Les Cahiers du GERAD ISSN: 0711–2440

GenoGraphiX-Log version 2.0 user guide

H. Usoof, C. Leblay, G. Caporossi

G–2020–68

December 2020

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2020
– Bibliothèque et Archives Canada, 2020

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2020
– Library and Archives Canada, 2020

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

GenoGraphiX-Log version 2.0 user guide

Gilles Caporossi [a]

Christophe Leblay [b]

Hakim Usoof [c]

[a] *GERAD & HEC Montréal, Montréal (Québec), Canada, H3T 2A7*

[b] *School of Languages & Translation Studies, University of Turku, FI-20014 Turku, Finland*

[c] *Department of Statistics & Computer Science, University of Peradeniya, 20400 Peradeniya, Sri Lanka*

hau@sci.pdn.ac.lk
christophe.leblay@utu.fi
gilles.caporossi@hec.ca

December 2020
Les Cahiers du GERAD
G–2020–68

Abstract : GenoGraphiX-Log 2.0 (abbreviation GGXLog) is a keystroke logging software that was developed as a collaboration between HEC Montréal, University of Turku, ITEM and GERAD. This project is mainly focused on combining Text Genetics (Greésillon, 1994) and Graph Theory (West, 1996) with Keystroke Logging in writing studies. Text genetics seeks to gather the material traces of the creative processes, to put them in relation to each other and to the products to which these traces lead; the idea is then to order them in a chronological sequence that reflects the stages of textual elaboration in order to identify the recurring elements and regularities. Graph Theory is a branch of mathematics which studies networks or vertices connected with edges; this can be used to represent data and relationships between the data. Furthermore, GGXLog emphases on using data visualisation (Sinar, 2015) in order to humanize writing data for learning and research. Data visualisation enables faster processing by the human brain, identifications of patterns, corelations, trends and allows the user to interact with the data. The GGXLog software is intended to be used by teachers, students, researchers and writing professionals.

GGXLog will record a writing session, store session data and analyse and visualize the session data. The programme stores data as a log (.log) file which can be loaded into the software for analysis and data visualization and as a tab-separated-value (.tsv) file for loading data into spreadsheet or statistical software. Furthermore, informant data, third-party application usage and the final text are stored as support data.

The software is based on the Java programming language and will work on Windows and Mac OS X operating systems.

Contents

List of Figures

List of Tables

1 Introduction

GenoGraphiX-Log 2.0 (abbreviation GGXLog) is a keystroke logging software that was developed as a collaboration between HEC Montréal, University of Turku, ITEM and GERAD. The idea of combining Text Genetics (Grésillon, 1994) and Graph Theory (West, 1996) with Keystroke Logging in writing studies was explored from one decade now (Caporossi and Leblay, 2011 ; Caporossi and Leblay, 2015 ; Bécotte *et al.* 2019), and proved to be a promising avenue. The current version is a step forward and aims at generalizing the usage of the methodology.

Text genetics seeks to gather the material traces of the creative processes, to put them in relation to each other and to the products to which these traces lead; the idea is then to order them in a chronological sequence that reflects the stages of textual elaboration in order to identify the recurring elements and regularities. Graph Theory is a branch of mathematics which studies networks or vertices connected with edges; this can be used to represent data and relationships between the data. Furthermore, GGXLog emphases on using data visualisation (Sinar, 2015) in order to humanize writing data for learning and research. Data visualisation enables faster processing by the human brain, identifications of patterns, corelations, trends and allows the user to interact with the data. The GGXLog software is intended to be used by teachers, students, researchers and writing professionals.

GGXLog will record a writing session, store session data and analyse and visualize the session data. The programme stores data as a log (.log) file which can be loaded into the software for analysis and data visualization and as a tab-separated-value (.tsv) file for loading data into spreadsheet or statistical software. Furthermore, informant data, third-party application usage and the final text are stored as support data.

The software is based on the Java programming language and will work on Windows and Mac OS X operating systems.

1.1 License, warranty, copyright & acknowledgement

1.2 Functionality

The three main functionalities of GenoGraphiX-Log 2.0 are:

I. Record writing session

GGXLog is capable of recording data in different writing contexts. The software natively supports free writing, translation and translation revision. Softcopy versions of text to be translated may be loaded onto the software for side by side view, and softcopies of translations may be loaded to the software for editing. All files that are loaded to the software must meet the following requirements:

- Text file
- Encoded as UTF-8

Note: Text files can be created or converted into the required form using Notepad on Microsoft Windows and TextEdit on Mac OS X.

II. Store session data

The session files are stored in a directory/folder in the location specified by the user. The default programmed directory is the Documents folder (see Section 4.0). The naming format of the specific session folder is "yyMMddHHmmss".

Example: *The folder name of a session started on 2020/09/08 at 13.23:38 would be 200908132338.*

Session data is stored in three main file formats:

- Log (*session_log.log*) file: Used by the software for analysis and data visualisation
- Session Data as a tab-separated-value (*session_data.tsv)* file: Used for loading the data into a spreadsheet or statistical software
- Final text of the session (*session_text.txt*): Can be opened in any text editor

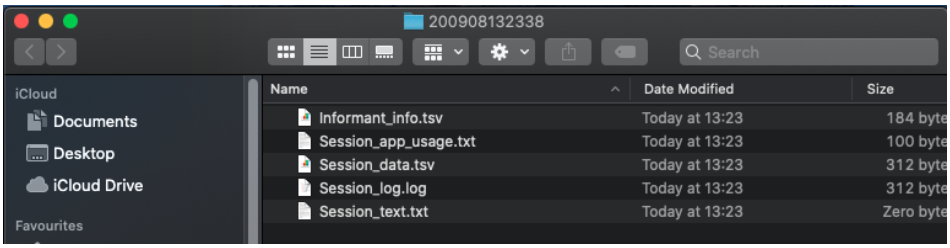


Figure 1.1: Session Data files

In addition to these three files, the software also creates two more files:

- Informant data (*Informant_info.tsv*) file: Can be loaded into a spreadsheet software
- Usage of 3rd party applications during the writing session (*session_app_usage.txt*): Can be opened in any text editor (see Section 5.3)

III. Analyse and visualise data

The primary analysis of data in the GGXLog software is based on Text Genetics and Graph Theory, with emphasis on the visualisation of the writing session as a graph. The developed graph is further analysed to identify patterns and statistics for study of the

session. However, the programme also provides other data representation formats and analysis used by writing researchers and linguists (see Sections 8.2 & 8.3).

1.3 Features

GenoGraphiX-Log 2.0 is designed to capture writing data, to store such data, and to process these data to produce data visualisations and analysis. Features of GGXLog are listed below:

I. Text Editor
This allows the recording of a writing session. The software natively supports three primary writing contexts:
- Free-writing (see Section 4.3)
- Translation (see Section 4.4)
- Editing Translation (see section 4.5)

In addition to the above writing contexts, the software can generate a real-time graph during the writing session for relatively short sessions (less than 10 minutes). In this setting, the graph will be drawn alongside the text editor in real time (see section 4.6).

II. Data Storage
- As LOG, TSV (tab-separated values) and Text files (see section 5.0)

III. Replay Session (see Section 7.0)

IV. Graph Representation (see section 8.1)
- Session Graph
- Graph Statistics
- Graph Pause Analysis
- Progressive Graph

V. Notations (see Section 8.2)
- Linear Text
- S-Notations

VI. Analysis (see Section 8.3)
- Statistics
- Burst and Efficiency
- Micro Level Analysis
- Edit Distance

VII. Export Data (see Section 9.0)
- GGXLog is capable of exporting visualisations, notations and analysis into common file formats such as MS Excel, MS Word, HTML, PDF (via Print option in web browser) and JSON.

1.4 System requirements

The specifications that GenoGraphiX-Log 2.0 has been tested on are given below. However, the software may run on older and lesser configurations without problems.

Intel or AMD 64-bit Processor 1.4GHz or above
4GB RAM
2GB Free Hard disk space
Screen resolution of 1366 x 768 or higher

Windows 10 or Mac OS X 10.13 or later
PDF Reader software and Safari, Google Chrome, Firefox or Opera web browser

Note: *The programme will run on lower configurations; however a higher performance configuration will be required if the real-time graph option is used. Running the real-time graph mode on lower performance computers may cause lagging and errors in the session log.*

1.5 Contact information

Website: www.ggxlog.net
Email: info@ggxlog.net

Hakim Usoof
Dept. of Statistics and Computer Science
University of Peradeniya
20400 Peradeniya, Sri Lanka
email: hau@sci.pdn.ac.lk

Christophe Leblay
School of Languages & Translation Studies
University of Turku, FI-20014 Turku, Finland
email: christophe.leblay@utu.fi

Gilles Caporossi
HEC Montréal, 3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec), Canada H3T 2A7
email: gilles.caporossi@hec.ca

2 Installation

Currently, installers are available for Microsoft Windows and Mac OS X. The Windows and Mac OS X versions of the software come bundled with Java 9.0.4 hence it does **not require** separate installation of Java.

The GGXLog2.jar file for running the application on Linux/Unix environments can be made available on request. However, the Unix/Linux environments will require Java RE 9.0.4 preinstalled. The 3rd party application monitoring function is not implemented for Unix/Linux environments.

2.1 Microsoft Windows

In order to install GGXLog on your Windows computer:

1. Visit www.ggxlog.net and download the GGXLog2.exe file. The GGXLog2.exe will likely be saved to the Downloads Folder or another specified location.
2. Double click on the **GGXLog2.exe** file.

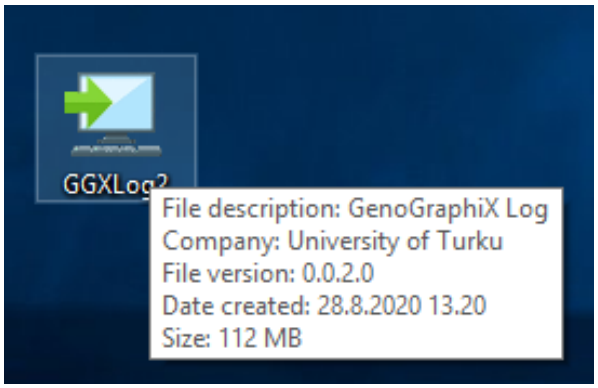


Figure 2.1: GGXLog2.exe installation file

Note: If the following Warning appears, click **More info** and Click **Run anyway**

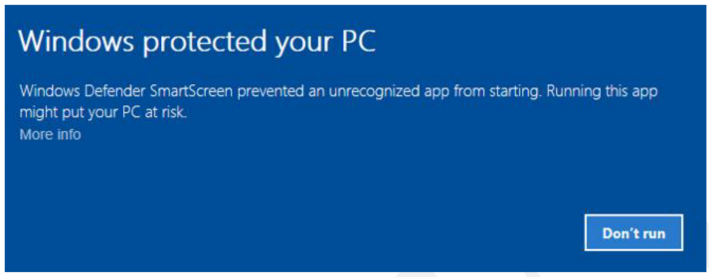


Figure 2.2: Windows Security Message

3. The default option is to overwrite any existing installations of GenoGraphiX-Log on the computer. Click on **Next.**

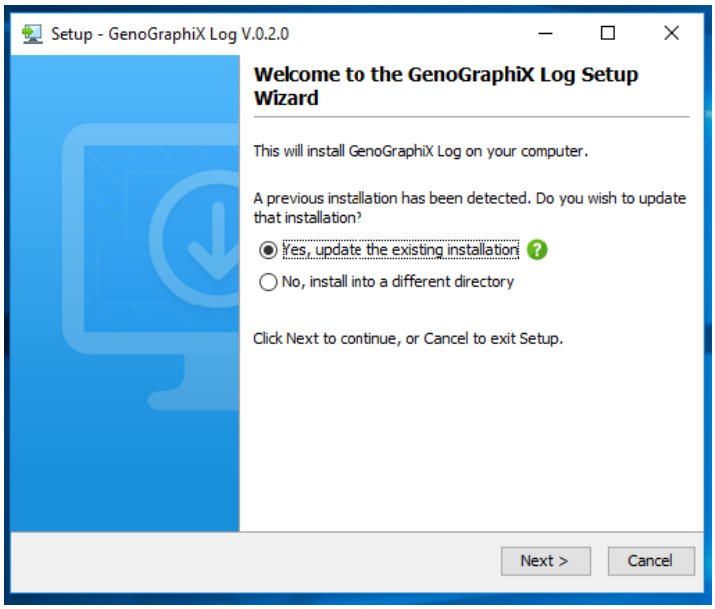


Figure 2.3: Installation prompt for Windows

4. The programme files will be copied into the programme directory as below.

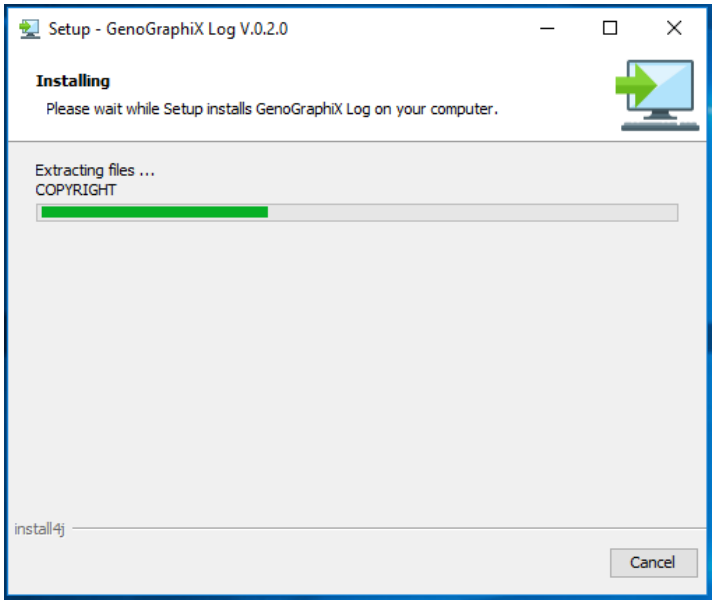


Figure 2.4: Windows installation progress

5. When all files are copied, click on **Finish.**

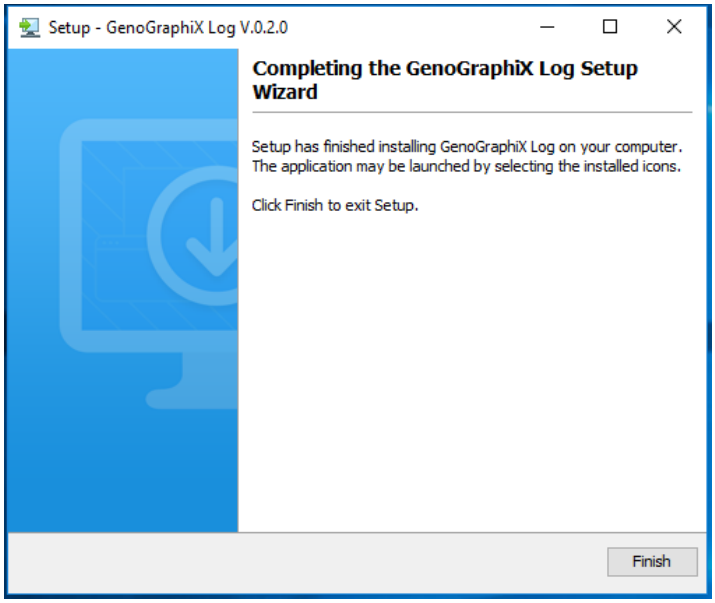


Figure 2.5: Windows installer completion

6. To launch GenoGraphiX-Log 2.0, click on the **Start Menu**, then click on **GGXLog2**.

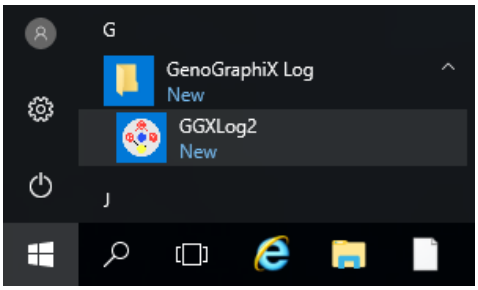


Figure 2.6: Launch GGXLog2 from Start Menu

2.2 Mac OS X

In order to install GGXLog on your Mac computer:

1. Visit www.ggxlog.net and download the GGXLog2.dmg file. The GGXLog2.dmg will likely be saved to the Downloads Folder or another specified location.
2. Double click on the **GGXLog2.dmg** file, then double click on the **GenoGraphiX-Log Installer.**

Figure 2.7: GGXLog2 Mac OSX installer

3. The default option is to overwrite any existing installations of GenoGraphiX-Log on the computer. Click on **Next.**

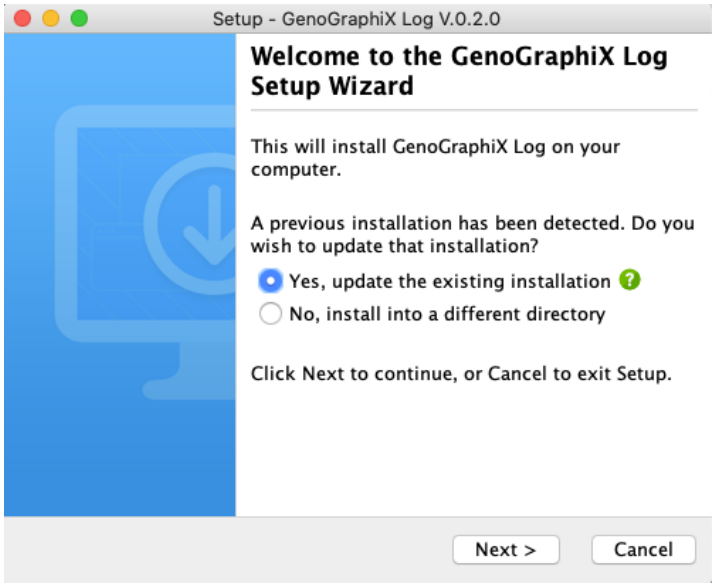


Figure 2.8: installation prompt for Mac OSX

4. The programme files will be copied into the programme directory as below.

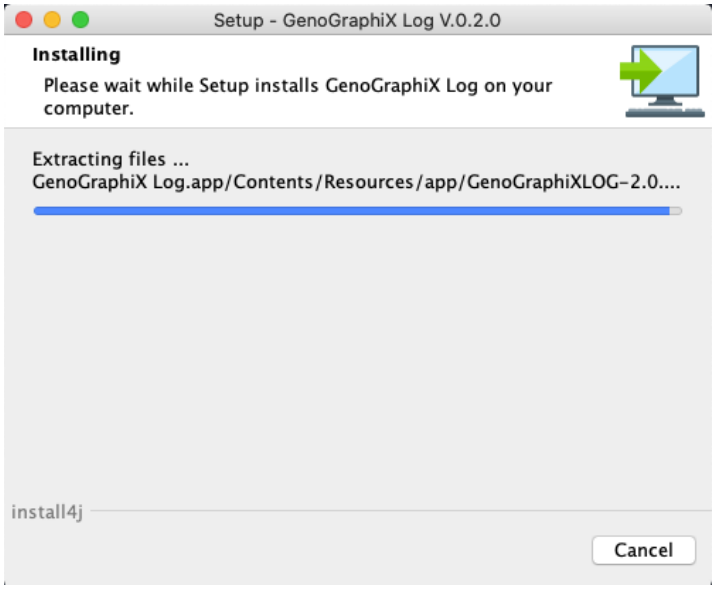


Figure 2.9: Mac OSX progress

5. When all files are copied, click on **Finish.**

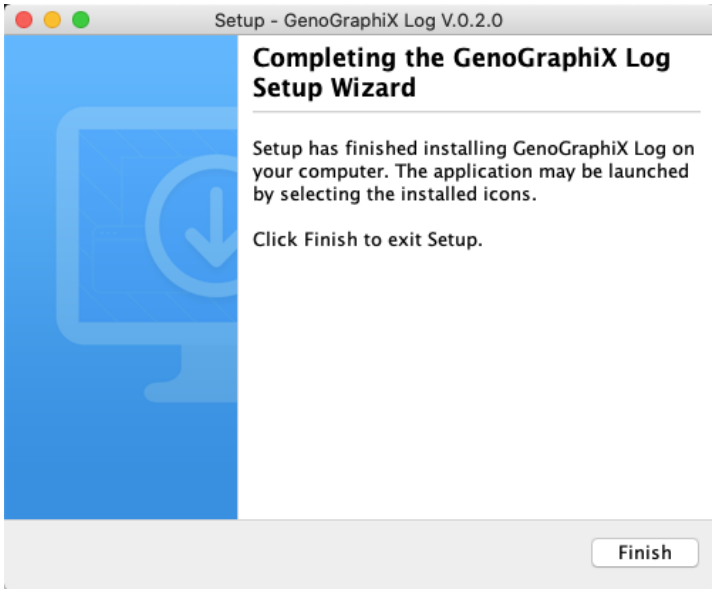


Figure 2.10: Mac OSX installer completion

6. To launch GenoGraphiX-Log 2.0, click on **Launchpad**  **or the Applications folder**, then click on **GenoGraphiX Log.**

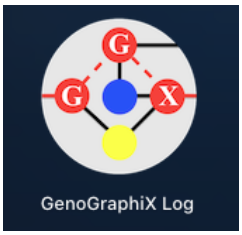


Figure 2.11: Icon from Launcher or Application folder

3 Interface

The interface is designed according to common interface design standards. The interface has five primary areas:

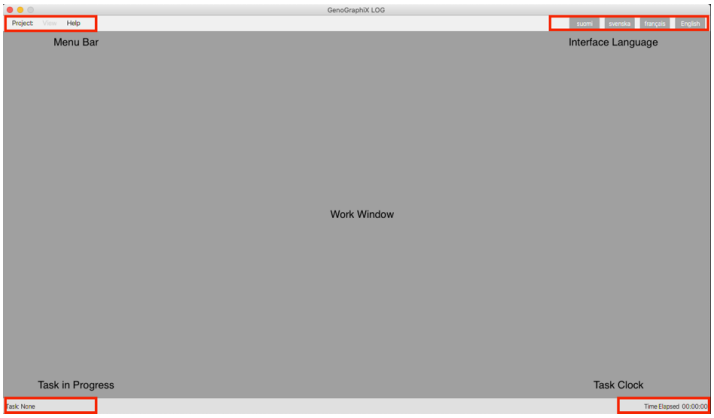


Figure 3.1: Main Window of GGXLog

1. **Main Menu** [Top left-hand corner]
   The Main Menu allows the user to select different features offered by the software. It is a hierarchical menu that permits **Project** creation and opening previous projects, **Viewing** analysis and visualisations of session data and accessing links for **Help** with the software.

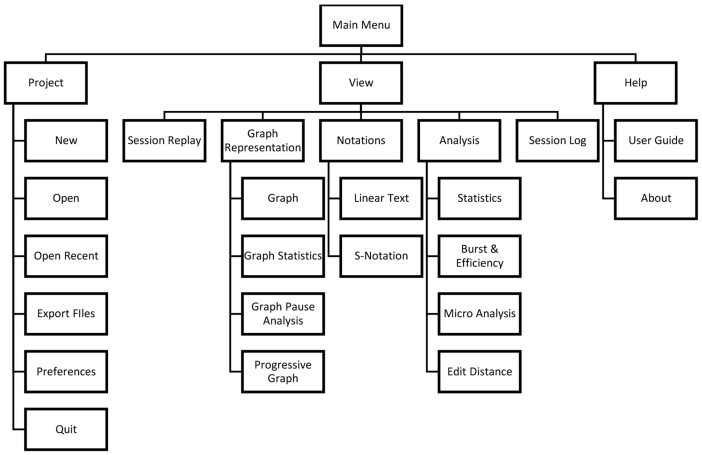


Figure 3.2: Main menu structure of GGXLog

During a writing session, all menus except **Help** will be disabled to safeguard against accidental exiting of the session without saving data. The **View** menu will enable only once the session log file is loaded into the programme.

2. **Interface Language Selection** [Top right-hand corner]
   Currently the software supports Finnish, Swedish, French and English. Other interface languages may be added in the future. To change the interface language, click on the flag.
3. **Work Window** [Centre]
   The work area is where the editor for the writing session is loaded. This window also allows for analyses to be viewed and session data to be visualized.
4. **Task in Progress** [Bottom left-hand corner]
   The task notification functions during a writing session or a session replay. It will display the current task being performed. During a writing session, it will display if the session is being recorded or paused. During replay, it will display the Player status.
5. **Task Clock** [Bottom right-hand corner]
   The task clock functions during a writing session. During a writing session, it will display the time elapsed since the click of the **Record** button. If the writing session is paused, the task clock will also be paused.

4 Logging a writing session

To start a logging session:

1. Click on **Project** menu, then select **New.**

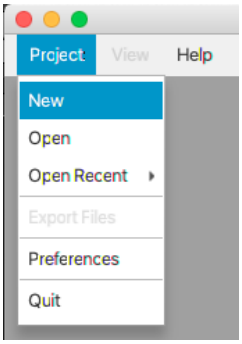


Figure 4.1: Project Menu

2. The Participant Information and Consent Form window will appear. Select the appropriate Project Settings for the planned writing session.

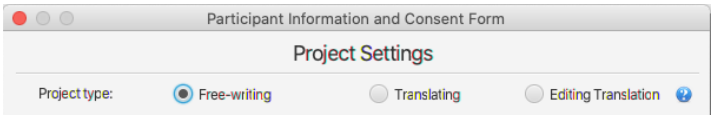


Figure 4.2: Project type selection

GGXLog supports three writing contexts or Project Types:

- **Free-writing**: This is intended for writing original text.
- **Translating**: This is intended for translating texts and provides the option of loading a softcopy of the source text onto the interface for convenient reading.

- **Editing Translation**: This is intended for editing already translated text. This option allows for loading softcopies of both the original source text and the translation onto the interface. The editing is done on the translation.

3. Select estimated text length. This option will restrict the use of Real-time Graph during a logging session. A short text is estimated to be approximately 10 minutes of writing.

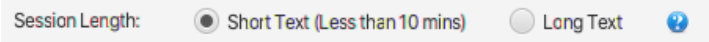


Figure 4.3: Expected writing session length

4. Select whether the Real-time Graph option should be on or off, in the scenario of a Short Text.

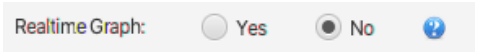


Figure 4.4: Realtime Graph option

WARNING! The Real-time Graph requires considerable computation power, which on a less capable computer may lead to lagging and errors in the session log. This may be used for short texts to help understand the writing process using graph visualisations.

5. Select the location where session files will be stored. The default location set in Preferences of the programme is typically the Documents folder.

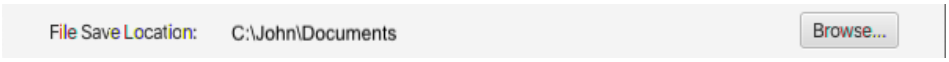


Figure 4.5: Set file save location

6. Fill in the Informant Information fields.

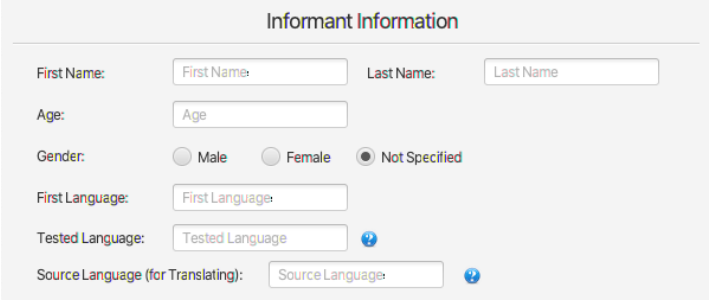


Figure 4.6: Informant information collection form

In this form:

- **First Language**: First language or native language of the informant
- **Tested Language**: The writing language
- **Source Language**: The language of the source text (applicable only for Translating and Editing Translations)

7. Read Informed Consent Form. If the writer agrees, tick **I grant consent.**
8. Click on the **Continue** button.

4.1 General process of writing

The writing interface is controlled by three buttons:

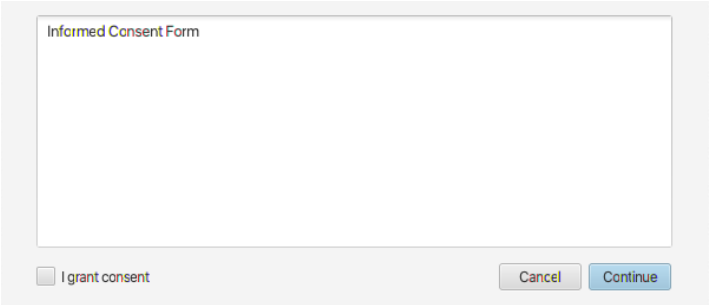


Figure 4.7: Informed Consent Form

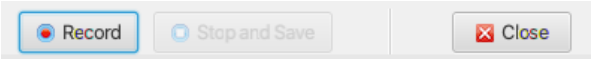


Figure 4.8: Record session option

- **Record/Pause**: Clicking Record starts the recording of the writing session. The text editing area is only enabled when **Record** is clicked. Clicking **Pause** will pause the writing session and clicking **Record** again will restart recording. All Project and View menus and the **Close** button are disabled when **Record** is clicked and only enabled again when **Stop and Save** is clicked.

Figure 4.9: Pause and Save session option

- **Stop and Save**: This button is enabled when recording a session. When clicked, the recording of the session is stopped and the session files are saved. To complete the session, confirmation must be given by clicking **Yes** when prompted by the Confirmation dialog box.

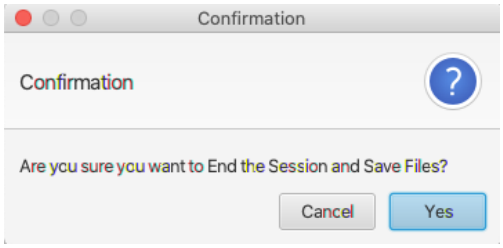


Figure 4.10: Confirmation for Save Session

- **Close**: This button is enabled only before starting a recording of a writing session. Clicking on it will exit the text editor interface. To exit the text editor, confirmation must be given by clicking **Yes** when prompted by the Confirmation dialog box.

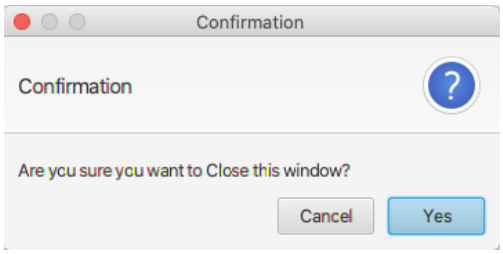


Figure 4.11: Confirmation for closing window

4.2 Support for special characters

GGXLog supports French **aigu**, **grave**, **circonflexe**, and **tréma** accents, which can be typed using dead keys if they are not possible as a single keystroke. The cedilla/cedille (Ç) can be typed on Microsoft Windows using **Ctrl+COMMA** and on Mac OS X **COMMAND+COMMA**, followed by **c** or **Shift+c** for capitalisation.

Dead keys are special modifier keys on the keyboard that typically associate an accent or diacritical sign to a basic letter. For example, the dead key acute (´) followed by the letter a (a) will produce á. Dead keys are dependent on the layout of the keyboard and keyboards may include only a few dead keys. GGXLog supports the following 16 dead keys;

"DEAD_ACUTE","DEAD_DIAERESIS", "DEAD_GRAVE", "DEAD_CEDILLA", "DEAD_TILDA", "DEAD_CIRCUMFLEX", "DEAD_ABOVEDOT", "DEAD_ABOVERING", "DEAD_BREVE", "DEAD_CARON", "DEAD_DOUBLEACUTE", "DEAD_IOTA", "DEAD_MACRON", "DEAD_VOICE_SOUND", "DEAD_SEMIVOICE_SOUND" and "DEAD_OGONEK"

4.3 Free-writing

The Free-writing option allows the writer to create original text. This would be the most common option used for writing and common writing research scenarios. The writing is done in Text Editor marked in red in the figure below.

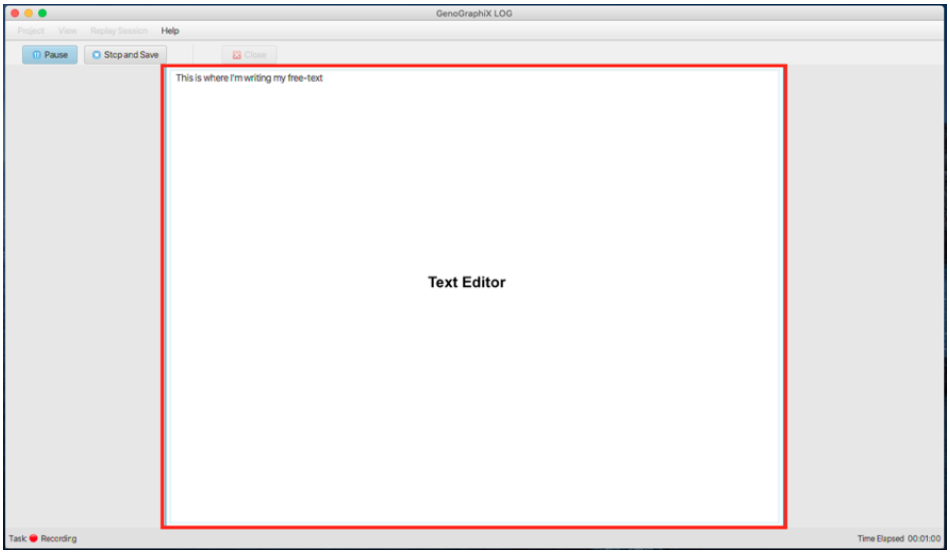


Figure 4.12: Free-writing session interface

4.4 Translation

The Translation option allows the writer to display a source text on the interface, which can then be translated side-by-side in the text editor. The source text is displayed in the area marked in green and the translation is done in Text Editor marked in red in the figure below.

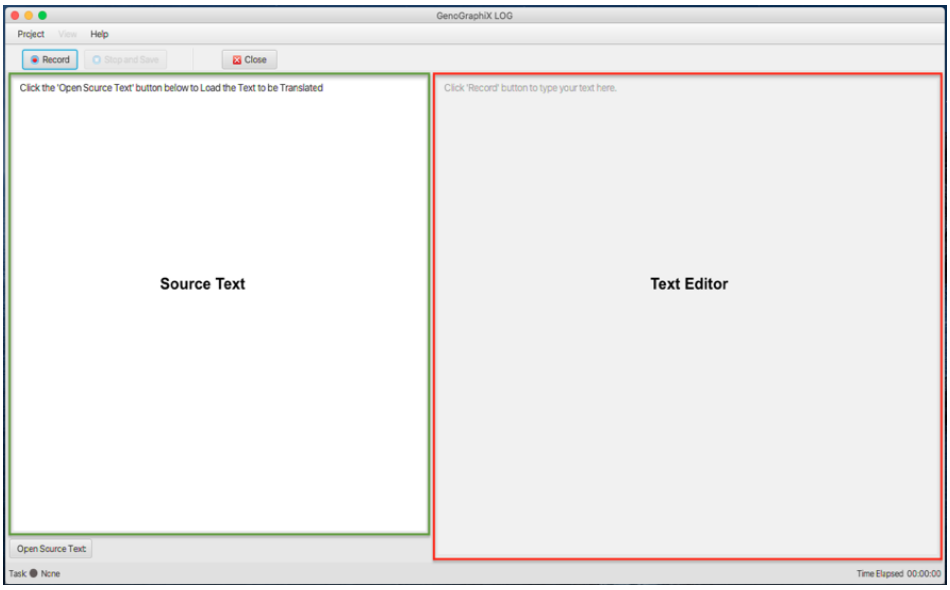


Figure 4.13: Translation session interface

To open a source text with the original text to be translated:

1. Click on the **Open Source Text** button in the bottom left-hand corner. It is important to note that only text files encoded in UTF-8 are supported by GGXLog.

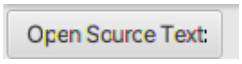


Figure 4.14: Open Source Text button

2. Select and open the source text from the **File Selection** window.

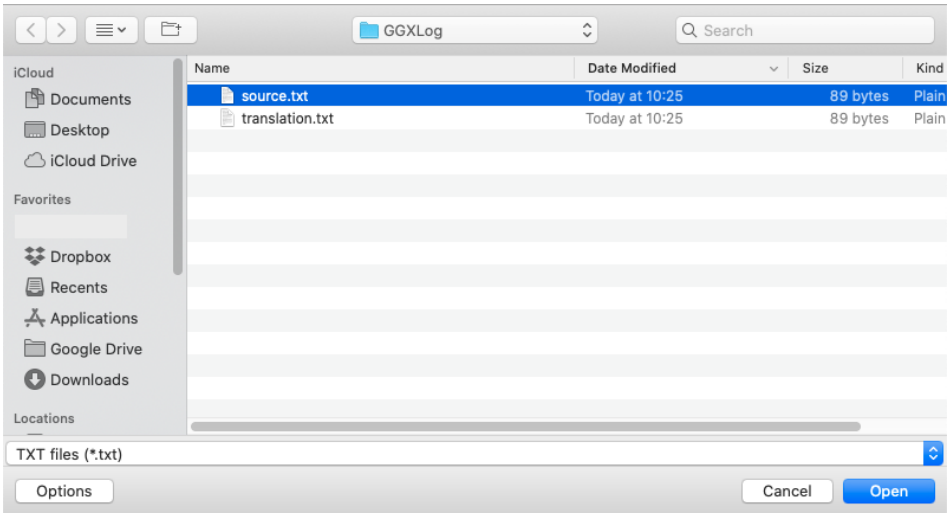


Figure 4.15: Source text file selection window

4.5 Editing Translation

The Edit Translation option allows the writer to display the original source text that was translated on the interface and the already translated text side-by-side in the text editor. The already translated

text can be edited and the editing session is recorded. The source text is displayed in the area marked in green and the editing of the translation is done in the text editor marked in red in the figure below.

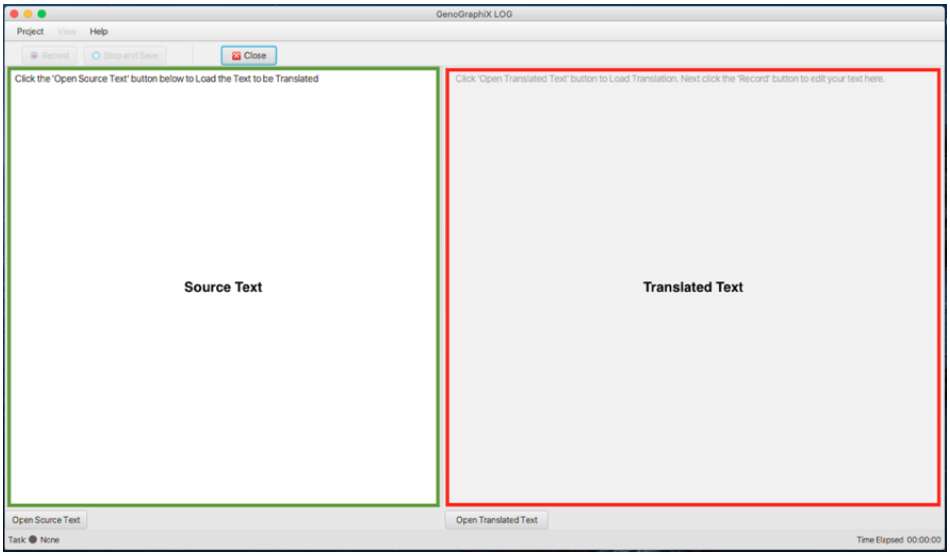


Figure 4.16: Translation Editing session interface

To open a source text with the original text that was translated:

1. Click on the **Open Source Text** button in the bottom left-hand corner.

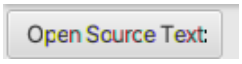


Figure 4.17: Open Source Text button

2. Select and open the source text from the **File Selection** window.

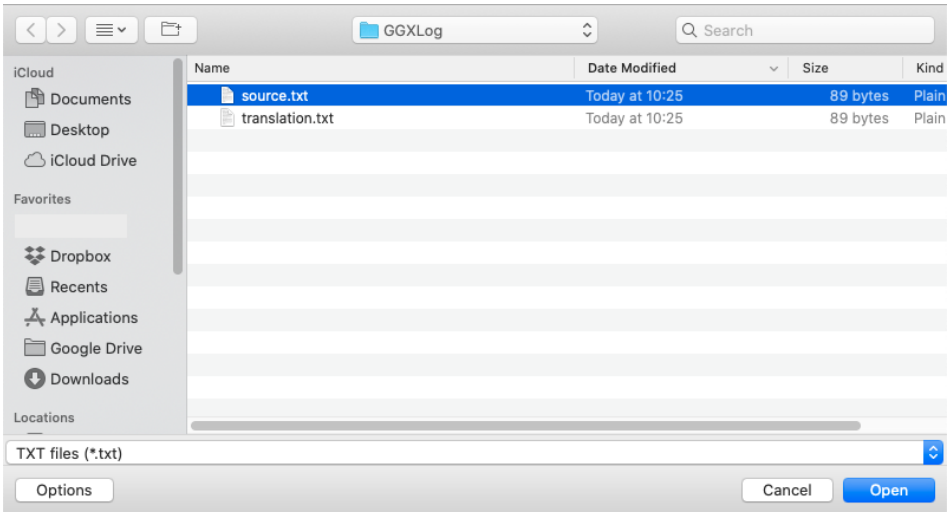


Figure 4.18: Source text file selection window

To open the translated text:

1. Click on the **Open Translated Text** button in the bottom-middle. As accuracy is key in GGXLog, it is imperative that the translated text file be encoded in UTF-8 encoding.

Figure 4.19: Open Translation text button

2. Select and open the translated text from the **File Selection** window.

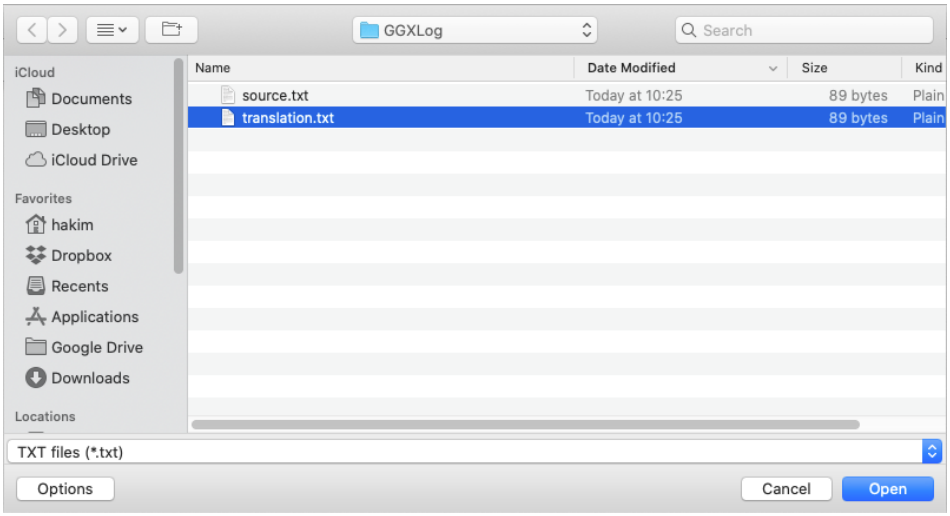


Figure 4.20: Translation text file selection window

Note: The **Record** button will only be enabled once a translation text is loaded into the text editor. Once the recording of the session is started, the **Open Translated Text** button will be disabled.

4.6 Real-time Graph

The Real-time Graph visualises the ongoing writing session as a graph adjacent to Text Editor. The Real-time Graph may be used for short texts, preferably for writing sessions lasting less than 10 minutes. Longer text sessions may cause lagging and errors in the session log. The Real-time Graph option is available for Free-writing and Translation project types.

To disable Real-time Graph during the writing session, untick the Real-time Graph checkbox.

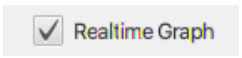


Figure 4.21: Realtime graph enable-disable button

Real-time Graphs will be shown as follows in the Free-writing interface:

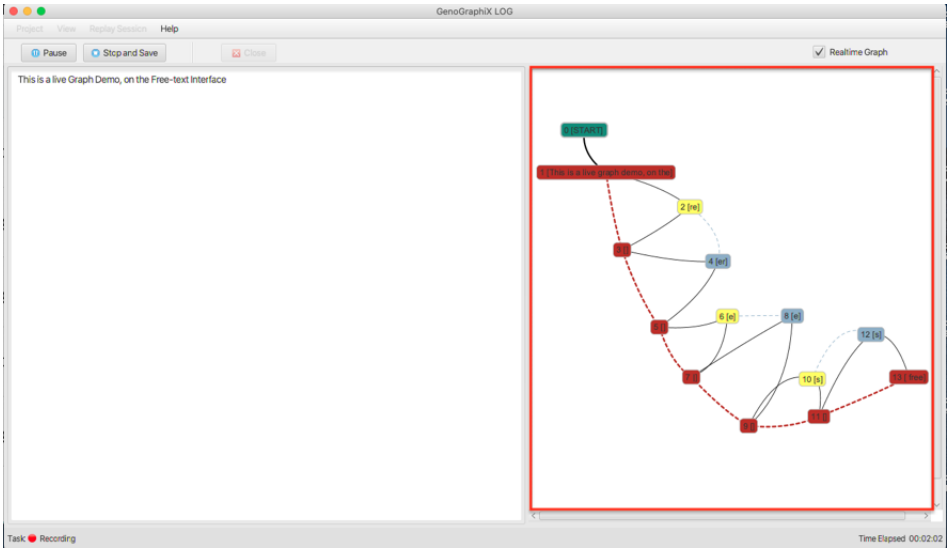


Figure 4.22: Realtime Graph in Free-writing interface

Real-time Graphs will be shown as follows in the Translation interface:

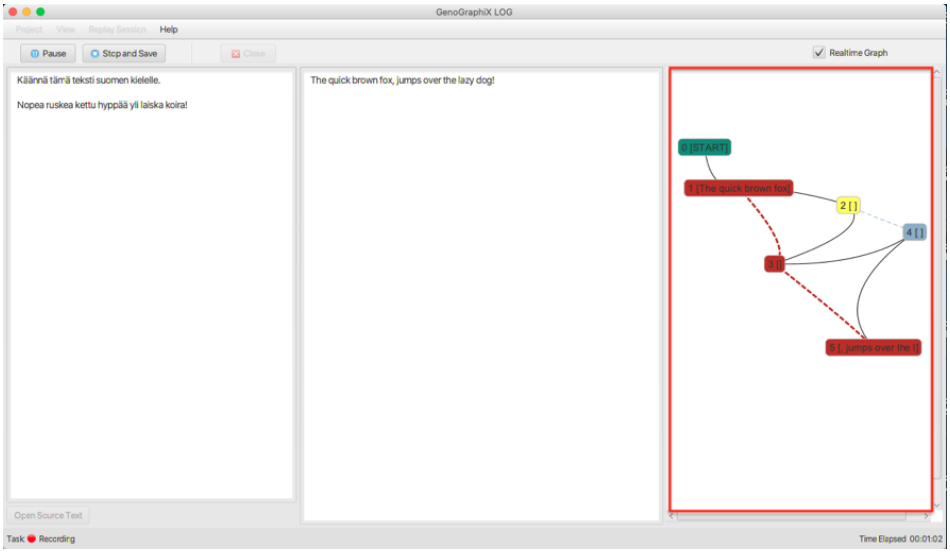


Figure 4.23: Realtime graph in Translation interface

4.7 3rd Party Application Logging

GGXLog is capable of recording 3rd party applications during the writing session. If the writer uses a 3rd party application, for example a web browser or a Word document, GGXLog will record the name of the 3rd party application, the title of the page and the corresponding index number in the *Session_log.log* file. This data is stored in the *Session_app_usage.txt* file. This event can be identified using the input label <APP_CHANGE> in the session log. This option will keep recording 3rd party application usage even if the writing session is paused using the **Pause** button.

4.8 Session pause logging

When the writing session is paused using the **Pause** button, GGXLog logs this event in the *Session_log.log* file. This event can be identified by the input label <SESSION_PAUSE> in the session log.

5 Data storage and session files

A session file is stored in the session directory. Each session will create a new directory with a timestamp with the format Year:month:date:hour:minute:second (yyMMddhhmmss). This is also the Session ID.

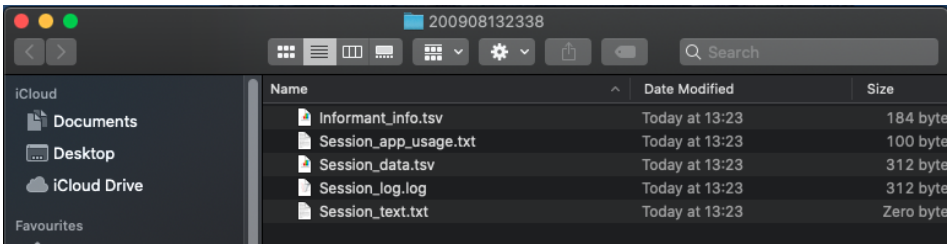


Figure 5.1: Session data files

A writing session creates five files:

1. Session Log: Stores the session data in .log format and is the data source opened by GGXLog

   WARNING! The session_log.log file should not be edited in other applications as this would cause errors when the file opened in GGXLog software.
2. Session Data: Stores the data in .tsv format and is identical to the data stored in the *Session Log* file
3. Informant Information: Stores information about the writer
4. Session App usage: Stores data of any 3rd party applications that are used by the writer during the session
5. Session Text: Stores the final text from the text editor when the **Stop and Save** button is clicked on at the end of the writing session

5.1 Session Log and Session Data

Session data is stored in three main file formats:

- **Session Log** (*session_log.log*) is the file used by the software for analysis and data visualisation. The Session Log consists of the following data:
  - **Session ID**: Same as the timestamp in the session directory
  - **Data and Time**: Date and time when the *Record* button is clicked
  - **Operating System**: The operating system the writing session was done on
  - **Locale**: The operating system language and the keyboard layout used
  - **Version**: Version of the GGXLog software
  - **Session Data**:
    - Index: User Input Event number
    - Device: Input device (e.g. Keyboard, Mouse, System)
    - Input: Input (e.g. Character, Mouse Click)
    - Start_Time: Input start time (e.g. Key press time)

* Start_clock: *Start_Time* in hour:minute:second format
* End_Time: Input end time (e.g. Key release time)
* End_clock: *End_Time* in hour:minute:second format
* Event_Time: Duration of the event (Duration between *Start_Time* and *End_Time*)
* Pause_Time: Duration between the end of an Input Event and the beginning of the next Input Event
* Pause_location: The location of the pause (e.g. within word, between words)
* Caret_position: Location of the cursor when the Input Event occurs
* Anchor_position: Position of the selection point (e.g. when a text is selected the selection would be between *Anchor_position* and *Caret_position*)
* Mouse_x: x-coordinate of the Mouse pointer in Text Editor
* Mouse_y: y-coordinate of the Mouse pointer in Text Editor
* Selected_text: The selected/highlighted text
* Replaced_text: The text that is replaced by another text (e.g. highlighted and typed over or pasted over)
* Gene_type: The genetic operation performed on the text

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Session ID: 310820104156, Date & Time:<START> :31-08-2020 10:41:56																
2																	
3	Log Operating System	mac	Locale	en_US	Version	0.2.0 (Beta)											
4	Index	Device	Input	Start_time	Start_clock	End_Time	End_clock	Event_time	Pause_time	Pause_location	Caret_position	Anchor_position	Mouse_x	Mouse_y	Selected_text	Replaced_text	Gene_type
5	0	<MOUSE>	<PRIMARY>	0	00:00:00	59	00:00:00	59	362	initial	0	0	145	57			navigate
6	1	<KEYBOARD>	SHIFT	1382	00:00:01	1635	00:00:01	253	1323	unknown	0	0					unknown
7	2	<KEYBOARD>	T	1515	00:00:01	1558	00:00:01	43	133	sentence beginning	0	0					append
8	3	<KEYBOARD>	h	1920	00:00:01	1964	00:00:01	44	362	within word	1	1					append
9	4	<KEYBOARD>	i	2311	00:00:02	2384	00:00:02	73	347	within word	2	2					append
10	5	<KEYBOARD>	s	2398	00:00:02	2491	00:00:02	93	14	within word	3	3					append
11	6	<KEYBOARD>	SPACE	2445	00:00:02	2563	00:00:02	118	47	between words	4	4					append
12	7	<KEYBOARD>	i	2611	00:00:02	2728	00:00:02	117	48	between words	5	5					append
13	8	<KEYBOARD>	s	2715	00:00:02	2775	00:00:02	60	104	within word	6	6					append
14	9	<KEYBOARD>	SPACE	2788	00:00:02	2894	00:00:02	106	13	between words	7	7					append

Figure 5.2: Session data log file format

- **Session Data** (*session_data.tsv)* is a tab-separated values file that is intended to be used for loading the data into a spreadsheet (e.g. MS Excel or Numbers), statistical software or text editor. The data in this file is identical to the session_log.log file.
- **Final Text** of the session (*session_text.txt*) stores the text developed during the writing session. The text in Text Editor at the time the *Stop and Save* button is clicked is saved in this file. This is a text that can be opened in any text editor or word processing software.

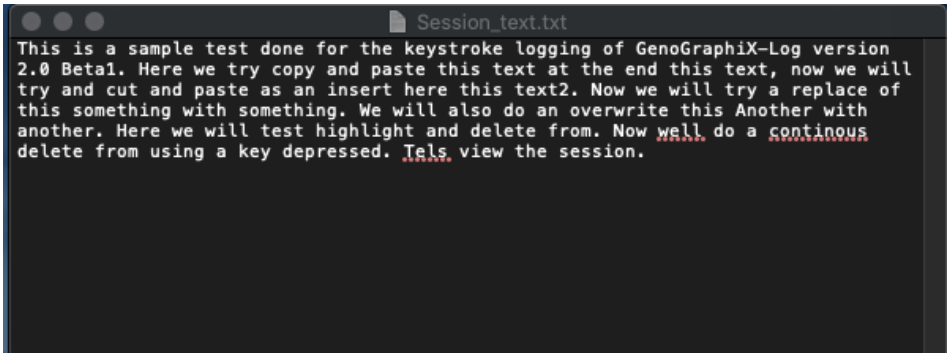


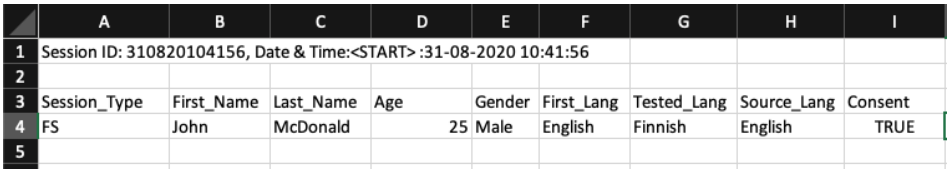
Figure 5.3: Session final product

5.2 Informant Data

The Informant Data (*Informant_info.tsv*) file stores the data entered by the writer at the beginning of the writing session. The informant data is stored as tab-separated values and can be loaded into a spreadsheet software or text editor.

The Informant Data file consists of the following data:

- **Session ID**: Same as the timestamp in the session directory
- **Data and Time**: Date and time when the *Record* button is clicked
- **Informant Data**:
  - Session_Type: A code created for the Project Type
    - The code is as follows:
      - The 1st character
        F: Free-writing, T: Translation and E: Edit translation
      - The 2nd character
        S: Short Text and L: Long Text
      - The 3rd character
        G: Real-time Graph enabled
    - E.g. FS: Free-writing, Short Text
  - First_Name
  - Last_Name
  - Age
  - Gender
  - First_Language: Native language of informant
  - Tested_Language: Session writing language
  - Source_Language: In the case of a translation or editing translation, the language of the source text
  - Consent: TRUE if the Informant agrees to the Consent form and ticks the "I grant consent" checkbox



	A	B	C	D	E	F	G	H	I
1	Session ID: 310820104156, Date & Time:<START> :31-08-2020 10:41:56								
2									
3	Session_Type	First_Name	Last_Name	Age	Gender	First_Lang	Tested_Lang	Source_Lang	Consent
4	FS	John	McDonald	25	Male	English	Finnish	English	TRUE
5									

Figure 5.4: Informant information file format

5.3 3rd Party Application Usage

The Session App Usage (*session_app_usage.txt*) file stores the information of 3rd party applications used by the writer during the writing session (see Section 4.7). This is a text that can be opened in any text editor or word processing software.

Session App Usage file consists of the following data:

- **Session ID**: Same as the timestamp in the session directory
- **Data and Time**: Date and time when the *Record* button is clicked
- **App Data**:
  - Log Index: This number corresponds to the index of the *Session Log* file (e.g. Log Index 54 in the *Session App Usage* file corresponds to the 54th index in the *Session Log* file)
  - Application: The name of the application or software used
  - Title: The title of the document open in the application

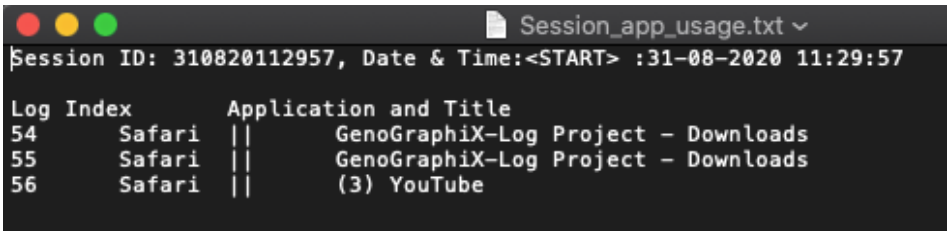


Figure 5.5: 3rd part application usage log format

53	<KEYBOARD>	SPACE	17378	00:00:17	17452	00:00:17	74	166
54	<SYSTEM>	<APP_CHANGE>	22071	00:00:22	22071	00:00:22	0	4619
55	<SYSTEM>	<APP_CHANGE>	36560	00:00:36	36560	00:00:36	0	14489
56	<SYSTEM>	<APP_CHANGE>	51622	00:00:51	51622	00:00:51	0	15062
57	<KEYBOARD>	SHIFT	62120	00:01:02	62405	00:01:02	285	10498

Figure 5.6: 3rd Party application usage as an event in the Session Log

6 Open project

To review a session, the session log file should be opened in the software.

To open a session log file:

1. Click on the **Project** menu, then select **Open.**

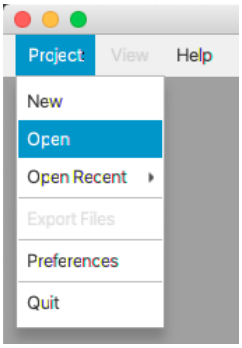


Figure 6.1: Project menu to Open existing session log

2. From the File Open window choose the session log file with the .log extension.

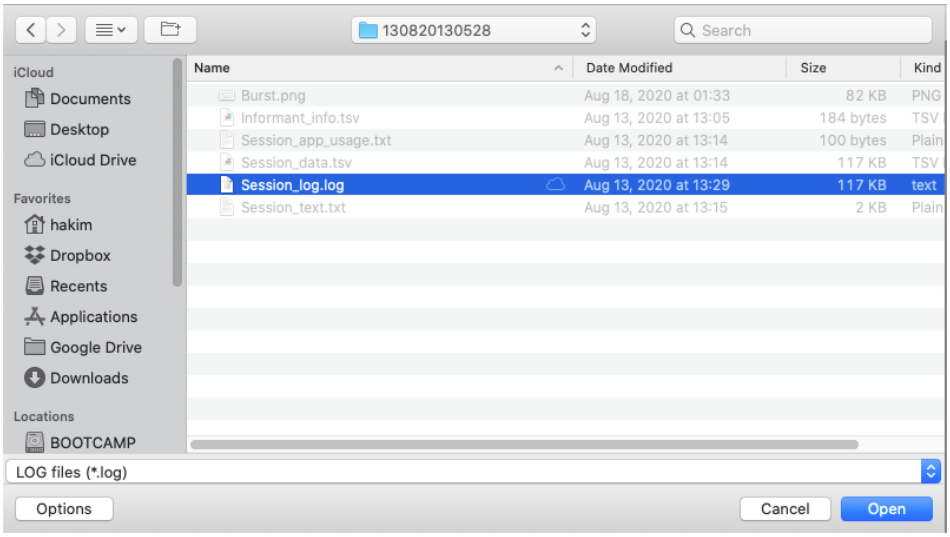


Figure 6.2: Session Log file selection window

3. Recently opened files may be directly opened by clicking on the file name in the **Open recent** menu.

When the session log is loaded to the software, the Session Replay interface will be displayed.

7 Session Replay

Session Replay is the default interface of the **View** menu. Replaying the session allows for observation of the writing session as seen on Text Editor. The Session Replay interface consists of five components:

1. Open File
   - Shows the file name and path of the currently opened file.
2. Controls
   - **Play/Pause**: Play and pause a writing session
   - **Start**: Go to the beginning of the session
   - **End**: Go to the end of the session
   - **Fast**: Increase the replay speed up to 500%
   - **Slow**: Decrease the replay speed down to 50%
   - **Close**: Close the open project
3. Progress Slider
   - The Progress Slider can be used to navigate to different points in the writing session.
4. Session Text
   - This window is where the Session Text will be displayed, similar to Text Editor during the writing session.
5. Time Elapsed/Session Duration
   - This shows the time elapsed in the replay of the writing session and the total duration of the entire writing session.

6. Status
    - The Status indicates Play/Pause, Replay Speed, etc.

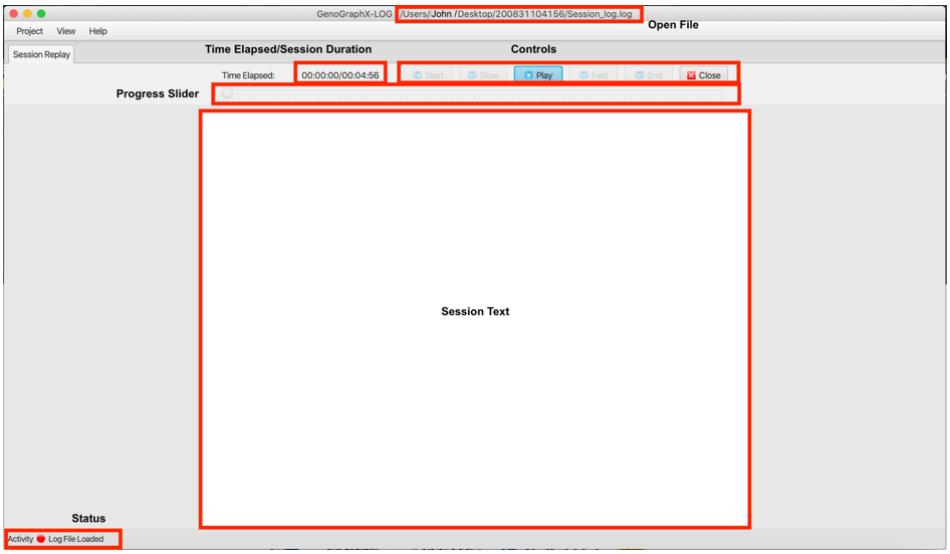


Figure 7.1: Session Replay interface

8 Session Analysis and Data Visualisation

The opened project can be viewed in different formats and aspects from the **View** menu. The views are categorised into five formats:

1. Replay Session: Allows for the replay of the writing session (see Section 7.0)
2. Graph Representation (see Section 8.1)
    - Session Graph: Visualises the writing session using **Graph Theory**
    - Graph Statistics: Provide numerical statistics and visual summaries of the Session Graph
    - Progressive Graph: Constructs the Session Graph in a progressive manner based on key events during the writing session
3. Notations (see Section 8.2)
    - Linear Text: Depicts the writing session using two notational formats, which summarises the sequence of events during the writing session using text and symbols
    - S-Notation: Is a representation by Py Kollberg (1996) used for analysing revisions and text production during the session
4. Analysis (see Section 8.3)
    - Statistics: Provides basics statistics for the session
    - Burst and Efficiency: Provides an analysis of bursts and efficiency between interruptions
    - Micro Analysis: Provides a micro-level analysis of events and text from the writing session
    - Edit Distance: Provides an analysis of the distance between two edits
5. Session Log: Displays the filtered set of fields from the session log file (see Section 8.4)

To select a particular view:

1. Click on the **Project** menu, then on **Open.**

2. From the File Open window, choose the relevant session log file with the .log extension (usually *Session_log.log* file).
3. Click on the **View** menu, then on the required view option.
4. If the interface consists of a Settings button, click on **Settings** to specify parameters for the view (this will enable the Process button and other options on the interface).
5. Click on **Save Settings.**

8.1 Graph representation

The graph representation in the GenoGraphiX-Log software uses Text Genetics and Graph-Theory from the domain of mathematics to represent a writing session as a set of **nodes** (vertices) and **edges** (links). A node represents a cluster of continuous events in the writing session (e.g. a sequence of characters inserted in a text, a sequence of characters appended to a text, a sequence of characters deleted from a text, or a pause in the writing). An edge represents a relationship between these nodes (e.g. chronological order, the flow with respect to the final text, or a removed text and the deletion).

Nodes are colour coded to represent different categories. The graph representation has 7 types of nodes:

1. **Append**: A node created when text is added at the end of the current text
   a. **Non-empty nodes**: Nodes that contain text
   b. **Empty nodes**: Nodes without text that are created during editing events and required to maintain graph structure
2. **Insert**: A node created when text is inserted between the beginning and the end of the text
3. **Removed**: A text or node removed from the original text
4. **Deletion**: A node containing the entire deleted text
5. **Displaced**: A node appearing when a text is moved from one place to another
6. **Copy**: A node created when copying text from one place to another
7. **Pause**: A pause with a duration greater than or equal to the specified cognitive pause duration (see Sections 8.1.2 and 10.0 on setting the cognitive pause duration)

Edges are colour coded to represent categories of relationships. A continuous line will represent chronological order while a discontinuous line will represent a relationship. Thus, a colour coded line may represent both chronology and relationship or just one of them. Edges with an arrow and a similar colour as an **Append** node represents the flow of the (final) produced text.

8.1.1 Graph visualisation of writing

Graph visualisation is a combination of graphical representations of scenarios that may occur during a natural writing session. The patterns help to easily identify which scenario occurred at a given instance and how it affected the text as well as its relationship to other events. A series of writing scenarios are exemplified below:

I. **Node 0** or the **[START]** node is the starting position of the Graph.

Figure 8.1: Graph Start node

II. An **Append** at the end of a text is depicted as below.

Figure 8.2: Graph Append node example

III. An **Insert** is depicted below. An insert that is done in a text will split an existing node into two parts with consecutive numbers (nodes 1 and 2), and the insert will have a greater node number (node 3). While chronological order will be maintained in a continuous line, the flow of produced text will include the inserted node between the split nodes. In this example, the initial text was "*This is a text.*" and the word "*simple*" was inserted to create "*This is a simple text.*".

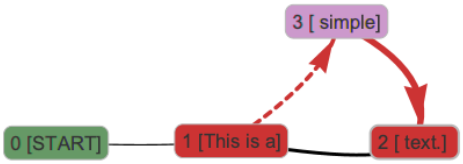


Figure 8.3: Graph Insert node example

IV. A **Delete** done using the *backspace* key is depicted below. The initial text has a word deleted; this splits the single node into three parts (nodes 1, 2 and 3) that will maintain the initial chronological order of the text. The produced text flow will now exclude the removed text (node 3), which will typically be coloured in a different shade. In addition, the deletion event (node 4) is now included in the order of chronology to emphasise when the deletion occurred. Furthermore, the reverse order of text in the deletion node represents that the backspace key was used for deletion.

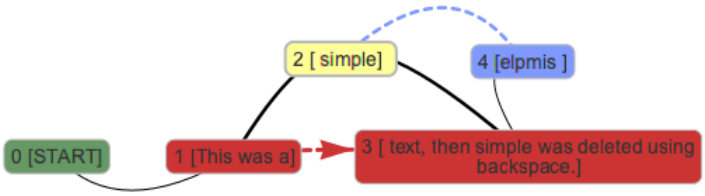


Figure 8.4: Delete with Backspace and Removed nodes example

V. A **Delete** done using the *delete* key or *highlight and deletion* is depicted below. The initial text has a word deleted; this splits the single node into three parts (nodes 1, 2 and 3) that will maintain the initial chronological order of the text. The produced text flow will now exclude the removed text (node 3), which will typically be coloured in a different shade. In addition, the deletion event (node 4) is now included in the order of chronology to emphasise when the deletion occurred. A highlight and deletion may be displayed when any existing text is highlighted using the mouse or keyboard and then removed using the backspace, delete, or any other character key, or when the text is replaced using a (cut-copy-paste) keyboard or mouse action.

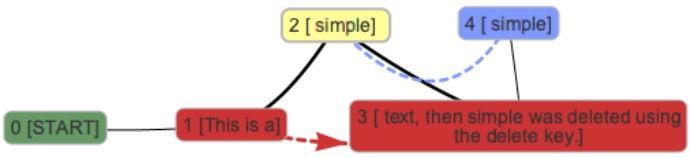


Figure 8.5: Delete with delete key and Removed nodes example

VI. An instance of **Copy and Paste** is depicted below. In the example below, a part of the existing text (node 1) is copied, then followed by an append (node 2). Next, the copied text is pasted (node 3) and finally followed by another append to the text (node 4). Here, the discontinuous edge between nodes 1 and 3 represents where the pasted text was copied from. This example also depicts a scenario in which the chronological order of events is similar to the produced text flow.

Figure 8.6: Copy and Paste example

VII. An instance of **Displacement** or **Cut and Paste** is depicted below. This scenario is treated similar to a deletion followed by an append or insert. Similar to a deletion, the existing node is split into 3 (nodes 1, 2 and 3) maintaining the chronological order of writing. The displaced text is depicted in a different shade (node 3), and the occurrence of the deletion is represented as a delete node (node 4). The displaced text is now represented as an append or insert, but in its own unique shade to highlight the displacement (node 5). The discontinuous line between the displaced node and the delete node shows the relationship between this deletion and append or insertion.

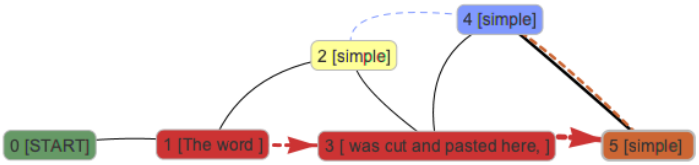


Figure 8.7: Displacement node example

VIII. When the **Show Pause** option is selected, pauses that are greater than or equal to the specified cognitive pause value (e.g. 2000ms) are represented by a **Pause** node (node 2). The pause time is indicated within the node. The pause is included in the chronological flow.

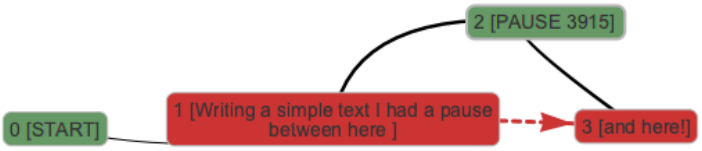


Figure 8.8: Pause node example

8.1.2 Graph Settings

The Graph Settings interface allows the specification of parameters that are used to create the Session Graph, Graph Statistics and Progressive Graph. This interface is common to all graph representations.

Changing the parameters will create unique graph representations. The parameters are as follows:

- **Pause Time**: This specifies the time in milliseconds that a pause is considered a cognitive pause.
- **Show Pauses**: This indicates pauses with a timespan greater than *Pause Time* that are shown as a Pause node in the Graph.
  WARNING! Showing Pause nodes will make the Graph larger and visually more complex.
- **Show Text**: This shows the text created during the sequence of events within the node.

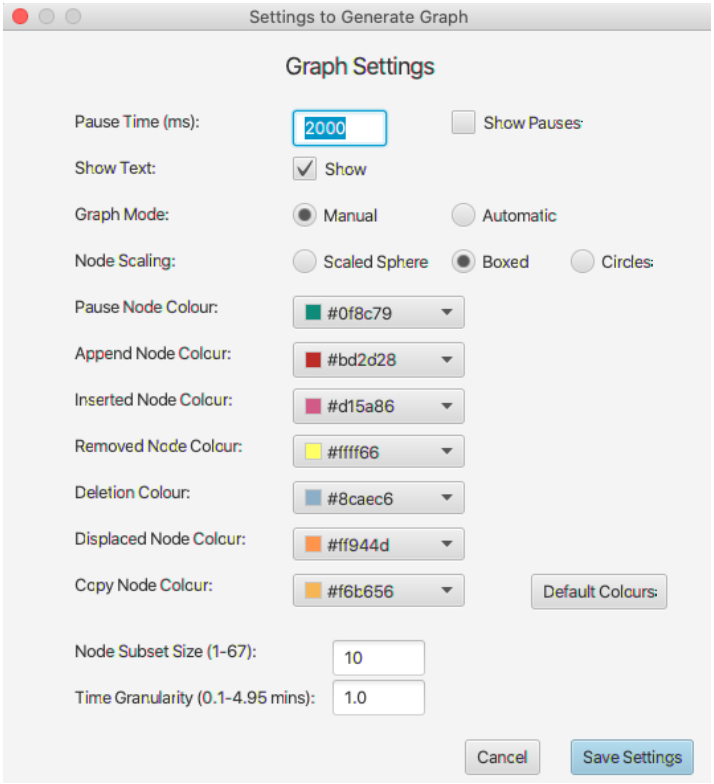


Figure 8.9: Graph settings form

- **Graph Mode**: The Graph will be drawn in manual or automatic mode. In automatic mode, GGXLog will draw the respective graph and the entire graph will be affected if nodes are manipulated using the mouse. In manual mode, GGXLog will draw the respective graph and only the specific node will be affected if a node is manipulated. This mode may be turned off in the Session Graph interface.
  IMPORTANT! In the case where there are a large number of edits and the Graph is complex, automatic mode would be advised.
  WARNING! Manipulation of the Graph in automatic mode requires a considerable amount of computational power.
- **Node Scaling**: The shape and size of the nodes are specified here. While boxes and circles will have text within the nodes, scaled spheres will have text outside the nodes if the Show Text option is selected. This mode may be changed in the Session Graph interface.
- **Node Colour**: This specifies the colour of each category of node.
- **Default Colours**: This will revert back to the default colours specified for the graph in the system.
- **Node Subset Size**: This parameter is relevant to Subset Node Distribution and Cumulative Node Distribution in Graph Statistics. This parameter specifies the size of the subset of nodes we consider. The value should be between 1 and the total number of nodes in the Graph, shown within parentheses for convenience.
- **Time Granularity**: This parameter is relevant to Node Distribution in Time Blocks and Cumulative Node Distribution over Time in Graph Statistics. The value should be greater than 0 and less than the total duration of the session in minutes, shown in parentheses for convenience.

Note: The colour choices are saved as a preference for the next time the software is used.

8.1.3 Session Graph

Session Graph visualises the writing session using Graph Theory.

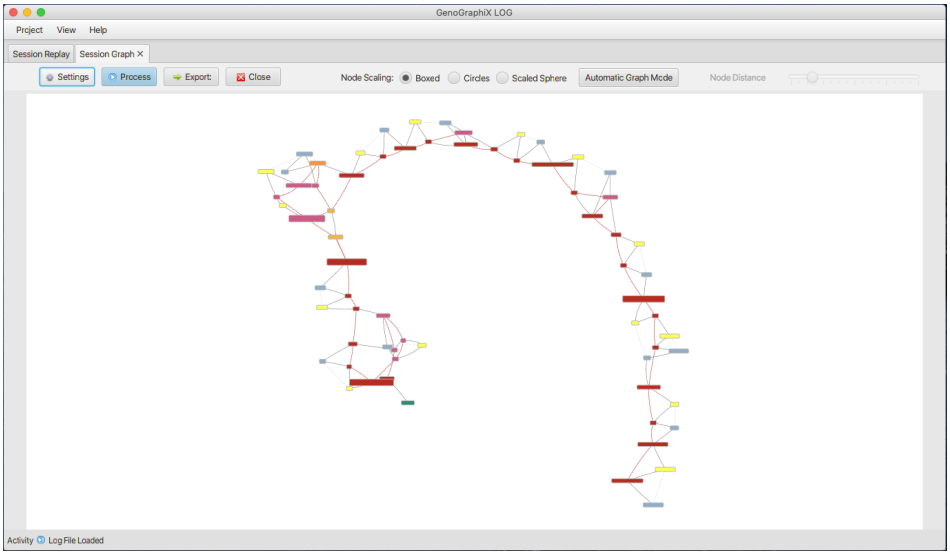


Figure 8.10: Session Graph interface

To view Session Graph:

1. Click on the **View** menu, then select **Graph Representation and** click on **Session Graph.**
2. Click on the **Settings** button to set parameters; if parameters are already set, click on the **Process** button.
3. To redraw the Graph, click on the **Process** button until a satisfactory graph is drawn with a convenient placement of the nodes.
4. To change the shapes of the nodes, click on the **Node Scaling** options.
5. To automatically redraw the Graph, click on the **Automatic Graph Mode** button. Click on this button again to disable automatic mode.
6. To increase the distance between nodes in *Automatic Graph Mode*, move the **Node Distance** slider to the right; move the slider to the left to reduce the distance.

NOTE: Options 4–6 are disabled when the graph is larger than the default 120-node size. This value may be changed in Preferences (see section 10.0).

7. Use the mouse scroll wheel to **Zoom-in or Zoom-out.**
8. Click on and drag the windows to **Move the Graph** around in the window.
9. Click on and drag a node to **Move the Node** around in the window.

8.1.4 Graph Statistics

Graph Statistics provides numerical statistics and visual summaries of Session Graph.

To view Graph Statistics:

1. Click on the **View** menu, then select **Graph Representation** and click on **Graph Statistics.**
2. Click on the **Settings** button to set parameters; if parameters are already set, click on the **Process** button.

Graph Complexity Measure provides the following details:

1. The *Nodes Per Minute* parameter provides a value that relates to the frequency of edits/revisions during a writing session, as nodes are created by edits/revisions.

$$\textbf{Nodes per Minute} = \frac{\text{Number of Nodes in the Session}}{\text{Session Duration}} \qquad \text{(Eq. 8.1)}$$

2. The Final Text to Chronology Edge Ratio provides insight into the number of nodes in the final text product against the entire text development and editing events. The maximum value will be 1 when there are no edits or pauses in the writing session. The greater the number of edits and pauses, the lower the value will be.

$$\textbf{Final Text Edges to Chronology Edges Ratio} = \frac{\text{Number of Final Text Edges}}{\text{Number of Chronological Edges}} \qquad \text{(Eq. 8.2)}$$

3. The Final Text to Total Edge Ratio provides insight into the number of nodes in the final text product against the complexity of the writing process measured by the total number of edges in the graph. The maximum value will be 1 when there are no edits or pauses in the writing session. The greater the number of edits and pauses, the lower the value will be; thus the lower the number the more complex the writing process. The number of deletes and displacements will have a greater impact on this value.

$$\textbf{Final Text Edges to Total Edges Ratio} = \frac{\text{Number of Final Text Edges}}{\text{Total Number of Edges in the Graph}} \qquad \text{(Eq. 8.3)}$$

Table 8.1: Graph Complexity measure

Parameter	Value
Nodes Per Minute [67 nodes in 00:04:56]	13.55
Final Text Edges to Chronology Edges Ratio [39:67]	0.58
Final Text Edges to Total Edges Ratio [39:120]	0.33

The Node Distribution table provides detailed information on how the nodes and characters are distributed in Session Graph.

Table 8.2: Node Distribution

Node Type	Number of Nodes	Percentage	Number of Characters	Percentage	Mean Node Length (Chars)
Non-Empty Append Nodes	15	22.39%	335	67.81%	22.33
Empty Append Nodes	11	16.42%	-	-	-
Insert Nodes	10	14.93%	78	15.79%	7.8
Displaced Nodes	1	1.49%	10	2.02%	10
Copy Nodes	2	2.99%	10	2.02%	5
Removed Nodes	14	20.9%	-	-	-
Delete Nodes	14	20.9%	61	12.35%	4.36
Pause Nodes	0	0%	-	-	-

The pie charts for *Node Distribution* and *Node to Character Distribution* graphically represent node and character distribution in the Session Graph.

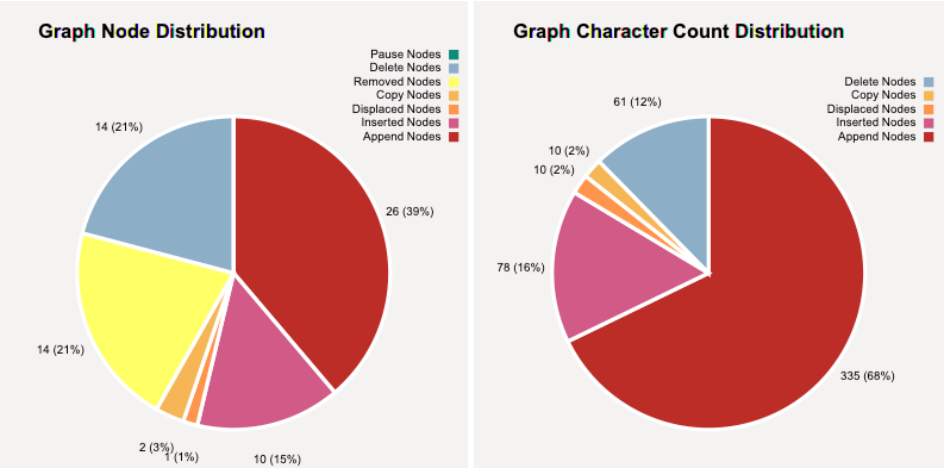


Figure 8.11: Node Distribution charts

The *Node Distribution between Current and Edit Point* chart graphically represents the node distance between the edit node and the last node of a Graph during the edit.

Note:

$$\begin{aligned}\textbf{Node Distance} = &\text{ Number of Nodes in the Graph at any Given Time of an Edit} \\ &- \text{ Node Number the affected by the Edit}\end{aligned} \tag{Eq. 8.4}$$

In the example below, there are 10 nodes in the Graph at a given time and an insert is done, which affects the sixth node of the graph:

$$\textbf{Node Distance} = \ 10 - 6$$

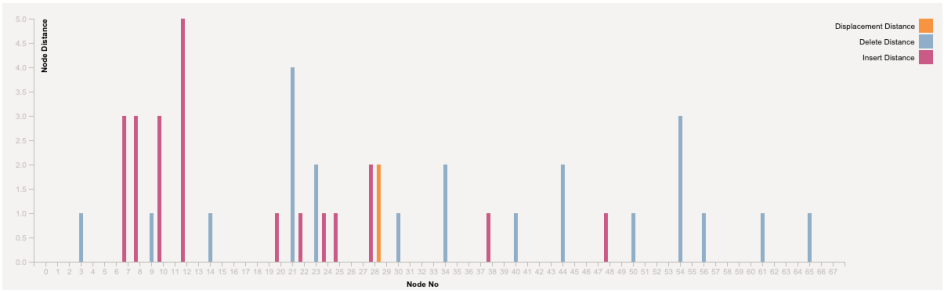


Figure 8.12: Node Distance between current and edit point chart

The *Subset Node Distribution* chart provides a graphical representation of the distribution of nodes in a block of nodes The block size is specified in *Node Subset Size* in **Graph Settings**.

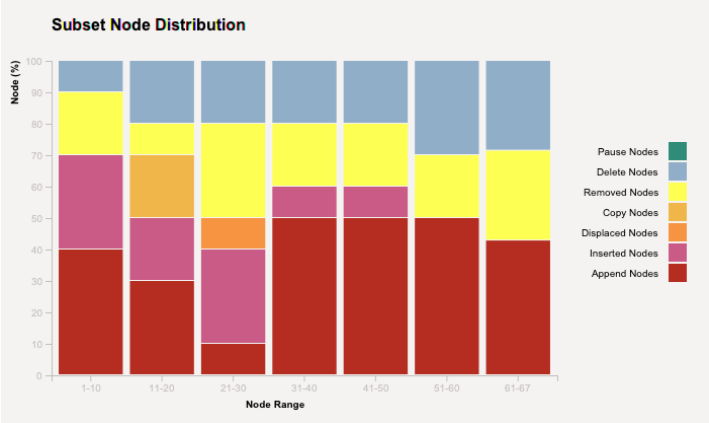


Figure 8.13: Node distribution in subsets chart

The *Cumulative Node Distribution* chart provides a graphical representation of the distribution of nodes in an incremental block. The block size is specified in *Node Subset Size* in **Graph Settings**.

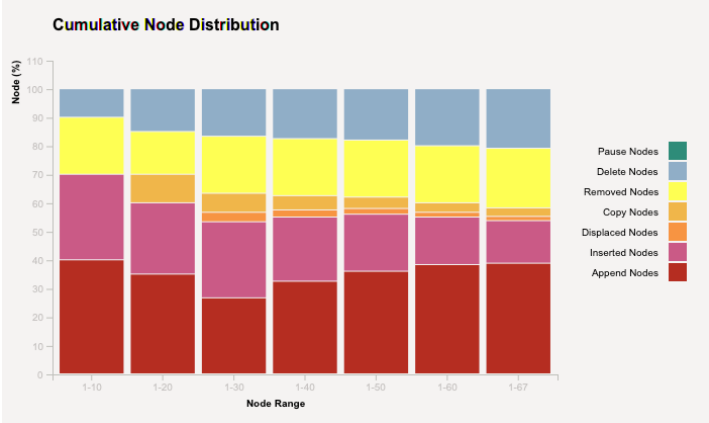


Figure 8.14: Cumulative node distribution of subsets increments chart

The *Node Distribution in Time Blocks* chart provides a graphical representation of the distribution of nodes in a specified duration. The duration is specified in *Time Granularity* in **Graph Settings**.

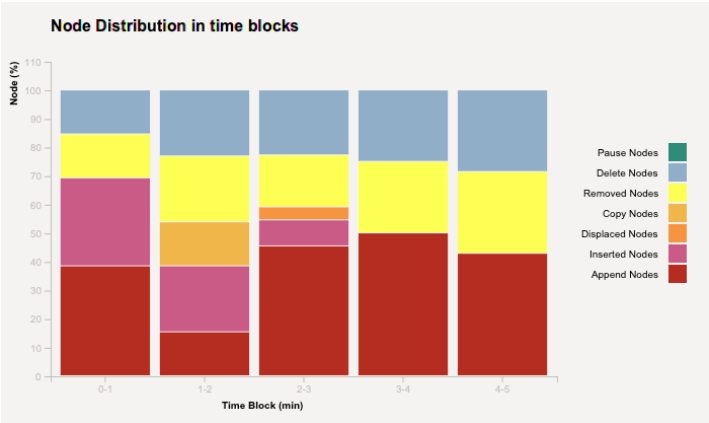


Figure 8.15: Node distribution in time blocks chart

The Cumulative *Node Distribution over Time* chart provides a graphical representation of the distribution of nodes in specified time increments. The duration is specified in *Time Granularity* in **Graph Settings**.

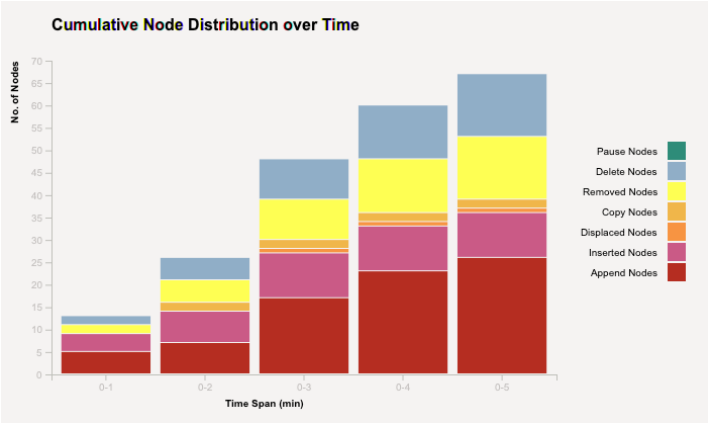


Figure 8.16: Cumulative node distribution in time block increments chart

The *Deletion vs. Node Distance Plot* chart provides a graphical representation of the distribution of deletions in the Session Graph. The chart represents four dimensions:

1. **Average Node Distance**: Node Distance between the Delete Node number (number of nodes at the time of deletion) and the number of nodes affected by the deletion
   - The smaller the value, the more immediate the deletion; the larger the value, the more delayed the deletion
2. **Characters Deleted**: Radius of the circle quantifies this parameter
3. **Relative Node Distance***:* Indicated by the darkness of the circle (i.e. Average Node Distance as a percentage of the total number of nodes when the deletion occurred)
   - In the example below, there were 10 nodes at the time of deletion and the deletion affected the sixth node:

$$\textbf{Relative Node Distance} = \frac{10-6}{10} \times 100\%$$

4. The Node ID of the Deletion and the numerical value of the Relative Node Distance: Displayed on Mouse Over

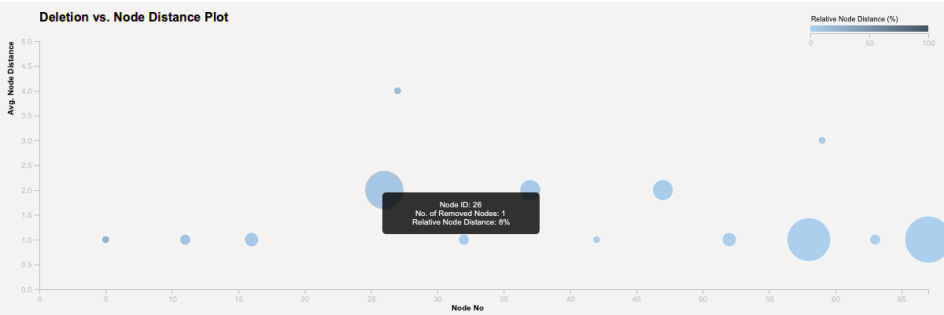


Figure 8.17: Deletion vs. Node distance chart

The *Insertion vs. Node Distance Plot* chart provides a graphical representation of the distribution of Insertions in the Session Graph. The Chart represents four dimensions:

1. **Node Distance**: Node Distance between the Insert Node number (number of nodes at the time of insertion) and the number of nodes affected by the insertion
   - The smaller the value, the more immediate the deletion; the larger the value, the more delayed the deletion
2. **Characters Inserted**: Radius of the circle
3. **Relative Node Distance***:* Indicated by the darkness of the circle (i.e. Average Node Distance as a percentage of the total number of nodes when the insertion occurred)
   - In the example below, there were 10 nodes at the time of insertion and the insertion affected the sixth node:

$$\textbf{Relative Node Distance} = \frac{10-6}{10} \times 100\%$$

4. The Node ID of the Insertion and the numerical value of the Relative Node Distance: Displayed on Mouse Over

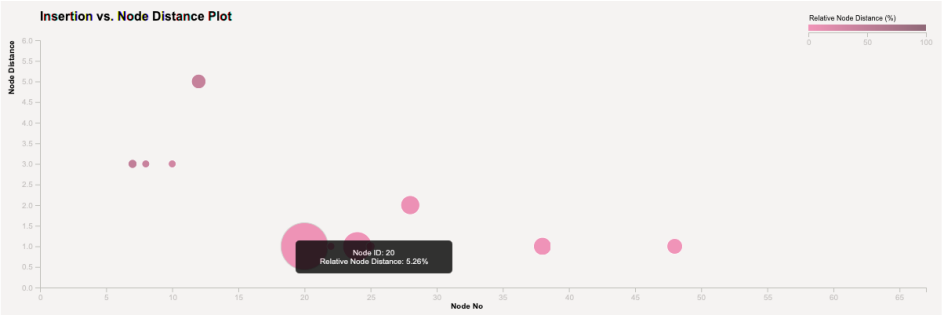


Figure 8.18: Insertion vs Node distance chart

8.1.5 Graph Pause Analysis

The Graph Pause analysis provides a visualization of pause category distribution and along with the caret position plotting of pauses. Pauses are considered based on the cognitive pause duration. Pause with a duration greater or equal to the specified cognitive pause duration are categorised based on location of the where the pauses occur. Pauses that cannot be categorised are listed as unknown. For example, a pause between a mouse-movement and a typing of a character is typically categorised as unknown.

To view the Graph Pause Analysis:

1. Click on the **View** menu, then select **Graph Representation** and click on **Graph Pause Analysis.**
2. Click on the **Settings** button to set parameters; if parameters are already set, click on the **Process** button.

Note: As Graph Pause Analysis depends on Pauses, the Show Pause option in Graph Settings will automatically be enabled and the Session Graph will be redrawn, and the Graph Statistics will be recalculated.

The chart represents six dimensions which are;

i. Pause Category (Colour coding)
ii. Node ID of the pause in the Session Graph (number above the circle)
iii. Pause duration (left y-axis)
iv. Caret position during the pause (right y-axis)
v. Time of the pause (x-axis)
vi. Event ID in the Session Log (on mouse over)

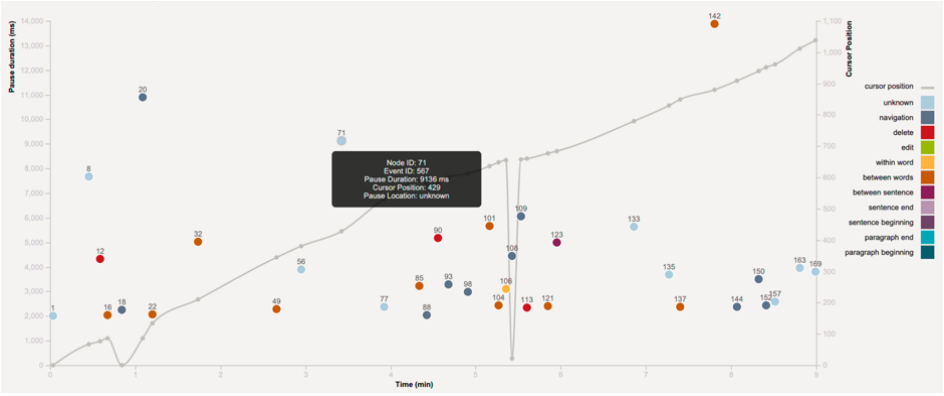


Figure 8.19: Graph Pause Analysis chart

The Graph Pause Analysis data is further summarised in the Pause Breakdown table. This data includes:

i. A count of the number of pauses in a category
ii. The count as a percentage of the total number of pauses
iii. Cumulative pause duration of each category
iv. The cumulative pause duration as a percentage of the total duration of all pauses
v. Mean pause duration for each pause category

Table 8.3: Pause Breakdown example

Pause Location	Count	Percentage of Total Pauses	Total Pause Duration(ms)	Percentage of Total Pause Time	Mean Pause Duration(ms)
paragraph beginning	0	0%	0	0%	-
paragraph end	0	0%	0	0%	-
sentence beginning	0	0%	0	0%	-
sentence end	0	0%	0	0%	-
between sentence	1	2.86%	4981	3.4%	4981
between words	10	28.57%	41378	28.28%	4137.8
within word	1	2.86%	3108	2.12%	3108
edit	0	0%	0	0%	-
delete	3	8.57%	11824	8.08%	3941.33
navigation	10	28.57%	40254	27.51%	4025.4
unknown	10	28.57%	44775	30.6%	4477.5

8.1.6 Progressive Graph

The Progressive Graph allows for the visualisation of the Graph at various stages of the writing session. The concurrent text is displayed in the text area below the graph.

To view the Progressive Graph:

1. Click on the **View** menu, then select **Graph Representation** and click on **Progressive Graph.**
2. Click on the **Settings** button to set parameters; if parameters are already set, click on the **Process** button.

To navigate the Graph progressively, use the navigation buttons below:

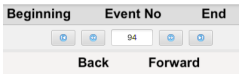


Figure 8.20: Progressive Graph navigation interface

The Progressive Graph represents the following data:

1. **Node Type**: Represented by colours specified in the Graph Settings
2. **Number of Characters in the Node**: Radius of the node
3. **Text in the Node**: Displayed the mouse is moved over the node

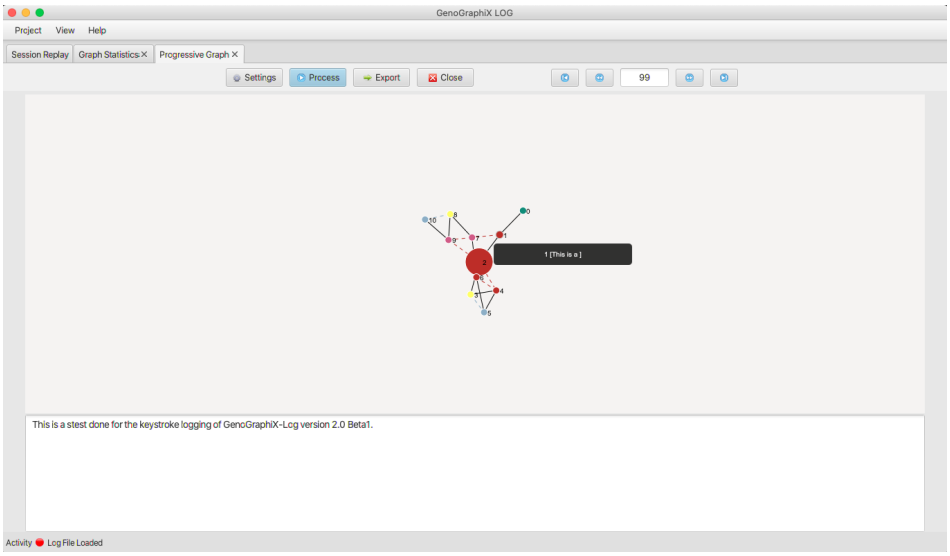


Figure 8.21: Progressive Graph interface

8.2 Notations

Notations represent the writing session using text and symbols. GGXLog supports three formats.

8.2.1 Linear Text

The Linear Text format represents all input events linearly. Text entries are represented as text, while other events are represented as symbols or notations. GGXLog supports two Linear Text formats.

To view the Linear Text:

1. Click on the **View** menu, then select **Notations** and click on **Linear Text.**
2. Click on the **Settings** button to set parameters; if parameters are already set, click on the **Process** button.
3. To switch between the two formats, click on the radio buttons **Format 1** or **Format 2** in the interface.

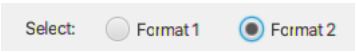


Figure 8.22: Linear Text format selection buttons

The **Settings** for Linear Text allows the user to specify:

- **Pause Time**: Represents any pause greater than the specified cognitive pause duration (in milliseconds)
- **Show Pauses**: Specifies if pauses should be displayed
- **Colour**: Specifies the colour used to represent each category of input
- **Default Colours**: Reverts back to the default colours specified for the graph in the system

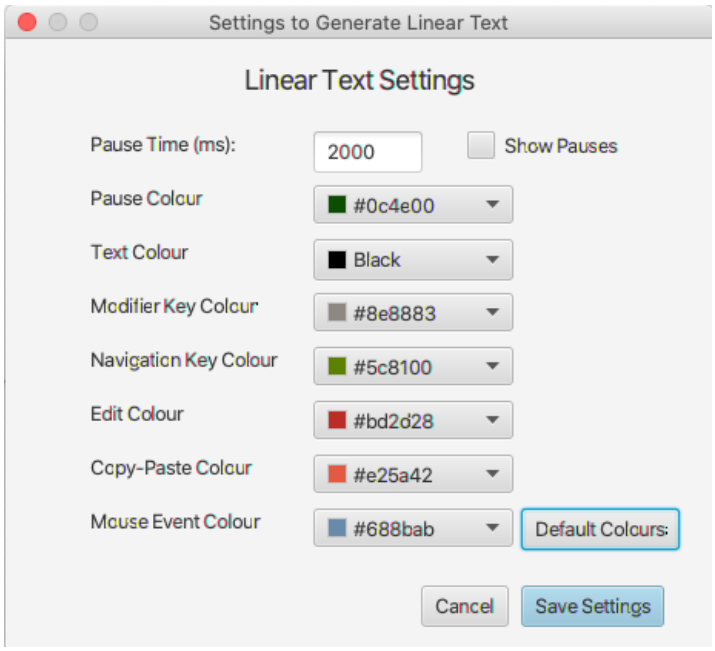


Figure 8.23: Linear Text settings form

Note: The colour choices are saved as a preference for the next time the software is used.

Symbols used for Linear Text representation are listed in the table below:

Table 8.4: Linear Text representations

Representation	Linear Text Format 1	Linear Text Format 2
Pause	[pause duration]	<pause duration>
Mouse Action	[▼][▲]	<MOUSE EVENT>
Mouse Cut	[▼]X[▲]	<MOUSE CUT>
Mouse Copy	[▼]C[▲]	<MOUSE COPY>
Mouse Paste	[▼]V[▲]	<MOUSE PASTE>
Backspace	◄	<BACKSPACE>
Delete	►	<DELETE>
Enter/Return	↲	<RETURN>
Shift	⇑	<SHIFT>
Left Arrow	←	<LEFT>
Right Arrow	→	<RIGHT>
Up Arrow	↑	<UP>
Down Arrow	↓	<DOWN>
Page Up	[PG_UP]	<PG_UP>
Page Down	[PG_DOWN]	<PG_DOWN>
Home	[HOME]	<HOME>
End	[END]	<END>
Cut	[Ctrl+X]	<CUT>
Copy	[Ctrl+C]	<COPY>
Paste	[Ctrl+X]	<PASTE>
Undo	[Ctrl+Z]	<UNDO>
Redo	[Ctrl+Y]	<REDO>

In the case of a continuous sequence of identical input events, the Linear Text notation represents them as below:

Table 8.5: Linear Text example notation

Example	Linear Text Format 1	Linear Text Format 2
5 Backspace presses in sequence	**5**◄	<BACKSPACE5>
8 left arrow keys	**8**←	<LEFT8>

Linear Text Format 1 is based on “linear view” in Translog-II (Carl, 2012)

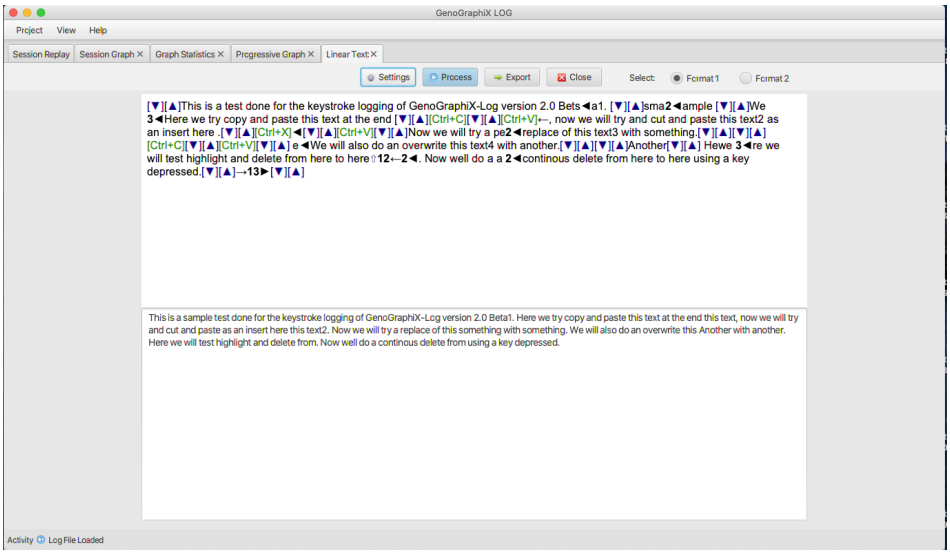


Figure 8.24: Linear Text Format 1 interface and example

Linear Text Format 2 is based on “linear text” in ScriptLog (Wengelin et. al., 2009)

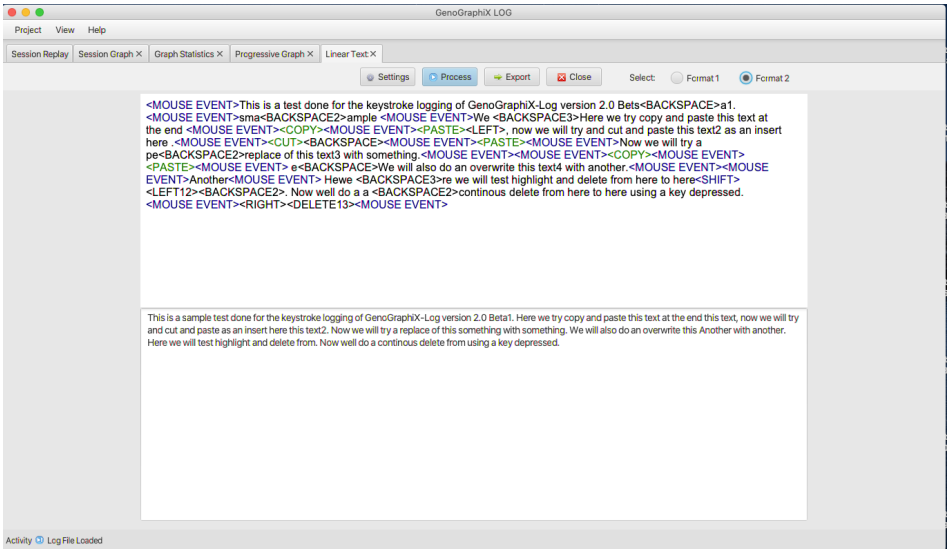


Figure 8.25: Linear Text Format 2 interface and example

8.2.2 S-Notation

S-Notation (Kollberg, 1996) represents text inputs, while primarily focusing on revision or edits based on *Inserts* and *Deletes*. Details about S-notation can be found here.

To view the S-Notation:

1. Click on the **View** menu, then select **Notations** and click on **S-Notation.**
2. Click on the **Settings** button to set parameters; if parameters are already set, click on the **Process** button.

The **Settings** for S-Notation allow the user to specify:

- **Minimum number of characters to include edit number on both markers**: Specifies the minimum number of characters required inside an edit for both markers to include edit numbers as superscript. This improves the legibility of the notation for considerably long revisions.
- **Minimum number of characters deleted to be considered as a revision**: Specifies the minimum number of consecutive delete/backspace key presses required for the edit to be considered a revision and not a correction. The default value is 0, thus all deletions are considered revisions (e.g. if 0, correcting a typographical error with a single backspace will still be considered a revision).
- **Consider pasting text from the Clipboard at the end of the text as an insert**: Specifies that any text that is copied and pasted at the end of the current text be considered an insert and not an append.

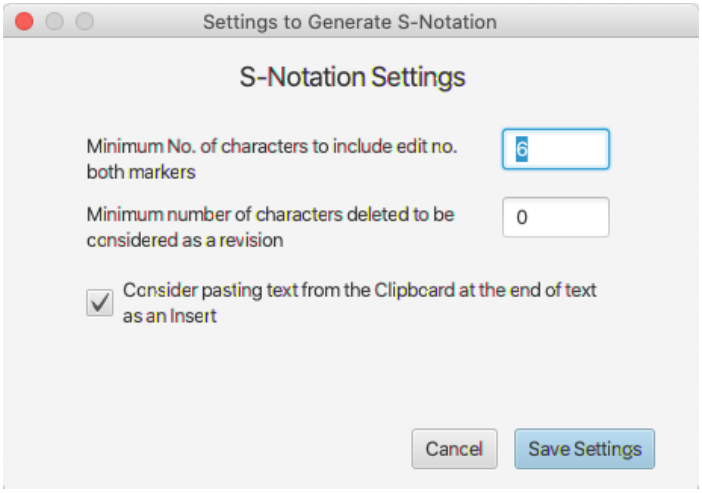


Figure 8.26: S-Notation settings form

In S-Notation:

- Square brackets represent a Delete, e.g. [this text was deleted].
- Curly brackets represent an Insert, e.g. {this text was inserted}.
- The superscript represents the edit number, e.g. $\{\text{this is an insert}\}^2$ OR $^2\{\text{this is an insert}\}^2$.
- The subscript and arrow represent cursor location prior to the specific edit, e.g. $\downarrow_2$.

The S-Notation interface consists of:

- Navigation

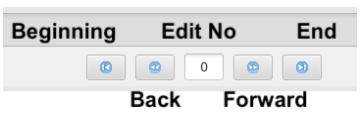


Figure 8.27: S-Notation navigation interface

Highlights in the *List of Edits* and the *S-Notation* markers provide convenient tracking of each edit.

- List of Edits (Marked in green)
  - Action: Insert or Delete
  - Position: Current caret/cursor position
  - Length: Number of characters inserted or deleted
  - Start Time: Start time of the edit
  - End Time: End time of the edit

  Clicking on Edit Number directly navigates to a particular Edit, which highlights the *Edit* and the *S-Notation* position.

- S-Notation (marked in red)
- Current text (marked in blue)

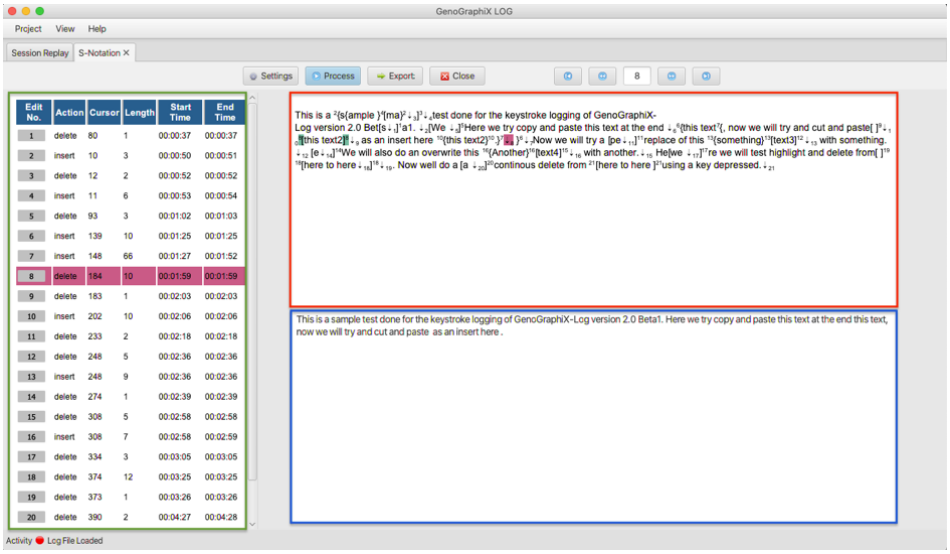


Figure 8.28: S-Notation interface and example

8.3 Analysis

Analysis provides common analyses used in writing research. The Session Log data is processed to provide meaningful information summarising the writing session.

8.3.1 Statistics

Statistics provides some basic analyses of the session data including:

- Event Analysis: Categorization of input events
- Time Analysis: Summarisation of inputs with relation to time
- Pause Analysis: Summarisation of pauses with relation to category and time
- Gene Analysis: Genetical categorization according to the effect of the input events on the text

To view Statistics:

3. Click on the **View** menu, then select **Analysis** and click on **Statistics.**
4. Click on the **Settings** button to set parameters; if parameters are already set, click on the **Process** button.

The **Settings** for Statistics allows the user to specify:

- **Pause Time**: This is the cognitive pause time in milliseconds. Any pause greater than the specified duration will be represented as a Pause in the notation. The default value is 2000ms, but may be changed in Preferences (see Section 10.0).
- **Required Statistics**: These specify which statistics to compute.

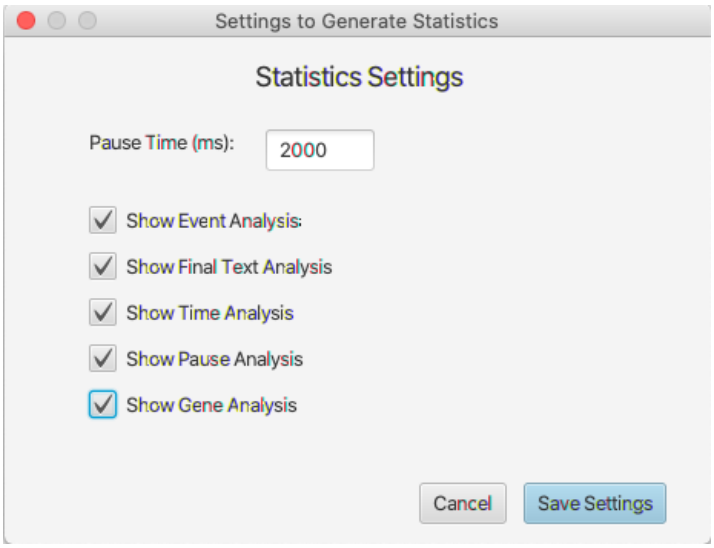


Figure 8.29: General Statistics settings form

Event Analysis lists the following statistics:

- Total Events: *All input events*
- Total Text Production: *Number of textual inputs*
- Total Text Eliminations: *Number of Deletes*
- Total Cut Events: *Number of Cut events (Both keyboard- and mouse-based)*
- Total Copy Events: *Number of Copy events (Both keyboard- and mouse-based)*
- Total Paste Events: *Number of Paste events (Both keyboard- and mouse-based)*
- Total Cursor Navigations: *Number of Navigation key presses*
- Total Mouse Events: *Number of mouse clicks*
- Total Miscellaneous Events: *Number of non-categorised events such as Modifier keys, System inputs, etc.*

Table 8.6: Event analysis example

Statistic	Value
Total User Events	553
Total Text Productions	465
Total Text Eliminations	30
Total Cut Events	1
Total Copy Events	2
Total Paste Events	3
Total Cursor Navigations	14
Total Mouse Events	17
Total Miscellaneous Events (e.g. Modifier Keys)	21

Final Text Analysis lists the following statistics:

- Character Count (incl. Spaces): *Total number of Characters in the Final text including spaces and punctuation*
- Characters per minute (incl. Spaces): *Character Count (incl. Spaces) divided by session duration*
- Character Count (excl. Spaces): *Total number of Characters in the Final text excluding Spaces*
- Characters per minute (excl. Spaces): *Character Count (excl. Spaces) divided by session duration*
- Word Count: *Total number of words in the text*
- Words per minute: *Total number of words divided by session duration*
- Mean word length: *Mean number of characters in the list of words in the final text*
- Std. Deviation in word length: *Standard deviation of the number of characters in the list of words in the final text*
- Sentence Count: *Number of sentences in the final text*
- Paragraph Count: *Number of paragraphs in the final text*
- Product to Process Ratio: *Ratio of the total number of characters in the final text including spaces as opposed to the total number of characters including spaces typed during the writing session*

Table 8.7: Final text/product analysis example

Statistic	Value
Character Count (incl. Spaces)	433
Character per minute (incl. Spaces)	87.55
Character Count (excl. Spaces)	352
Character per minute (excl. Spaces)	71.17
Word Count	82
Words per minute	16.58
Mean Word length	4.21
Std. Deviation in Word length	2.4
Sentence Count	6
Paragraph Count	1
Product to Process Ratio [Character Count (incl. Spaces):Total Text Productions]	0.93

Time Analysis lists the following statistics:

- Total Duration: *Duration of the writing session in hour: minute: second format*
- Events per Minute: *Mean number of inputs per minute*
- Text Production per Minute: *Mean number of textual inputs per minute*

Table 8.8: Time analysis example

Statistic	Value
Total Duration (h:m:s)	00:04:56
Events per Minute	111.82
Text Production per Minute	94.02

Pause Analysis lists the following statistics:

- Total Pause Time: *Cumulative duration of all pauses*
- Mean Pause Time: *Mean Pause duration (Total Pause Time / Total Number of Pauses)*
- Number of Cognitive Pauses: *Number of pauses with duration greater than the specified cognitive pause duration*
- Total Cognitive Pause Time: *Cumulative duration of cognitive pauses*

- Mean Cognitive Pause Time: *Mean cognitive pause duration*
- Percentage of Cognitive Pauses: *Percentage of cumulative cognitive pause duration against total pause duration*
- Percentage of Cognitive Pauses: *Percentage of cumulative cognitive pause duration against session duration*
- Percentage of Cognitive Pauses: *Percentage of the number of cognitive pauses against the total number of input events*

Table 8.9: Pause analysis example

Statistic	Value
Total Pause Time (h:m:s)	00:04:02
Mean Pause Time (ms)	439.39
Number of Cognitive Pauses	17
Total Cognitive Pause Time (h:m:s)	00:01:43
Mean Cognitive Pause Time (ms)	6095.12
Percentage of Cognitive Pauses (vs. Total Pause Duration)	42.64%
Percentage of Cognitive Pauses (vs. Session Duration)	34.92%
Percentage of Cognitive Pauses (vs. Total User Events)	3.07%

Gene Analysis lists the following statistics:

- Total Appends: *Total number of input events that append text*
- Total Inserts: *Total number of input events that insert text*
- Total Deletes: *Total number of input events that delete text*
- Total Displacements: *Total number of input events that displace/move text from one point in the text to another*
- Total Replacements: *Total number of input events that replace or overwrite a block of text in the current text*
- Total Copy: *Total number of input events that copy and paste text within the text*
- Total Navigation: *Total number of input events that navigate the caret/cursor in the text*
- Total Selects: *Total number of input events that select/highlight text*
- Total Unknown: *Total number of input events that may not have a direct implication on the text (e.g. Modifier Keys, system inputs)*

Table 8.10: Gene analysis example

Statistic	Value
Total Appends	383
Total Inserts	81
Total Deletes	31
Total Displacements	0
Total Replacements	2
Total Copy	2
Total Navigation	14
Total Selects	17
Total Unknown (e.g. Modifier Keys)	23

8.3.2 Burst and Efficiency

Burst and Efficiency (Miller, Lindgren & Sullivan, 2008) provides an analysis of writing bursts between interruptions and the efficiency of such bursts.

- **Burst**: This is the number of characters typed between two interruptions (pauses of revisions). A burst is specified as the number of typed characters. It excludes modifier keys and dead keys.
- **Efficiency**: This is the number of characters per minute within a burst.

$$\textbf{Efficiency} = \frac{\text{Number of Characters in the Bust}}{\text{Duration of the Burst}} \qquad \text{(Eq. 8.5)}$$

To view Burst and Efficiency:

1. Click on the **View** menu, then select **Analysis** and click on **Burst and Efficiency.**
2. Click on the **Settings** button to set parameters; if parameters are already set, click on the **Process** button.

The **Settings** for Burst and Efficiency allow the user to specify:

- **Pause Time**: Represents any pause greater than the specified cognitive pause duration (in milliseconds)
- **Select Interruptions**:
  - Specifies if pauses are considered interruptions
  - Specifies which input events are considered interruptions

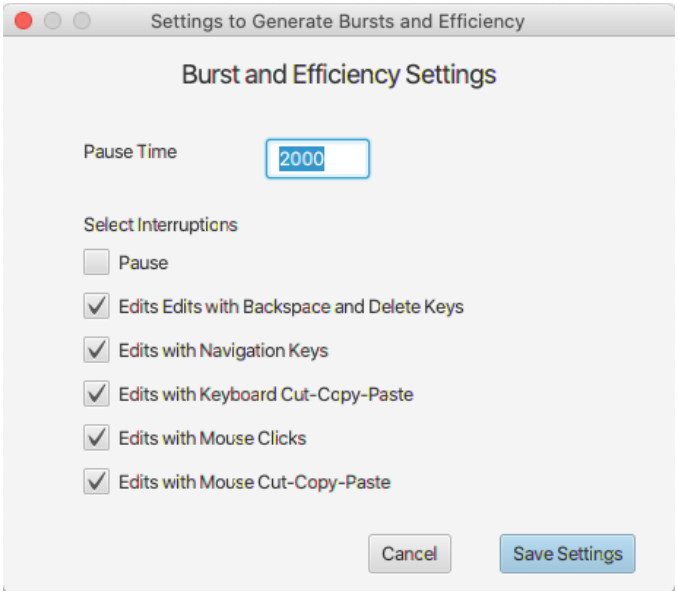


Figure 8.30: Burst and Efficiency settings form

The Burst and Efficiency chart visualises the burst and efficiency distribution throughout the writing session.

- The left y-axis specifies the burst size and is related to the bar and value above the bar.
- Bursts are categorised into Append (text added to the end) and Insert (text added within the text).
- The right y-axis specifies the efficiency and the corresponding dots indicate the efficiency of each burst.
- The discontinuous line represents the 10% Trimmed Mean Burst Efficiency of the entire writing session. This mean value is calculated by excluding 10% of the lowest and 10% of the highest efficiency values in the set and then calculating the mean of the remainder. This reduces outliers in the efficiency data set.
- The x-axis indicates time.

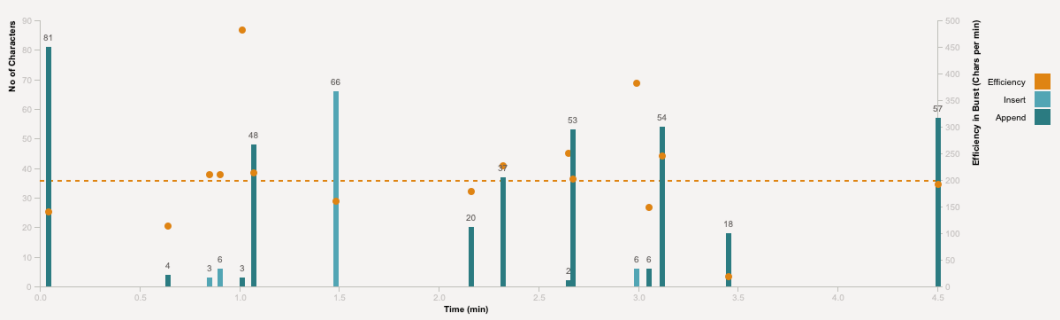


Figure 8.31: Burst and Efficiency chart

The Session Burst Parameters table provides a summary of bursts for the writing session.

Note: The values differ based on the Cognitive Pause Value and considered interruptions specified in Settings.

Table 8.11: Session Burst parameters example

Parameter	Value (Chars per min)
10% Trimmed Mean of Burst Efficiency	198.09
Mean Burst (Append)	31.92
Mean Burst (Insert)	20.25
Mean Burst (Overall)	29

The Event List table provides a list of events as performed actions. Burst (No. of Characters), Burst Duration and Efficiency are calculated only for append and insert events, which are the events that produce text using keyboard inputs.

Table 8.12: Event list for burst and efficiency calculation

Event	Action	Start Time	No. of Chars	Burst Duration (sec)	Efficiency (Chars per min)
1	append	00:00:01	81	34.85	139.47
2	backspace	00:00:37	-	-	-
3	append	00:00:37	4	2.12	113.48
4	insert	00:00:50	3	0.85	210.77
5	backspace	00:00:52	-	-	-
6	insert	00:00:53	6	1.71	210.53
7	append	00:00:59	3	0.37	481.28
8	backspace	00:01:02	-	-	-
9	append	00:01:03	48	13.47	213.79
10	paste_append	00:01:25	-	-	-
11	insert	00:01:27	66	24.78	159.81
12	cut	00:01:59	-	-	-
13	backspace	00:02:03	-	-	-
14	paste_insert	00:02:06	-	-	-
15	append	00:02:08	20	6.75	177.78
16	backspace	00:02:18	-	-	-

8.3.3 Micro Analysis

Micro Analysis provides an input by input analysis of the writing session.

To view Micro Analysis:

1. Click on the **View** menu, then select **Analysis** and click on **Micro Analysis.**
2. Click on the **Settings** button to set parameters; if parameters are already set, click on the **Process** button.

The **Settings** for Micro Analysis allow the user to specify:

- Predefined input patterns: The ^ position is the considered a transition (e.g. ". A" would relate to the "._+Sa" pattern and may be interpreted as the beginning of a sentence)
- Specific inputs: e.g. Uppercase characters

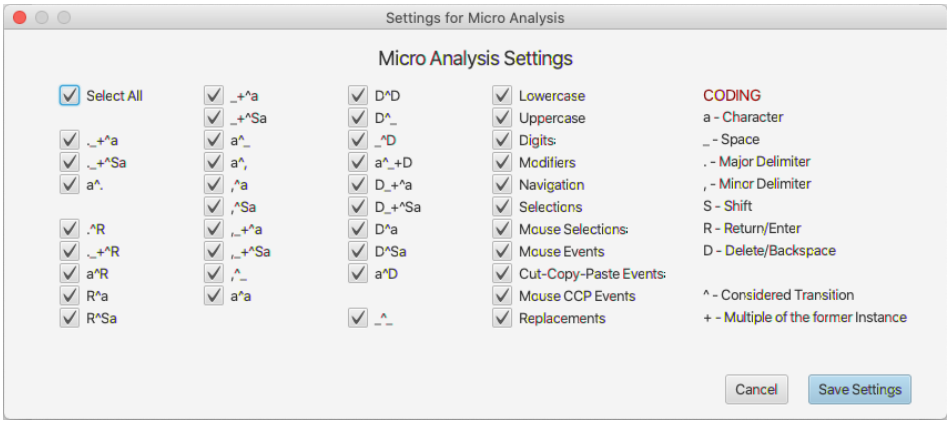


Figure 8.32: Micro Analysis settings form

The Micro Level Analysis indicates:

- The number of occurrences of a pattern or input
- Total pause time of the considered transition pattern or input
- Mean pause time of the considered transition pattern or input

Note: The pause time calculation does not differentiate between cognitive pauses and non-cognitive pauses and considers pauses altogether.

Table 8.13: Micro level analysis example

Sequence	No of Occurrences	Total Pause Time	Mean Pause Time
._+^a	0	0	-
._+^Sa	1	841	841
a^.	5	4163	832.6
.^R	0	0	-
._+^R	0	0	-
a^R	0	0	-
R^a	0	0	-
R^Sa	0	0	-
_+^a	77	27874	362
_+^Sa	5	1810	362
a^_	84	16126	191.98
a^,	1	663	663
,^a	0	0	-
,^Sa	1	242	242

8.3.4 Edit Distance

Edit Distance provides an analysis of the distance between two consecutive edits.

To view Edit Distance:

1. Click on the **View** menu, then select **Analysis** and click on **Edit Distance.**
2. Click on the **Settings** button to set parameters; if parameters are already set, click on the **Process** button.

The **Settings** for Edit Distance allow the user to specify which events are considered Edits.

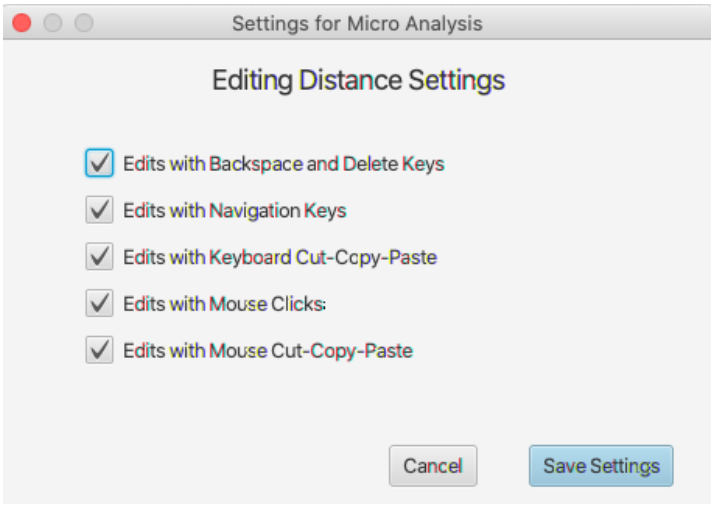


Figure 8.33: Editing Distance settings form

The Editing Distance Analysis indicates the:

- Time the edit occurs.
- Event number in the Session Log.
- Number of events between two consecutive events.
- Amount of time between two consecutive events in milliseconds.
- Edit (input type).

Table 8.14: Editing Distance example

Time	Event No	Edit Distance	Edit Time Distance (ms)	Input
00:00:00	0	0	0	<PRIMARY>
00:00:37	88	88	37606	BACK_SPACE
00:00:43	93	5	5677	<PRIMARY>
00:00:52	97	4	9338	BACK_SPACE
00:00:52	98	1	270	BACK_SPACE
00:00:56	105	7	3966	<PRIMARY>
00:01:02	110	5	5843	BACK_SPACE
00:01:02	111	1	196	BACK_SPACE
00:01:03	112	1	375	BACK_SPACE
00:01:19	162	50	16132	<PRIMARY>
00:01:24	164	2	4871	COMMAND+
00:01:24	165	1	514	<PRIMARY>
00:01:25	167	2	1072	COMMAND+
00:01:27	168	1	1771	LEFT
00:01:55	235	67	27396	<PRIMARY>
00:01:59	237	2	4645	COMMAND+
00:02:03	238	1	3960	BACK_SPACE

The following table provides the codes for Input in the Editing Distance analysis:

Table 8.15: Coding of events for Editing Distance calculation

Input	Setting	Event description
<PRIMARY>	Edits with Mouse Clicks	Primary mouse button clicks
<PRIMARY_DOUBLE>	Edits with Mouse Clicks	Primary mouse button double clicks
<SECONDARY>	Edits with Mouse Clicks	Secondary mouse button clicks
CONTROL+	Edits with Keyboard Cut-Copy-Paste	Cut, Copy or Paste using keyboard shortcuts (Windows & Linux)
COMMAND+	Edits with Keyboard Cut-Copy-Paste	Cut, Copy or Paste using keyboard shortcuts (Mac OS X)
BACK_SPACE	Edits with Backspace and Delete	Backspace key presses
DELETE	Edits with Backspace and Delete	Delete key presses
LEFT, RIGHT, UP, DOWN, PAGE_UP, PAGE_DOWN, HOME, END	Edits with Navigation Keys	Navigation key presses
<MENU_CUT>, <MENU_COPY>, <MENU_PASTE>	Edits with Mouse Cut-Copy-Paste	Cut, Copy or Paste event with mouse

8.4 Session Log

The Session Log in the software allows the viewing of the log file in the software.

Index	Device	Input	Start	Clock	End	Clock	Event Duration	Pause Duration	Pause Location	Caret Position	Anchor Position	x	y	Selected Text	Replaced Text	Gene
0	<MOUSE>	<PRIMARY>	0	00:00:00	59	00:00:00	59	362	initial	0	0	145	57			navigate
1	<KEYBOARD>	SHIFT	1382	00:00:01	1635	00:00:01	253	1323	unknown	0	0	0	0			unknown
2	<KEYBOARD>	T	1515	00:00:01	1558	00:00:01	43	133	sentence beginning	0	0	0	0			append
3	<KEYBOARD>	h	1920	00:00:01	1964	00:00:01	44	362	within word	1	1	0	0			append
4	<KEYBOARD>	i	2311	00:00:02	2384	00:00:02	73	347	within word	2	2	0	0			append
5	<KEYBOARD>	s	2398	00:00:02	2491	00:00:02	93	14	within word	3	3	0	0			append
6	<KEYBOARD>	SPACE	2445	00:00:02	2563	00:00:02	118	47	between words	4	4	0	0			append
7	<KEYBOARD>	i	2611	00:00:02	2728	00:00:02	117	48	between words	5	5	0	0			append
8	<KEYBOARD>	s	2715	00:00:02	2775	00:00:02	60	104	within word	6	6	0	0			append
9	<KEYBOARD>	SPACE	2788	00:00:02	2894	00:00:02	106	13	between words	7	7	0	0			append
10	<KEYBOARD>	a	4066	00:00:04	4169	00:00:04	103	1172	between words	8	8	0	0			append
11	<KEYBOARD>	SPACE	4230	00:00:04	4350	00:00:04	120	61	between words	9	9	0	0			append
12	<KEYBOARD>	t	5414	00:00:05	5458	00:00:05	44	1064	between words	10	10	0	0			append
13	<KEYBOARD>	e	5536	00:00:05	5595	00:00:05	59	78	within word	11	11	0	0			append
14	<KEYBOARD>	s	5730	00:00:05	5789	00:00:05	59	135	within word	12	12	0	0			append
15	<KEYBOARD>	t	5896	00:00:05	5955	00:00:05	59	107	within word	13	13	0	0			append
16	<KEYBOARD>	SPACE	6045	00:00:06	6150	00:00:06	105	90	between words	14	14	0	0			append
17	<KEYBOARD>	d	6690	00:00:06	6750	00:00:06	60	540	between words	15	15	0	0			append
18	<KEYBOARD>	o	6841	00:00:06	6930	00:00:06	89	91	within word	16	16	0	0			append
19	<KEYBOARD>	n	7050	00:00:07	7109	00:00:07	59	120	within word	17	17	0	0			append
20	<KEYBOARD>	e	7184	00:00:07	7260	00:00:07	76	75	within word	18	18	0	0			append
21	<KEYBOARD>	SPACE	8729	00:00:08	8819	00:00:08	90	1469	between words	19	19	0	0			append

Figure 8.34: Session log example

9 Exporting files

GGXLog allows files to be exported into other formats. Files may be exported once a session file has been loaded and each setting is specified. Files can be exported in two ways.

* IMPORTANT!

- Safari, Firefox, Google Chrome and Opera are the only web browsers that will accurately render exported files. One of these browsers should be set as the system's default browser.
- Files exported to a web browser can be Exported as *Portable Document Format (PDF)*, printed into PDF, saved as a *Web Archive* file, captured as Screen Shots or printed out for future use.

A. Each view contains an **Export** button. To export files from the particular view:

1. Click on the **Export** button.

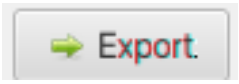


Figure 9.1: Export button on interfaces

2. Specify the **save location** and the **filename**.

This approach allows for the exporting of the files listed in the table below.

View	Exported Data
Session Graph	Session Graph as a HTML file opened in the web browser*
Graph Statistics	Graph Statistics as a HTML file opened in the web browser*
Graph Pause Analysis	Graph Pause Analysis as a HTML file opened in the web browser*
Linear text	Linear Text (format 1 and 2) saved as HTML files
S-Notation	S-Notation saved as a Microsoft Word document
Statistics	Statistics saved as a Microsoft Excel file, in which each worksheet in the file represents a selected table
Burst and Efficiency	Burst and Efficiency as a HTML file opened in the web browser*
Micro Analysis	Micro Analysis saved as a Microsoft Excel file
Editing Distance	Edit Distance saved as a Microsoft Excel file

B. From the **Export Files** menu

1. Click on the Project menu, then select Export Files.
2. Choose the files to export, then click on the Export button.

3. The files will be saved to the folder of the opened Session Log file, shown as the File Save Location.

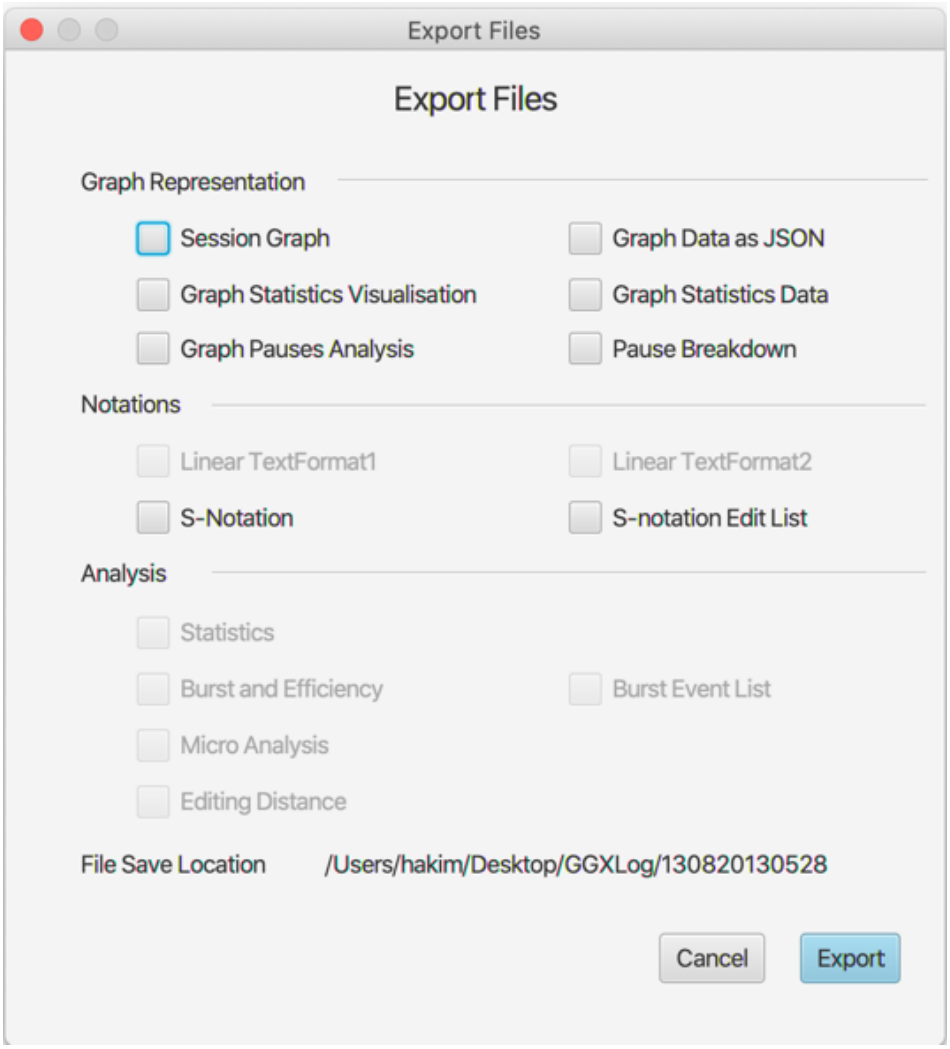


Figure 9.2: Export Files interface

Note: The exporting checkbox is only active if the relevant settings are saved in each interface, as the data created varies according to the specified settings.

This option allows for the following files to be exported:

- Session Graph as a HTML file opened in the web browser*
- Graph Statistics Visualisation as a HTML file opened in the web browser*
- Graph Data as JSON saves the Session Graph as a **GraphData.json** file that can be used to load the Graph data into Excel (only on MS Windows) or to visualise the Graph in another application that supports json format.
- Graph Statistics Data as **GraphStatisticsData.xls,** which includes the Table data
- Graph Pause Analysis as a HTML file opened in the web browser*
- Graph Pause Breakdown as a Microsoft Excel file as **GraphPauseBreakdown.xls**
- Linear Text (format 1 and 2) saved as HTML files as **LinearText1.html** and **LinearText2.html**
- S-Notation saved as a Microsoft Word document file as **S-Notation.doc**

- S-Notation Edit list as a Microsoft Excel file as **S-NotationEditList.xls**
- Statistics saved as a Microsoft Excel file as **Statistics.xls**. Each worksheet in the file represents a selected table
- Burst and Efficiency as a HTML file opened in the web browser*
- Session Burst Parameters and Event List as a Microsoft Excel file as **Burst-Efficiency.xls**
- Micro Analysis saved as a Microsoft Excel file as **MicroAnalysis.xls**
- Edit Distance saved as a Microsoft Excel file as **EditDistance.xls**

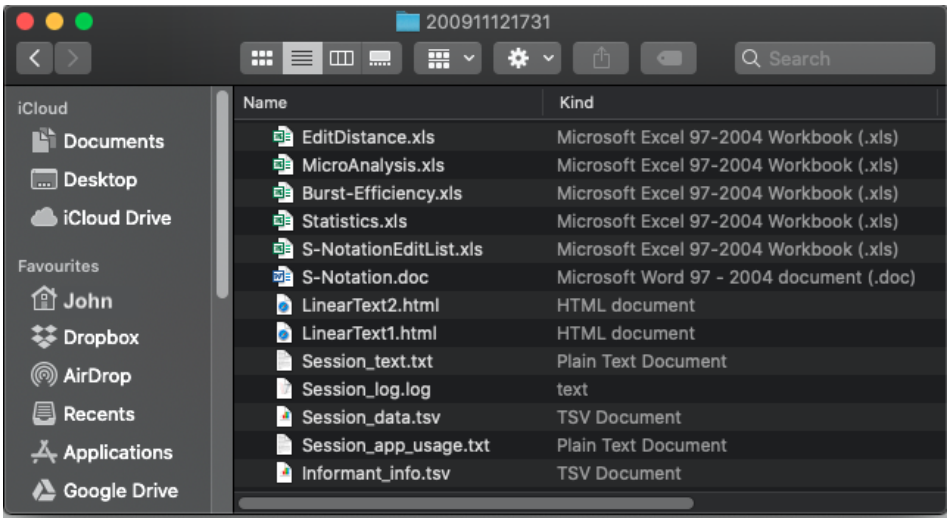


Figure 9.3: Exported files to Open session log folder

10 Preferences

Preferences permit the setting of common parameters. Preferences are categorised into three main sections:

1. General Settings (Interface)
   - **Interface Language**: Specifies the language used for the interface and currently supports Finnish, Swedish, French and English
   - **Application Folder**: Specifies the location on the computer where session files are stored
   - **Delete Working Folder on Exit**: Stores files that are created during Graph Visualization, Notations and Analysis. If ticked, the folder will be deleted upon exiting the programme
   - **Default font size in Text Window**: Specifies font size in Text Areas to help legibility

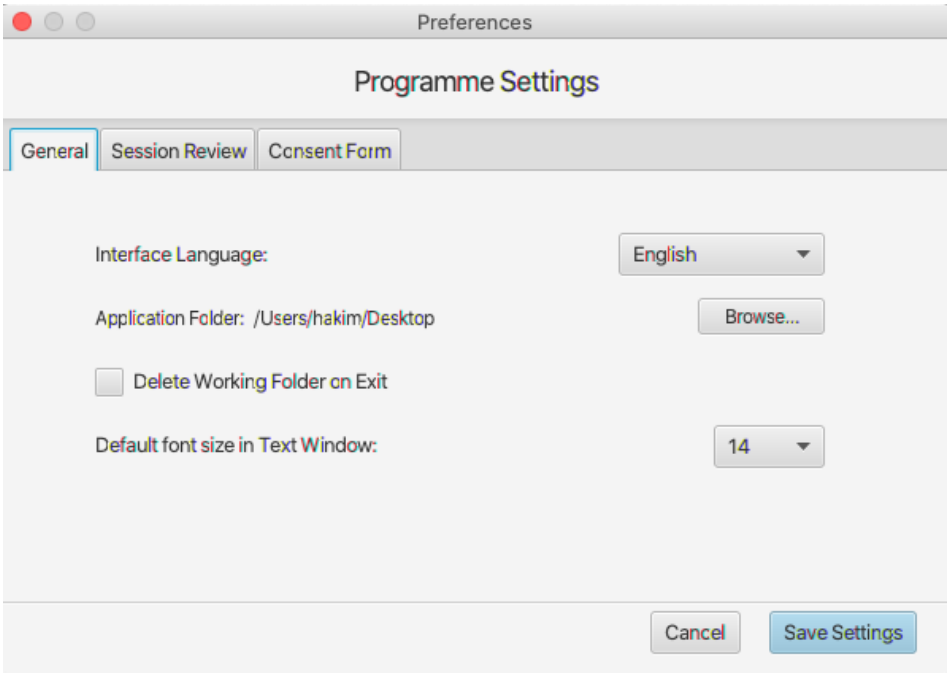


Figure 10.1: General Programme settings form

2. Session Review Settings

   - **Default Cognitive Pause Time**: This value, in milliseconds, will be used in all Settings where cognitive pause is specified; however, the pause value can be changed in the individual settings as required.
   - **Always Consider Cognitive Pauses**: If ticked, the default in all settings is to show or consider cognitive pause.
   - **Maximum number of Graph Nodes for automatic representation**: As the number of nodes increases and the Graph visualisation becomes complex, this threshold will be used to switch to a simpler representation of the Session Graph.
   - **Frequency to check other application use during writing**: This specifies, in seconds, the frequency with which 3rd party application usage information is logged during a writing session.
   - **Consider pasting text from the Clipboard at the end of text as an insert in S-notation**: If ticked, text that is cut/copied and pasted at the end of the text will be considered an Insert Event and not an Append.

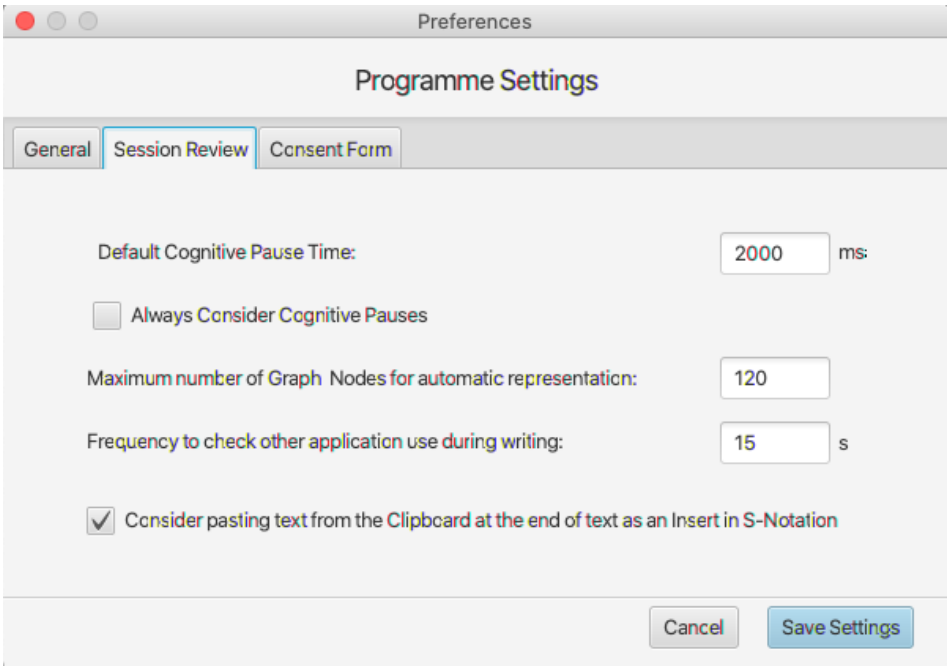


Figure 10.2: Session review settings form

3. Consent Form Text
   - **Consent Form Text**: This can be modified so that a custom Consent Form text will be displayed to the Informant at the beginning of the writing session. The text will be used on the computer the software is installed in until changed.
   - **Revert to Default Text**: This reverts to the default consent form text packaged with the software.

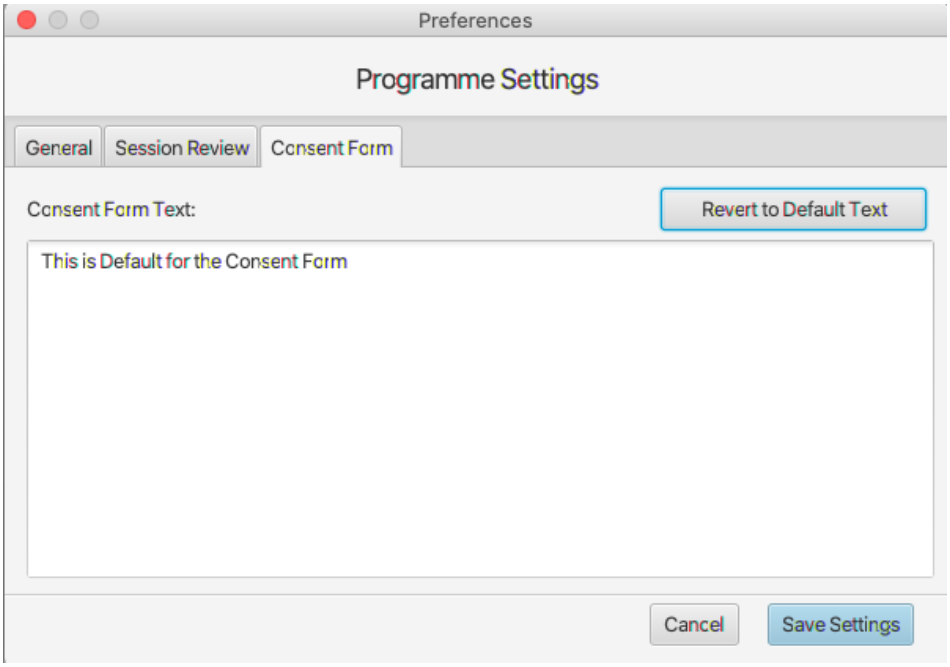


Figure 10.3: Custom Consent form text input interface

11 Tips

These are some tips that may help in making GGXLog more efficient.

- Tip 1. Open File name
  The full path of the open file is given in the **Title bar** of the Programme.
- Tip 2. Graph layout
  The Session Graph with rectangle nodes will have different layouts each time it is drawn. Hence, press the **Process** button until the graph layout is satisfactory and legible.
- Tip 3. Improve performance during graph manipulation.
  To manipulate the graph if the number of nodes is large;
  1. Export the graph to your web browser (Using the **Export** button in the interface or the **Export** menu)
  2. Close the Session Graph tab using the "**x**" on the graph tab
  3. Manipulate form the browser and print to PDF.

  Web browsers are more efficient in manipulating the graph as well as provides full screen access.
- Tip 4. Saving Session Graphs, Graph Statistics, Burst & Efficiency chart.
  To save sessions graphs and other data visualisations;
  1. Export the visualisation to your web browser (Using the **Export** button in the interface or the **Export** menu)
  2. Use **Save As** or **Save Page As** menu on the web browser to save your visualisations as web pages
     - On Safari as *Web Archive*
     - On Firefox and Google Chrome as *Web Page, Complete*

Note: The node placement of the Session Graph will not be saved. If a reference is required, print to PDF using the **Print** menu of the web browser or obtain a screen shot.

12 References

Bécotte, H.S., Caporossi, G., Hertz, A. & Leblay, C. (2019). Writing and rewriting: Keystroke logging's colored numerical visualization. In E. Lindgren & K. P. H. Sullivan (Eds.), Observing writing: Insights from keystroke logging and handwriting, Studies in Writing (pp. 95–122). Leiden: Brill Academic Publishers.

Caporossi, G. & Leblay, C. (2011). Online Writing Data Representation: a Graph Theory Approach. In J. Gama, E. Bradley et J. Hollmén (Eds.) Lecture Notes in Computer Sciences 7014 (Advances in Intelligent Data Analysis X), Springer Heidelberg, Dordrecht, London, New York, 80–89.

Caporossi, G. & Leblay, C. (2015). A graph theory approach to online writing data visualizaion. In G. Cislaru (Ed.) Writing(s) at the Crossroads: The Process-Product Interface. Amsterdam: John Benjamins. 171–181.

Carl, M. (2012). Translog-II: A program for recording user activity data for empirical translation process research. In Proceedings of the eighth international conference on language resources and evaluation. Istanbul, Turkey.

Grésillon, A. (1994). Éleéments de critique génétique. Lire les manuscrits, Paris, PUF-CNRS.

Kollberg, P. (1996). Rules for the S-notation: a computer-based method for representing

revisions. Stockholm, Sweden: IPLab, Royal Institute of Technology (KTH).

Miller, K., Lindgren, E., & Sullivan, K. (2008). The psycholinguistic dimension in second language writing: Opportunities for research and pedagogy using computer keystroke logging. TESOL Quarterly, 42(3), 433–454.

Sinar E.F. (2015), Data visualization. Tonidandel S., King E., Cortina J. (Eds.), Big data at work: The data science revolution and organizational psychology, Routledge, New York (2015), pp. 115–157

Wengelin, A., Torrance, M., Holmwvist, K., Simpson, S., Galbraith, D., Johansson, V., et al. (2009). Combined eyetracking and keystroke-logging methods for studying cognitive processes in text production. Behavior Research Methods, 41(2), 337–351.

West, D. B. (1996), Introduction to Graph Theory, Prentice-Hall, Englewood Cliffs, NJ.