

VCode and VData

Tools for Video
Annotation



Documentation by
Joshua Hailpern and Joey Hagedorn
University of Illinois at Urbana Champaign



VCode and VData

Tools for Video Annotation

VCode and VData are a suite of “open source” applications which create a set of effective interfaces supporting the video annotation workflow . Our system has three main components: VCode (annotation), VCode Admin Window (configuration) and VData (examination of data, coder agreement and training). The Design of VCode and VData was grounded in existing literature, interviews with experienced coders, and ongoing discussions with researchers in multiple disciplines. This document provides a description of the interface of VCode and VData, as well as documentation on set-up, and execution.

We greatly appreciate you using our system, and would love to hear any comments you have! From all of us in the Social Research Group in the Department of Computer Science at the University of Illinois, welcome to VCode and VData!

Joshua Hailpern, Joey Hagedorn, and Karrie Karahalios



Contents

Forward
Contents

Chapter 1: VCode and VData
Features
System Requirements
Installation

Chapter 2: Interfaces
VCode Main Window
VCode Admin Window
VData

Chapter 3: Set-Up
Data Files

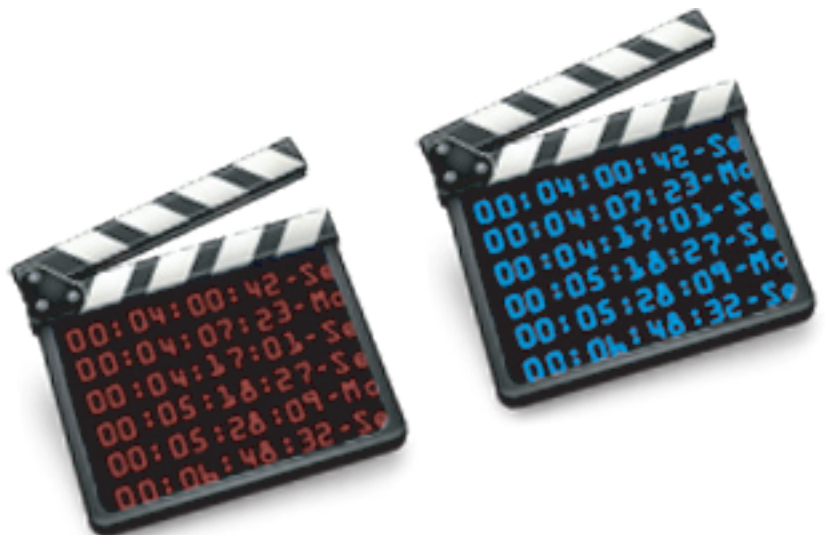
Chapter 4: Using VCode

Chapter 5: Using VData

Chapter 6: Exporting Data
VCode
VData

Chapter 1:

Installation & System Requirements



Features

Multi-Video Stream Support:

Quite often, there are multiple streams of video that coders must annotate. This can be different camera angles, or even screen capture video on a computer. VCode presents one main video at full size, and a dock with other streams playing in real time. When a docked stream is clicked on, it repositions itself into the main video window, while the video which was the previous focus, scales down to the dock, thus equating visual importance with relative size and visual weight.

Different Events for Different Needs:

When annotating a video, we often have to capture different types of data. VCode supports multiple types of event annotation: ranged, momentary and notes/phonetic transcription. A ranged event is one which extends over a period of time (marking action start and duration). Momentary marks have no duration, and thus represent one specific moment in time. Comments can be attached to any mark, allowing additional observations, levels/ranking, or phonetic transcription (through onscreen phonetic keyboard). Any mark with a comment has a inverted outlines to signify that it has a comment attached.

Timeline & Secondary Data

The heart of VCode is the timeline. Events are graphically represented here by diamonds. The spatial-linear presentation allows users to not only see chronologically the location of their events, but also the relative position of one mark to another. Each “track” or “dependent variable” is a different color, allowing for quick and easy assessment of the annotations made to date. To avoid scrolling and maximize the use of screen real-estate, momentary events overlap, thus optimizing screen usage, while providing enough area for track isolation.

In addition to the annotator’s events, additional secondary data can be displayed on the timeline, for example a waveform of the audio from the video. If data (from a computer, or even another coder’s annotations) is logged in a separate log file, this data can be displayed graphically as a bar, line or scatter plot. As a result, video annotators can use the best information available, when making their decisions when to code.

Features

Multiple Playback Modes

Continuous playback is not always the preferred method of analyzing a video. Often multiple modes of playback are utilized; continuous or standard playback, continuous interval playback (play for N seconds, then stop), and skip interval playback (jump N seconds, then stop). This allows the video to be divided into smaller segments for annotation of events that are more difficult to pinpoint (i.e. when a smile starts or ends). Though conceptually simple, manipulations of video using a standard VCR is often described as annoying, and due to hand eye coordination, repeatability & reliability may suffer.

Seperate Admin Window + Templates

To ensure consistent configuration between coders and sessions, all administrative features are consolidated in a single window. The expected workflow is such that a researcher would setup a single coding document with all the variables to be used on all the videos. This template would then be duplicated (with media and log files inserted for each trial). By using this model, large video files only need to be on a hard drive in one location, rather than embedded in the VCode file. Through the Admin Window, the name, color and hot key of each track can be set through this list presentation. Tracks can be enabled as ranged events through a check box in this interface. The Administration Window is also where a researcher specifies videos and data file to be coded, as well as secondary data for contextual annotation. These elements are specified and synchronized through a drag and drop interface, all of which is hidden from the coder to prevent configuration corruption.

Coder Agreement Support

Critical aspects of the video coding workflow (training, reliability, and accuracy) revolve around demonstrating agreement between coders. VData (Figure 3) is a separate executable application specifically targeted to aid researchers in training and agreement analysis of coded data produced in VCode. VData calculates user agreement simply by dragging and dropping in VCode files. Users can set the tolerance variable to accommodate for variability in the mark placement by the coders.

It is not uncommon for multiple tracks or variables to be measuring slight variations on a theme (e.g. smiling vs. large smile vs. grin), thus VData implements a track-merging feature which allows opportunities on two distinct tracks to be treated indistinguishably. For a holistic view, researchers can select tracks to be added into a total agreement calculation. In other words, if analysis determines that a single track is not reliable or it is determined that a given track will not be used in the future, it can be easily excluded from the total agreement calculation.

Features

Coder Conflict Resolution Support

We have optimized coder training and reliability analysis by providing a graphical mechanism to directly compare annotations of two coders. VData can create a VCode session containing specific tracks of two individual coders for side-by-side comparison. The visual, side by side, representation of the data makes it easy to recognize systematic errors in context and detect differences between two coders markings. This reduces the time necessary to locate discrepancies and discuss the reasons why they might have occurred. It is necessary to keep records of these agreement analyses performed with VData by text export. Maintaining export at each stage of the process provides additional transparency and maintains traceability of results that come out of the system.

System Requirements

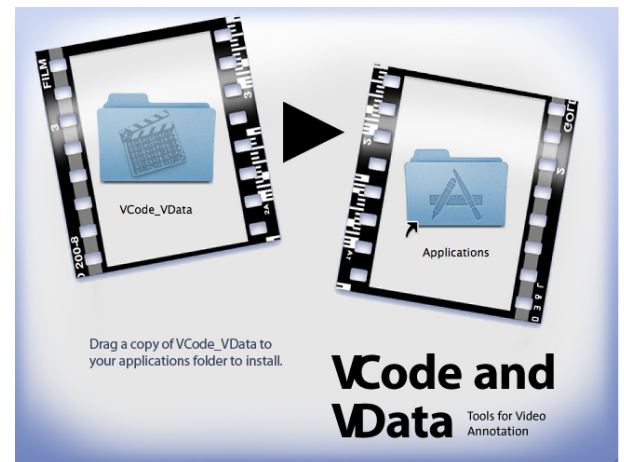
Mac OS 10.5.x (Leopard)
1.8 Ghz or Faster Processor (Intel or PPC)
Video in .mov format
1 Gig of Ram

Installation

- 1 Open the DMG file by double-clicking



- 2 Drag the folder on the left into the applications folder on the right.



- 3 Run VCode and VData by clicking on their respective icons in your applications folder.



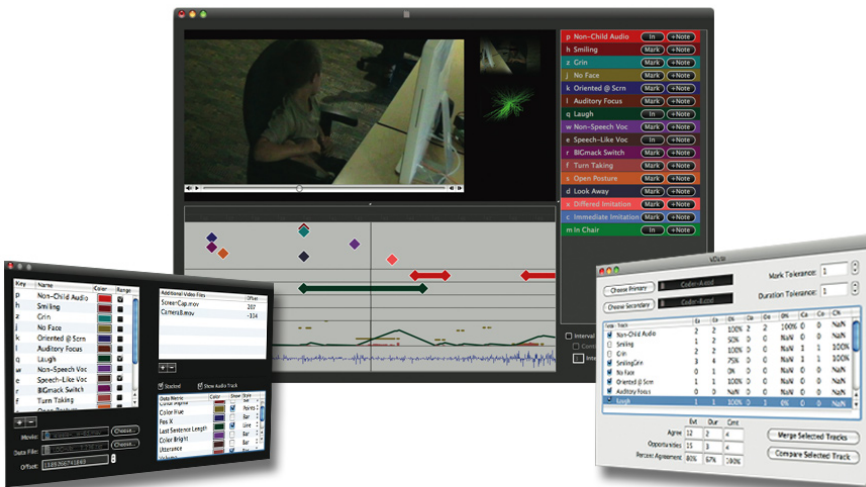
VCode



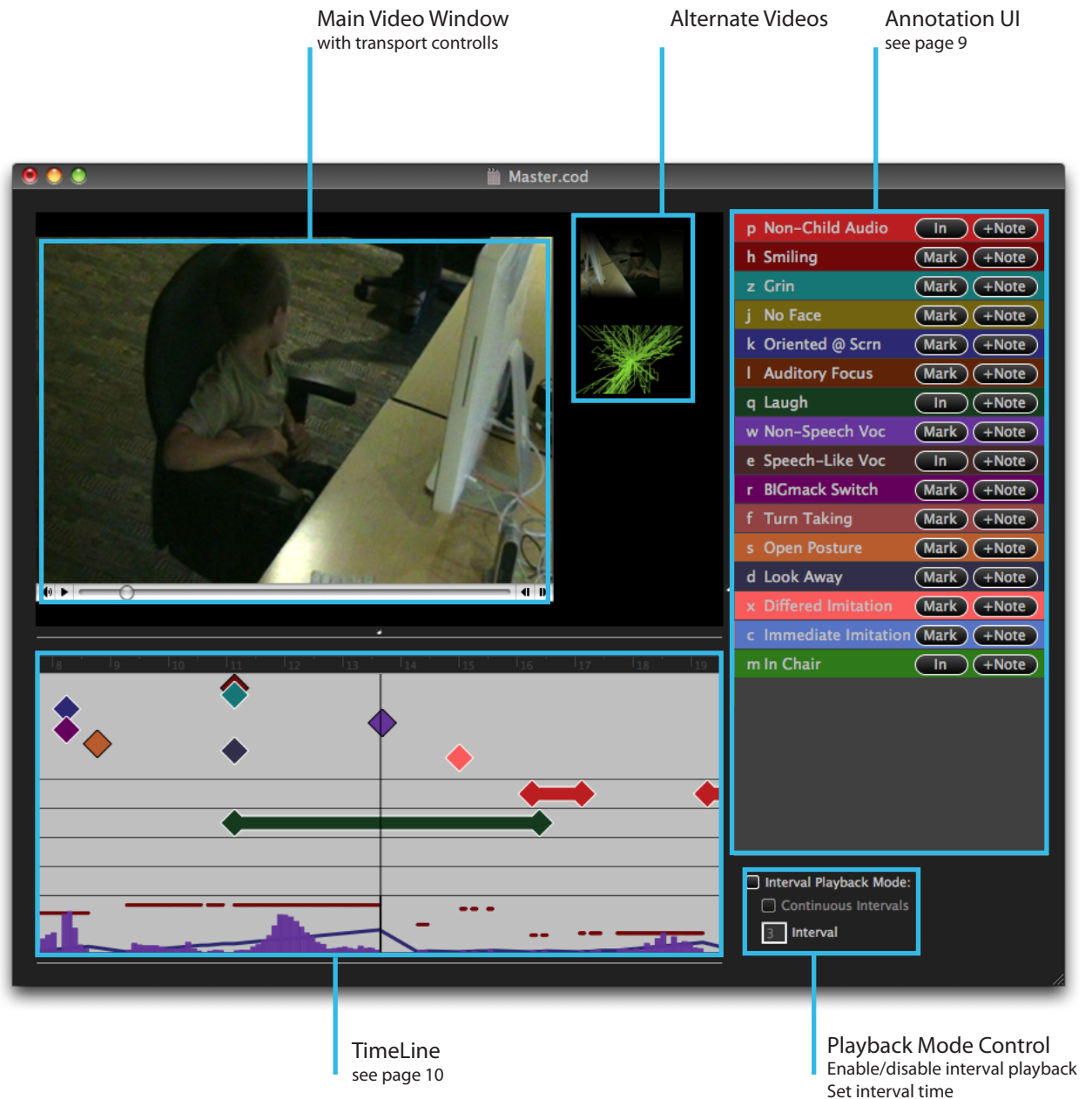
VData

Chapter 2:

Interfaces



VCode



To facilitate multiple video streams VCode presents one main video at full size, and a dock with other streams playing in real time. When a docked stream is clicked on, it repositions itself into the main video window, while the video which was the previous focus, scales down to the dock, thus equating visual importance with relative size and visualweight.

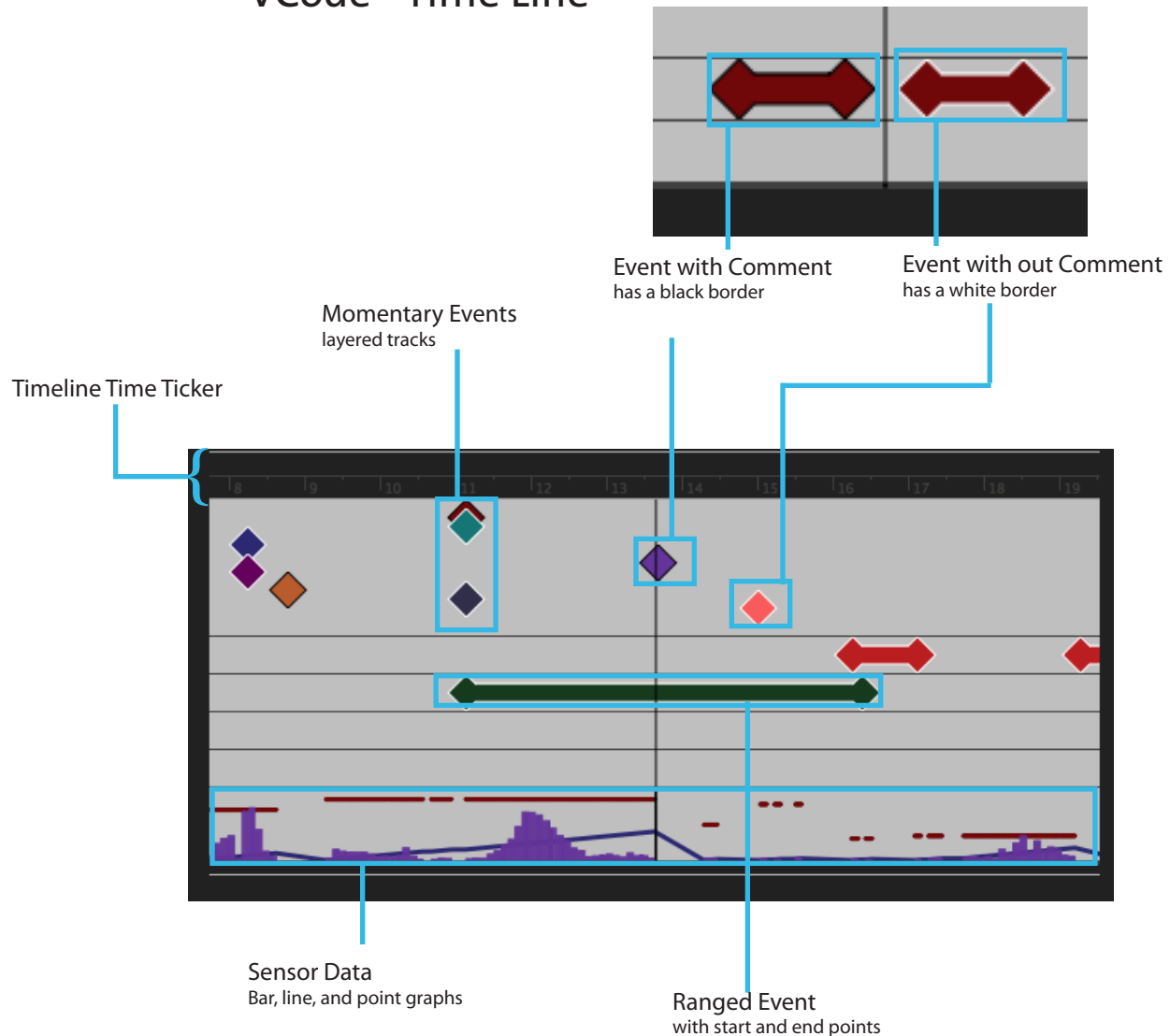
Annotations can be inserted into the timeline via UI buttons or keyboard hot keys. To optimize the typically complex transport controls we isolated the key activities that coders need execute and provided controls limited to play/pause buttons, coarse and fine grained play-head positioning, and step controls.

VCode - Annotation UI



When annotating a video, two different classes of coding events emerge: ranged and momentary. A ranged event is one which extends over a period of time (marking action start and duration). Momentary marks have no duration, and thus represent one specific moment in time.

VCode - Time Line



The heart of VCode is the timeline. Events are graphically represented here by diamonds. The spatial-linear presentation allows users to not only see chronologically the location of their events, but also the relative position of one mark to another. Each “track” or “dependent variable” is a different color, allowing for quick and easy assessment of the annotations made to date. To avoid scrolling and maximize the use of screen real-estate, momentary events overlap, thus optimizing screen usage, while providing enough area for track isolation.

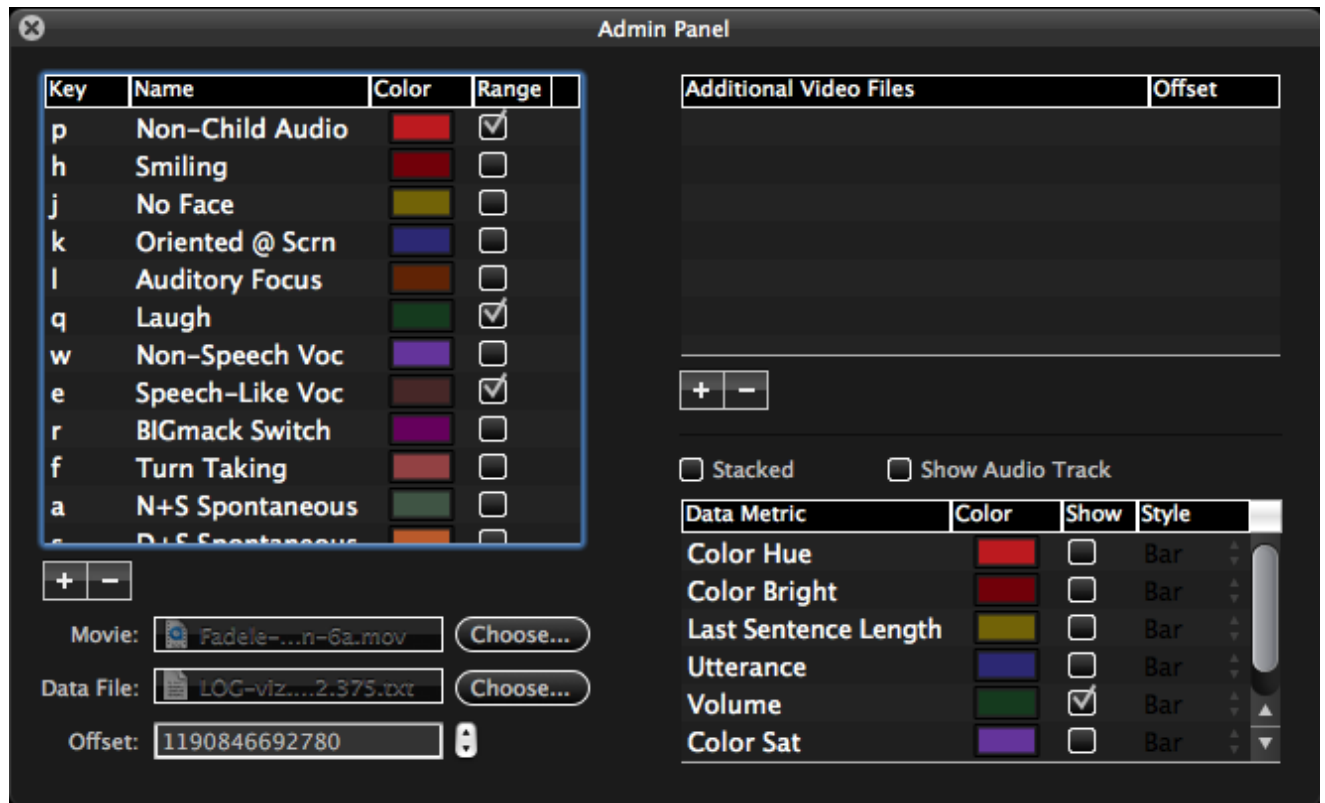
In addition to the annotator’s events, additional secondary data can be displayed on the timeline, for example a waveform of the audio from the video. If data (from a computer, or even another coder’s annotations) is logged in a separate log file, this data can be displayed graphically as a bar, line or scatter plot. As a result, video annotators can use the best information available, when making their decisions when to code.

VCode - Comments



Comments can be attached to any mark, allowing additional observations, levels/ranking, or phonetic transcription (through onscreen phonetic keyboard). Any mark with a comment has a inverted outlines to signify that it has a comment attached.

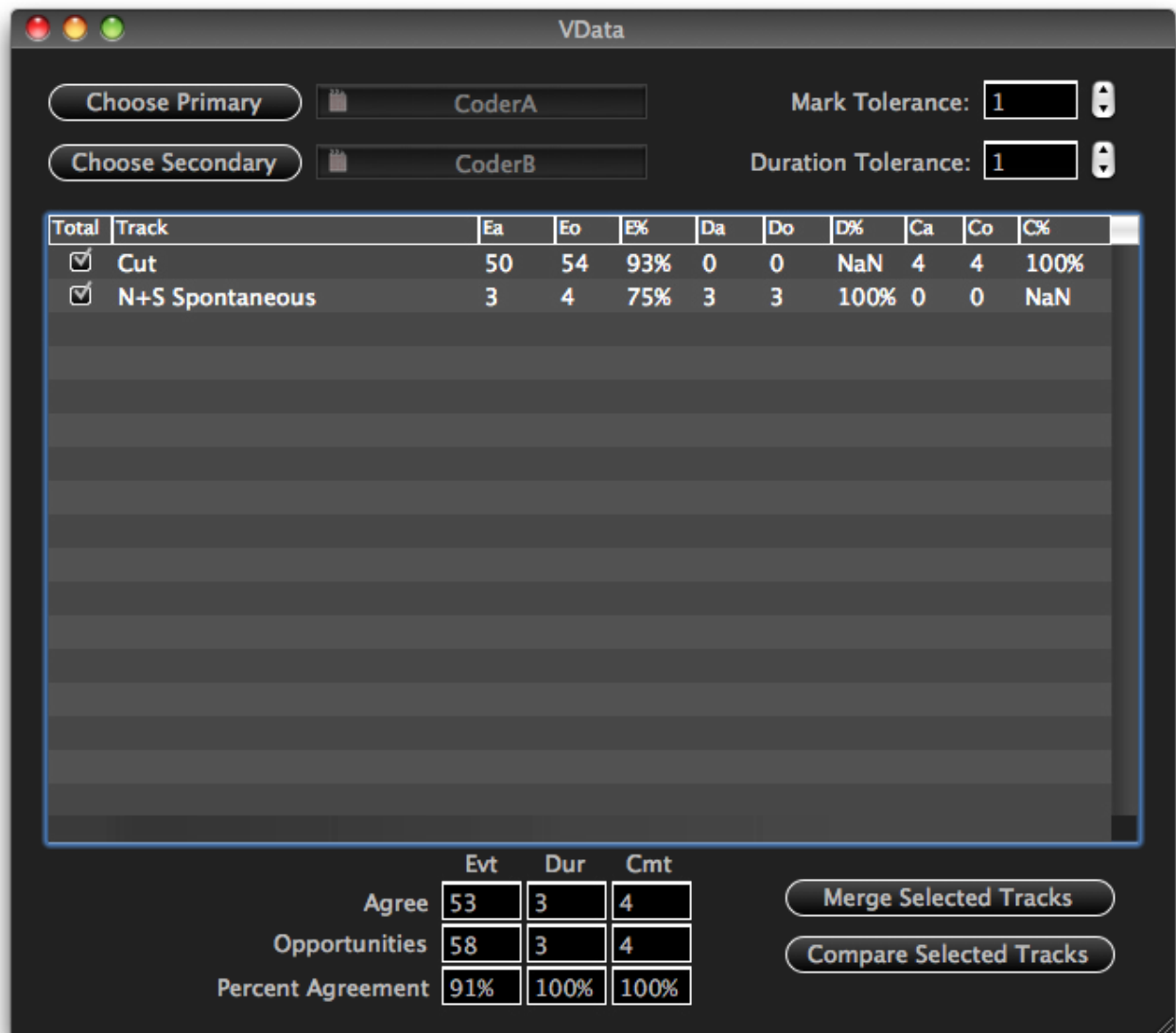
VCode Admin Window



To ensure consistent configuration between coders and sessions, all administrative features are consolidated in a single window. The expected workflow is such that a researcher would setup a single coding document with all the variables to be used on all the videos. This template would then be

VCode Admin Window

VData



Chapter 3:

Set-Up



Data Files

Free Line for any notes

Track Names Comma Separated
(first number must be a time stamp)

Track Data Comma Separated
(first number must be a time stamp)

1188659823961: Log Created - viz.Viz_Explode Play Echo(false)
Time,Number of Sounds Played,Last Sentence Length,Pos Y,Pos Z

1188659825430,0,1,-32,-59,272,100,100,100,0.47808772,1
1188659825479,0,2,-32,-59,272,100,100,100,0.47808772,1
1188659825500,0,3,-32,-59,272,100,100,100,0.49171424,1
1188659825518,0,4,-32,-59,272,100,100,100,0.49171424,1
1188659825534,0,5,-32,-59,272,100,100,100,0.49171424,1
1188659825548,0,6,-32,-59,272,100,100,100,0.49171424,1
1188659825563,0,7,-32,-59,272,100,100,100,0.49171424,1
1188659825579,0,8,-32,-59,272,100,100,100,0.37980285,1
1188659825593,0,9,-32,-59,272,100,100,100,0.37980285,1
1188659825607,0,10,-32,-59,272,100,100,100,0.37980285,1
1188659825624,0,11,-32,-59,272,100,100,100,0.37980285,1
1188659825643,0,12,-32,-59,272,100,100,100,0.37980285,1
1188659825657,0,13,-32,-59,272,100,100,100,0.37980285,1
1188659825679,0,14,-32,-59,272,100,100,100,0.28072453,1
1188659825694,0,15,-32,-59,272,100,100,100,0.28072453,1
1188659825708,0,16,-32,-59,272,100,100,100,0.28072453,1
1188659825721,0,17,-32,-59,272,100,100,100,0.28072453,1
1188659825739,0,18,-32,-59,272,100,100,100,0.28072453,1

Chapter 4:

Using VCode



Coming Soon...

Chapter 5:

Using VData



Coming Soon