EMViz

Early Music Visualization

Aaron Carter-Ényì, PhD
Department of Music, Morehouse College
carterenyi@gmail.com

Supported by:

ACLS

NATIONAL ENDOWMENT FOR THE HUMANITIES

About

Currently implemented in MATLAB and deployed as a standalone executable for Windows (using MATLAB Runtime). The project was funded in part by the American Council of Learned Societies (ACLS) and the National Endowment for the Humanities (NEH).

EMViz (Early Music Visualization) provides built-in pattern recognition for symbolic music data (MIDI) based on a contour recursion algorithm by Carter-Enyi (2016) producing visualizations of musical form using arc diagrams, as proposed by Wattenberg (2002). The algorithm brings together contour theory (Morris 1987, Quinn 1996, Schultz 2013) with studies of melodic accent (Thomassen 1982, Huron 2006). Symbolic music data (.midi, .xml) from various sources (including ELVIS at McGill and the Yale Classical Archives Corpus) may be imported, analyzed and visualized in a matter of minutes. Arc diagram visualizations in the supplemental materials include music from the *Liber Usualis*, Josquin des Prez and J. S. Bach.

For more information on the contour algorithm:

- Carter-Ényì, A. (2016). Contour Recursion and Auto-Segmentation. *Music Theory Online*, *22*(1).

For more information on the arc diagram visualization:

- Wattenberg, M. (2002). *Arc diagrams: Visualizing structure in strings.* Paper presented at the Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on.

Installation

The Windows application is available on the GitHub repository:
https://github.com/carterenyi/emviz

Note: While there is not a standalone application available for Mac at this time, Mac users with MATLAB installed and licensed may download the folder “EMVizMATLAB”, add it to your MATLAB path and run the EMVizGUI.m file.

Detailed instructions for installation on Windows:

1. You will need a web connection to complete installation because MATLAB Runtime (also free) will also be downloaded and installed when you run the application installer
2. At the link above, download the “EMVizWindows” folder or “EMVizWindows.zip” (and unzip)
3. Find the “AppInstaller” folder and double-click “MyApplicationInstall_web.exe”.
4. Because this software is not from an “App Store”, you will likely need to override some security preferences after expanding/unzipping and clicking on the.exe, to do this right-click or control-click and select “Run as administrator”
5. The installation process (which requires an internet connection) may take 5 to 20 minutes depending on the download speed of your internet connection (it is downloading MATLAB Runtime so your computer can interpret the source code)

Instructions

Note: Instructions for testing basic functionality are on the next page.

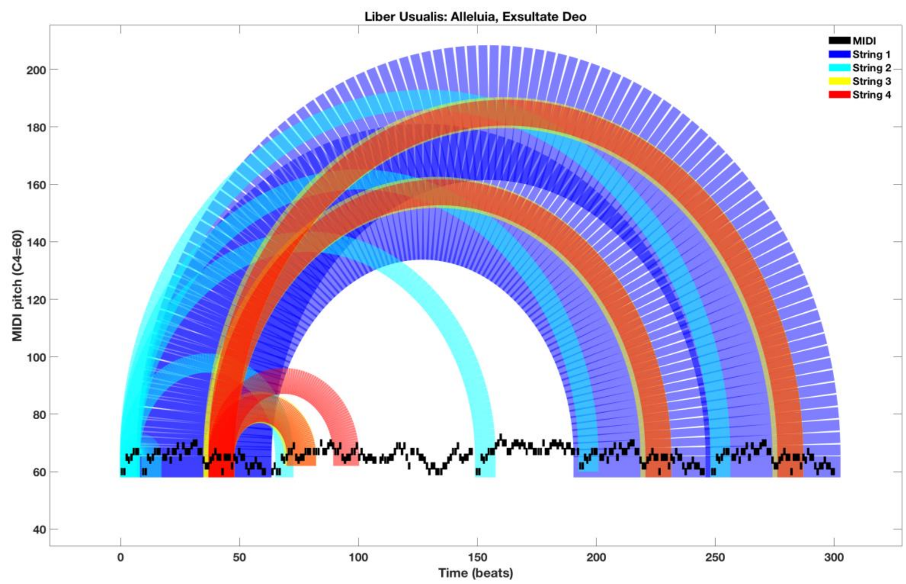
1. **Select MIDI File (Button)**: opens a finder window to select a MIDI file, you can search for ".mid" or for file type "MIDI Audio", or navigate to sample MIDI provided with the application
2. **Current MIDI File (Display Text)**: will display the filename for the MIDI file you have selected
3. **Minimum Cardinality (Text Entry Box)**: allows you to select the shortest string length you are interested in. Generally, five notes is the default minimum because three or four-note contours are too generic. If you know you are interested in longer strings (on the order of 10 or 20 events), then set a higher minimum for results closer to your analytical goals.
4. **Use Pitch; Use Durations; Use Both (Selection Box)**: allows you to select what type of pattern-recognition algorithm to run: (1) *Pitch* only makes binary contour comparisons of pitch height within a local window of two degrees of adjacency; (2) *Durations* only makes binary comparisons of note duration within a local window of two degrees of adjacency; (3) *Both* combines pitch contour and durational contour to find patterns.
5. **Run Analysis and Plot** (Button): triggers the algorithm you selected (Pitch, Durations, or Both) to run on the file listed as the **Current MIDI File**. This takes anywhere from a few seconds to about a minute depending on the size of the file (if the application does not return a diagram after a minute, it is not functioning).
6. **Enter Title of Piece (Text Entry Box)**: allows you to name your diagram (this can be entered before **Run Analysis and Plot** or **Replot**)
7. **Replot** (Button): enables you to selected strings in the listbox in the bottom right corner of the GUI (see 9 and 11)
8. **Export Data as CSV** (Button): exports all strings found in a table format that can be opened in Excel or any text editor.
9. **Strings Found** (Selection List Box): select from all strings found for either 10 or 11
10. **Plot String** (Button): plots and plays the string selected in 9.
11. **Add String To List (Button) / Enter name for String (Text Entry Box)**: adds the string selected in 9 to a list of user-specified strings (for 7. Replot). If no name is entered, then it will be labeled String*X*.

Testing

Before importing MIDI files of your own or those found through the Internet, it is recommended that you test basic functionality using one of the provided MIDI files, specifically:

LiberUsualis_Alleluia_Exsultate.mid

1. Click "Select the MIDI file"
2. Use default settings (i.e. selection box on "Use Pitch" and minimum cardinality at "5")
3. Click "Run Analysis and Plot"
4. Wait for analysis (this file is small so run time should be 5 to 10 seconds)
5. When the diagram appears, compare it to the image below.
6. Click "Export Data as CSV", navigate to the folder with the MIDI file, open the CSV file with the same filename (ideally, with Microsoft Excel) and compare to the sample CSV output below.



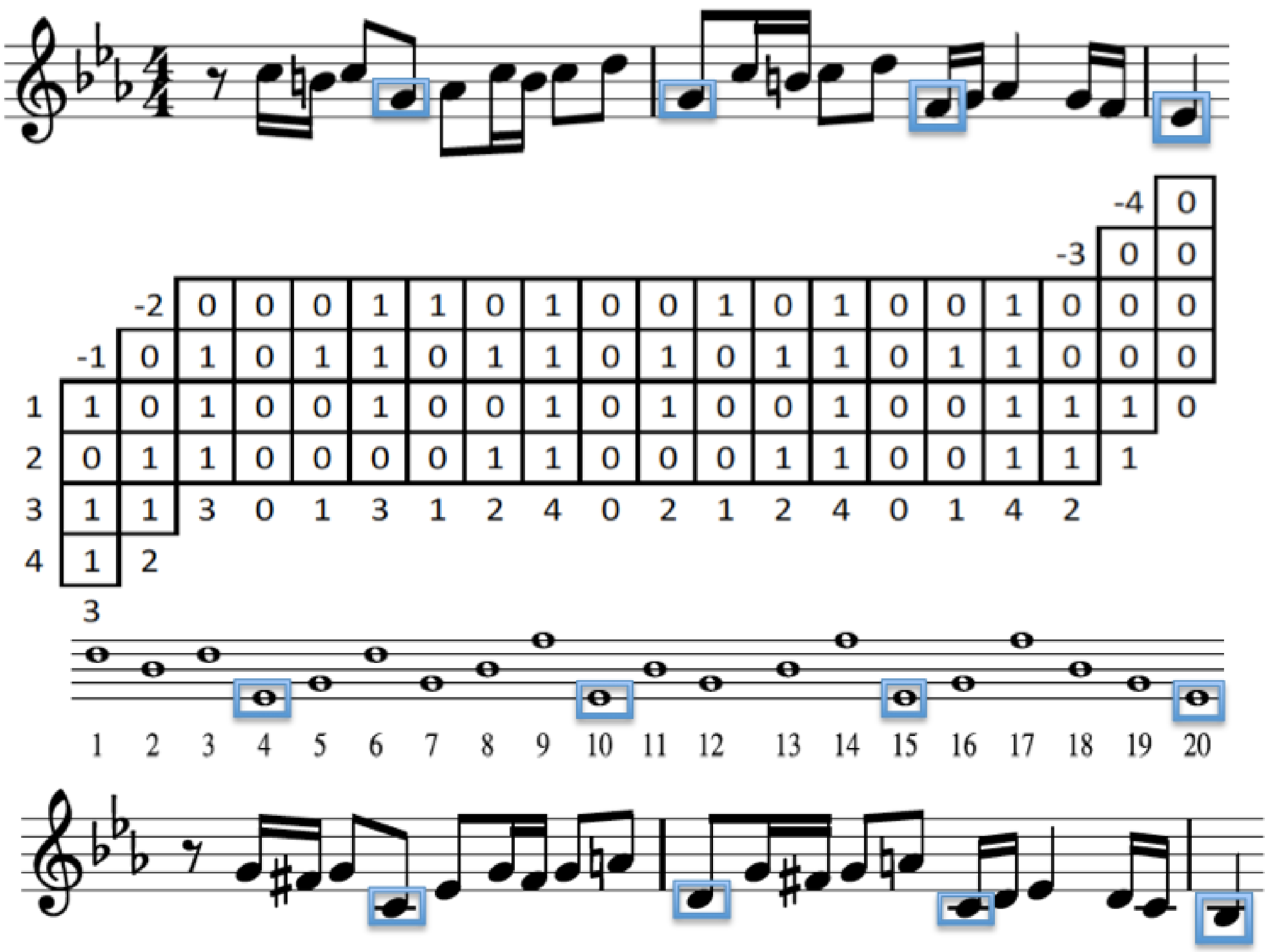
CSV Output:

String	Cardinality	Channel	Onset	Pitch
1	39	1	9	60
1	39	1	192.8	62
1	39	1	248.3	60
2	7	1	0	60
2	7	1	9	60
2	7	1	65.3	60
2	7	1	149.3	60
2	7	1	192.8	62
2	7	1	248.3	60
3	10	1	35.3	62
3	10	1	69.3	65
3	10	1	219.1	62
3	10	1	274.6	62
4	9	1	37.3	64
4	9	1	70.3	67
4	9	1	89.8	67

4	9	1	221.1	64
4	9	1	276.6	64

Theoretical Basis

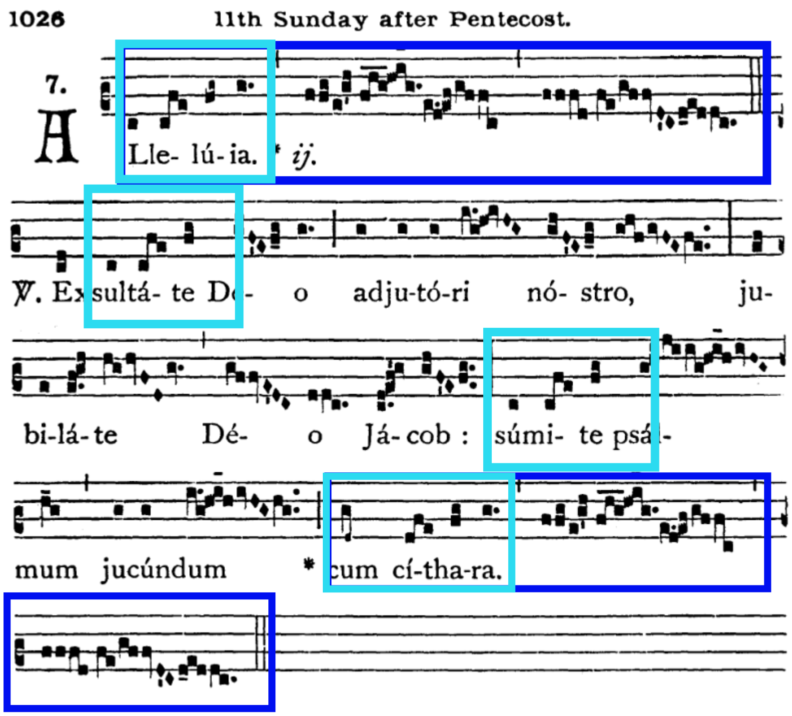
The contour algorithm behind EMViz uses local contour comparisons within a 2-degree radius of each focus pitch. This makes use of Quinn's (1997) binary C+ comparison, where 0 is equal or below and 1 is above the reference pitch. In the examples below, the subject and tonal answer of Bach's C-minor Fugue (BWV 847) are shown to be equivalent strings using this method (they both produce the same "continuous" matrix).



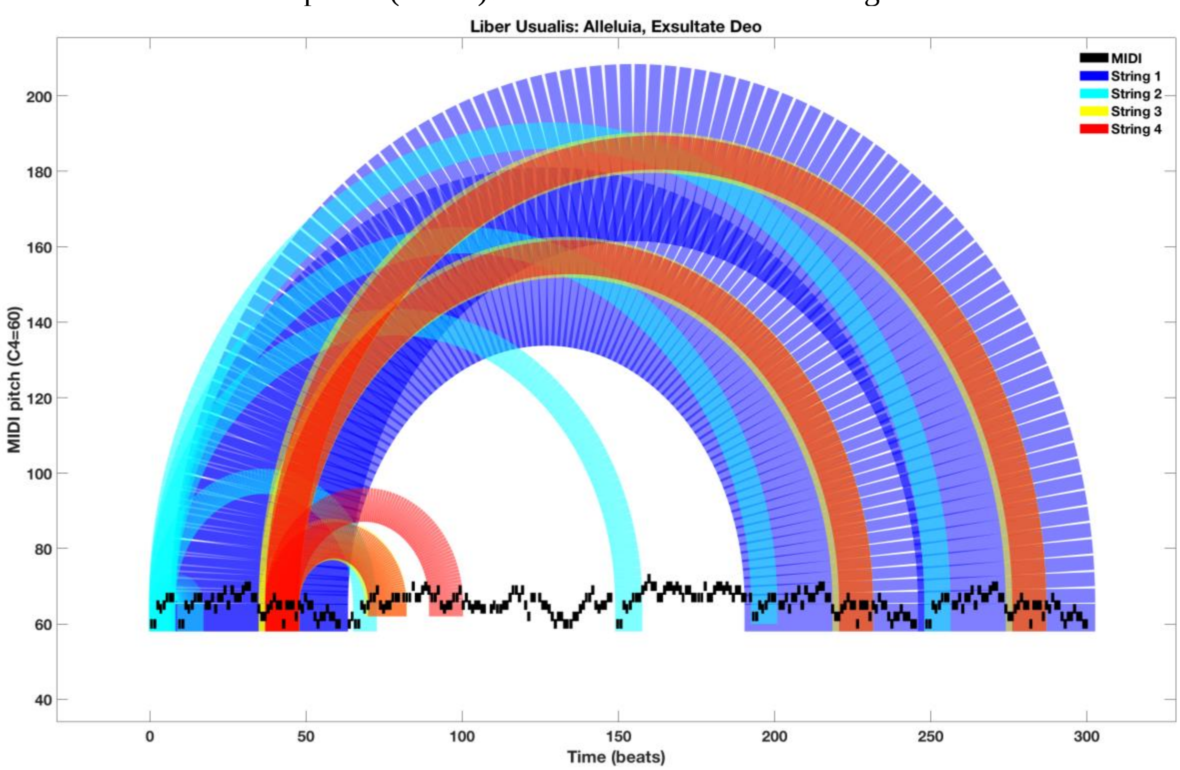
Examples

In arc diagrams, notes appear as small rectangles at the bottom of the visualization, if multiple voices are present, they are color-coded. Iterations of repeated contour patterns are connected with arcs.

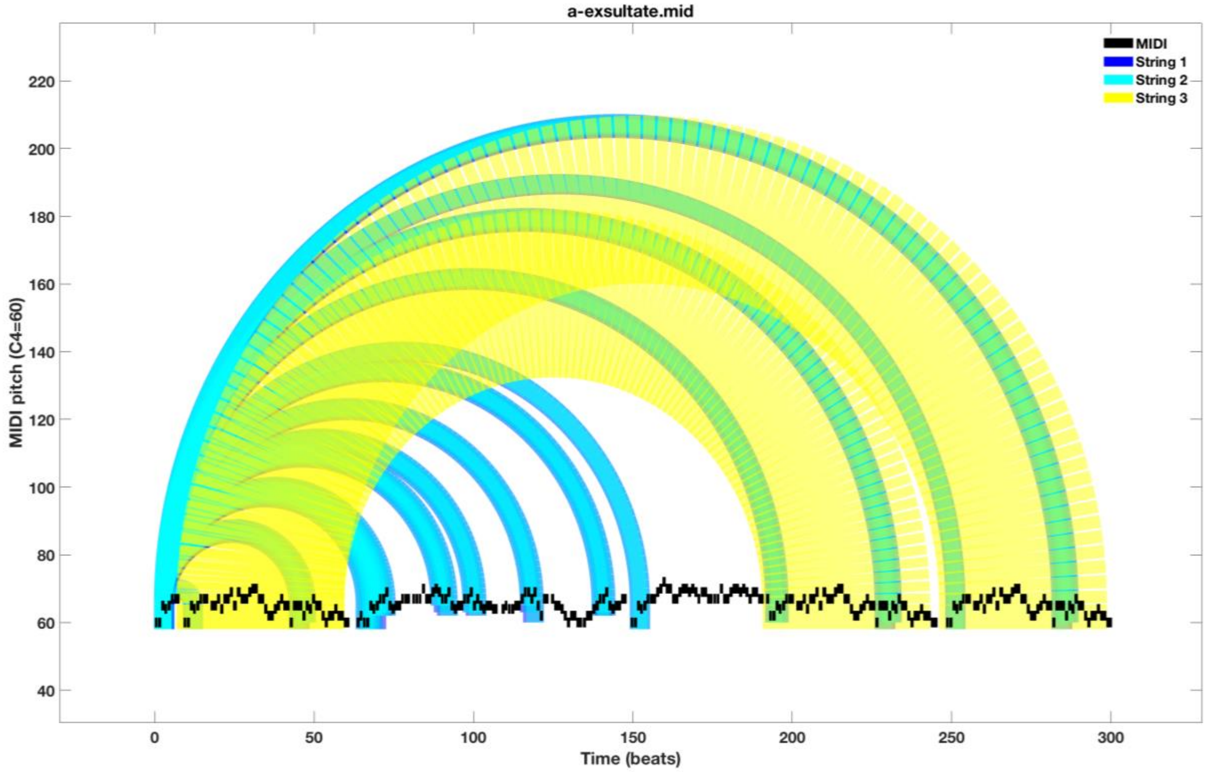
Example 1: *Liber Usualis: Alleluia – Exultate Deo*: Visualization of full performance (e.g. https://youtu.be/oa1XLOW55RE) with incipit at the beginning and repetition of "Alleluia" at the end.



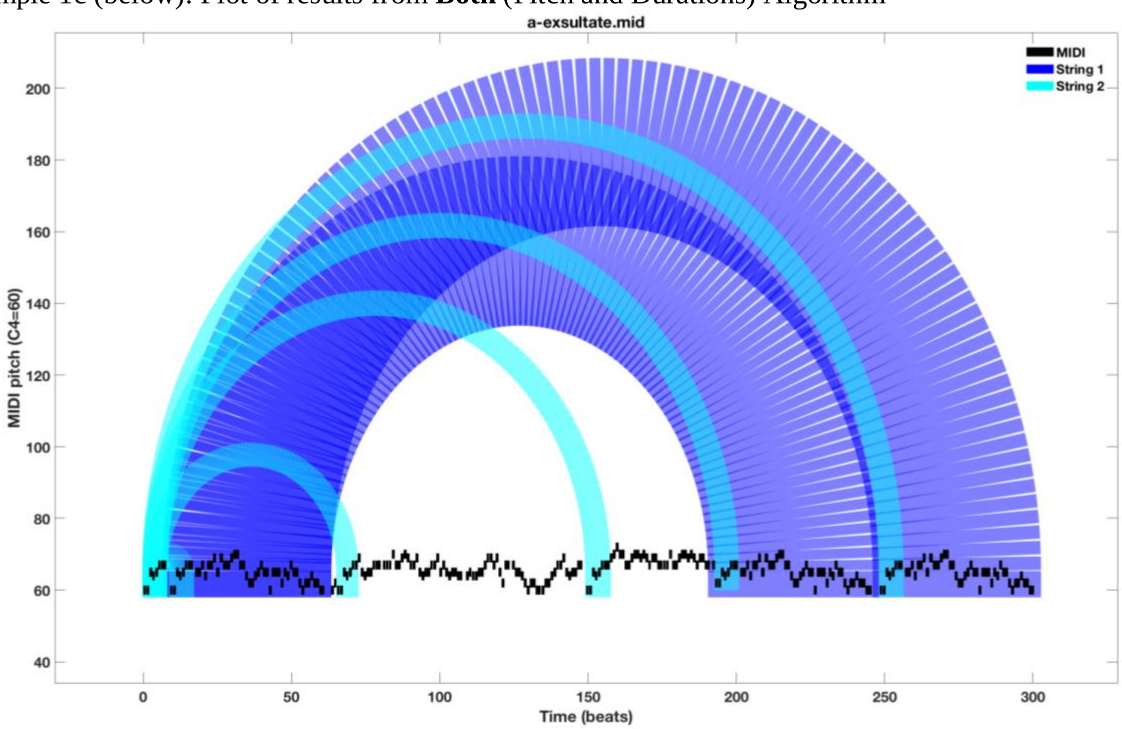
Example 1a (below): Plot of results from **Pitch** algorithm



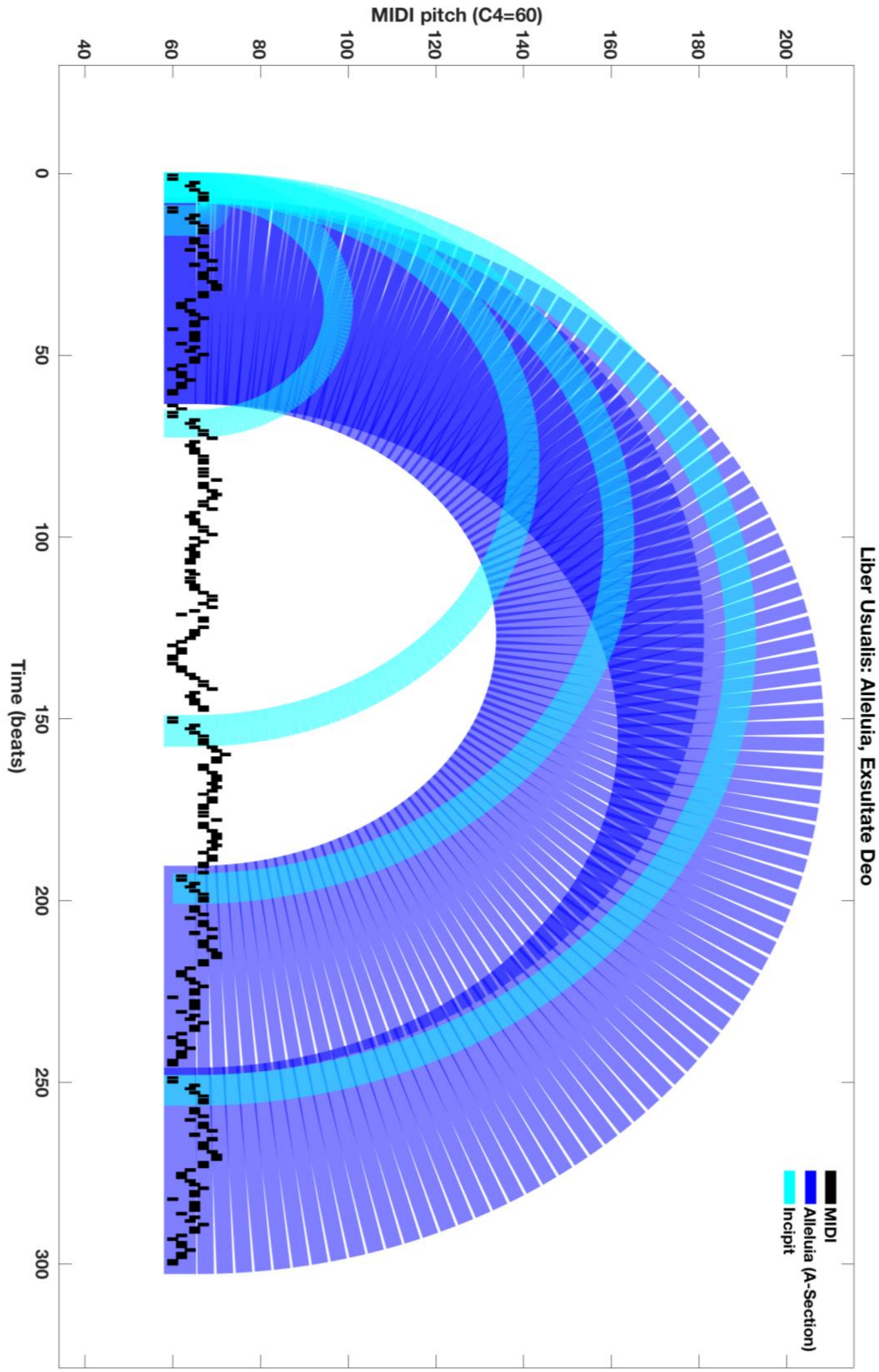
Example 1b (below): Plot of results from **Durations** algorithm



Example 1c (below): Plot of results from **Both** (Pitch and Durations) Algorithm



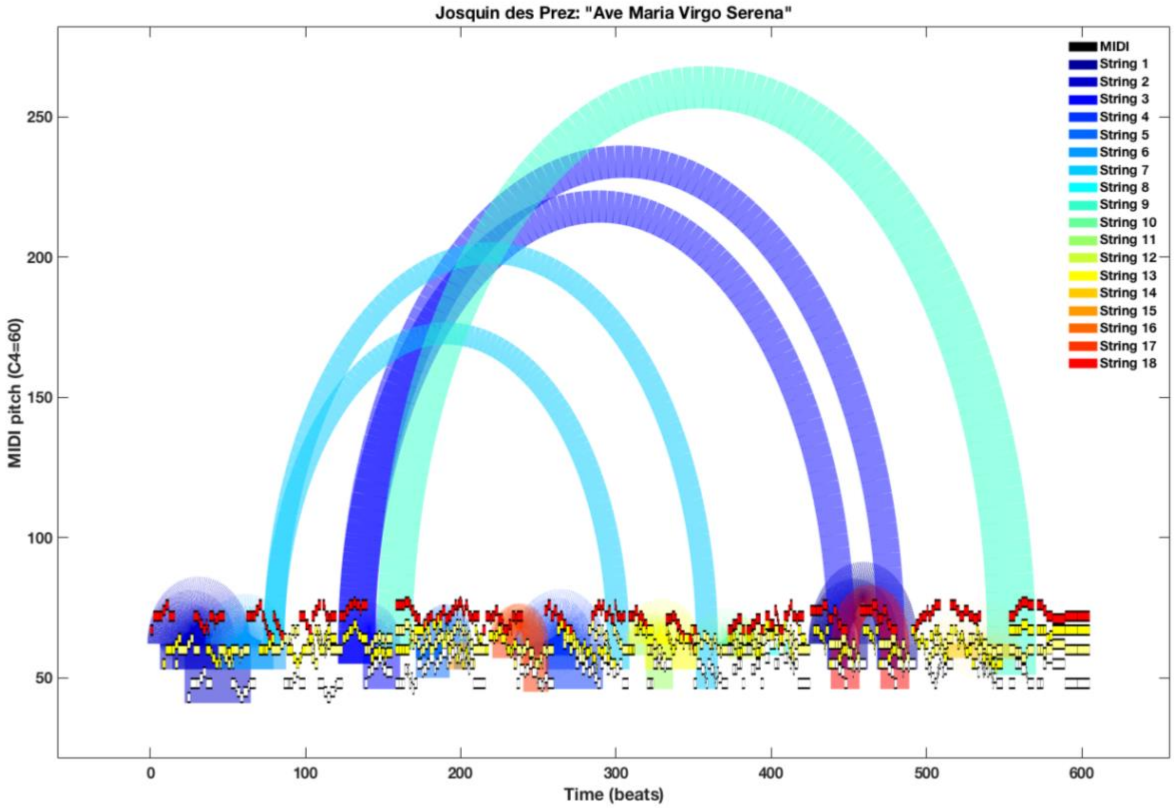
Example 1d (below): replot of selected strings (from **Pitch** algorithm) with labels added for legend (top right)



Examples 1e: Exported CSV for **Pitch** algorithm analysis of “Alleluia, Exsultate Deo” (EXAMPLE 1):

String	Cardinality	Channel	Onset	Pitch
1	39	1	9	60
1	39	1	192.8	62
1	39	1	248.3	60
2	7	1	0	60
2	7	1	9	60
2	7	1	65.3	60
2	7	1	149.3	60
2	7	1	192.8	62
2	7	1	248.3	60
3	10	1	35.3	62
3	10	1	69.3	65
3	10	1	219.1	62
3	10	1	274.6	62
4	9	1	37.3	64
4	9	1	70.3	67
4	9	1	89.8	67
4	9	1	221.1	64
4	9	1	276.6	64

Example 2: Josquin des Prez "Ave Maria Virgo Serena" with all strings found shown



Example 3: Johann Sebastian Bach "Fugue in c minor" (BWV 847) with three strings selected (String 1: Subject; String 2: Countersubject 1; String 3: Countersubject 2)