliding window analysis option, subsequences of the selected DNA sequence are saved in History/Window_Analysis/Fetched folder, with the naming convention, W<window size>_<startIndex>_<input fasta file name>.fasta. For the above example, <window size> will range from 300 to 302 and <startIndex> will range from 1 to 8882 for window size 300, 1 to 8881 for window size 301 and 1 to 8880 for window size 302. Encoded file corresponding to each subsequence is saved in History/Window_Analysis/Encoded folder with the naming convention <subsequence file name>_<code for encoding scheme selected>.fasta. Transformed file corresponding to each encoded file is saved in History/Window_Analysis/Transformed folder with the naming convention <encoding file name>_<code for transformation scheme selected>.fasta. For each window size, a matlab script is generated for plotting all the transformed files corresponding to the window size in History / Window_Analysis / Matlab_Files folder. Naming convention for the matlab script file is W<window size>_<name of the fasta file selected>_<code for encoding scheme>_<code for transformation scheme>_fasta.m. For the above example, three Matlab script files will be generated corresponding to window sizes 300, 301 and 302 respectively. Running Matlab script corresponding to window size 300 will automatically plot the transformation files one by one at the interval of 0.2 second. Pressing the q key at any time during the plotting will stop the plotting at that particular transformation file. When space bar is pressed, it will start plotting again at the interval of 0.2 second. Plot corresponding to one of the transformation files is shown in Figure 29.

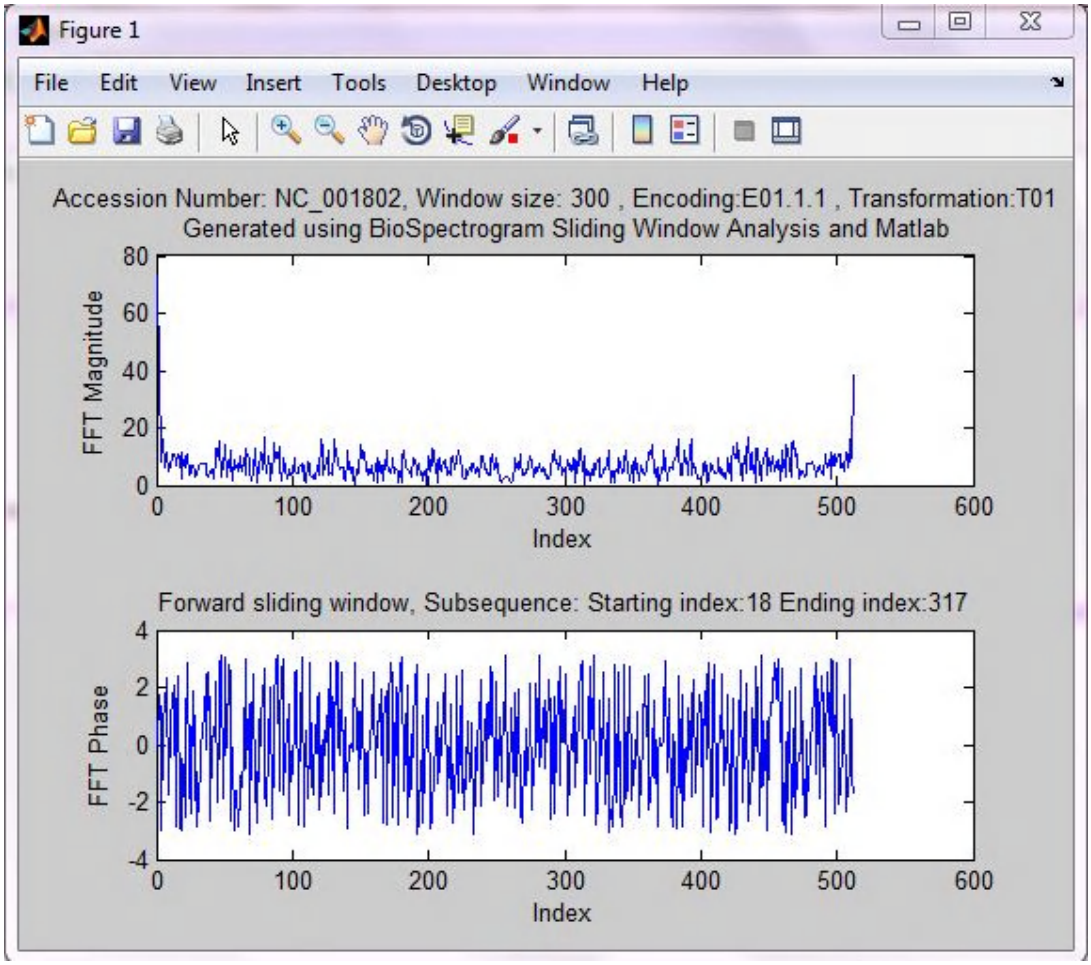

Figure 29: Matlab plot obtained using the script generated by sliding window analysis

The figure contains axes labels and title indicating the input file accession number, window size, encoding scheme selected, transformation scheme chosen, Forward/Backward sliding window and starting index and ending index for the subsequence corresponding to the current plot.

8.2 Stagnant Window Analysis

Stagnant window analysis can be used to extract a subsequence of the selected sequence and generate power spectrum from all its indicator sequences. For example, stagnant window analysis is chosen for NC_001802.fasta file, as shown in Figure 30.

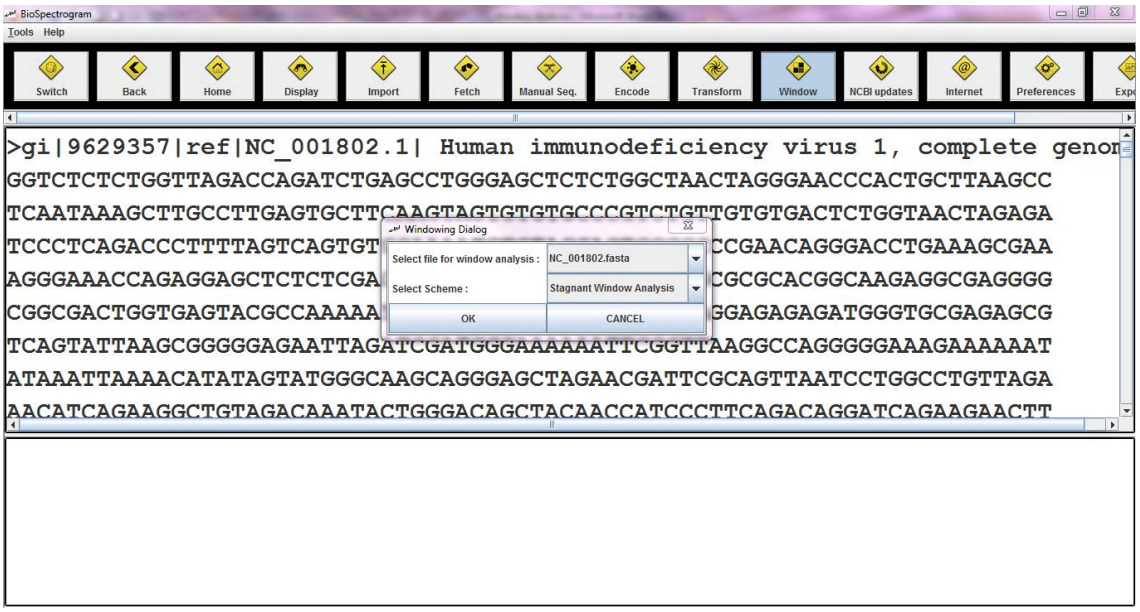

Figure 30: Stagnant Window Analysis

In the window analysis dialog box, when OK is clicked after selecting NC_001802.fasta from first dropdown menu and Stagnant window analysis from the second dropdown menu, a dialog box, similar to the one in Figure 22, appears indicating the number of characters in the selected DNA sequence. Clicking on OK, two new dialog boxes will appear, one after the other, asking for starting index and ending index of the stagnant window, as depicted in Figures 31 and 32. Thereafter, a new dialog box, shown in Figure 33, appears asking if the user wants to generate a power spectrum of the indicator sequences, which is the sum of the squares of magnitudes of Discrete Fourier Transforms of all the indicator sequences. If the user chooses to generate power spectrum, OK button in the windowing dialog box changes to "Analyzing…" thereby indicating that computation is going on. As soon as computation is over, a dialog box displaying the successful generation of power spectrum and Matlab script to plot it along with paths of the intermediate files generated will be displayed, as shown in Figure 34 and the subsequence will be displayed on the upper pane of the biospectrogram window. If the user chooses not to compute the power spectrum, the subsequence will be displayed in the upper pane of the window. Naming convention in the stagnant window analysis: Subsequence from starting index <startIndex> to the ending index <endIndex> is saved in History/Fetched folder by the name W_<startIndex>s_<endIndex>e_<input fasta file name>.fasta. If power spectrum is generated, indicator sequences are saved in History/Encoded folder as <subsequence fasta file name>_<indicator sequence code>.fasta and sum of power spectrum of all indicator sequences is saved as: <subsequence fasta file name>_W01.fasta in History/Transformed folder. Matlab script file for plotting power spectrum is saved as <subsequence fasta file name>_W01_fasta.m in History/Matlab_Files folder.

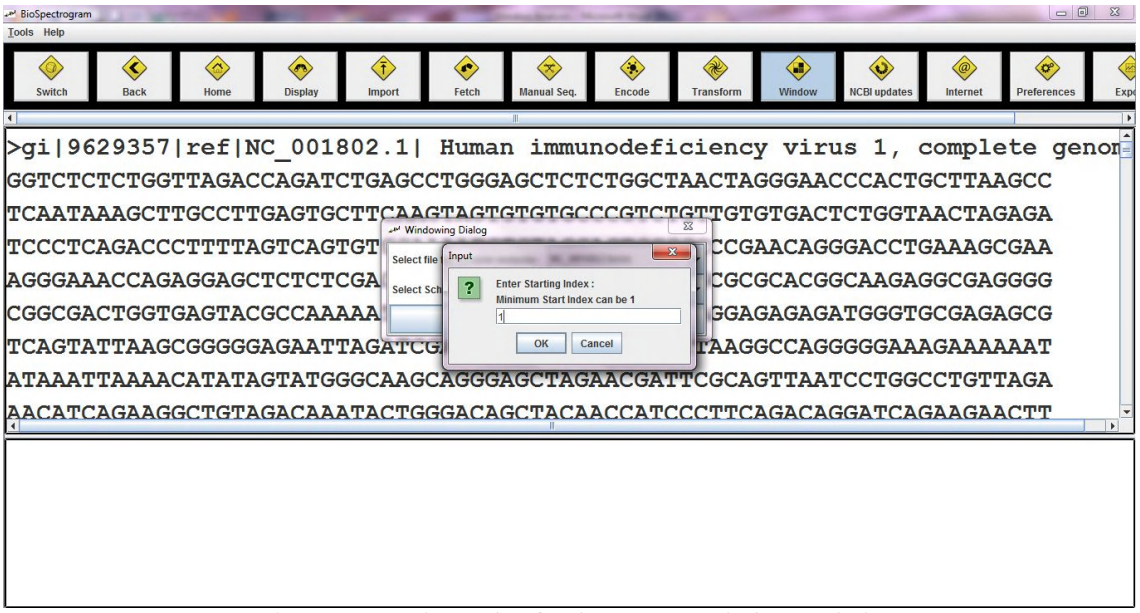

Figure 31: Starting Index for the stagnant window analysis

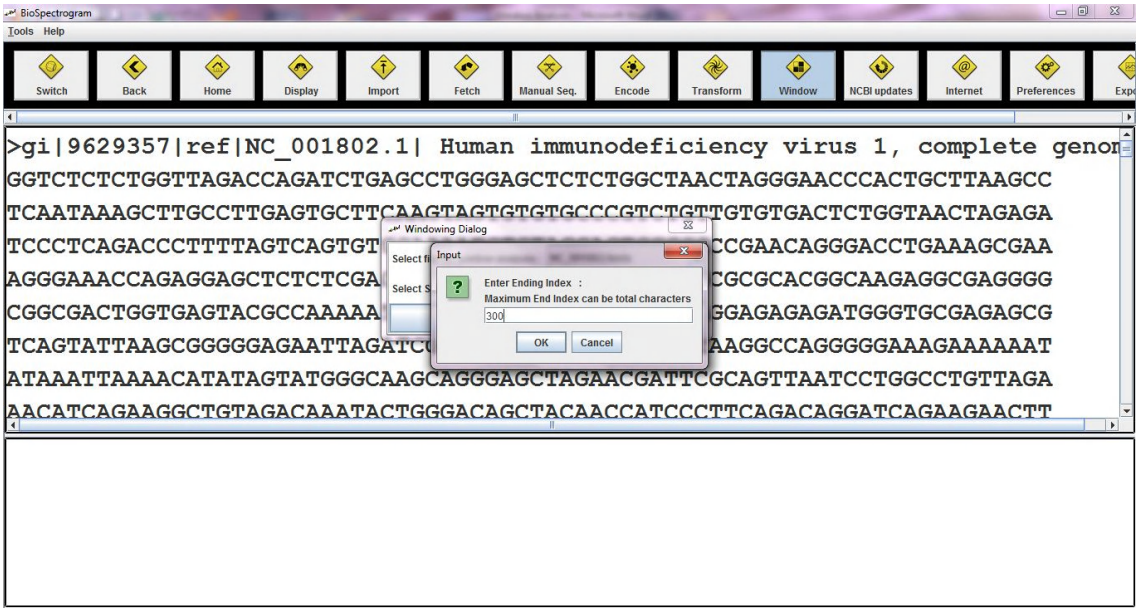

Figure 32: Ending index for stagnant window analysis

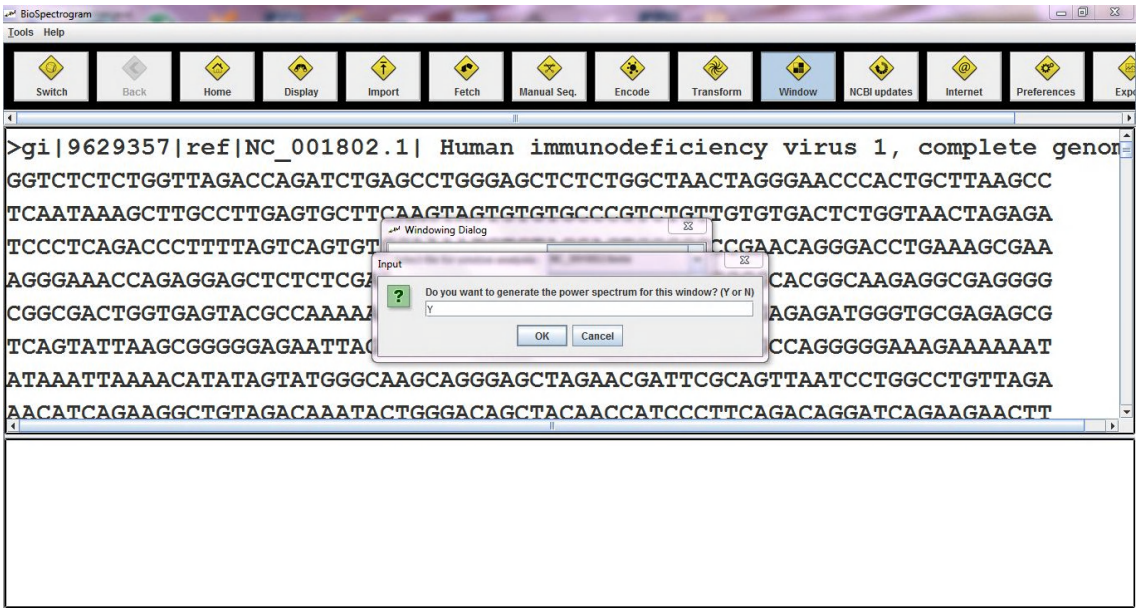

Figure 33: Power spectrum option for stagnant window analysis

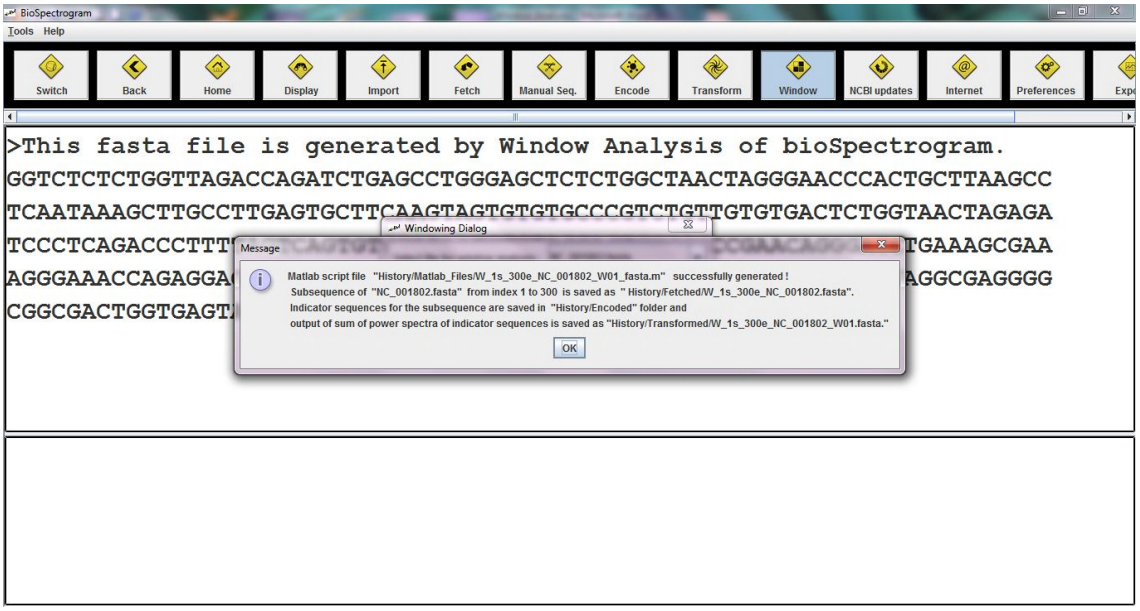

Figure 34: Successful generation of power spectrum and Matlab script file for plotting the spectrum

By running the Matlab script generated by stagnant window analysis for NC_001802.fasta file, the plot obtained is shown in Figure 35 along with the axes labels, and title containing the accession number, subsequence starting index and ending index and the title (Sum of Power spectrum of all indicator sequences).

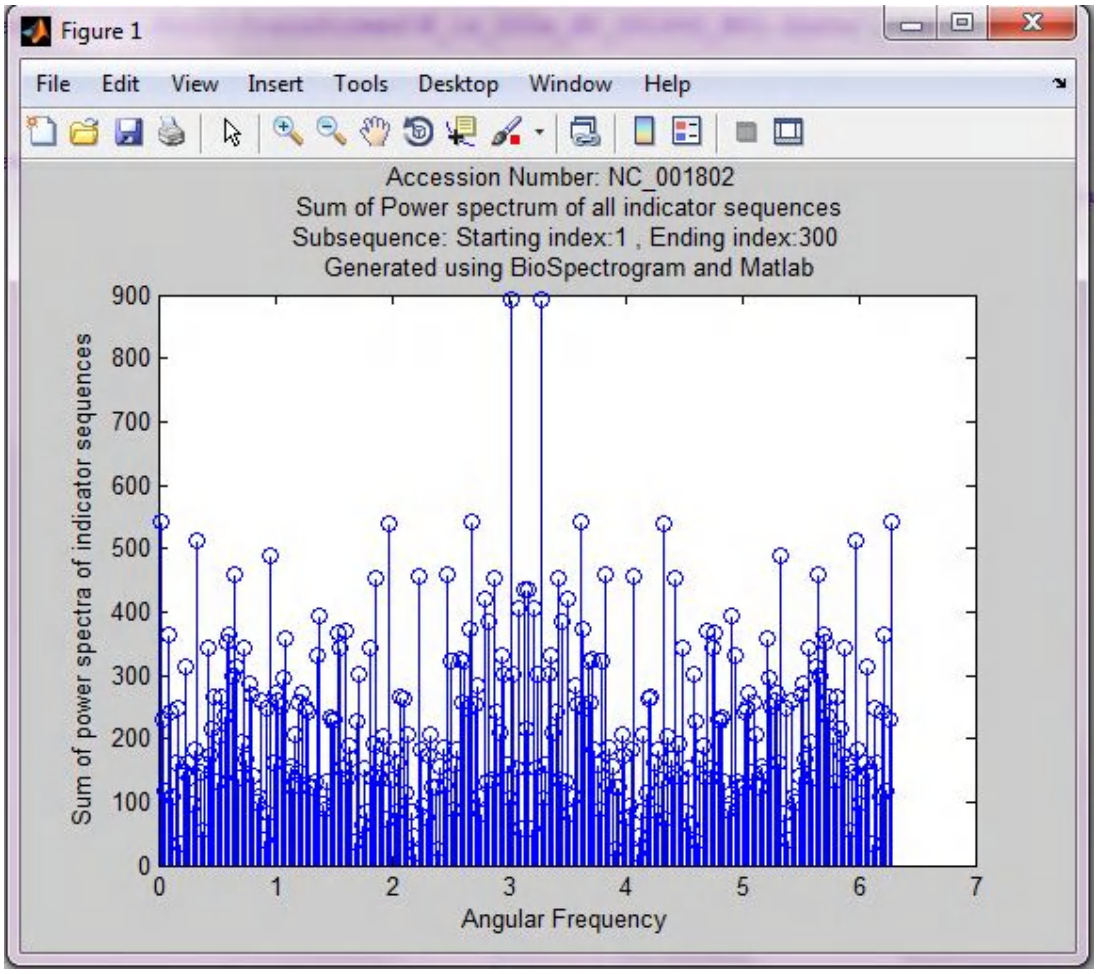

Figure 35: Sum of Power spectrum of indicator sequences plotted using Matlab script generated by stagnant window analysis

8.3 C Yin, Yau Gene Prediction

When the user selects this feature from the dropdown menu for the X02323.fasta file in windowing dialog, as shown in Figure 36, a dialog box, similar to the one shown in Figure 22, appears indicating the number of characters in the DNA sequence in the selected fasta file. Clicking on OK, a new dialog box appears asking for window size for gene prediction, as displayed in Figure 37. Clicking on OK after entering 300 as the window size, the OK button of the windowing dialog is replaced by percentage showing current percentage progress of the computation, which is shown in Figure 38.

During gene prediction, the sum of power spectra of all the indicator sequences of the subsequences of the original selected sequence with the given window size is calculated and 3-base periodicity property is checked.

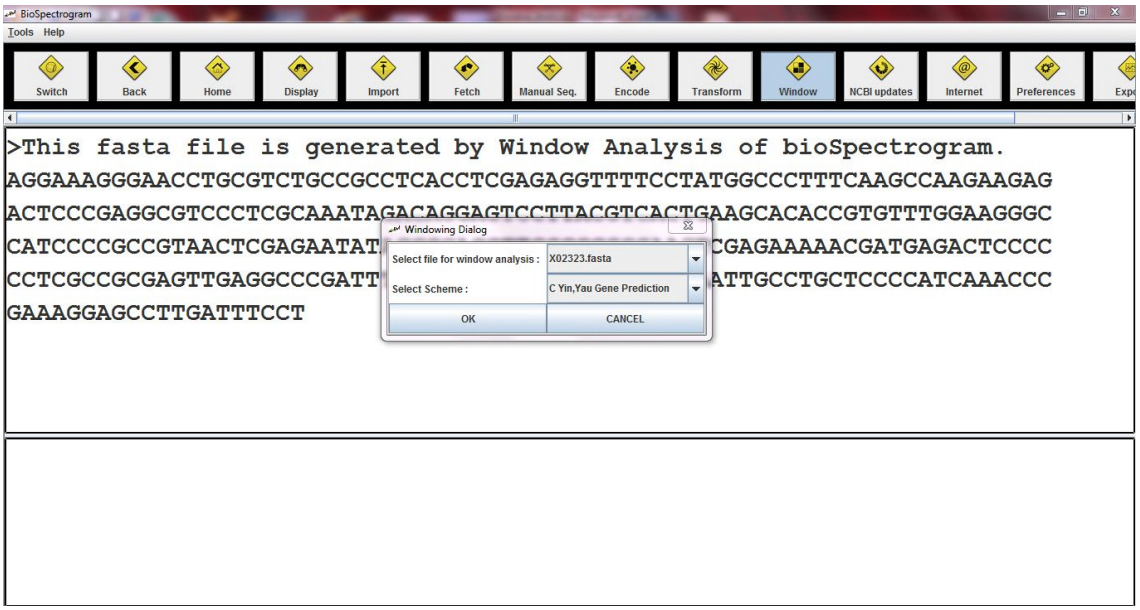

Figure 36: C Yin, Yau gene prediction

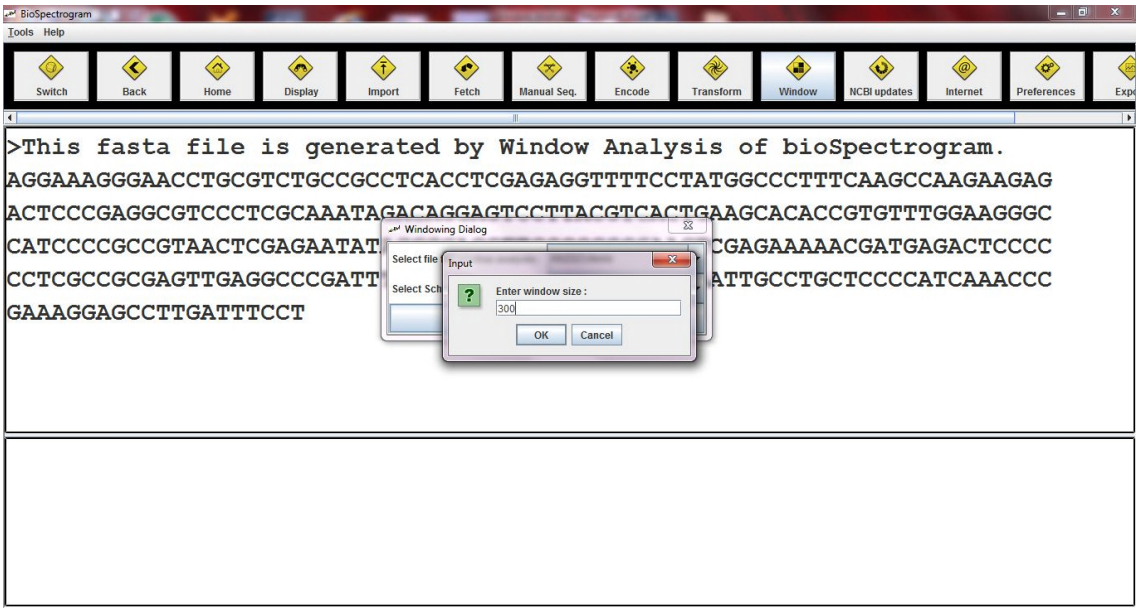

Figure 37: Window size for C Yin, Yau Gene Prediction

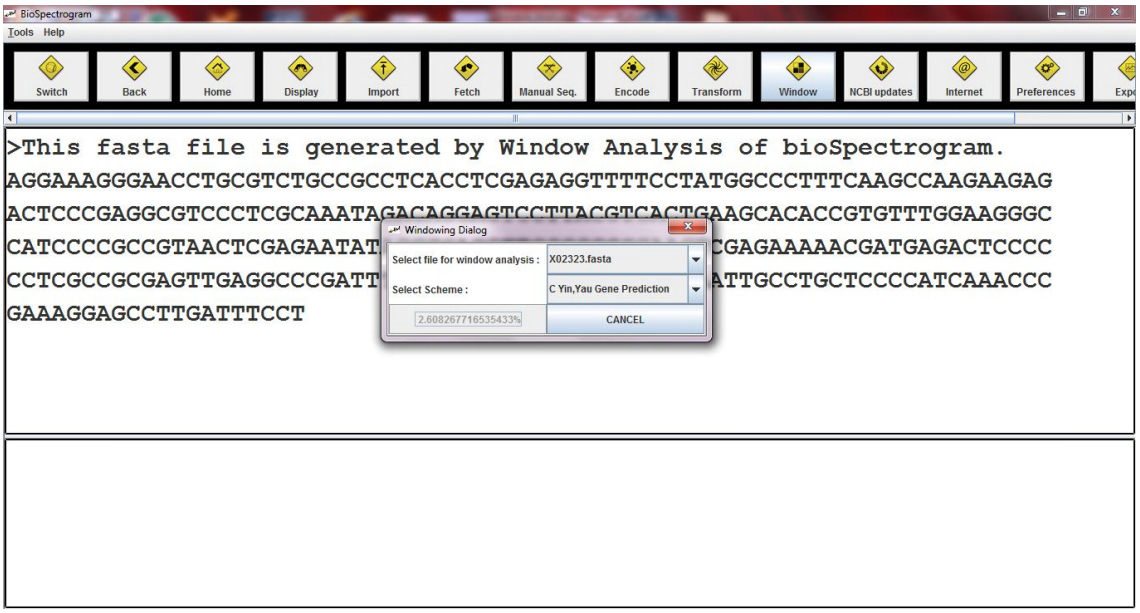

Figure 38: Percentage Progress of Gene Prediction

The start index and end index of the subsequences meeting having the property are saved in a file named <name of selected fasta file>_geneprediction.fasta in History/Gene_Prediction folder. At the end of computation, a message indicating successful generation of matlab script file is displayed, as shown in Figure 39, and the upper pane of BioSpectrogram displays the selected fasta file and the lower pane displays the gene prediction results at the end of the computation.

The sum of power spectra of indicator sequences are saved in History/Gene_Prediction/Transformed folder with the name W<window size>_<name of selected fasta file>_<startIndex>_W01.fasta, where startIndex varies from 1 to <number of characters in selected sequence> - <window size>+1.

Matlab script file for plotting the sum of the power spectra of all subsequences one by one is saved in History/Gene_Prediction/Matlab_Files folder by naming convention W<window size>_<name of selected fasta file>_W01_fasta.m. Running the Matlab script will automatically plot sum of power spectra of indicator sequences of each subsequence one by one at the interval of 0.2 second, one of which is shown in Figure 40. The user can stop the plotting at a particular subsequence by pressing "q" and can resume the automatic plotting by pressing space bar. The plot includes the axes labels along with accession number, window size, starting index and ending index of the subsequence.

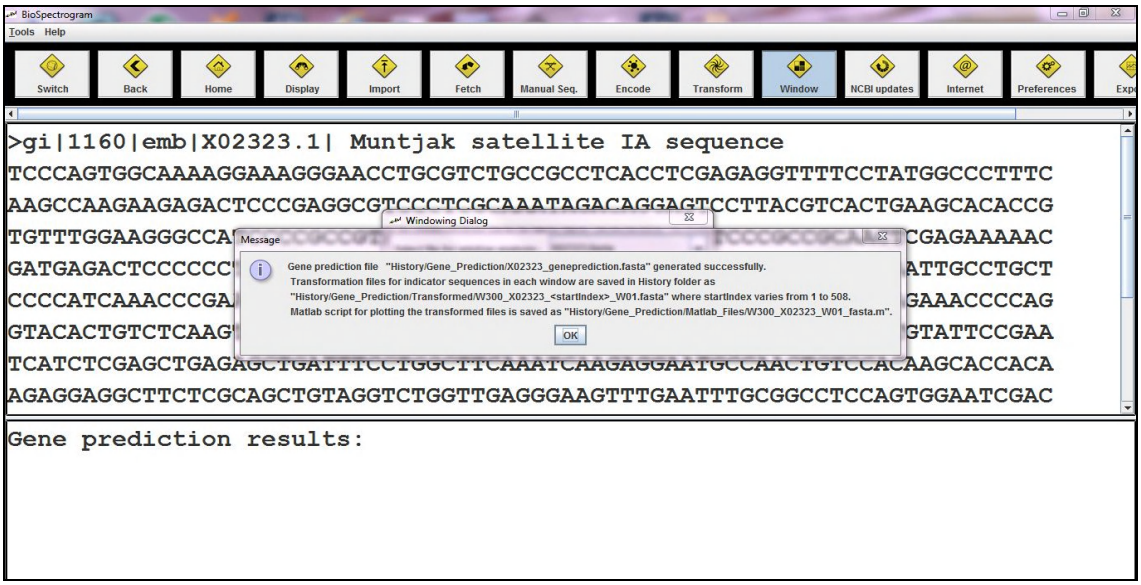

Figure 39: Message displaying successful completion of Gene Prediction and generation of Matlab script files

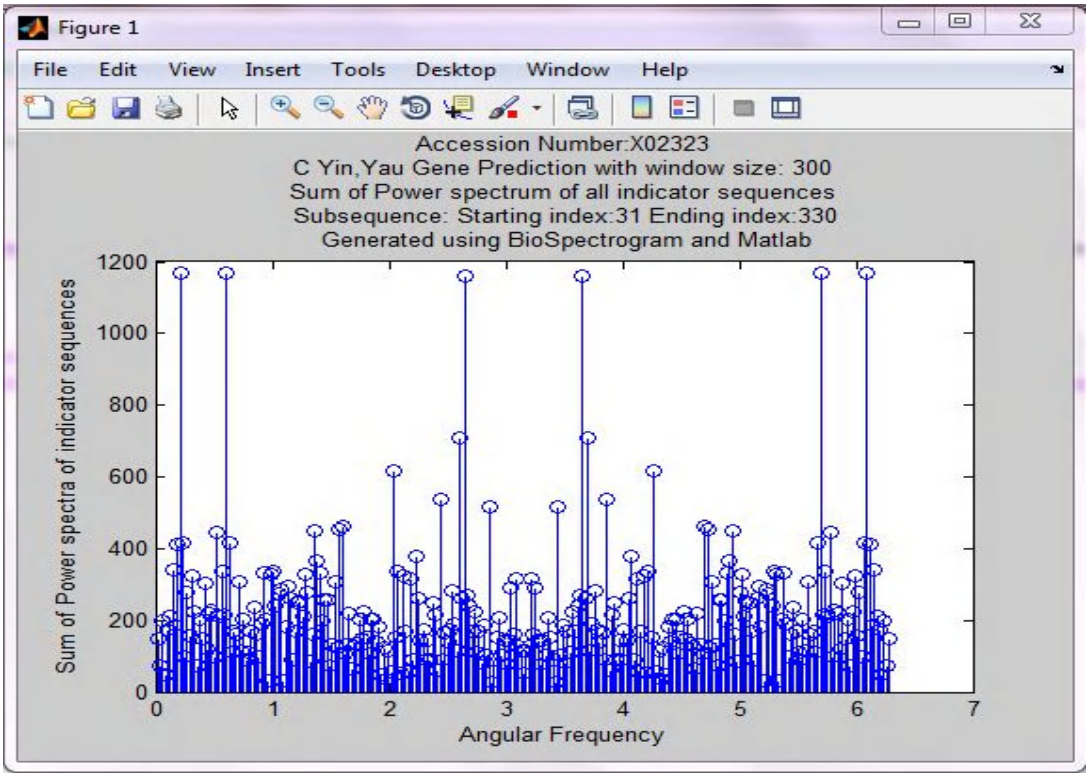

Figure 40: Sum of power spectra for subsequence 31-330 using C Yin, Yau Gene Prediction on X02323.fasta with window size 300

9 NCBI Updates

Screenshots in Figure 41 shows the button to update the fetched history. Purpose behind this is that database of the protein is highly dynamic. It keeps changing. So we have also provided utility to update all the files in the user history at once. This simply re-downloads all the files that are in the user history and replace them in place of older ones. This operation requires more time than

usual, so it is recommended that user perform this operation only when there is access of time and internet speed. Figure 41 screenshot shows the same message so that user does not perform this operation by mistake. For more security, we have put another yes/no input dialog after pressing "OK" on this one. After user enters "y", we are simply downloading all files again and on the successful updating we are showing confirmation dialog which will display message saying all files updated successfully.

Figure 41: Screenshot showing the NCBI update dialog

10 Check Internet Connectivity

The screenshot shown in Figure 42 shows the scenario when user presses the second last button with @ symbol on it. This button checks the internet connectivity of the user. On pressing the button application simply try to connect to the NCBI server itself because ultimately we also need to check the whether the database server is up or not.

You can see that the button is highlighted showing that is has been pressed. There is also tiny message box displaying the status of user's internet connection. Internet connection is required only of user needs to download some new files from NCBI server. This feature is also used by default when user used "Fetch" functionality.

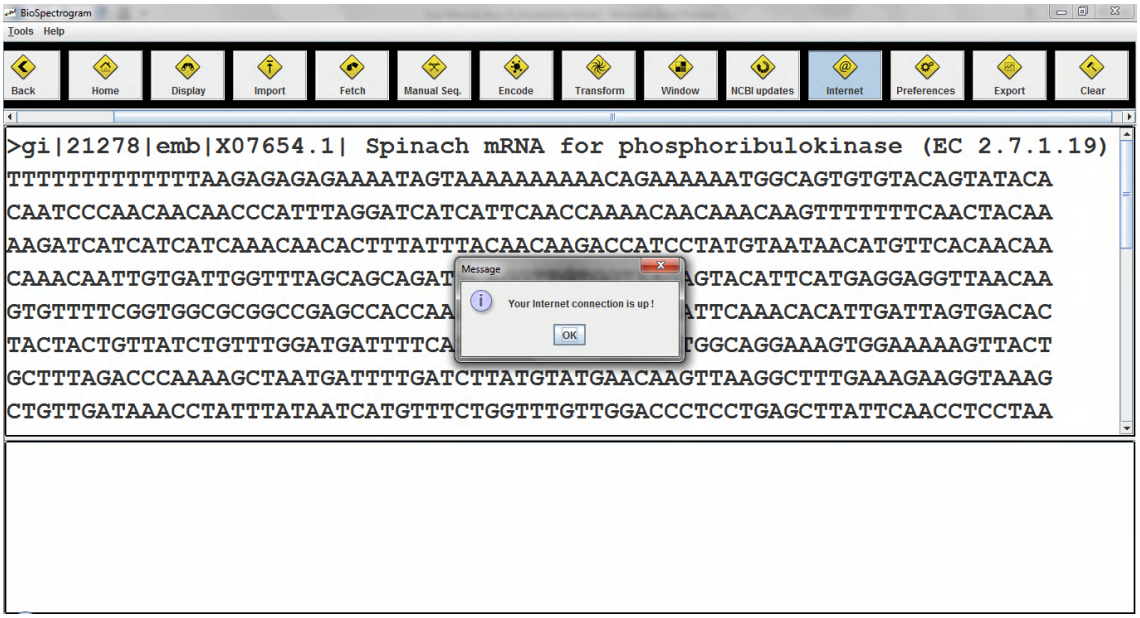

Figure 42: Screenshot for Internet connection is up dialog

11 Preferences

After this, we have shown the screenshot of Preferences Dialog Box in Figure 43. Preference box contains various preferences like checking amount of memory used for the fetched files, encoded files, transformed files, maximum allowed files to be kept in the fetched, encoded and transformed directory, check current maximum number of files allowed in fetched, encoded and transformed directory and finally change the font size of the pane. Minimum value for font size is 10 and the maximum it can take is 100. If user enters invalid input, an error message will show. In case user doesn't remember the function of the buttons, there are tool tips provided for the each and every button. If you look at the screen shot for the preferences box, you will see very short description in the tool tip.

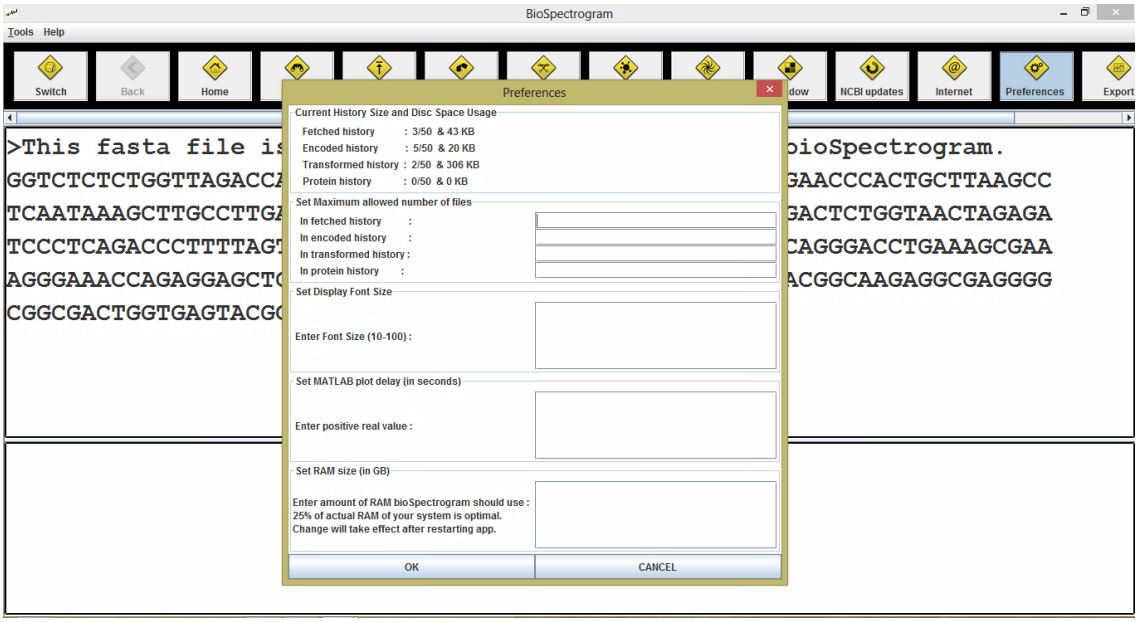

Figure 43: Screenshot showing preferences

After giving valid inputs, all the settings are saved and message confirming that is displayed as shown in Figure 44.

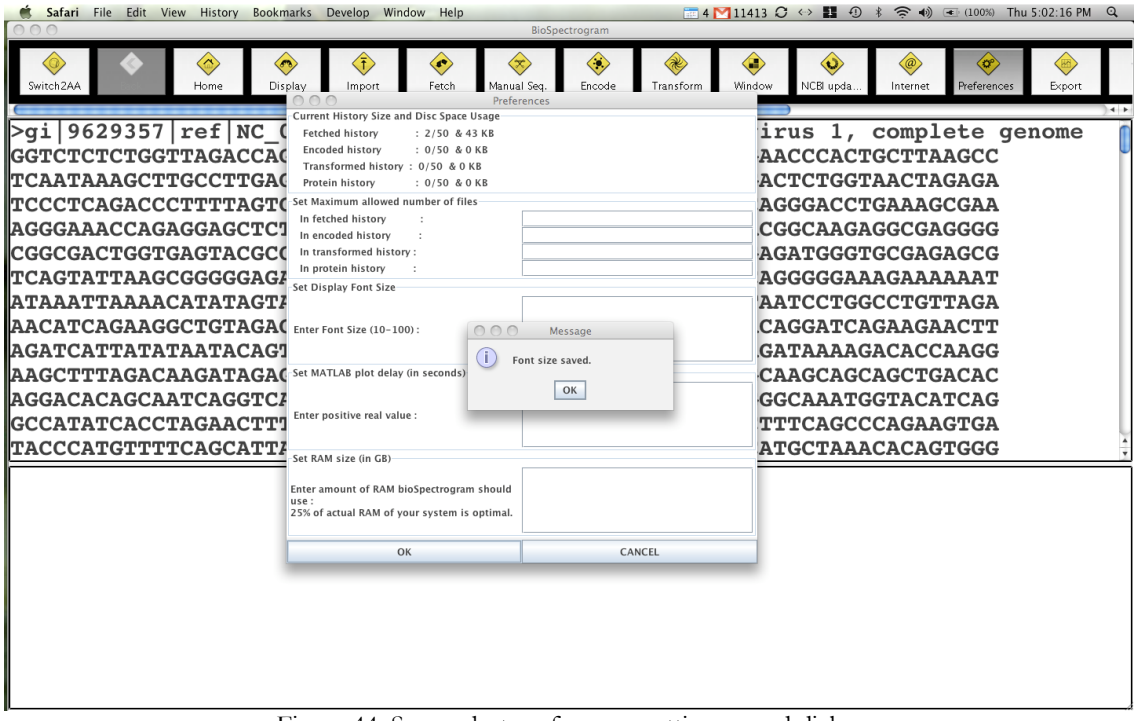

Figure 44: Screenshot preferences settings saved dialog

12 Clear History

This is a very simple function. As the name suggests, clear history button simply cleats all the files in all the directories. It is like resetting application for the first time use. When clutter of different fetched, encoded, transformed, exported to MATLAB files, files generated by different

window analysis reaches a situation which is out of control for user, this feature really comes very handy. Since it can be harmful for some of the user as it deletes all the files, we have put one extra confirmation dialog to decide whether user actually wants to delete all the files or it was just a false click.

Screenshot in Figure 45 shows the screenshot of the confirmation dialog when button "Clear" is pressed. You can see that the button "Clear" is highlighted as it was pressed.

Figure 45: Screenshot showing clear dialog

13 Display a Fetched File

This option is used to display any of the fetched files at any point of time during the application run. On clicking on a button saying "Display", a small dialog appears with single dropdown menu containing all the fetched file names. Screenshot of that is shown in Figure 46. User just has to select any of the file from the list and press "OK". Immediately after that, the selected file will be displayed in the upper input display pane.

Figure 46: Screenshot showing a display dialog

14 Switch to Protein Mode (Amino Acids mode)

Bio spectrogram can also process protein sequences. Very first button the left is used as a Switch to change the mode of operation. Right now we are supporting only two modes. One is DNA mode and the other is protein mode. All the operation explained in the user manual applies to only DNA sequences as long as we are working in default "DNA" mode. Once user clicks "Switch" button, application enters into "Protein" mode.

In protein mode, we enable all the operations that are supported to protein sequences and disable all the operations that are not supported currently. For example, Windowing feature is not available in protein mode. Actually, rest all of the features are available in protein mode with only exception that three of the features behaves differently which are as following.

1. Import
2. Encode
3. Display

In import a file, there is no change from the user point of view. Only change is that when user imports a Protein file, it should not go to standard fetched folder instead it will go to folder "/History/Protein" which is quiet reasonable. User can only import Protein file, fetch can only download or rather recognize DNA files. Screenshot in Figure 47 shows the file selector dialog box. You can see that the "Window" button is disabled and icon of switch also changes depending on the mode it is in.

Figure 47: Screenshot for importing a file from local hard drive

In encode functionality user will get similar dialog box with two dropdown menus but only change in it is that first dropdown menu will contain Protein files (from directory /history/protein) and second dropdown list will contain two of the encoding that is available in our software (Protein Indicator Encoding & Protein Electro Ion Encoding).

Screenshot in Figure 48 shows the dialog box that will appear on user's screen when encode function is used in Protein Mode.

Figure 48: Screenshot showing encoding dialog box for proteins

Finally, in display function, we only need to display the files that are in protein's directory. So that is the only difference in the display in protein mode and DNA mode. Screenshot in Figure 49 shows that there are only two files in protein history that are listed in the dropdown menu.

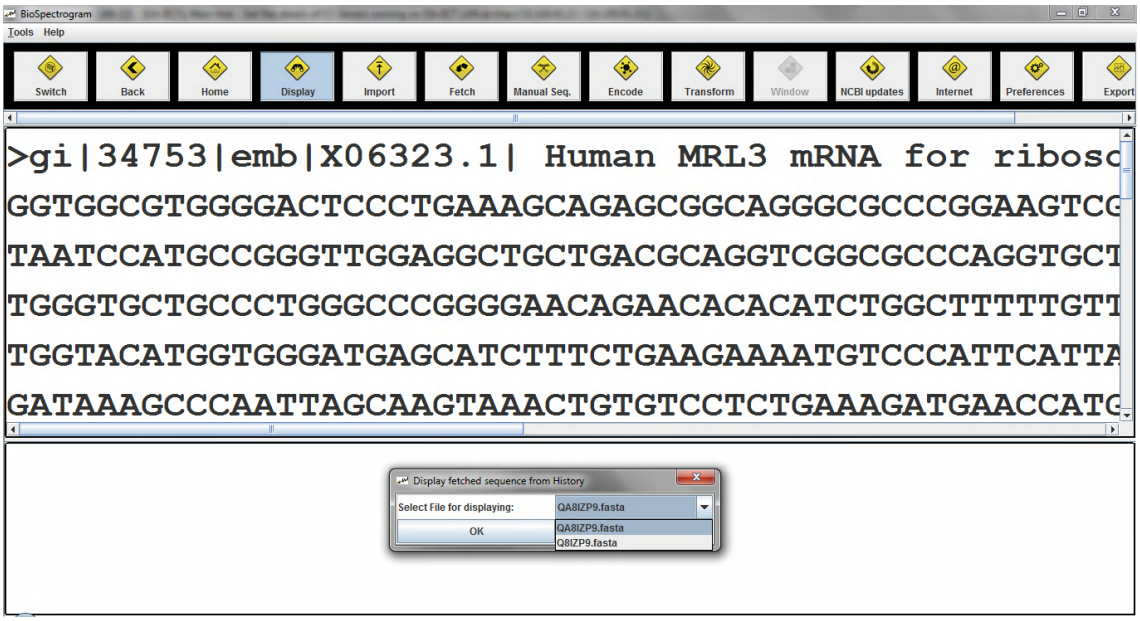

Figure 49: Screen shot showing files in protein history

In short, there is not much difference as far user is concerned. Only thing as a user has to remember is that whenever a raw, unprocessed protein file is needed, the function will display list from protein's directory and whenever it does not use any fetched files directly, its functionality does not change. Once, protein file is encoded, it is converted to number format from which one can not distinguish whether encoding was done on DNA file or a protein file. Although, naming conventions we are using can help user to know about original file, encoding applied, transformation applied etc.

15 Tools Menu

Now we are showing the screenshot of different menus available in the software. First menu is called "Tools". Some users like to use tools by clicking on the name if the tool rather than using buttons with symbols. So, in the tools menu we have given list of all functionality that is being achieved by the buttons in the UI.

There is only one extra entry in tools menu which is not in the toolbar. That entry is to exit the application which is very standard option. Screenshot below shows the menu bar showing all the options.

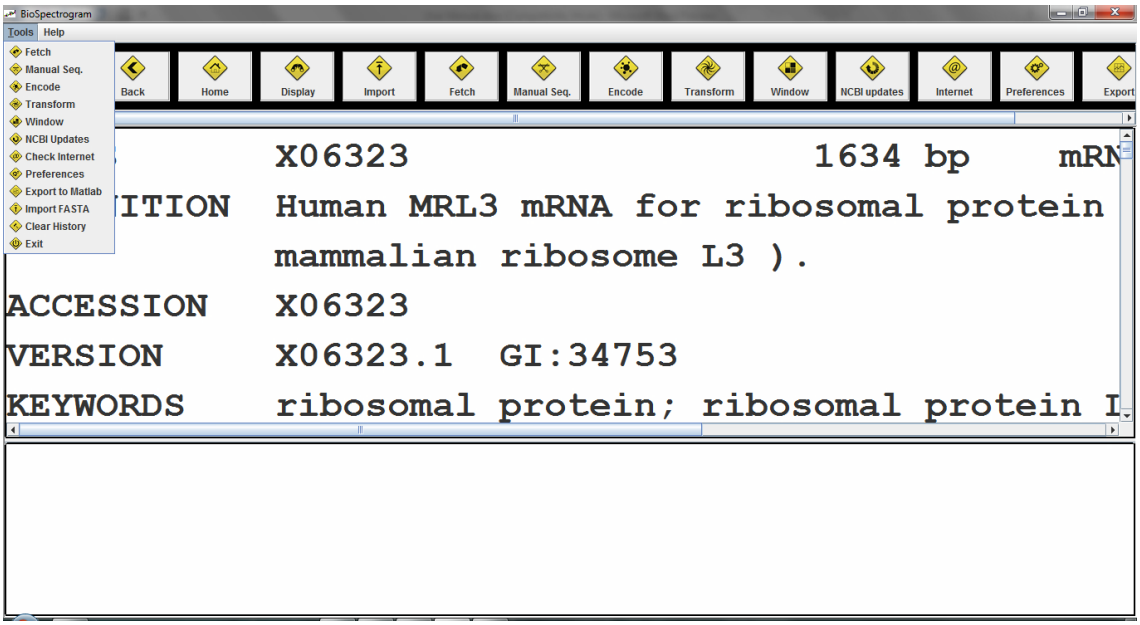

Figure 50: Screenshot showing tool bar menu

16 Help Menu

Second menu is the "Help" menu which is very common in any software. There are four options in the Help Menu.

1. User Manual
2. Software Update
3. Naming Conventions
4. About

First option, user manual should open this user manual in the default pdf reader of user's system. Second option is right now not very relevant but it will be once there are more versions released. It opens a default browser in the user's PC and opens a webpage of our application's download page. If user is using older version then he should upgrade to the newer version which can be learned from that webpage. Naming conventions is another PDF file which contains all the codes of Encodings and Transformations for quick access.

Finally about option recognize the contributors in this project. There is a dialog box that opens up as shown below, which contains information like logo of the software, version of the software, name of the software, Credits button and URL of the software. On pressing "Credits", it opens a PDF document in user's default pdf reader.

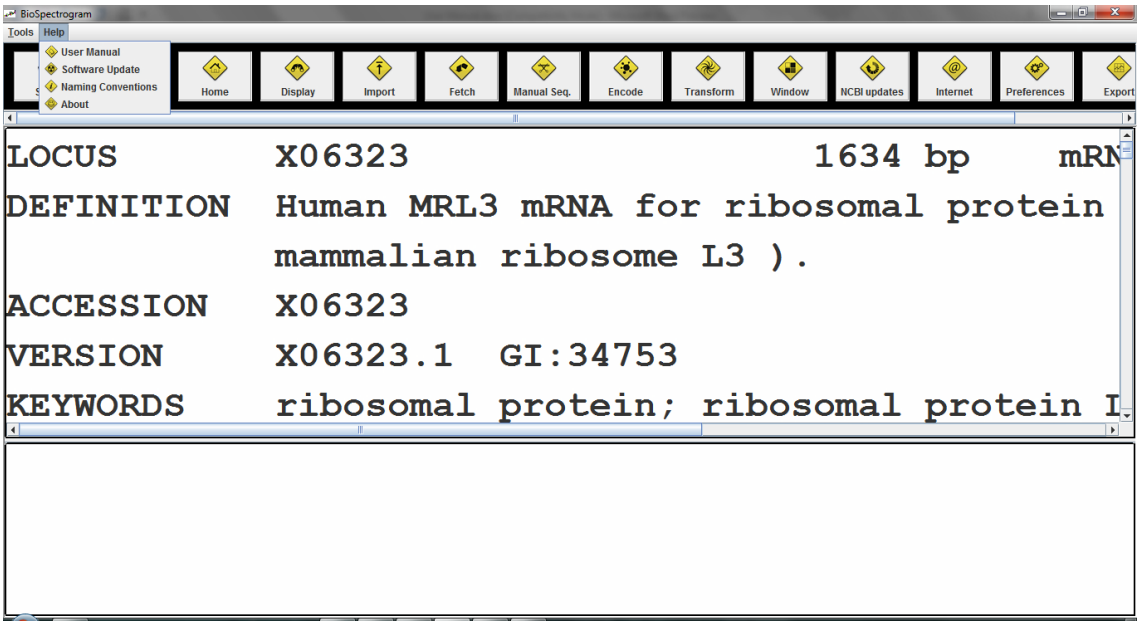

Figure 51: Screenshot showing help menu bar

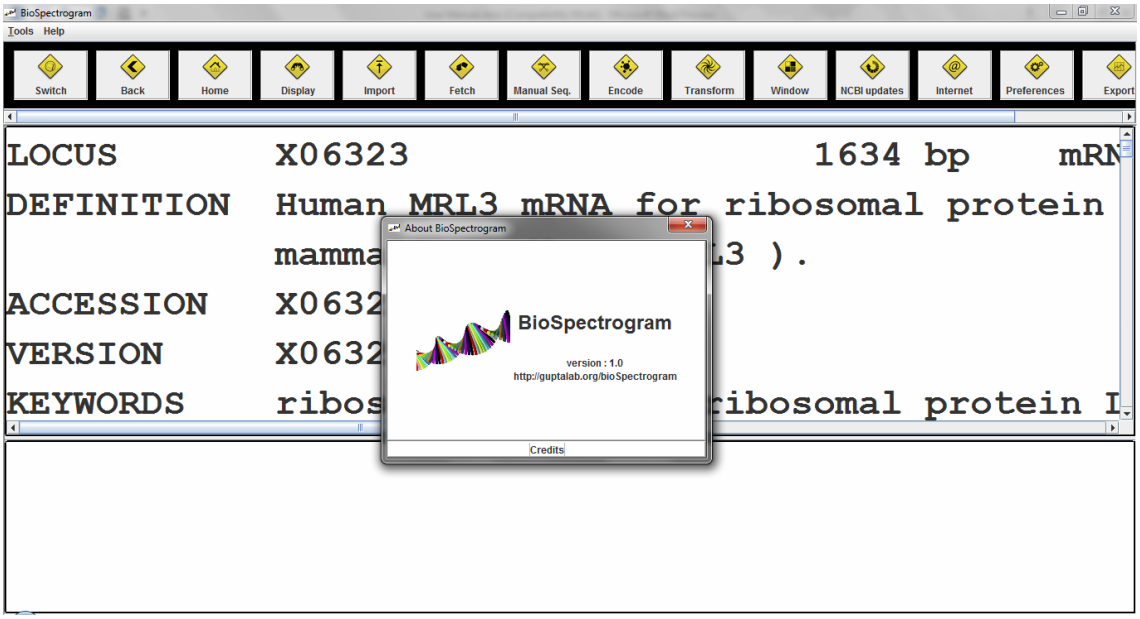

Figure 52: Screenshot showing about menu

17 MAC version

Since, we wanted to make our application platform independent, we chose Java as the primary developing language for our application. Now, in the Linux and Windows environment all the functionality will appear same but there is a slight change in MAC version of the software. Although, Java itself is platform independent, there are some UI related issues in MAC. All the screenshots shown in this user manual are taken on Windows 7 running system. There is only version of software but some UI component won't work in MAC so we have tweaked the UI when application is being run on MAC system and we are saying it as if it was a different version but it is not. Our application internally takes care of all the OS related changes. We have two menus as explained above. Now, we are removing menu bar entirely when application is

executed in MAC system. Now, in the tools menu, all the functionality is anyway present in the toolbar itself except "exit" button. But exit action can be done using standard "X" button in any application. Main problem is in Help Menu. None of those four options are included in the tools menu in the original application, so we added four additional buttons in Toolbar but only MAC users will be able to see them. To browse buttons conveniently we have put a horizontal scrollbar in the tools menu itself.

18 Directory Structure, Images, help files

There is one root directory "biospectrogram" and three sub directory namely 1. biospectrogram/ History, 2. biospectrogram / Help and 3. biospectrogram / Icons.

History directory will have seven different subdirectories to organize different files for convenient access. Four of them correspond to four main functions 1. Fetch 2. Encode 3. Transform 4. Plot. There is one for Protein and two more directories for Window Analysis and Gene Prediction. In Window analysis, there are again four sub directories for four primitive functions of our application. In gene prediction, there are only two subdirectories one for Transformed files and the other for export to MATLAB files.

Help directory contains all the help files in pdf format. Some of them are used in the application and some of them are just for users' general information. There are two build script one for Windows and the other for Linux/MAC. There are Coding conventions, User Manual, Credits files in the same folder. Readme file will guide user on how to run the application using source code.

Icons folder contains all the images used in the application. User is requested not to delete or modify any of the files from any of the directory directly. Root folder contains all the source code for the application. If you lose some files and application stops working just download a newer copy from internet and restart your work however any of the history cannot be recovered. List of some popular accession numbers is given in Annexure 1.

19 Support, Feedback and Distribuation

Product demo video is available at the homepage of Biospectrogram. Users are requested to contact Manish K. Gupta at the email: mankg@computer.org for feedback and any other issues with the software. A sample test data of genomes is available in the fetched folder of the distributions of the software. Two platform specific installers (Windows and Mac) are available on the project home page along with source code with open source license agreement. We plan to have discussion forums etc. for users at the home page. Enjoy the software!

20 References

[1] M. Akhtar, J. Epps, and E. Ambikairajah. On DNA numerical representations for period3 based exon prediction. In *Genomic Signal Processing and Statistics, 2007. GENSIPS 2007. IEEE International Workshop*, pp. 1 –4, June 2007.

[2] D. Anastassiou. Genomic signal processing. *Signal Processing Magazine, IEEE*, 18(4):8– 20, Jul 2001.

[3] A. K. Brodzik and O. Peters. Symbol-balanced quaternionic periodicity transform for latent pattern detection in DNA sequences. In *Proceedings. (ICASSP '05). IEEE*

International Conference on Acoustics, Speech, and Signal Processing, 2005., pp. 373– 376. Philadelphia, Pennsylvania, USA, IEEE, 2005.

[4] N. Chakravarthy, A. Spanias, L. D. lasemidis, and K. Tsakalis. Autoregressive modeling and feature analysis of DNA sequences,. *EURASIP JASP,*, 1:13–28, 2004.

[5] P.D. Cristea. Phase analysis of DNA genomic signals. *Proceedings of the International Symposium on Circuits and Systems, 2003. ISCAS '03.*, 5:V–25–V–28, 2003.

[6] Paul Dan Cristea. Large scale features in DNA genomic signals. *Signal Process.*, 83(4):871–888, April 2003.

[7] B. Liao. A 2d graphical representation of DNA sequence,. *Chem Phys Lett,*, 401:196– 199, January 2005.

[8] J. Ning, C.N. Moore, and J.C. Nelson. Preliminary wavelet analysis of genomic sequences. In *Bioinformatics Conference, 2003. CSB 2003. Proceedings of the 2003 IEEE*, pp. 509–510, Aug. 2003.

[9] N. Rao and S.J. Shepherd. Detection of 3-periodicity for small genomic sequences based on ar technique. In *Communications, Circuits and Systems, 2004. ICCCAS 2004. 2004 International Conference on*, volume 2, pp. 1032 – 1036 Vol.2, June 2004.

[10] G. L. Rosen. Signal processing for biologically inspired gradient source localization and DNA sequence analysis,. *PhD thesis, Georgia Institute of Technology,*, Aug. 2006.

[11] B. D. Silverman and R. Linsker. A measure of DNA periodicity. *Journal of Theoretical Biology*, 118:295–300, 1986.

[12] P. P. Vaidyanathan. Genomics and proteomics: A signal processors tour. *IEEE Circuits Syst. Mag*, 4:6–29, 2005.

[13] Richard F. Voss. Evolution of long-range fractal correlations and 1/f noise in DNA base sequences. *Physical Review Letters*, 68(25):3805+, June 1992.

[14] S. T. Yau, J. Wang, A. Niknejad, C. Lu, N. Jin, and Yee-Kin Ho. DNA sequence representation without degeneracy. *Nucleic Acids Research*, 31(12):3078–3080, June 2003.

[15] R. Zhang and C. T. Zhang. Z curves, an intuitive tool for visualizing and analyzing the DNA sequences. *J. Biomol. Struct. Dyn.*, 11(4):767–782, 1994.

[16] Zhu-Jin Zhang. DV-curve: a novel intuitive tool for visualizing and analyzing DNA sequences. *Bioinformatics*, 25(9):1112–1117, March 2009.

Annexure-1

Some popular accession numbers*

Sr. No.	Organism	Size (Approx.)	Description	Accession Number
1	Phage phiX174	5368 bp	1st Viral genome	NC_001422.1
2	Human mtDNA	16571 bp	1st Organelle genome	NC_012920.1
3	Lambda Phage	48502 bp	Important virus model	NC_001416.1
4	HIV	9193 bp	AIDS retrovirus	NC_001802.1
5	H. influenzae	1830 Kb	1st bacterial genome	NC_016809.1
6	M. genitalium	580 Kb	Smallest bacterial genome	NC_000908.2
7	S. cerevisiae	12.5 Mb total size	1st eukaryotic genome (Chromosome XV)	NC_001147.6
8	E. coli K12	4.6 Mb	Bacterial model organism	NC_000913.2
9	C. trachomatis	1042 Kb	Internal parasite of eukaryotes	AE001273.1
10	D. melanogaster	180 Mb	Fruit fly, model insect ((Chromosome 2L)	NT_033779.4
11	A. thaliana	125 Mb	Thale cress, model plant (Chromosome III)	NC_003074.8
12	H. Sapiens	3000 Mb total size	Human (Chromosome XI)	NT_009237.18
13	SARS	29751 bp	Coronavirus	NC_004718.3

* User can find more accession numbers from the entrez browser at NCBI.