QualCoder

QualCoder is free software for qualitative data analysis

Copyright Notice

1. Edition July 2020

Author: Colin Curtain

ccbogel@hotmail.com

Table of Contents

1 What is QualCoder

QualCoder is free, open source software for qualitative data analysis.

With QualCoder you can code text and images, write journal notes and memos. You can categorise codes into a tree-like hierarchical categorisation scheme. Coding for audio and video ican be performed and requires the VLC media player.

Reports can be generated for text coding and for coder comparison using the Cohen's Kappa statistic. A graph displaying codes and categories can be generated to visualise the coding hierarchy. Most reports can be exported at html, open document text (ODT) or as plain text files.

QualCoder is written in python 3 using Qt5 for the graphical interface. A Sqlite database is used to store the coding data.

QualCoder is available from https://github.com/ccbogel/QualCoder. There is also a wordpress site for QualCoder at https://qualcoder.wordpress.com/. Debians for Linux instillation are stored at https://github.com/ccbogel/QualCoder-Debians

1.1 Why use QualCoder?

Firstly, QualCoder is free. You do not need to pay for a commercial software license. QualCoder is easy to use and contains the core functionalities required for qualitative analysis. You use Linux. QualCoder has been developed on Linux and also works on Windows. There are many Free Open Source Software (FOSS) supporters who are willing to use and support FLOSS development.

2 Downloading files and dependencies

Download the latest QualCoder from GitHub: https://github.com/ccbogel/QualCoder. Unzip the folder. The main program code is in the *QualCoder* folder. The Examples folder contains some example files which can be loaded into a test QualCoder project. There are an install.sh and a QualCoder.desktop file which are used to create a Linux Launcher.

2.1 Preparatory Downloads

2.1.1 Microsoft Windows

You will need to download and install Python 3. Then install PyQt5, lxml and Pillow modules. A Python 3 release can be obtained here: https://www.python.org/downloads/windows/. Finally, you need to have the VLC media player installed. You can get this from the Windows Store or from their website: https://www.videolan.org/vlc/download-windows.html Please ensure that the Python and VLC downloads are in the same architecture – that is – 32 bit or 64 bit for both. Otherwise QualCoder will not work.

Once a Python 3 release is installed you should be able to install the required modules using the following command in the Windows Command prompt (The prompt is under Windows System):

```
python -m pip install pyqt5 lxml Pillow ebooklib ply chardet pdfminer.six
openpyxl
```

Sometimes there are problems recognising the audio/video VLC library file: libvlc.dll
Some solutions are to add the path of the file to PATH here:
https://stackoverflow.com/questions/42045887/python-vlc-install-problems?noredirect=1

2.1.2 Linux

I have created QualCoder.deb packages for easy install on Debian/Ubuntu systems. The Debians for Linux instillation are stored at https://github.com/ccbogel/QualCoder-Debians Choose the most recent as it will have better features and fewer bugs.

If you want to install manually, follow these instructions to create a program launcher:

Go to your downloads folder. Make the *install.sh* script executable. The example script below assumes you are in your home directory. Open a terminal and type the following to install QualCoder:

```
cd Downloads/QualCoder
```

Run the *install.sh* file. The install process will ask for your permission to install QualCoder into the /usr/share directory. It will also download the additional python modules.

```
./install.sh
```

The install.sh script will also install various modules as well as the VLC media player. If you do not use the install script, you need to run these commands to install the modules:

```
sudo apt install python3-pip python3-pyqt5 python3-lxml python3-pil vlc qpdf
python3-ebooklib python3-ply python3-six python3-chardet

sudo python3 -m pip install pdfminer.six openpyxl
```

2.1.3 Mac OS

Install Python3 and VLC.

Install the Python dependencies using `pip`:

```
pip install pyqt5 lxml pillow six ebooklib ply chardet qpdf
pdfminer.six openpyxl
```

There is no desktop icon launch right now for QualCoder on Mac OS. Open a new Terminal window in the directory and launch with `python qualcoder.py`.

3 Starting QualCoder

3.1 Linux

If you have successfully installed a Debian package or successfully run the install.sh script there will be a program launcher. Double-click to start.

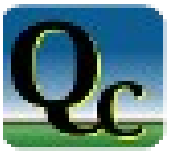

Figure 1: QualCoder launcher

Alternatively, using the terminal, move to the inner QualCoder folder and open the qualcoder.py file:

```
python3 qualcoder.py
```

3.2 Windows

In Windows create a desktop shortcut to the qualcoder.py file and double click to open.
Alternatively, using the command prompt, move to the directory and open the qualcoder.py file:

```
python3 qualcoder.py
```

3.3 Mac OS

Using the terminal prompt, move to the directory and open the qualcoder.py file:

```
python qualcoder.py
```

3.4 Backups

QualCoder has several backup options that can be changed in the settings. QualCoder can make a backup of a project every time the project is opened. This option can be modified where only non-audio/video files are backed up, this option might be chosen for speedy but less complete backups.

Another option is to not backup the project every time it is opened. However, I recommend that you make a backup before doing any substantial changes, such as reorganising codes and categories.

4 How to use

There are a few sample files in the Examples folder. These can be used to test importing files of different document formats and of importing an image. There are also example files to test importing case attributes, and for importing a survey.

4.1 Create a new project

Create a new project under the 'Project' menu, call it test. The project will be saved as *test.qda*. Test.qda is actually a folder containing subfolders which hold the database and other files. Additionally, in your home folder QualCoder will create a .qualcoder folder with a QualCoder.log file for logging events, a config.ini file which will contain the current coder's name, preferred fonts and a preferred working directory. There will also be a recent_projects.txt file which stores the most recdntly opened file. When opening QualCoder the most recent file will automatically be opened.

Under the *Project* menu click on *Project memo*. This is a memo about your project. Type "A test QDA project" and click the OK button.

You will notice the main window displays the following, with a menu bar at the top:

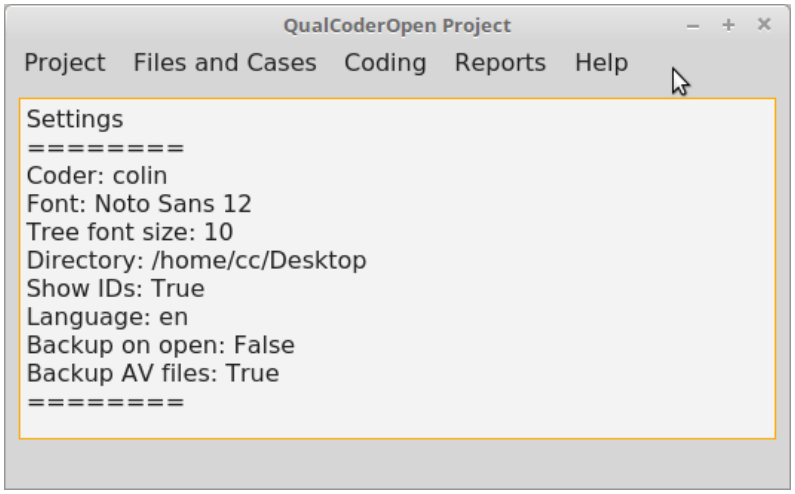

Figure 2: Main window

The main window displays various changes that are made when using QualCoder.

4.2 Settings

The settings dialog allows you to change the coder name. Multiple coders can code the same text. Checking the Show IDs box displays ID columns for data stored in various tables. The default language of QualCoder is English. Currently French, German and Spanish are alternative options. These alternatives have not been thoroughly reviewed for accuracy of translations. Once the

language is changed, close then re-open QualCoder to see the changed language.

Project backup options are set here too. When the backup every time project is opened is checked, an additional option is to choose to backup the audio/video files. For completeness of backups it is recommended to back up audio and video, however this may slow down opening of the software if you have many large files.

The backup name is changed to: *projectname_BKP_yyyymmdd_hh.qda* Where *hh* is 24 hour time. This also means multiple backups are not done within an hour. Too many backups was a nuisance issue. So a new backup cannot overwrite another backup created within the same hour.

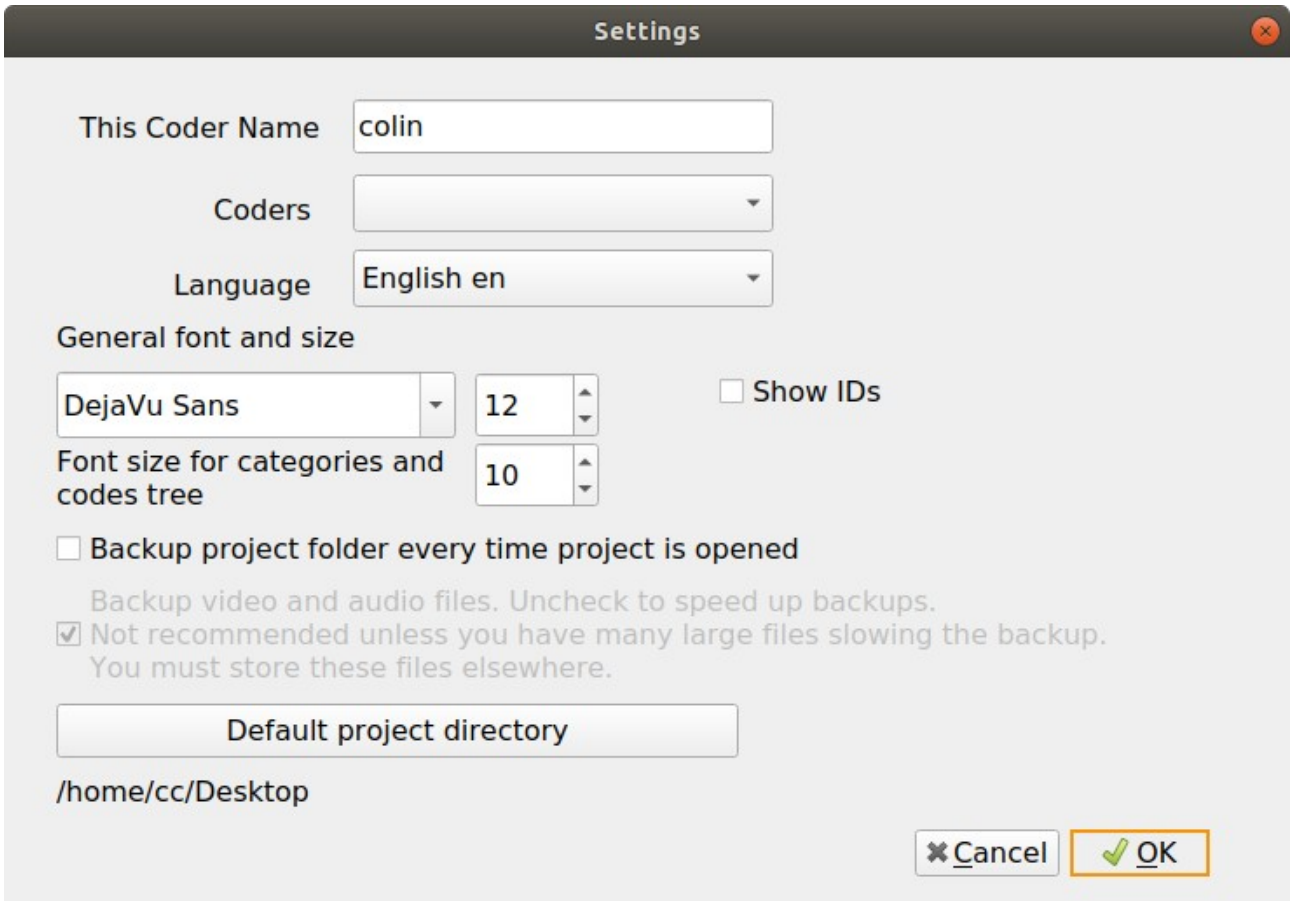

Figure 3: Settings dialog

4.3 Files and Cases Menu

4.3.1 Manage Files

You will usually want to load text files into QualCoder prior to coding. You can import text from plain text documents, docx, odt, html, htm, epub and pdf documents. Html text is loaded but may need further editing to suit, as formatting will not match the original page. Pdf importing can

sometimes be problematic. Large Pdf files take a long time to import. Another option is to manually enter text from within QualCoder. You can also load image files in the following formats: jpg, jpeg and png.

Video (mov, mp4, wmv format) and audio files (wav, mp3 format) can be imported. Loading an audio or video file will also automatically create a blank text transcript file. This file will have the same name as the audio or video file, but have a '.transcribed' suffix, shown in the image below. Initially, this text transcription will be empty. You can transcribe the file your self or get the file professionally transcribed and copy and paste the text into this file.

A right-click context menu allows you to view, export, delete a current file, Figure 4. The menu also allow you to re-order the files according to alphabet, date, file-type and alphabet, or to show only selected attribute types, if the right-click occurs in an attribute column.

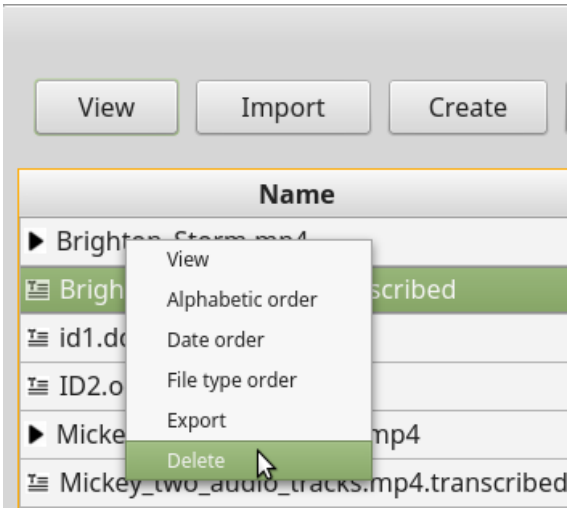

Figure 4: File manger context menu

Attributes

Attributes are variables that can be used to describe or classify the files. These can be added here or through the Manage Attributes menu option.

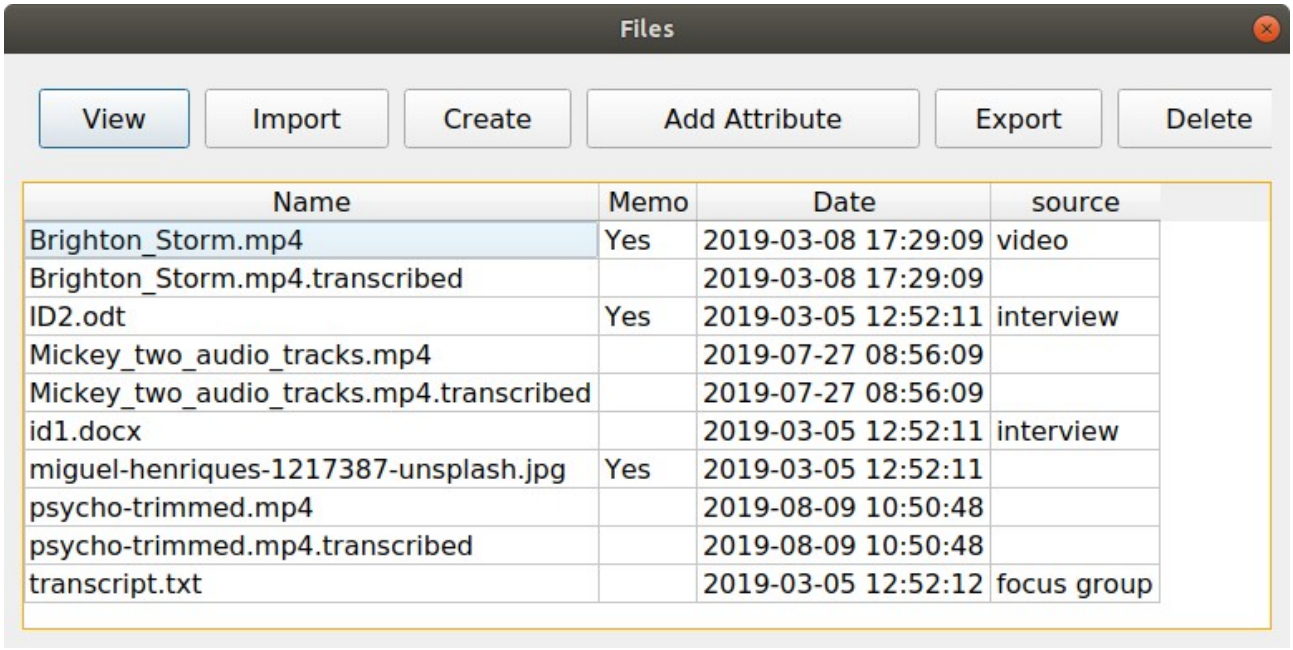

Figure 5: Manage files dialog

As a practical example: Open the *Manage Files* dialog. In the Examples folder import the following files: ID1.docx, ID2.odt, transcript.txt and the miguel-henriques.jpg.

4.3.2 Text file editing

Text files can be edited easily providing no coding or annotation or case assignment have been done with the text file. If the file has already had some coding or annotations assigned, the sections of the text file will be shown underlined in red. Editing text is now restricted to small text selections up to 20 characters. This occurs by selecting some text then right click and select the edit text option.

4.3.3 Viewing audio video

You can open an audio or video file to view. The video file might contain multiple audio tracks. There is a drop down box that allows you to choose another audio track.

Viewing opens two dialogs, one for viewing the video and one for the controls, shown in Figure 7. The transcribed text file must have the same name as the video file, but have a '.transcribed' suffix. The transcribed text file is stored within the sqlite database, but can be exported to a text file.

If you have a .srt file (a translation file that is read by VLC) you can place this alongside the video inside the project folder, in the video folder, shown in the image below. When the video is played, the translation wording will be shown as subtitles in the video. Also, if you open the .srt file in a text editor, copy and paste this into the video.mp4.transcribed text file. Then this text will be shown as the transcription for the video.

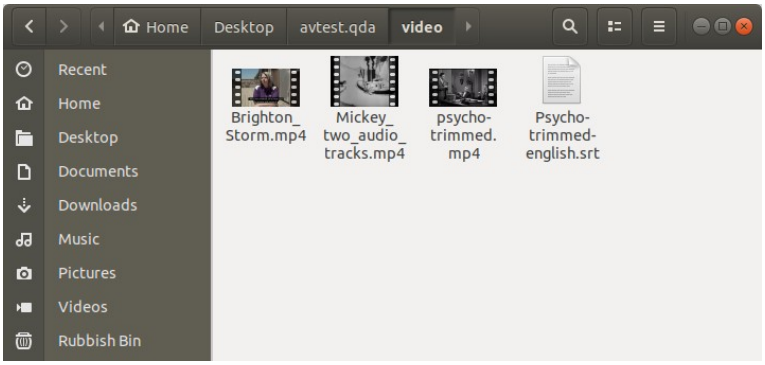

Figure 6: srt file location

Transcriptions should ideally contain timestamps indicating when the text is being pronounced during the video. The following formats are recognised by QualCoder, where SSS are milliseconds:

[hh:mm:ss], [mm:ss], [hh.mm.ss], [mm.ss], #hh:mm:ss.SSS#

hh:mm:ss,SSS --> hh:mm:ss,SSS

4.3.4 Transcribing audio and video

Transcriptions should contain timestamps and speaker names indicating who is speaking. Speaker names are bracketed in this format: [name]. Dots '.' and colons ':' cannot be used in speaker names.

Manually transcribing audio and video is helped with some keyboard shortcuts. You must be in the Transcription text box (this box will have an orange outline showing) to make the shortcuts work. Transcribing and adding or editing text can only occur if the existing text has no codes or annotations.

The shortcuts available are:

ctrl + r	Rewind 3 seconds
ctrl + s	Stop/Start toggle audio/video
ctrl + t	Insert timestamp in this format: [hh.mm.ss]
ctrl + n	Add a speaker name. This also pauses the audio/video.
ctrl + 1 to 8	Insert speaker name in this format: [name]

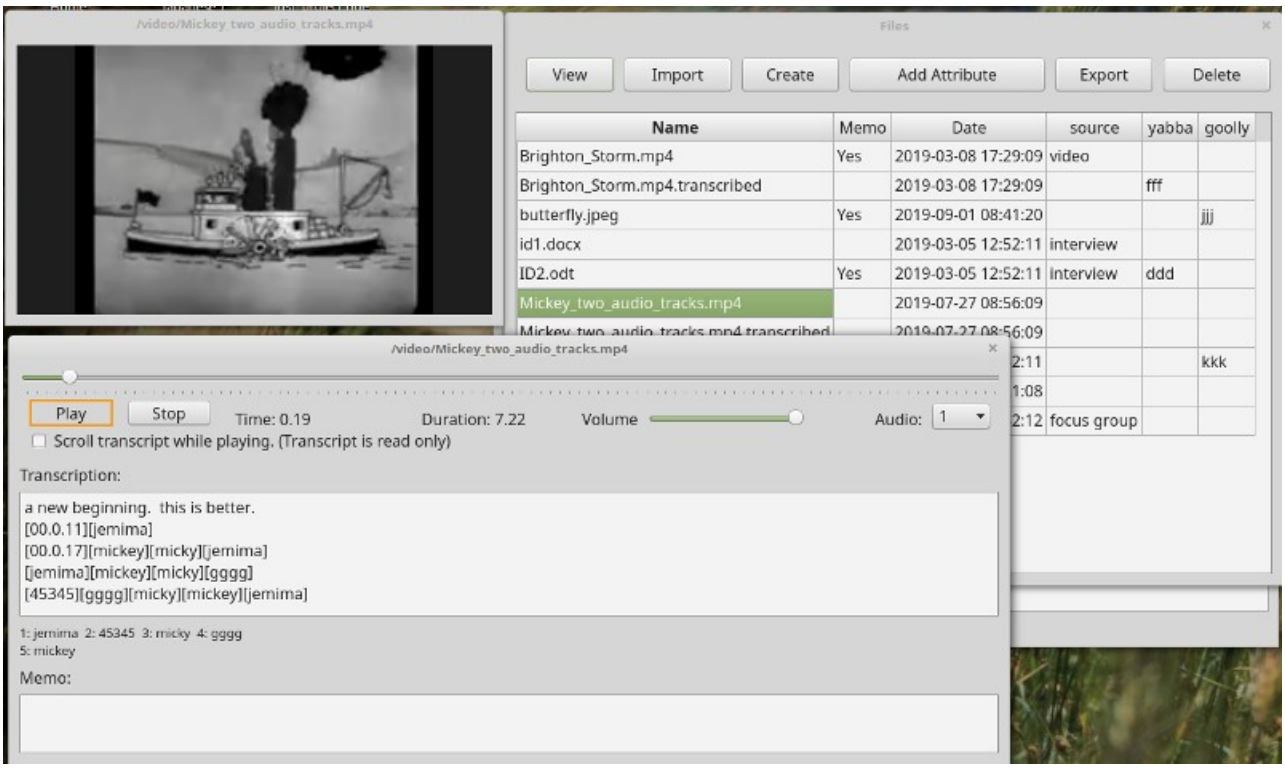

Figure 7: Video and control dialogs when viewing audio or video

4.3.5 Manage Cases

Open the *Manage Cases* dialog, shown in Figure 8.

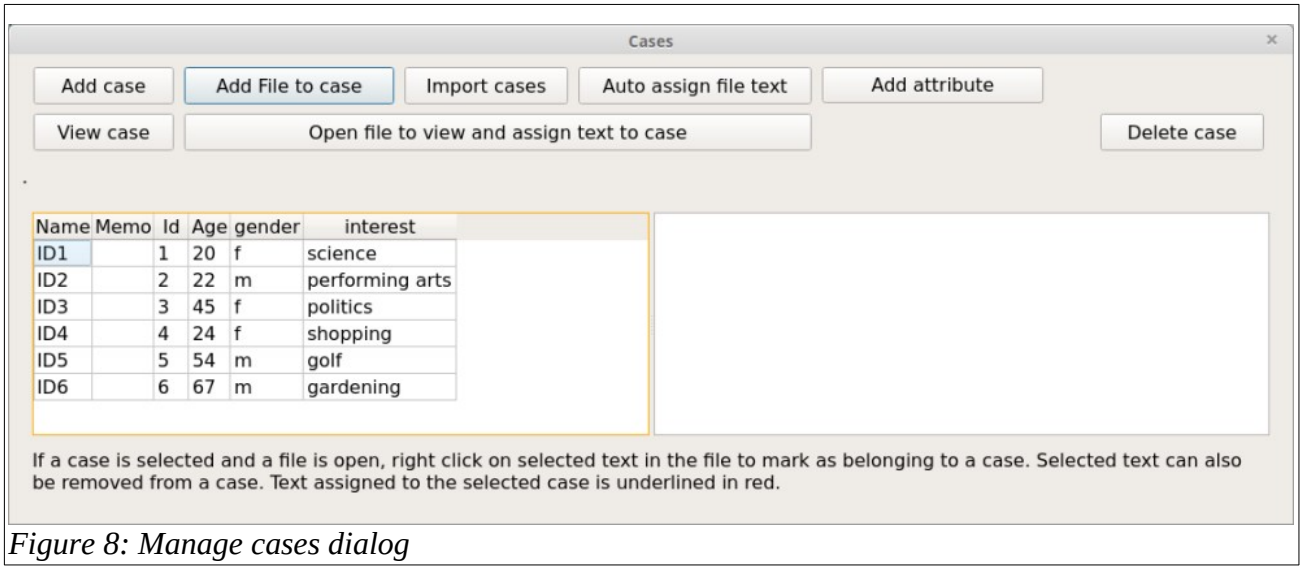

Figure 8: Manage cases dialog

Cases are useful for seeing text and imaged linked to particular cases and for assigning attributes such as age and gender to interview participants. You can add a memo of notes to each case and rename a case by double-clicking on a case name.

Practical example: Add the three students (or cases) here by clicking the *Add case* button. Call each student: ID1, ID2 and ID3.

Now, add each student's file to each student. Click on a case, say ID1, then click the *Add file to case* button. Select the ID1.docx file. Add the ID2.odt file to ID2 and add the miguel-henriques.jpg to ID3.

Use the *Open file to view and assign text to case* button and open file ID1.docx and select case ID1. You will see the text will be underlined in red which indicates this text is associated with this case.

Now open the transcript.txt file. Notice the transcript begins with a student id inside square brackets [ ].

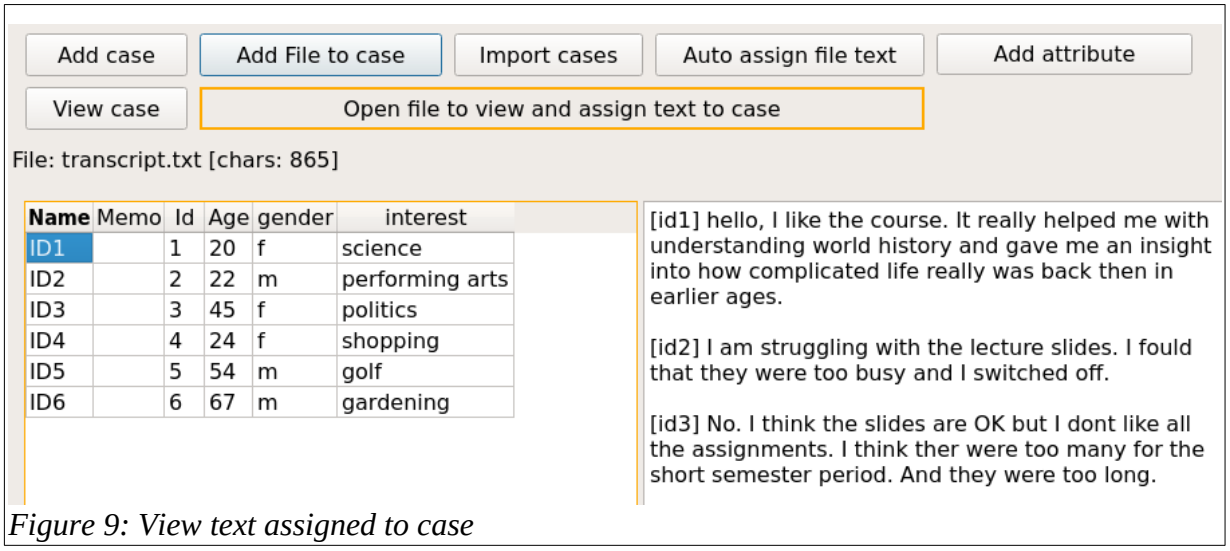

Figure 9: View text assigned to case

Try automatically assigning text to ID1 by selecting case ID1 then clicking the *Auto assign file text* button. You will be asked which file or files to assign the case to. Select *transcript.txt* from the list. Next you need to enter the start and end marks. The start mark will be '[id1]' and the end mark will be '[', note this is case sensitive so that is why you use the lower case here.

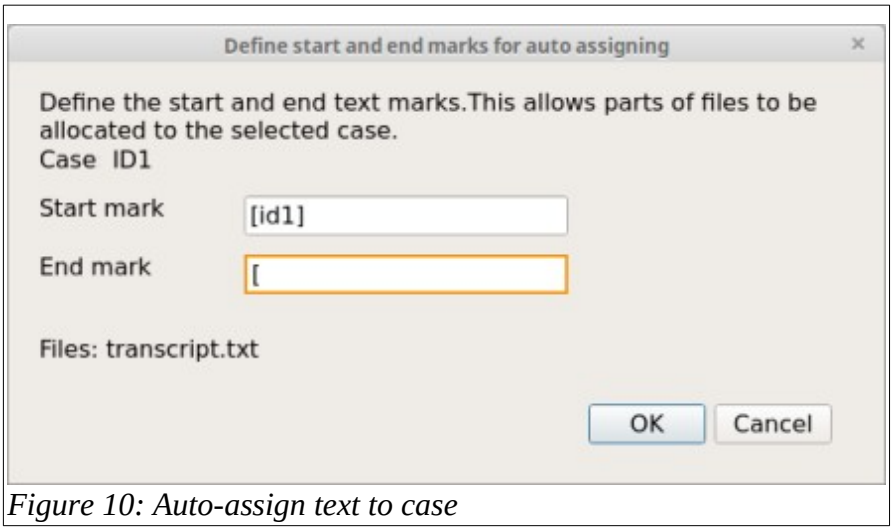

Figure 10: Auto-assign text to case

Now select the case ID1 and see the assigned text.clicking in a code gives a menu of options such as displaying the memo, or displaying all the case or file text that has been coded with the selected

code.

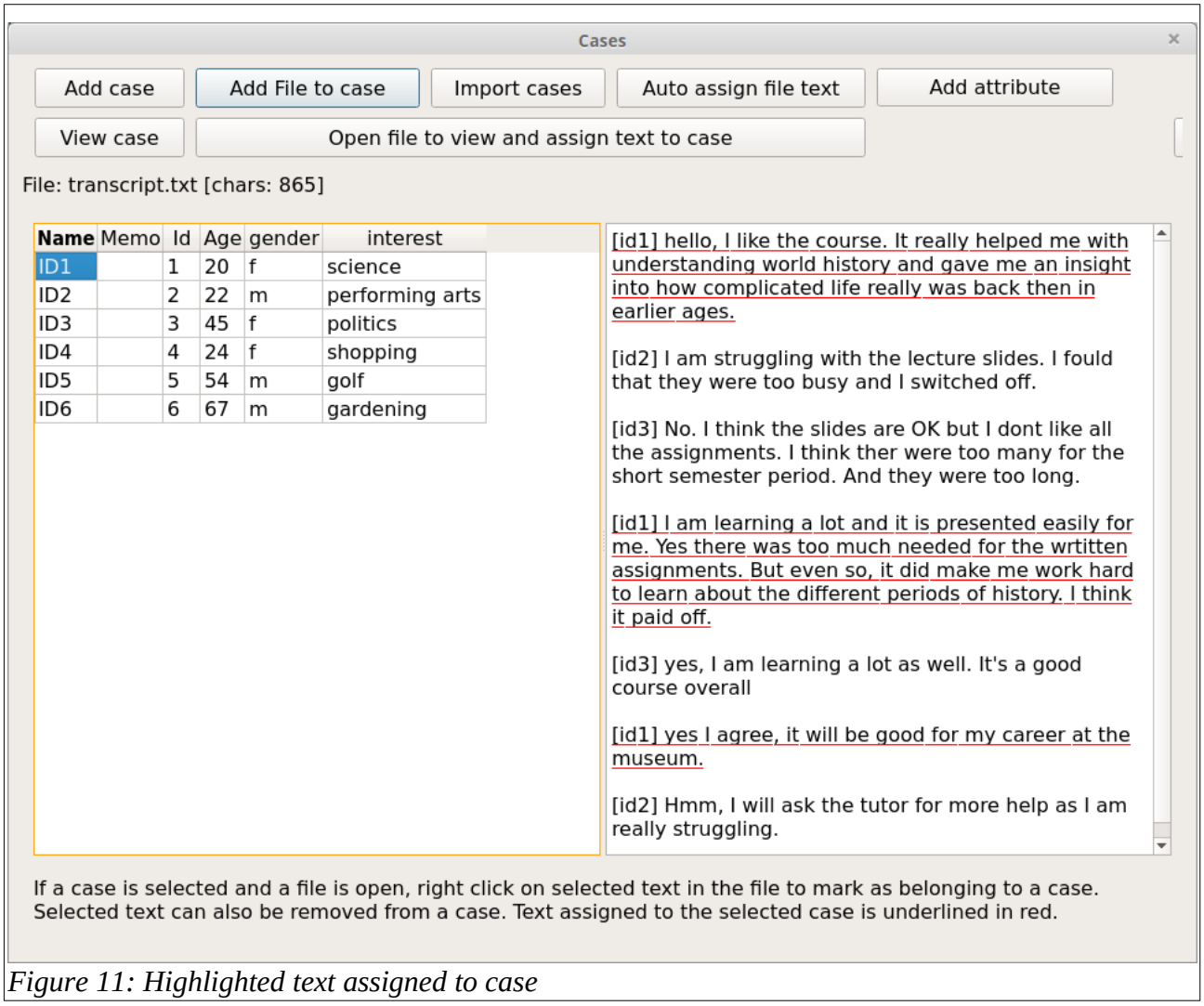

Figure 11: Highlighted text assigned to case

You can also select a case, manually highlight text, and right-click to mark – or assign -the text to the selected case.

Importing attributes for the cases. You can import attributes from a CSV file. The first row must contain the attribute headings. The first column must contain matching case names (for this example: ID1, ID2, ID3). Note that this is case sensitive. Open the attributes.csv file in the Examples folder to see how it is laid out.

For this example – delete all the cases shown. Then, import the attributes from the attributes.csv file through the *Import Attributes* dialog. You will again need to link the files and file text to each case.

4.3.6 Import survey

This asks for a CSV or Excel (xlsx format) file to be imported. You can try importing the survey.csv file in the Examples folder. You do need to have the python module *openpyxl* installed

to load Excel files.

Survey files in a CSV format, that is comma delimited format, or another delimiter (tabs are commonly used too). For tab-delimited files type *ta*, *tb*, or *tab* in the Delimiter box so that QualCoder knows the csv file is tab delimited. The first row must contain the headings for attributes. The first column must contain the unique identifiers for each survey respondent.

QualCoder will determine if the other columns (attributes) are Numeric or Character. QualCoder cannot determine if an attribute is qualitative data. You must right-click and change the field type from *character* to *qualitative* for those fields that need to be qualitative.

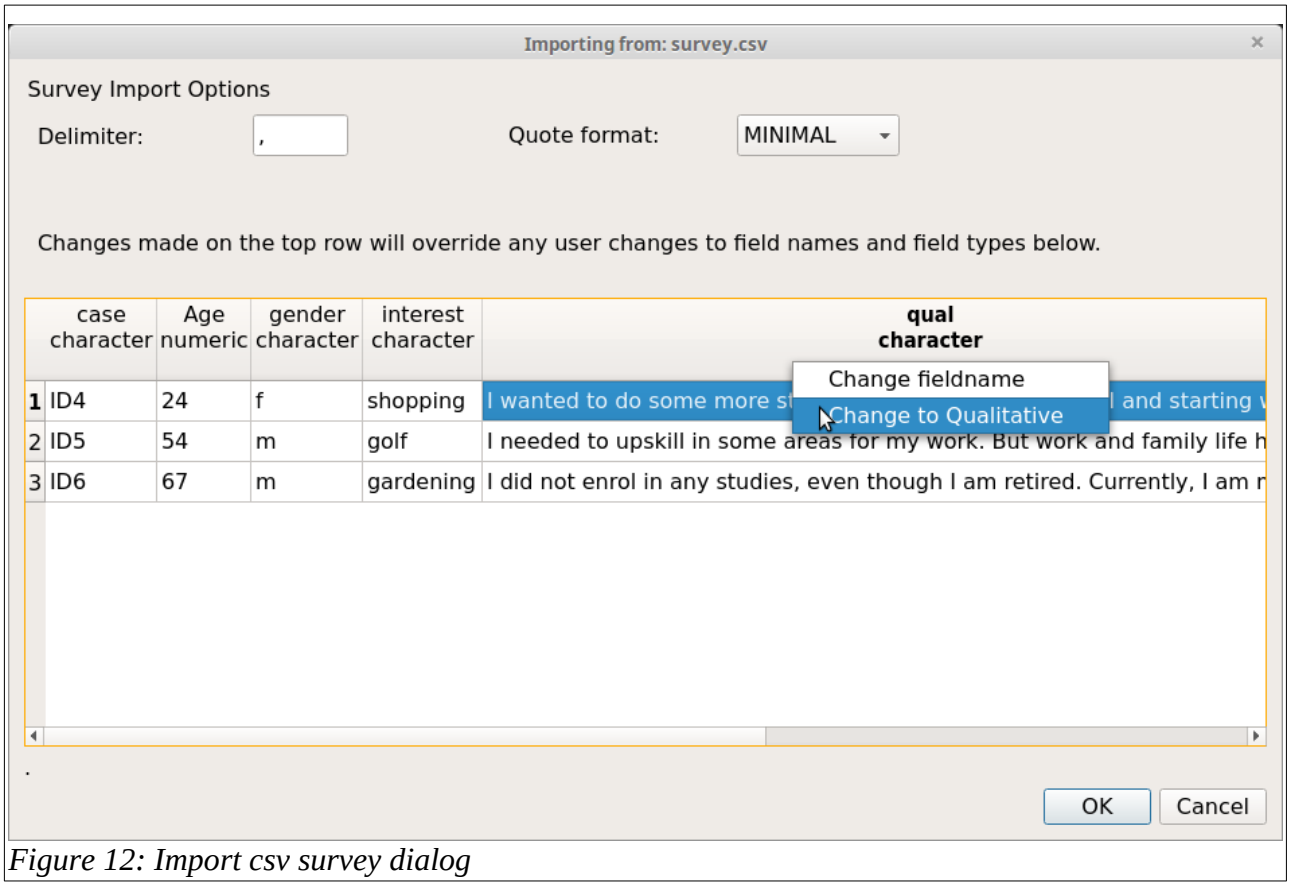

Figure 12: Import csv survey dialog

When you click on the OK button, the cases and their attributes will be added. The qualitative column will be converted to a file which will be named with the column name plus the current date and time. Each respondent's row will be prepended with [the unique id] so that you can identify each respondent. Also, the text for each respondent will be automatically linked to the corresponding case.

4.3.7 Attributes menu

Attributes are variables associated with files or cases. They can be useful to add context to the text analysis. Open the *Manage Attributes* dialog. You can add, delete, rename and add memo notes to attributes.

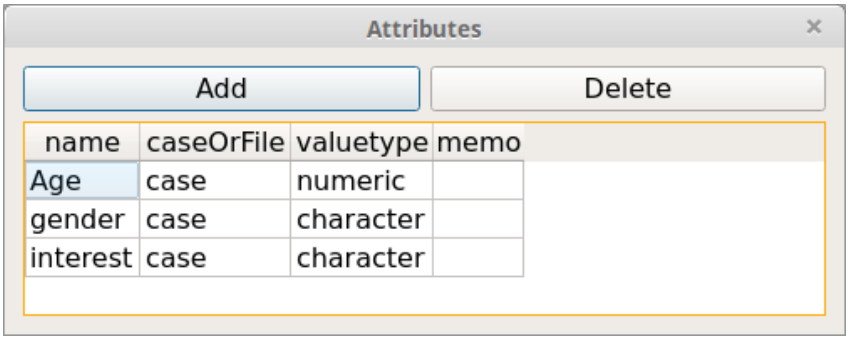

Figure 13: Attributes dialog

It is not the intention for QualCoder to perform statistical analyses of the attributes. It would be preferable to use dedicated statistical software such as R (https://www.r-project.org/) or other such software.

4.3.8 Coding text

Select *Code text* from the *Coding* menu. This is the central dialog for assigning codes to text. Once text segments are coded, hovering the mouse over the coding shows the code name as a tooltip. Clicking on the coded segment also shows the code name. Press the *View File* button to select a file to open for coding. Create a new code by right-clicking in the left hand window. Codes can also be assigned a colour by right-clicking on the code and selecting the *change code colour* option. Other options from the right-click menu include adding a memo to the code, deleting the code, renaming, adding a new code, and adding a new category.

The easiest way to code text, is to select some text, then left-click with the mouse on a code.

A second way is to select a code, then select some text. Right-click and mark the text to assign it to the selected code. Hover the mouse pointer over coded text to see a tooltip of the code. Coded text can be uncoded by clicking on the text segment and pressing the *Unmark* button.

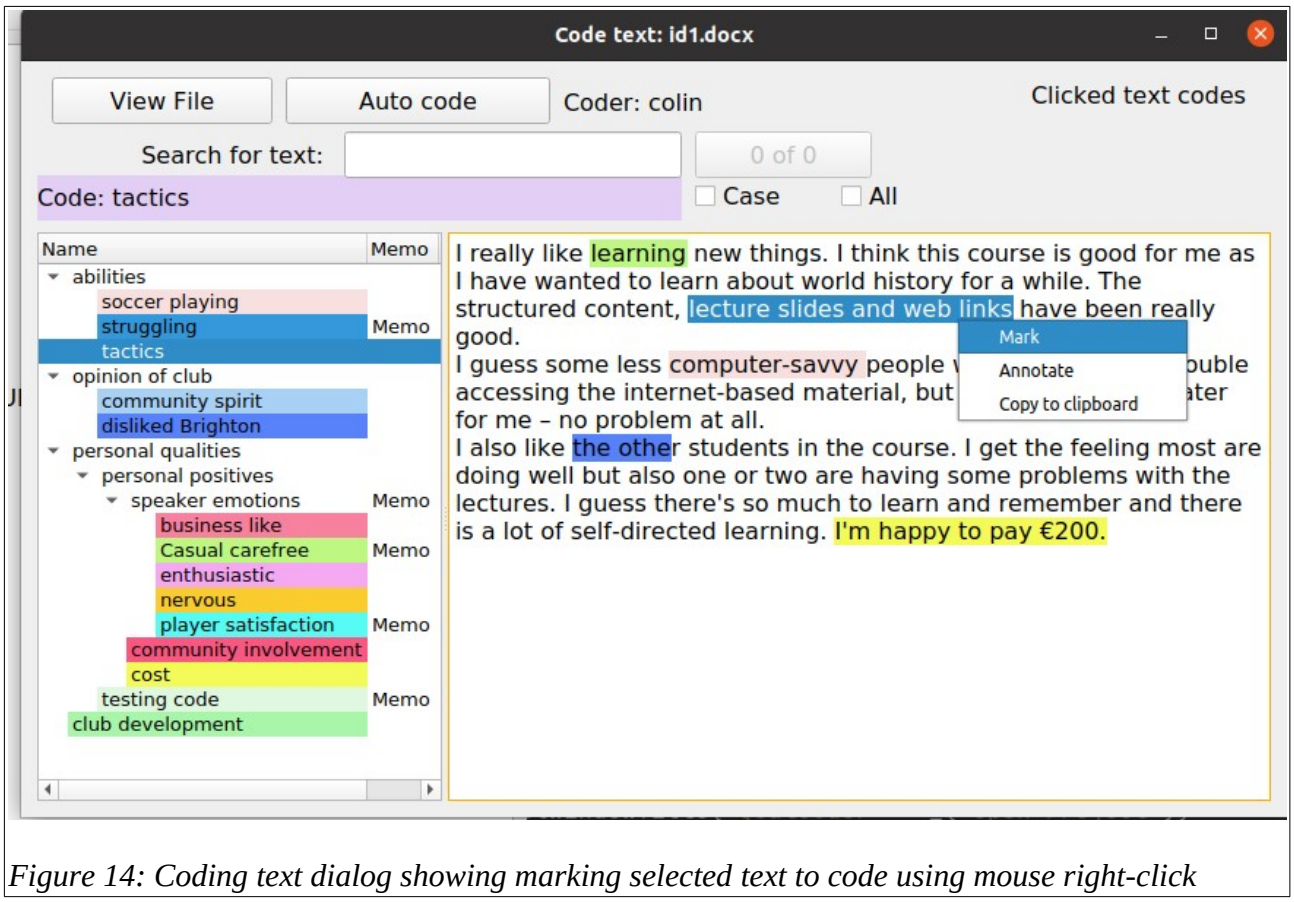

Figure 14: Coding text dialog showing marking selected text to code using mouse right-click

In the above Figure, If you click on coded text the *Clicked text codes* section shows the code(s) for that text in button(s). Clicking the button will underline text that has been coded with the code and select that code in the tree. The currently selected code in the tree is also shown above the tree.

Add an annotation (like a memo for a text segment) to a text selection. The text will become **bold** to mark the position of the annotation. To re-open an annotation, select some of the bolded-text and right-click to get the Annotate option.

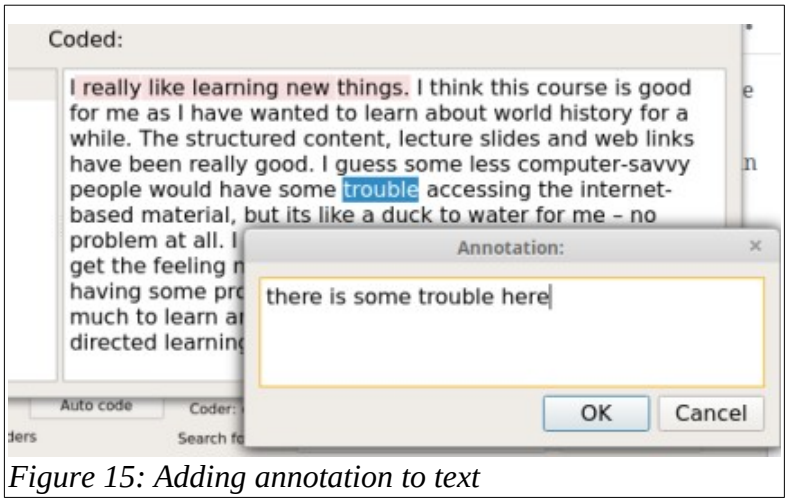

Figure 15: Adding annotation to text

You can also automatically code text segments using the *Auto Code* button. Enter the text you want

to autocode. A dialog asks for one or more files to autocode. All matching text will be assigned the selected code. Multiple sections of text can be assigned by autocode using the pipe '|' symbol. For example, *politics|politicians* can be assigned to the same code at the same time.

4.3.9 Categories and codes

Categories are used to organise codes. Categories are organised hierarchically in a tree structure. You can move codes into categories and move categories into larger categories. You can move categories and codes out of their current position. Codes and categories can be merged by dropping a code onto a code or a category onto a category. Categories and codes can be assigned memos

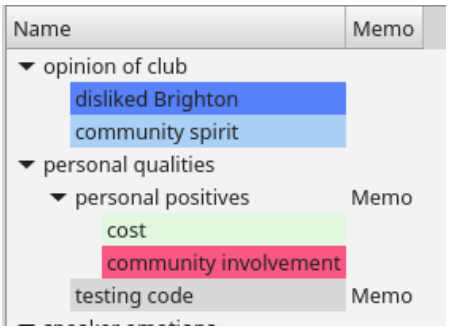

Figure 16: Categories and codes

4.3.10 Coding images

Images can be coded in a similar way to text coding. Select a code. Left-click and drag to highlight the area you want to assign to the code. Right-click will open a menu where you can remove the coding or add a memo. At the bottom of the screen there is a slider control to re-size the image. Dashed coded rectanges are colored to match the code color.

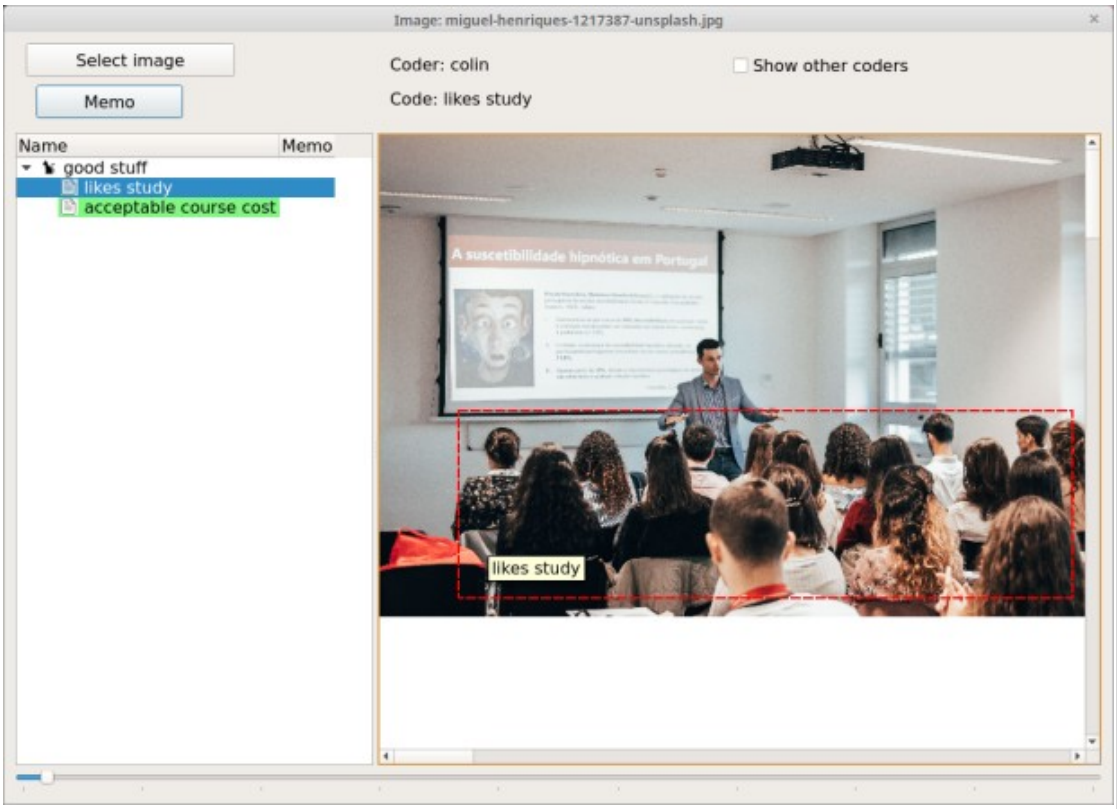

Figure 17: Picture coding showing code tooltip

4.3.11 Coding audio and video

When coding audio or video, two windows are displayed, shown below. One window has the audio or video playing. The other window has the controls and a button to begin and end a coded segment. The controls window shows the codes in the bottom left pane and the .transcribed file text is shown in the bottom right pane. The transcription text can also be coded and annotated in this window. Right-clicking on a timestamp will give you a menu option to go to that section of the video. If a video has multiple audio tracks, you can change the audio track too. Checking the scroll checkbox allows the transcript to scroll in time with the video, based on detected timestamps, however, you need to uncheck the scroll to be able to code the transcript.

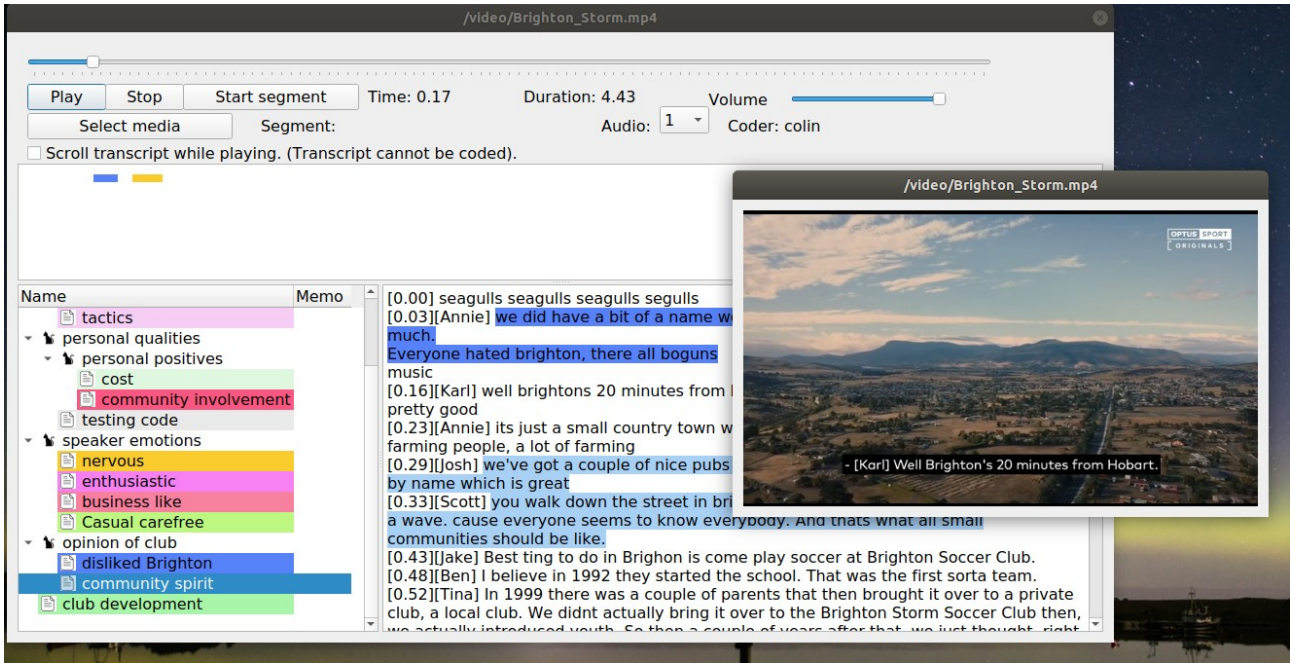

Figure 18: Audio/video coding dialogs

Once a segment is made using the *Start segment/Stop segment* button this can be assigned to a code by right clicking on the relevant code and assigning the segment.

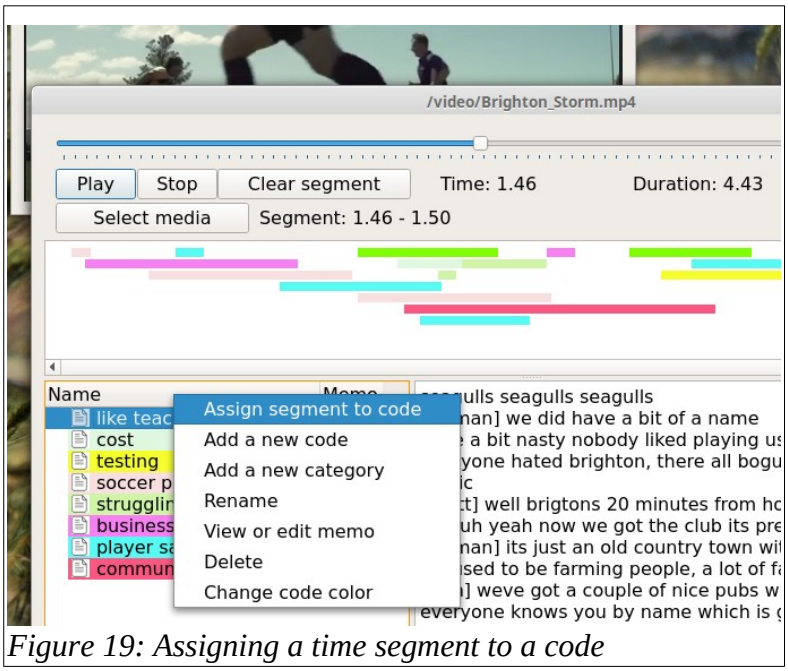

Figure 19: Assigning a time segment to a code

Coding stripes for the coded segments are shown in the upper pane. Hovering the mouse over each stripe shows the code name, time segment and any memo attached to that coded segment. Right clicking on a coded segment stripe shows a menu that can be used to edit the memo or delete the coded segment or play from this point. Coded stripes are shown on various lines so that they do not overlap.

The transcript text can also be coded and annotated in this dialog. However, when the Scroll

transcript check box is checked, this cannot be performed. Playing the video when this is check will scroll the transcript using the timestamps in time with the video.

Linking text to coded segments

There are two ways to link text and coded segments.

One way is to select text (it can be coded or preferably uncoded) then assign it to a segment. This is done in two steps shown in Figure 20. First select the text and right-click P*repare text link to segment*. Then right-click on the receiving segment, choose *Link text to segment*, to assign the text.

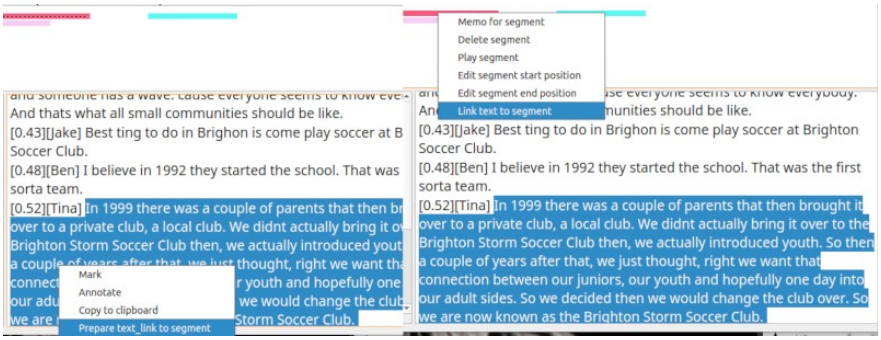

Figure 20: Assign a text selection to a segment

The second way is to select a segment and link it to a text selection via a similar process.

4.3.12 Codebook

A codebook is a list of your codes. Each code memo should detail the reasons for the purpose of each code. The codebook can be exported to a text file. The codebook also shows the frequency of the codes used (from all coders). Example codebook output is shown below. As codes and categories are put in a tree like structure the double minus ‘–‘ indicates the subordinate codes and categories within a category.

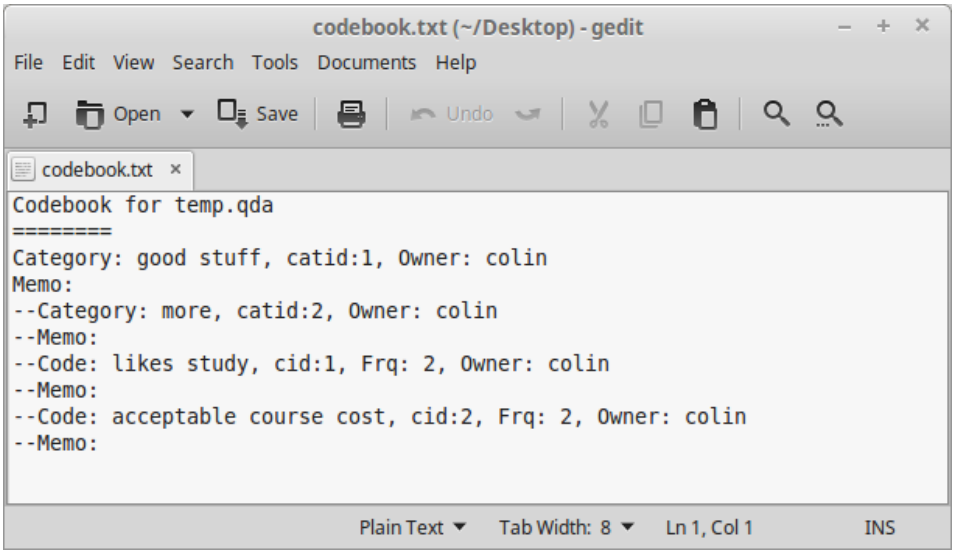

Figure 21: Codebook text file

4.3.13 Node graph

A graph of categories and codes is displayed. There are several options to change what is shown – such as Black and White, All or selected categories. Ther are two display styles – list view which is ordered with categories and codes or circular view. The circular view fans the codes and categories out, but you may need to move some aroudn as they can overlap of you have many.

Each code or category can be moved around by clicking near the edge of the box and dragging it around. Clicking on the text allows you to temporarily change the code or category wording. Right-clicking in a code gives a menu of options such as displaying the memo, or displaying all the case or file text that has been coded with the selected code.

Right-clicking on a line allows you to change the thickness, change to dotted line style or change the colour to red. This might be useful for emphasis.

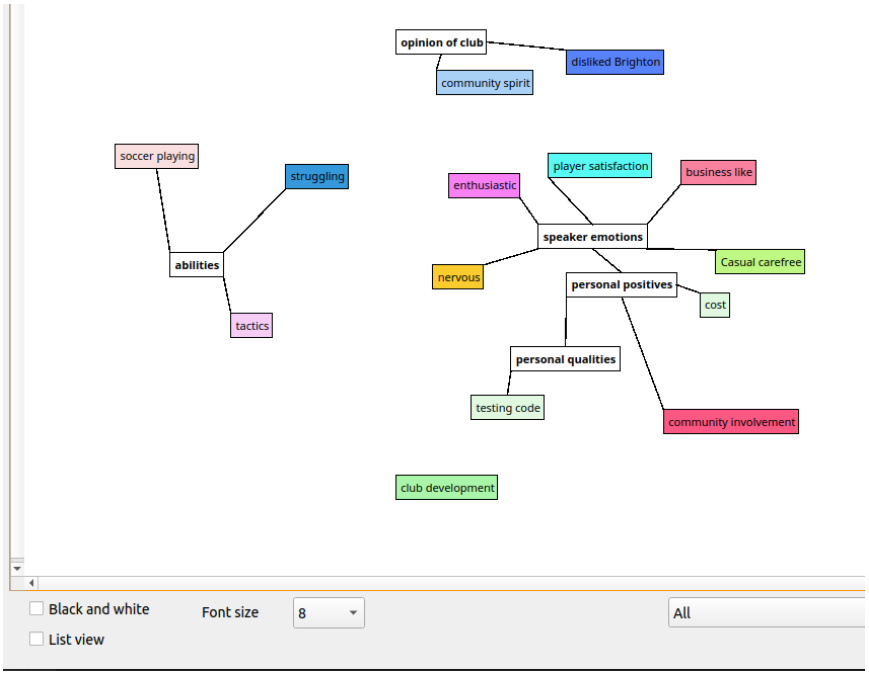

Figure 22: Circular view of categories and codes

4.4 Reports

4.4.1 Coding Reports

This dialog gives a list of coded text based on your selections. One or more codes need to be selected and one or both of *coder 1* and *coder 2* need to be selected. The *Search* button will present the results. If a category is selected all codes in that category are also selected. Multiple categories can be selected using the mouse and the 'ctrl' button. Codes and categories can be sorted to help find the codes of interest.

Codings can be narrowed down by using the *File selection Case selection* or *Attribute selection* buttons. Only codings within the selected files or cases will be presented. If text is entered into the *Search text* field, only codings which contain the matching text will be presented. Codings can also be narrowed down to case selections using the *Case selection* button. Note that when case selections are used, file selections are ignored. Also when file selections are used, case selections are ignored.

The *Search Text* box also limits the reported codings. The search text looks for any codings in text that contain the search text. The search text also looks for matching text in memos for coded image areas and coded audio/video segments. The search text function must be used in combination with *File selection, Case selection* or *Attribute selection*.

The *Attribute selection* button opens a dialog window where you can select attributes for files and cases, for example: only interview files for people aged > 60.

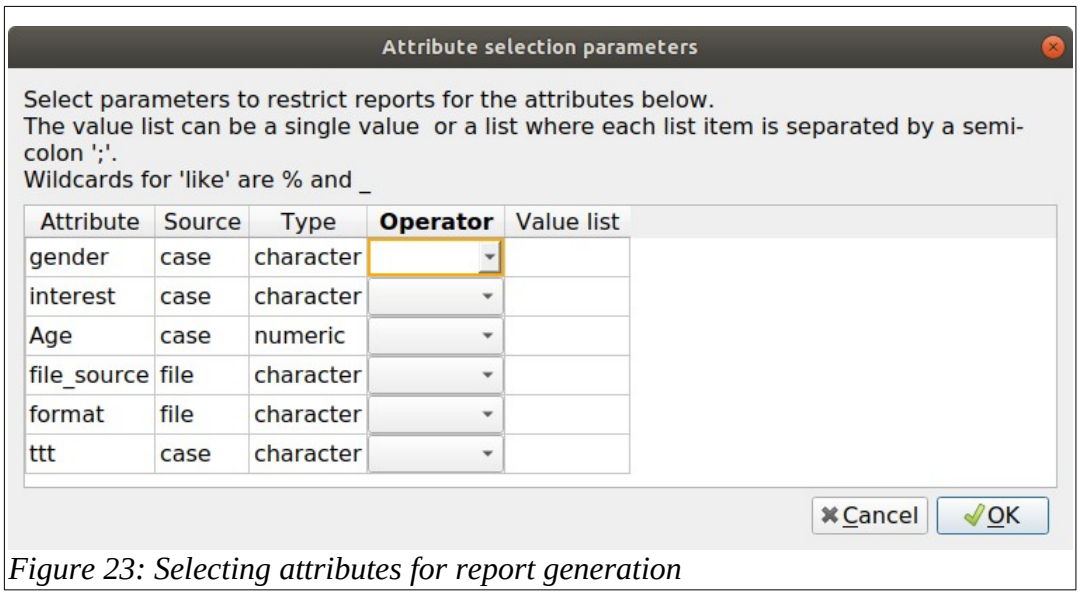

Figure 23: Selecting attributes for report generation

If you are creating a report based on Case selection, then the code tree pane is hidden. Instead two panes are shown. The left pane shows the codings in a list, the right pane shows a matrix with one case on each row and top-level categories in each column, shown as below:

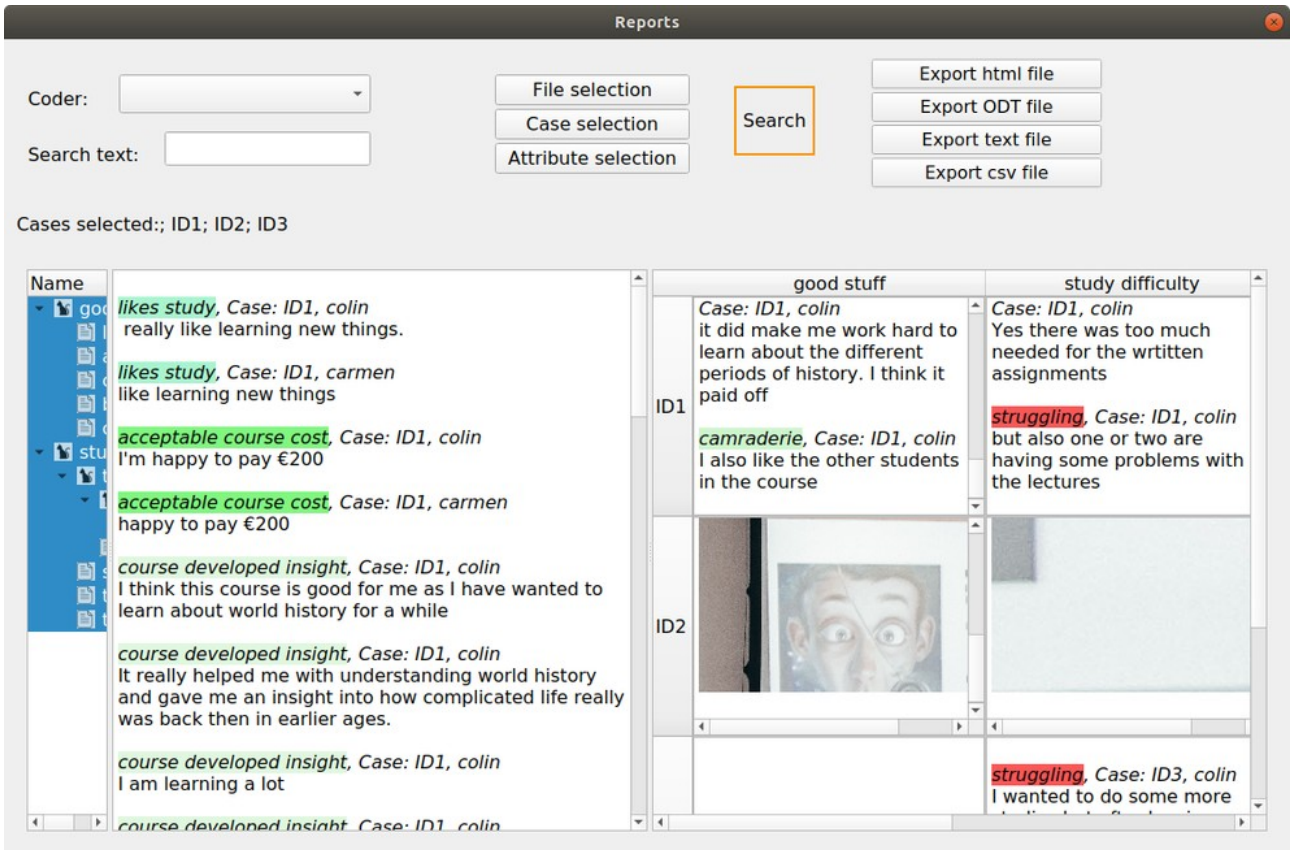

Figure 24: Case report showing code results and case by category matrix

Each of the three panes can be widened or narrowed by dragging with the mouse on the bar that splits each section.

Reports can be exported to text, open document (ODT) or html files. HTML files are provided as the main html file and a supporting folder which provides, images and audio or video media. Reports can also be exported as a csv file, where each column is a code.

Currently case matrices are not able to be exported.

4.4.2 Coding Comparison

This option shows the similarities and differences between two coders. Select two coders and click the *run comparisons* button. Coder compariosn is only available for coded text, not coded media files. For each code:

Agreement % shows agreement for a combination of coded and non-coded text characters.

A and B % shows agreement for the only the coded text characters divided by the total characters in the text.

Not A and Not B % shows the total of the uncoded text divided by the total characters in the text.

Disagree % shows the percentage of all the coded and non-coded text that did not match between coders. It is the same as 100 – the Agree %.

Cohen's Kappa is calculated based on the information in Wikipedia
https://en.wikipedia.org/wiki/Cohen%27s_kappa

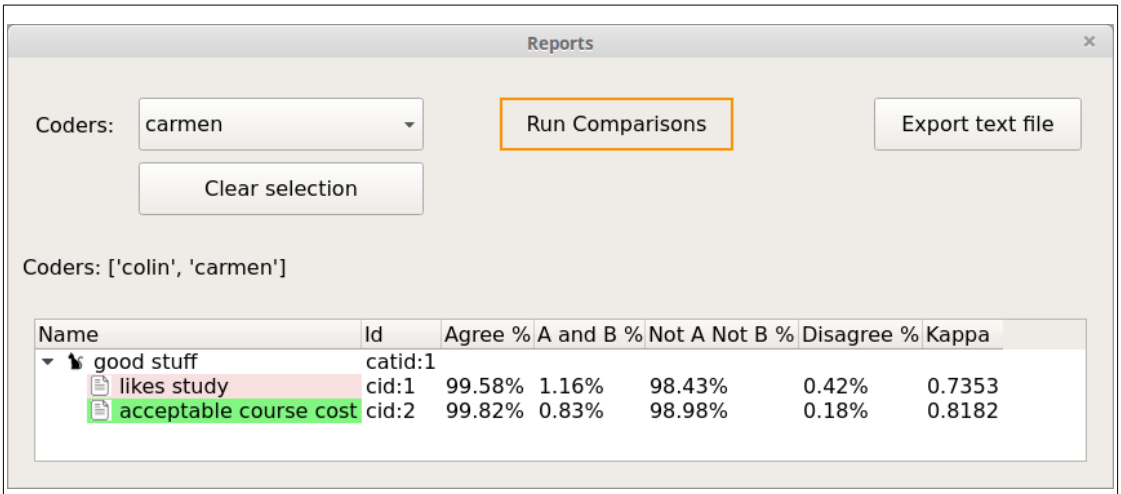

Figure 25: Coder comparison dialog

4.4.3 SQL Statements Dialog

This dialog contains three panes. The top pane is where SQL statements are entered and the bottom pane contains the results of queries. The left pane contains tables and field names. Double-clicking on a field name adds it to the SQL statement. Results can be exported to a text file. If you are not familiar with SQL take care as you will be able to update and delete the data as well as select data. Note: Some Unicode symbols are not converted to plain text and are ignored.

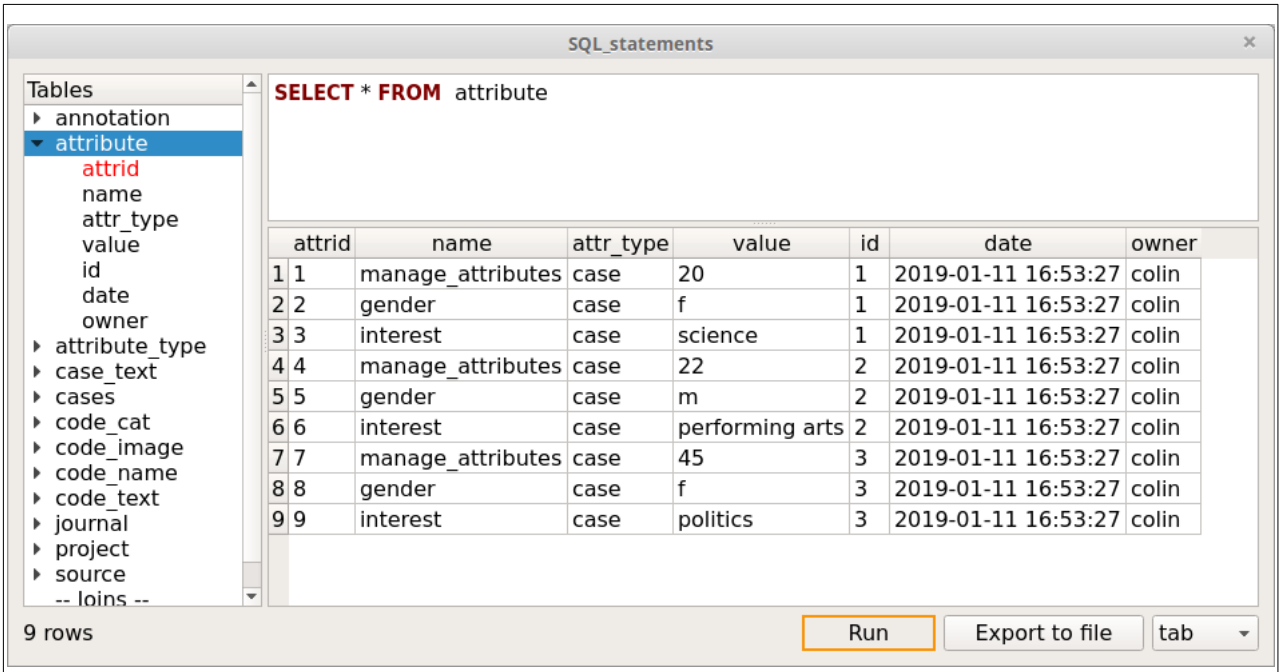

Figure 26: SQL dialog showing tables and fields on left, SQL statement and results

Most table fields are text. The following fields are integer: *anid, avid, attrid, caseid,catid, cid, fid, id, imid, jid, pos0,pos1, x1, y1, width, height.*

4.5 Multiple dialog windows

It is possible to have multiple dialog windows open. This is useful if you have run a report, or have a file open for coding, and you want to enter your thoughts into a journal entry. However there are some limitations.

Some dialog windows are priority windows, so swapping between windows will not be possible, these restricted windows are: Management of files, cases or attributes, survey import and confirmation dialogs (e.g. confirm delete) and project memo.

The second limitation is that changes in text editing in one dialog may not be reflected in the other dialogs. **Warning:** if you have the same journal entry open in two windows, the last window that is closed will be the entry that is saved. If you have the same text file opened for editing or transcribing in more than one dialog, the last one that is closed will be the entry that is saved.

5 Project import and export

5.1 REFI-QDA

The Rotterdam Exchange Format Initiative (REFI) is an open standard for exporting and importing codebooks and projects from and to different computer-aided qualitative data analysis software. The website for the exchange initiative is www.qdasoftware.org

Currently, QualCoder is able to export and import a codebook. Exportation of a project and importation of a project are experimental and do not meet the full REFI-QDA standard. For example: greater than 2GB files are not stored externally on export.

Importation of project.qpdx files partly works, but at least for Nvivo, coding stripes do not correctly match text selections. External files greater than 2GB may not be imported into QualCoder project. Features such as sets and graphs cannot be imported as this functionality is not withing QualCoder. Other data formats are not preserved, for example date or boolean would be converted to text data. To import, close any currently opened project first.

5.2 RQDA

Projects made with RQDA (http://rqda.r-forge.r-project.org/) can be directly imported. All data except for file categories are imported.

Close any opened project. Then in the Main Menu under Project click on RQDA Project import. You will be asked to create a new project- so enter a name. Then you will be asked to select the RQDA project file. QualCoder will then import the data.

6 Other details about QualCoder

The qda data folder contains folders for imported documents, images, audio and video. It also contains the sqlite database, named data.qda, to store coding data.

QualCoder creates QualCoder.log and config.ini and recent_projects.txt files inside a .qualcoder folder in your home directory. The config.ini file contains the name of the current coder, a default working directory, font choice and language. The log file records program errors and some user actions. QualCoder is written in python 3 using Qt5 for the graphical interface.

6.1 Acknowledgements

Ronggui Huang and Zhang Gehao for creating RQDA, which inspired this software. Mike MacCana for the source code for the docx module. User: bit4 on stackoverflow who presented the source code to convert html to text. Pdfminer3k: Copyright (c) 2004-2010 Yusuke Shinyama <yusuke at cs dot nyu dot edu> ebooklib: Aleksandar Erkalović (https://github.com/aerkalov). The VideoLAN team for the bindings to VLC. To various members on github for supporting this project.

6.2 Publications citing QualCoder

Local–global linkages: Challenges in organizing functional communities for ecosocial justice. Joel Izlar, Journal of Community Practice 27(3-4) 2019

Barriers to Health: Understanding the Barriers Faced by Community Intervention Projects. Vera Landrum, The University of Southern Mississippi 2020, Available from: https://aquila.usm.edu/cgi/viewcontent.cgi?article=1772&context=masters_theses

Framing food geographies. S Ramsay, Masters Thesis, Stockholms Universitet 2020

7 Future plans

Currently QualCoder has been used on Ubuntu 19/04, Linux Mint 19.04, Lubuntu 18.04, Windows 10 and MacOS. Limited testing has been performed on Mac OS.

In Windows, reports exported in ODT format are okay, but some images may overlap when the ODT file opened with Microsoft Word rather than opened with LibreOffice. There continue to be occasional issues in Windows with the software finding the VLC libvlc.dll file for audio/video work.

Further testing on different operating systems is required.

Some potential plans for the future are to add the following functionality:

General:

- Provide software in other languages for widgets and documentation. This is underway and some language options are available.
- Improve the REFI-QDA project import and export to make it standards compliant.
- Improve packaging for Linux via Debian files and Windows 10

Reports:

- Possibly look at text mining functionality, word clouds, word visualisations
- Possibly word counts
- Alternative ways to visualise codes

8 About the author

Hello, my name is Colin Curtain and I am a pharmacist and lecturer from Australia. I have many interests including clinical pharmacy, computer programming, research, statistics and clinical decision support. I completed a PhD evaluating computerised clinical decision support in 2014. When doing my PhD I used R as the statistics program of choice. This is where my interest in qualitative data analysis and the use of RQDA came from, which ultimately led to this project.

Originally when doing my PhD I analysed qualitative survey data via a thematic approach using RQDA. I then thought this could be reproduced in Python, so I scripted an earlier version called PyQDA which worked OK at the time.

I thought I would share QualCoder in the hope that it may help others. Bugs are possible and functionality could be further extended. I only work on the programming for this in my spare time.

If and when you use QualCoder and publish your results, I would really appreciate it if you let me know the bibliographic information of your work.