Voice Allocation with poly~

and the *bach* Library for Computer-Assisted Composition (CAC)

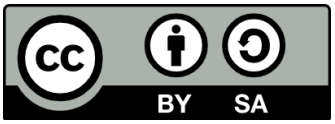
A MaxMSP Patching Odyssey

ENT3320 — *Interactive VR, Music, Sound, and Memory*

New York College of Technology, CUNY City Tech
Spring 2020

Professor Louis Goldford
LGoldford@citytech.cuny.edu

Computer Music Center, Columbia University



CONTENTS

Voice Allocation with poly~ **1**

1. Proportional Notation GUI **3**
1.1 Introducing bach.roll 3
1.2 Composing with bach.roll 4
1.3 Simple playback with bach.ezmidiplay 5

2. Synthesis with the bach Library **6**
2.1 Monophonic Synthesizer 6
2.1.1 Monophonic Synth Subpatcher 7
2.2 Polyphonic Synth with poly~ 8
2.2.2 poly~ Subpatcher 9
2.2.3 Monitoring the Mute-/Busy-States 10
2.2.4 poly~ Subpatch Folder Structure 11
2.3 Adding Control Over Synth Parameters 12
2.3.1 poly~ Subpatch With Ring Modulation 13

3. A Boiler For poly~ Instruments **14**
3.1 poly~ Subpatch Scaffolding 15
3.2 Voice Trigger Simulation 16
3.3 Data Stream Simulation 17
3.4 Example Data Inside Coll 18

4. What Is A Note? — Part 1 **19**

5. What Is A Chord? **20**

6. MIDI vs. Midicents (Microtonality) **21**

7. Playing With Beat Frequencies **22**

8. What Is a Note — Part 2 **23**
8.1 bach “Slot Machines” Tutorial — Part A 24

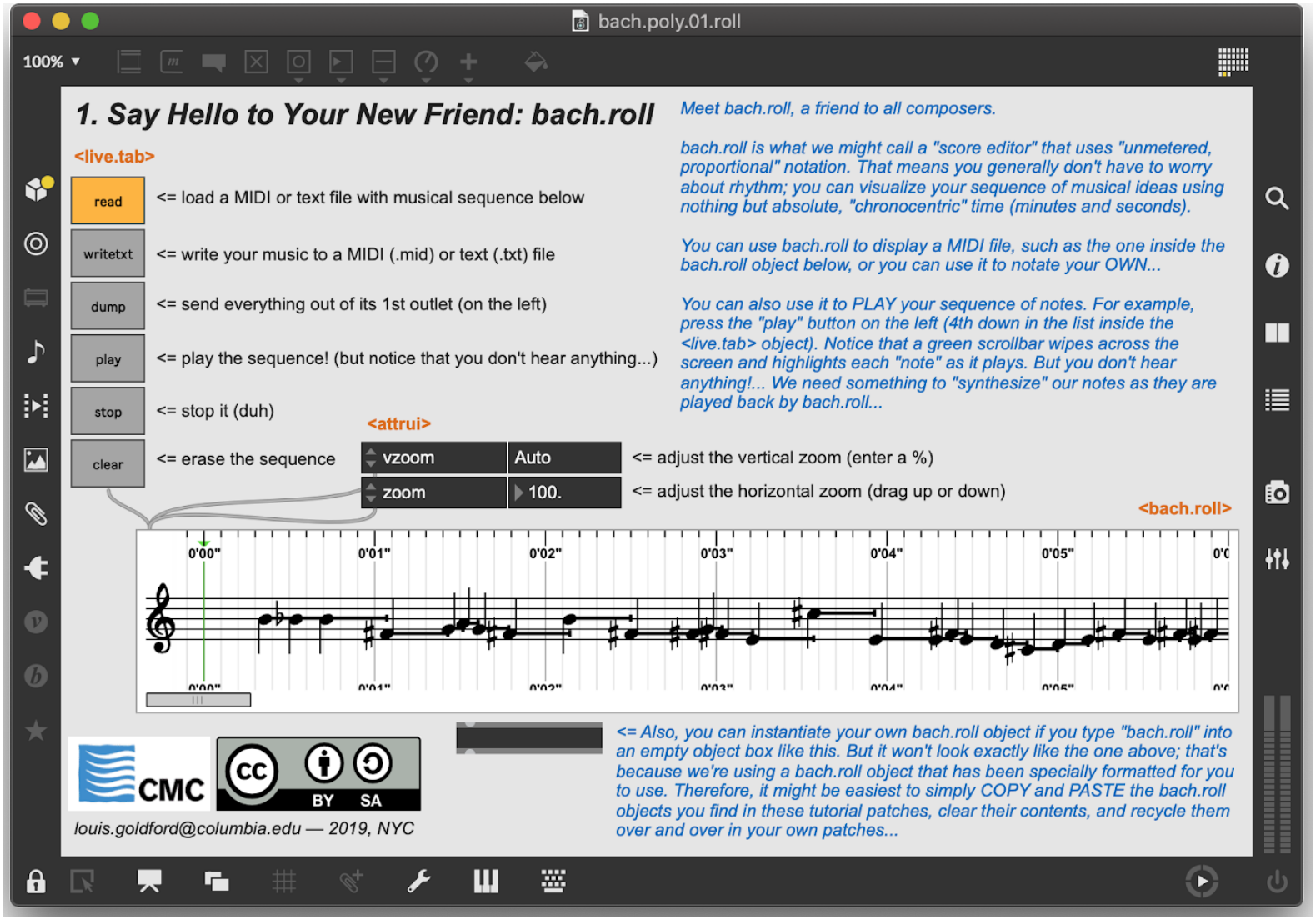
9. Controlling poly~ With bach Slots **25**
9.1 poly~ Subpatch 26

10. Sketch Vs. Score **27**

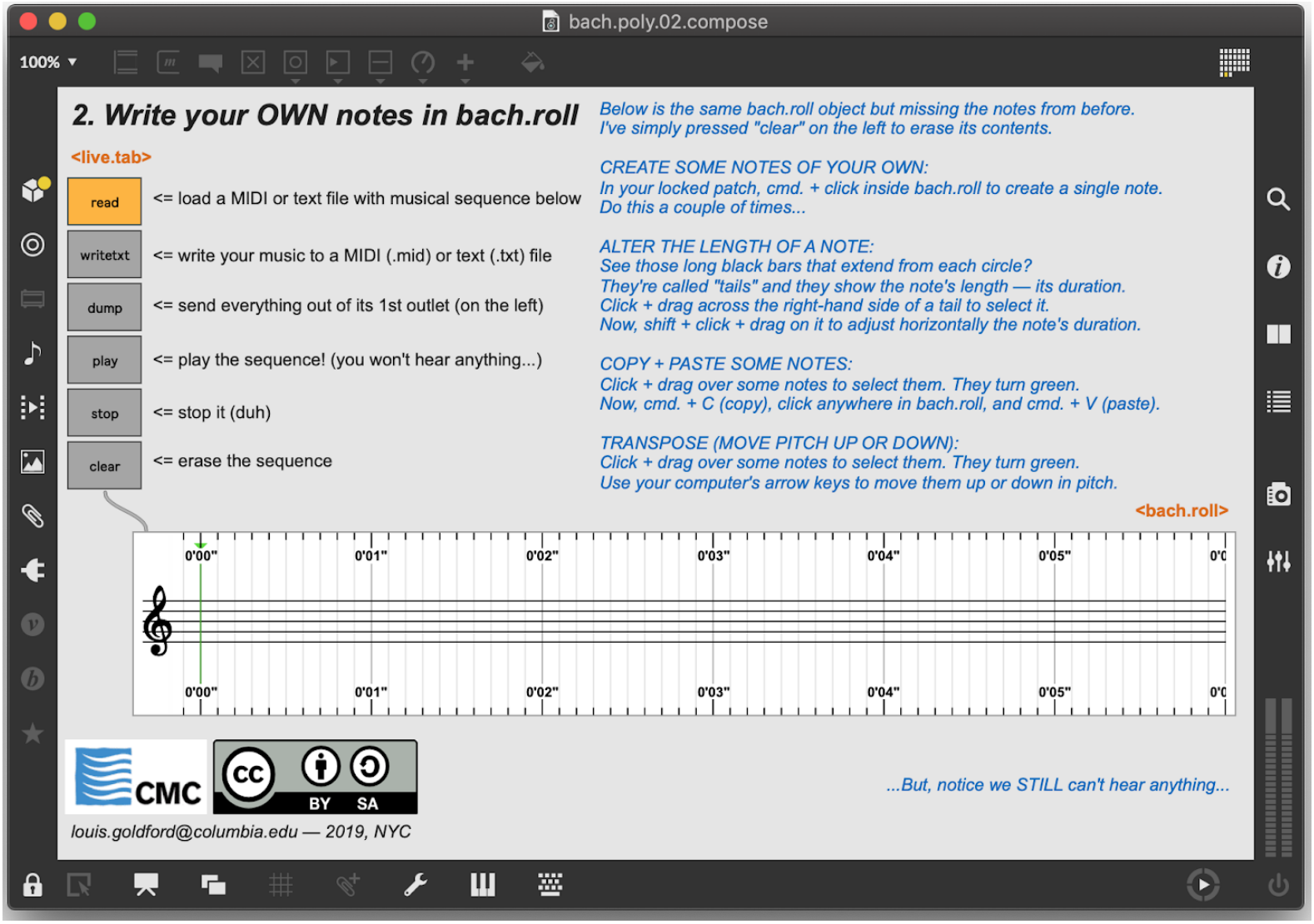
11. How to Record Audio from Max **28**

1. Proportional Notation GUI

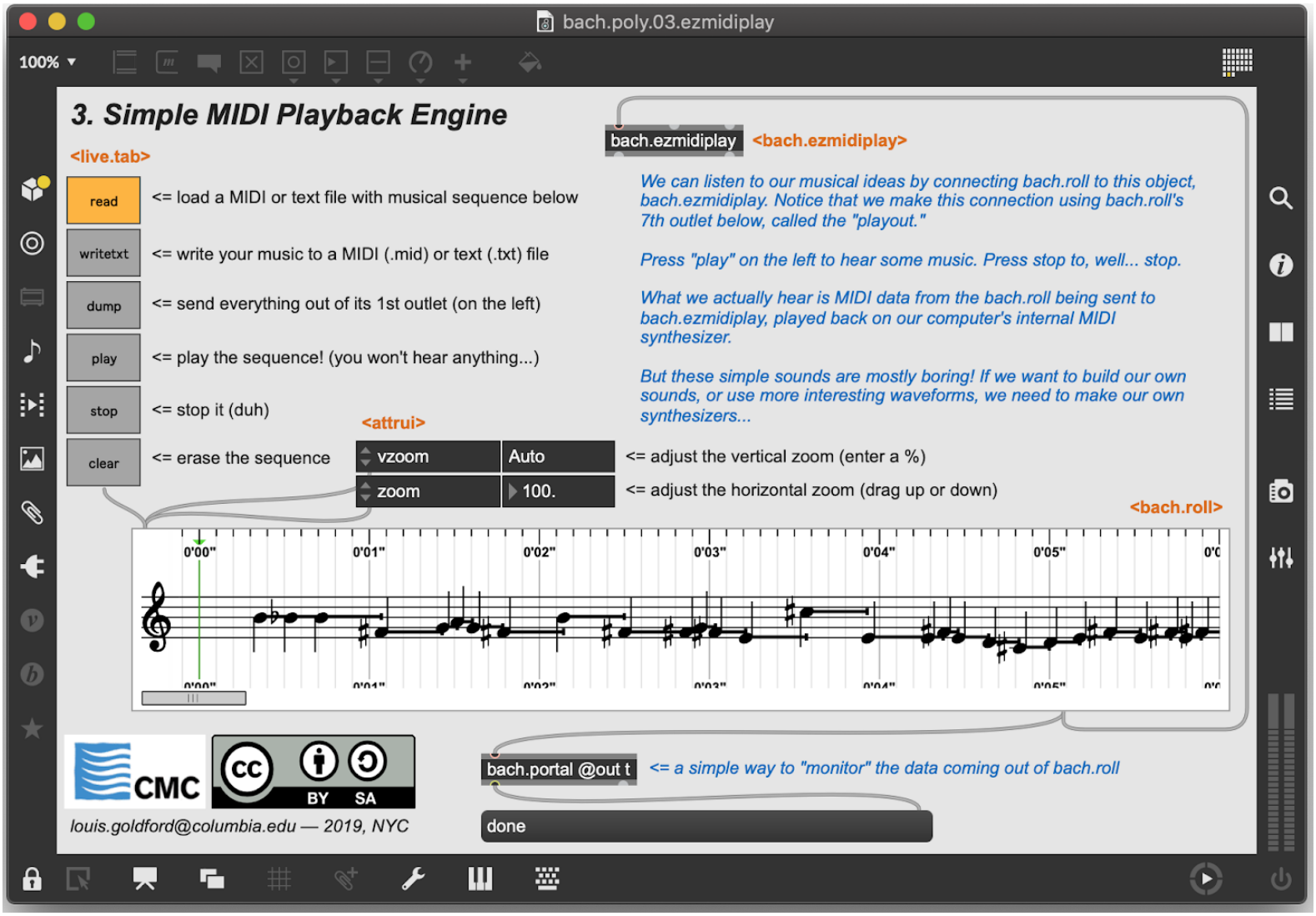
1.1 Introducing bach.roll



1.2 Composing with bach.roll

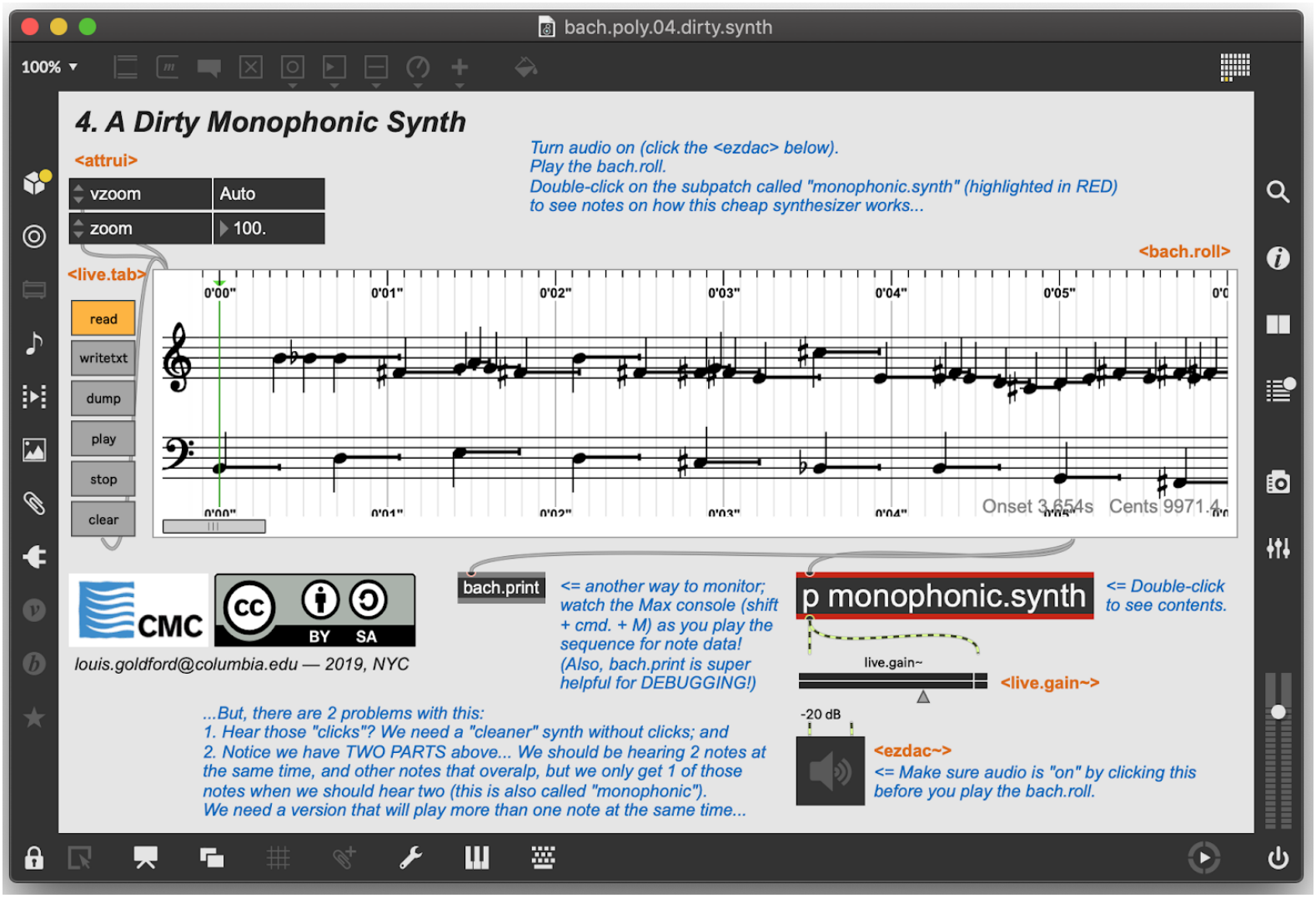


1.3 Simple playback with bach.ezmidiplay

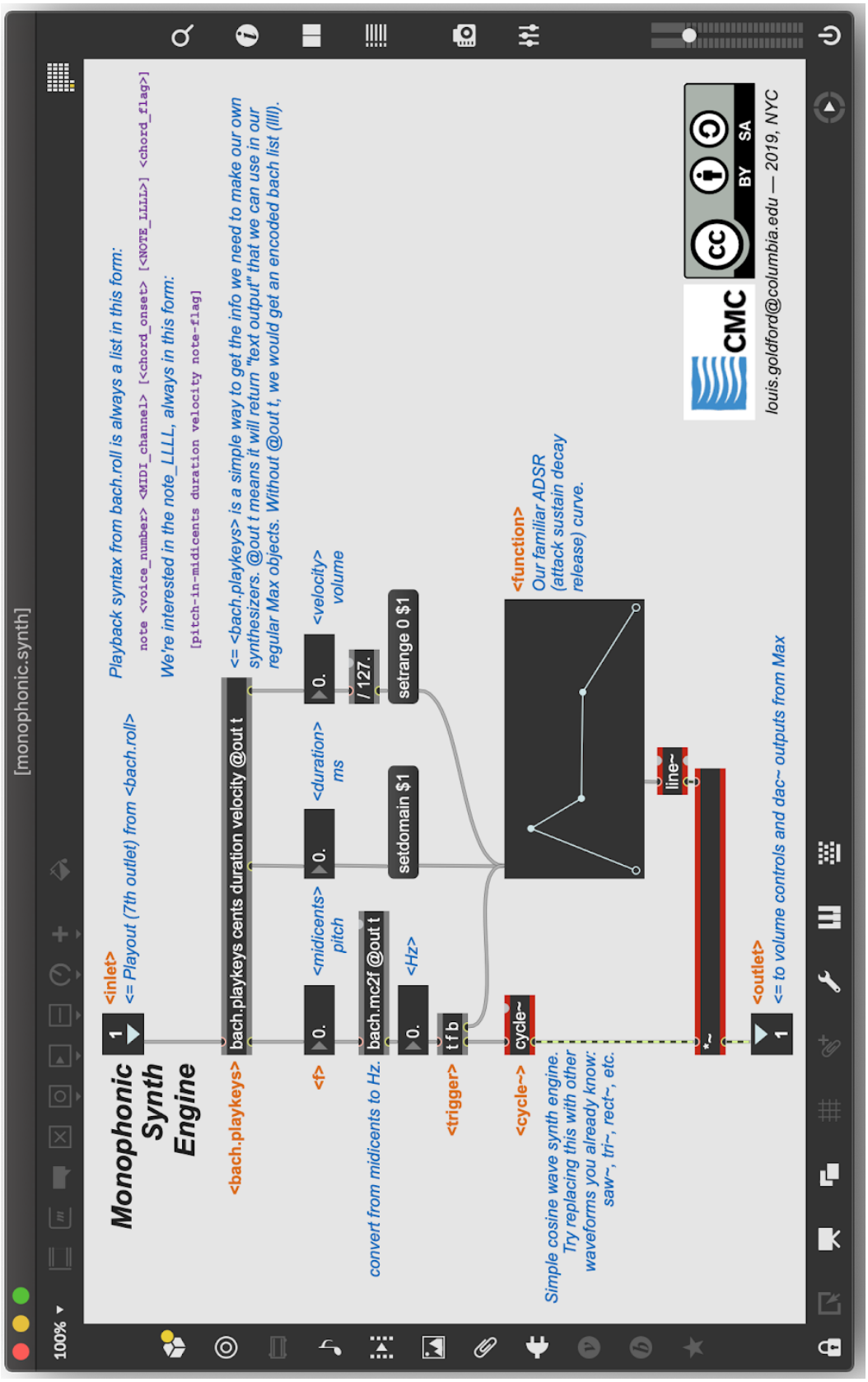


2. Synthesis with the *bach* Library

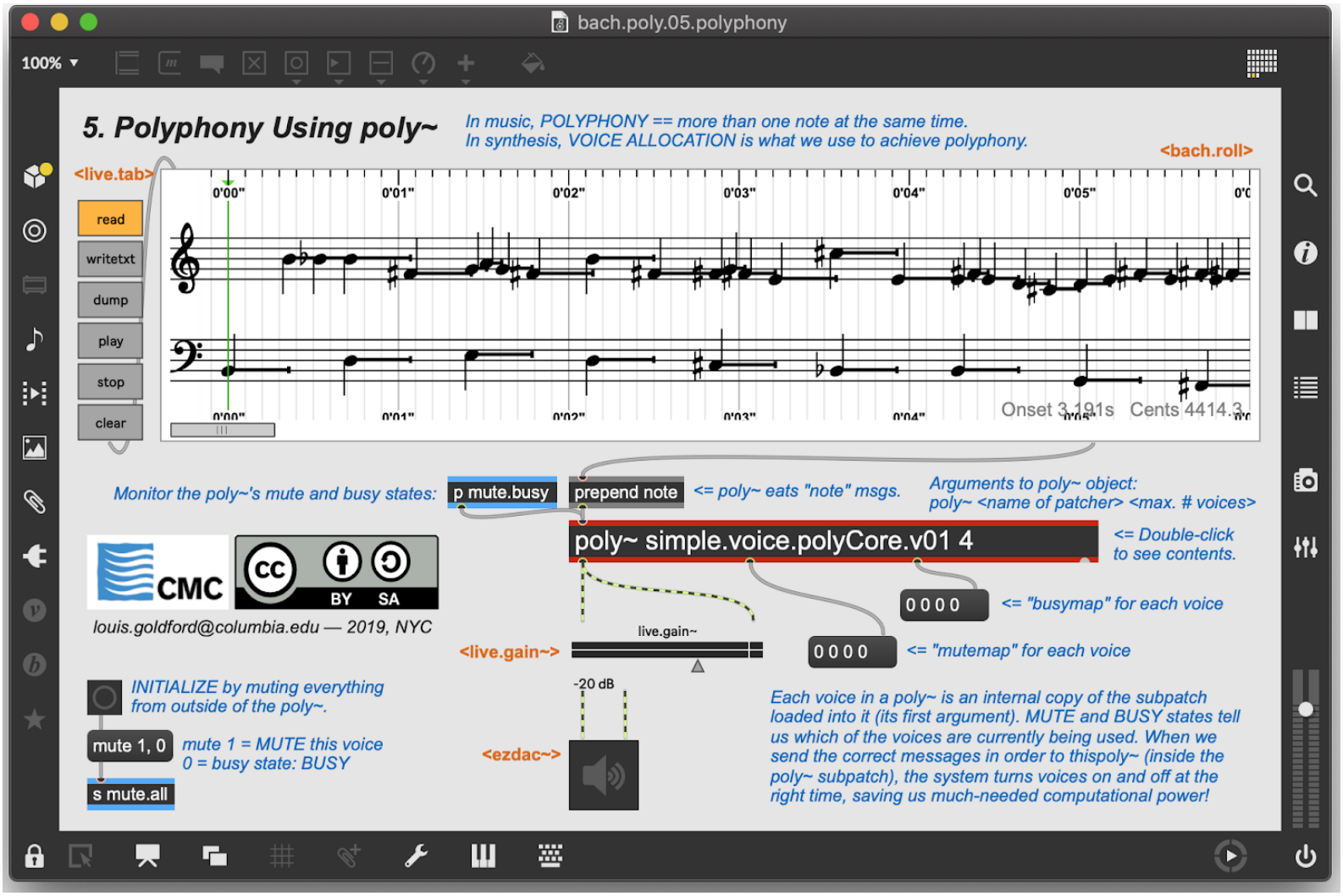
2.1 Monophonic Synthesizer



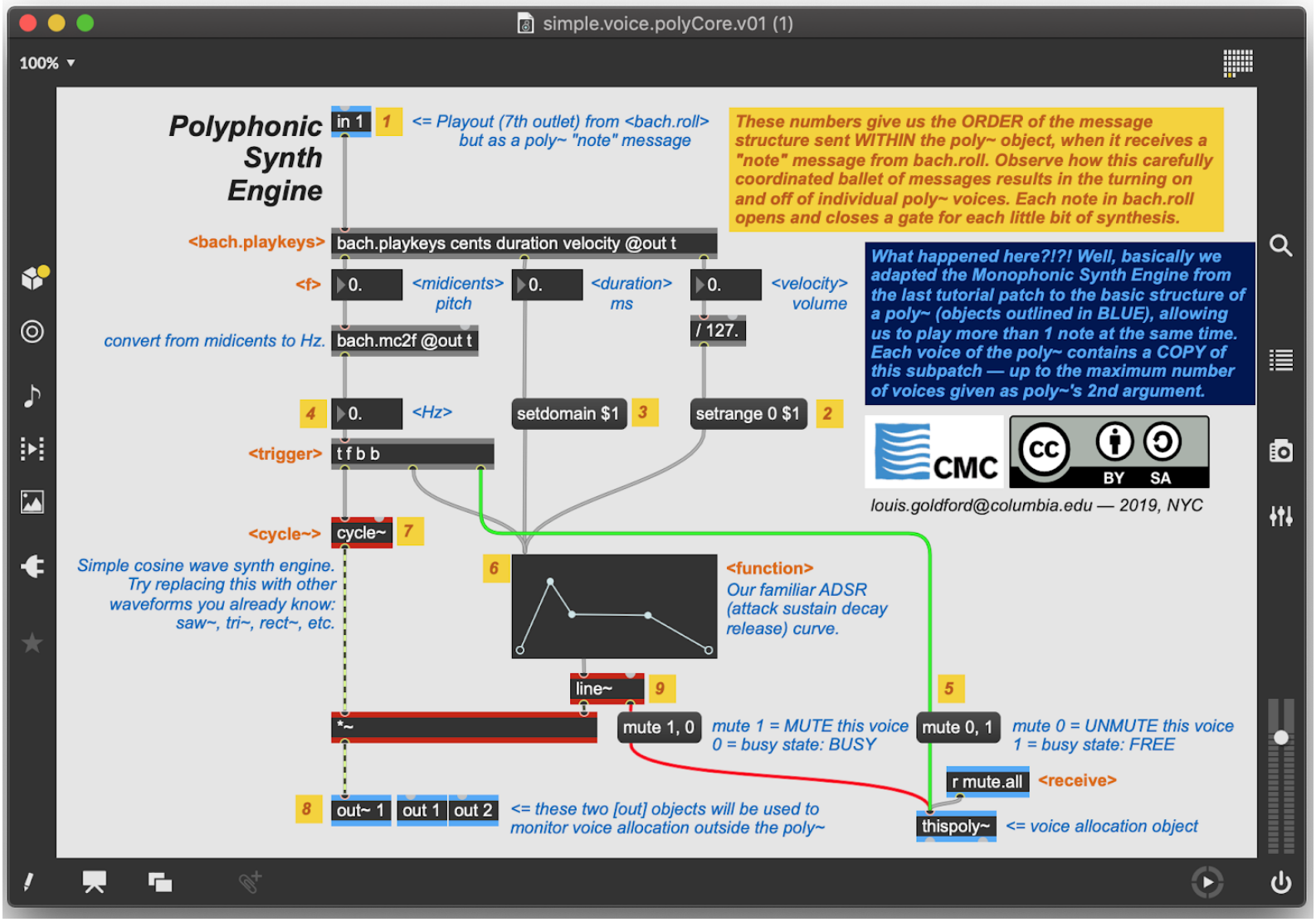
2.1.1 Monophonic Synth Subpatcher



2.2 Polyphonic Synth with poly~

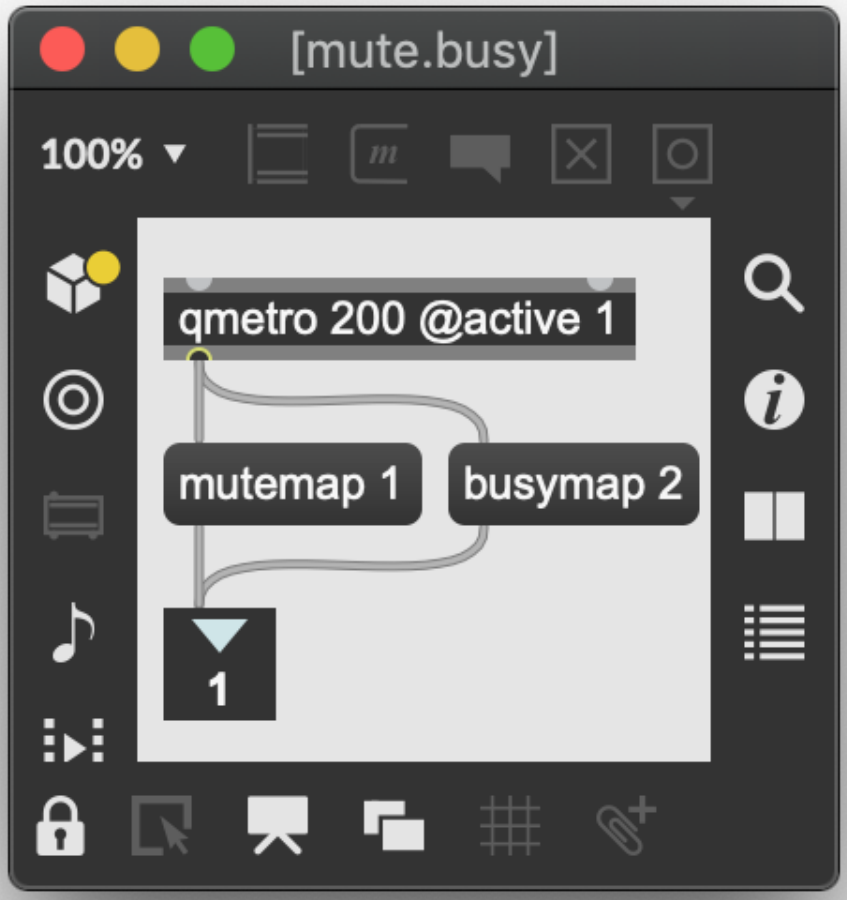


2.2.2 poly~ Subpatcher



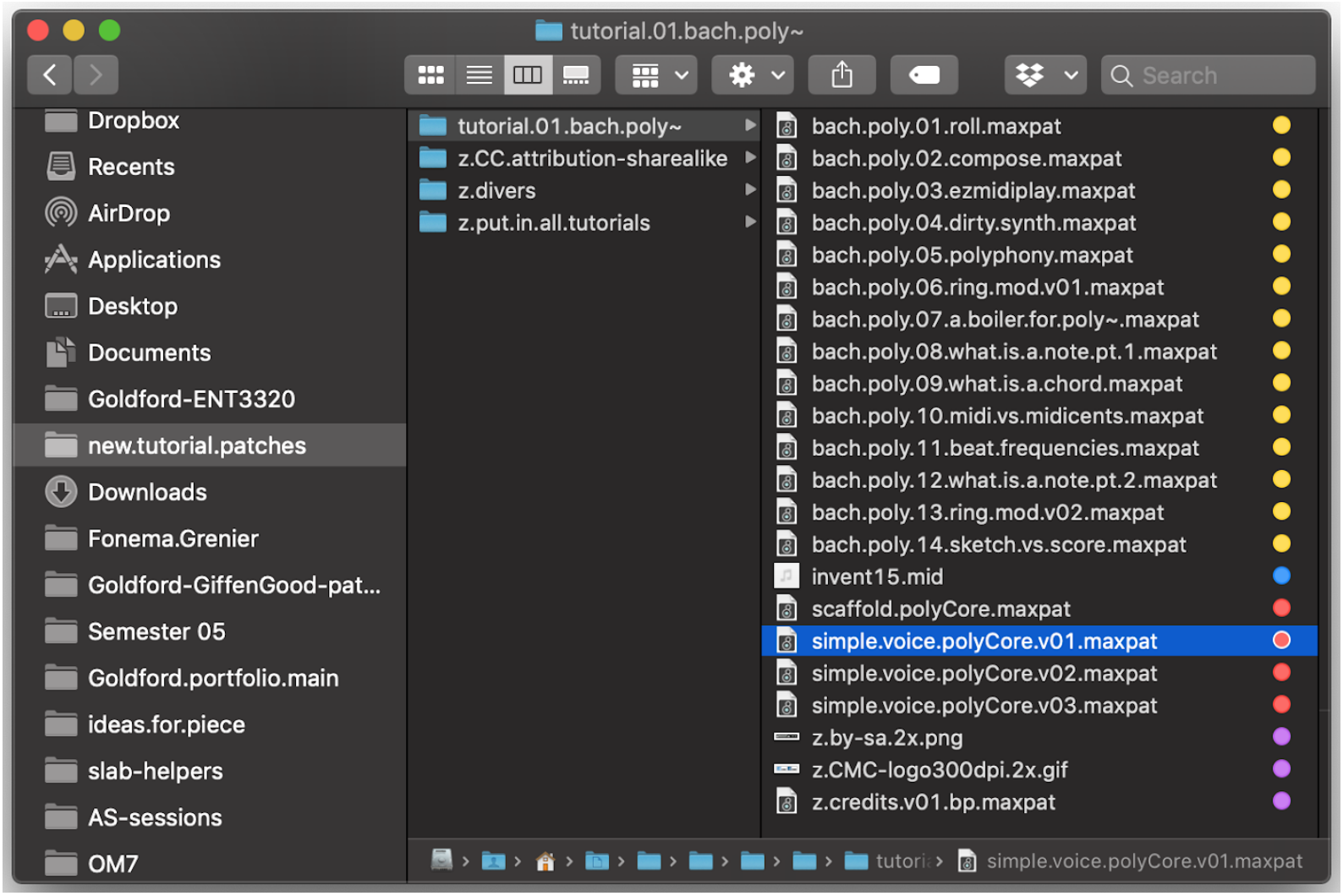
2.2.3 Monitoring the Mute-/Busy-States

By adding this subpatcher (below) to your main patch and connecting it to your poly~ (see 2.2 above), and an additional two extra outlets to the poly~ subpatcher itself (see 2.2.2 above), one can easily monitor the "mute" and "busy" map for all voices of the poly~, the total number of which was given as the 2nd argument to the poly~ object in the main patch.

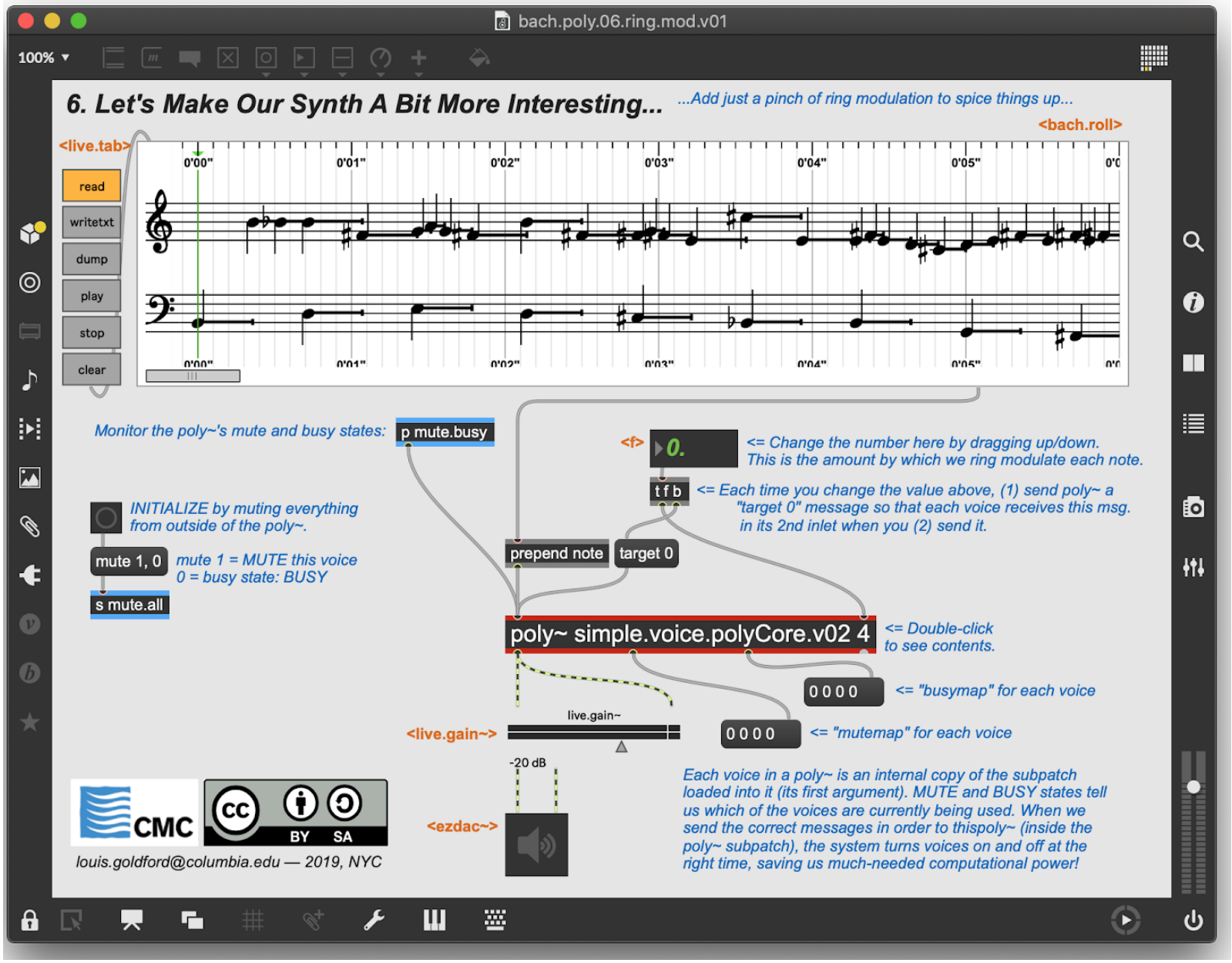


2.2.4 poly~ Subpatch Folder Structure

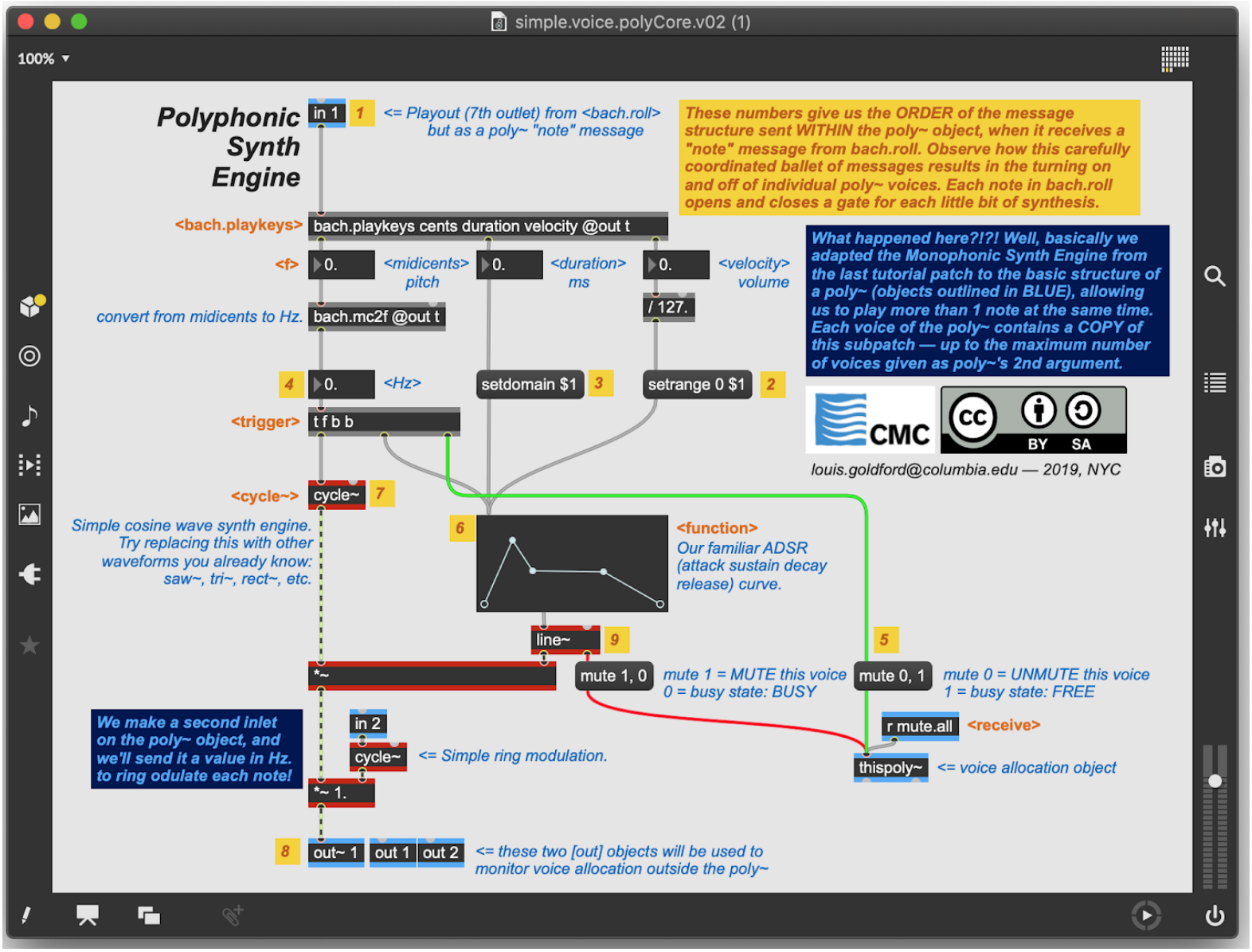
The main patch containing your poly~ object (see 2.2 above) and the subpatch loaded by the poly~ object and given as its 1st argument (see 2.2.2 above), must both be contained in the same directory, at the same level:



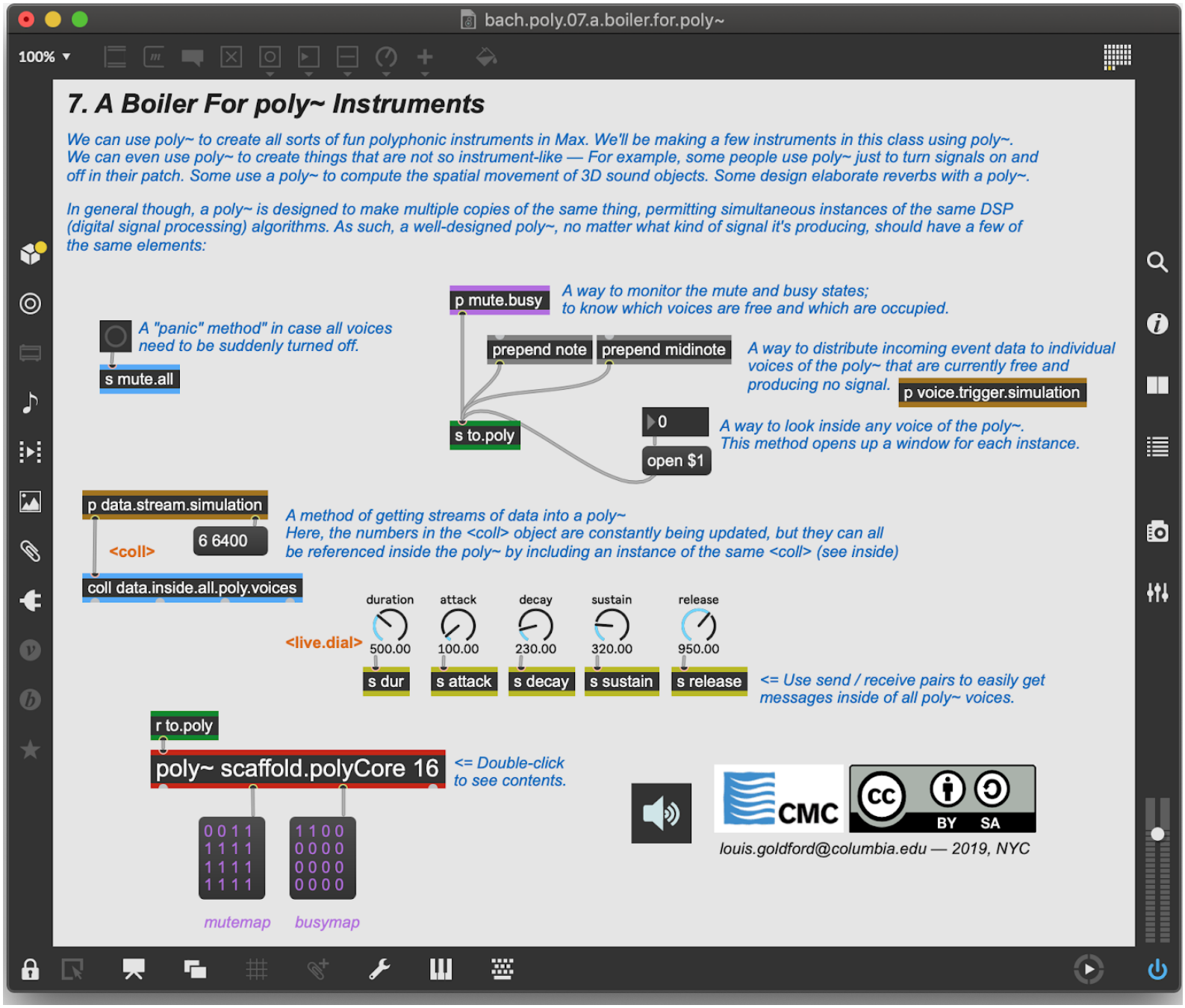
2.3 Adding Control Over Synth Parameters



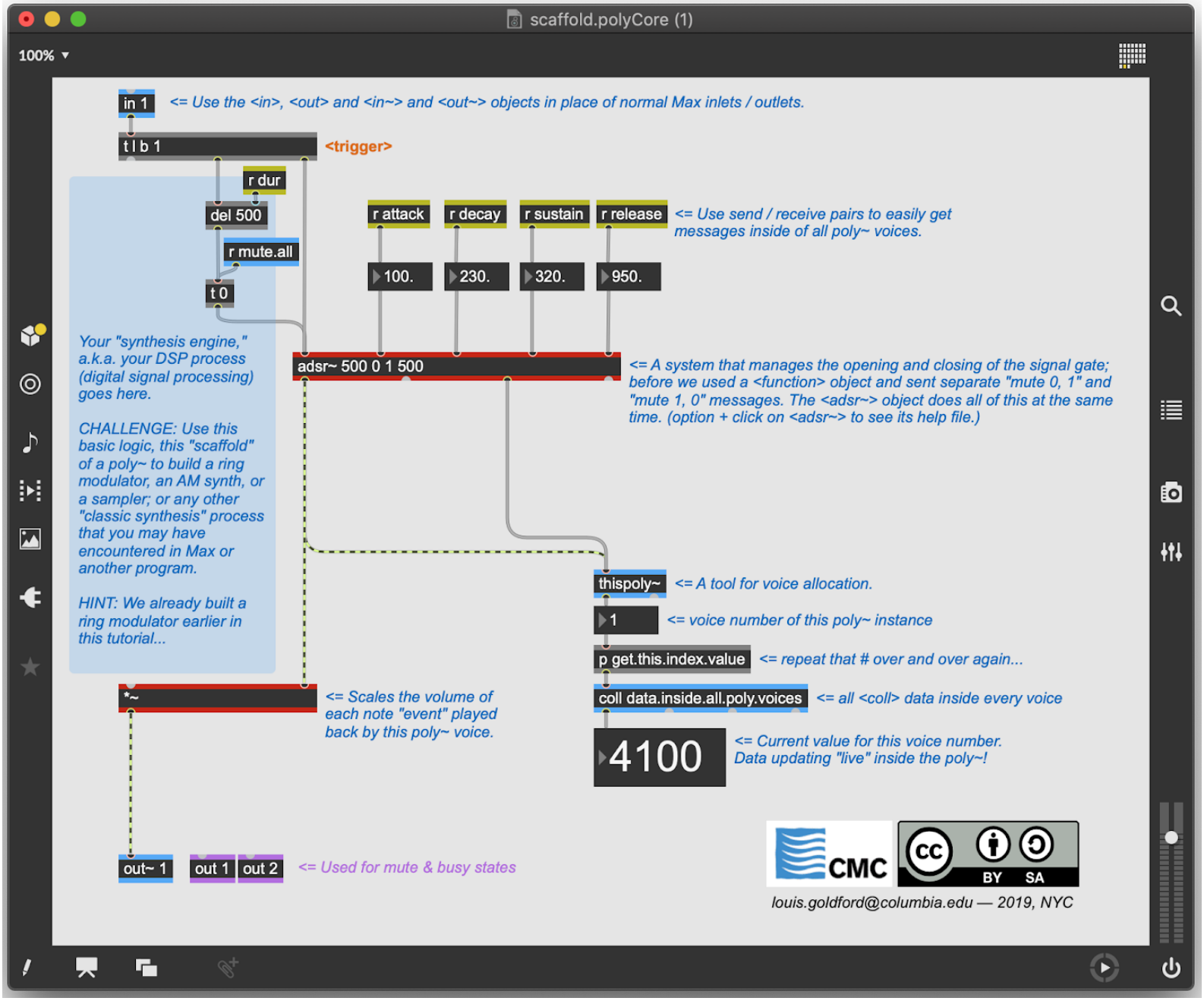
2.3.1 poly~ Subpatch With Ring Modulation



3. A Boiler For poly~ Instruments

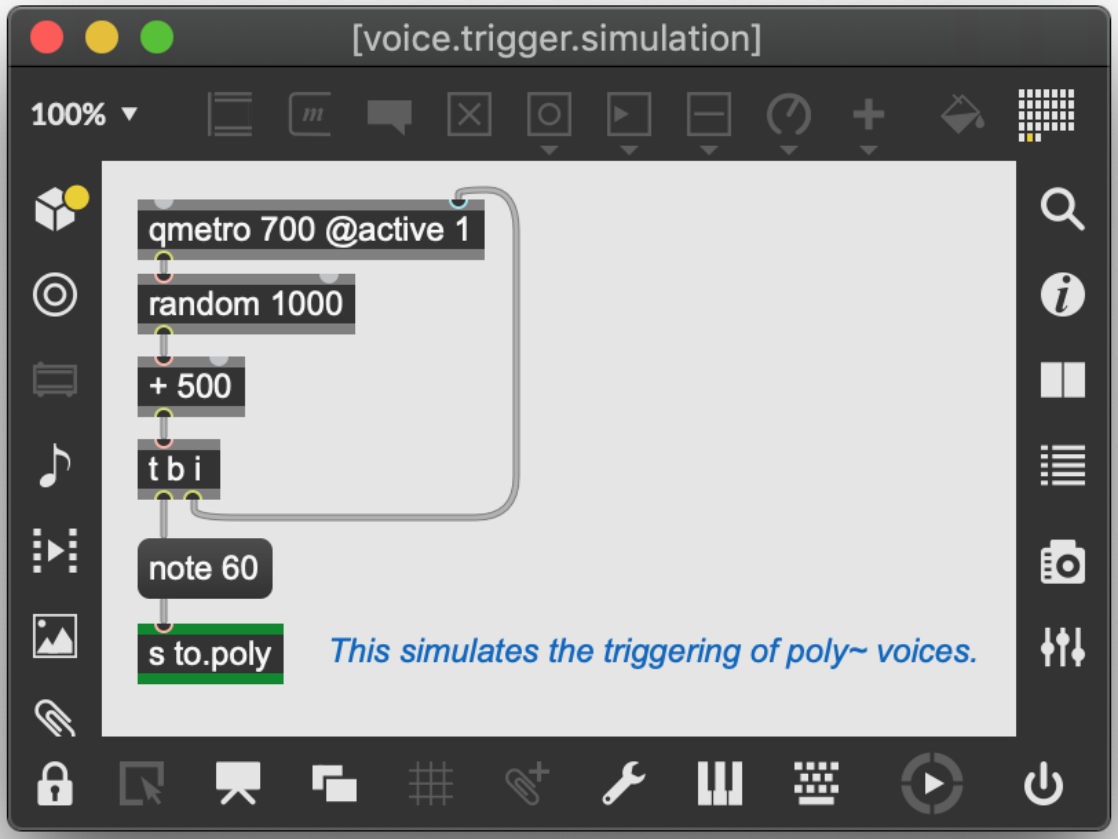


3.1 poly~ Subpatch Scaffolding



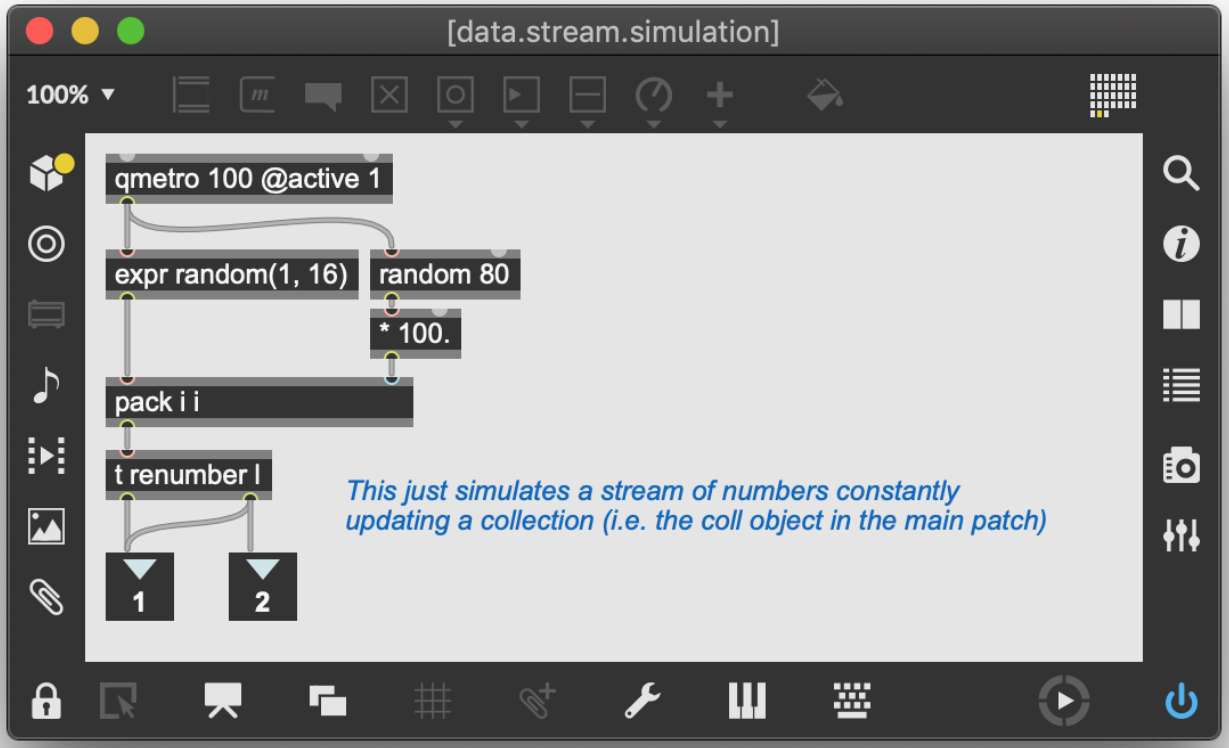
3.2 Voice Trigger Simulation

Use a MIDI keyboard or a bach.roll in place of this:

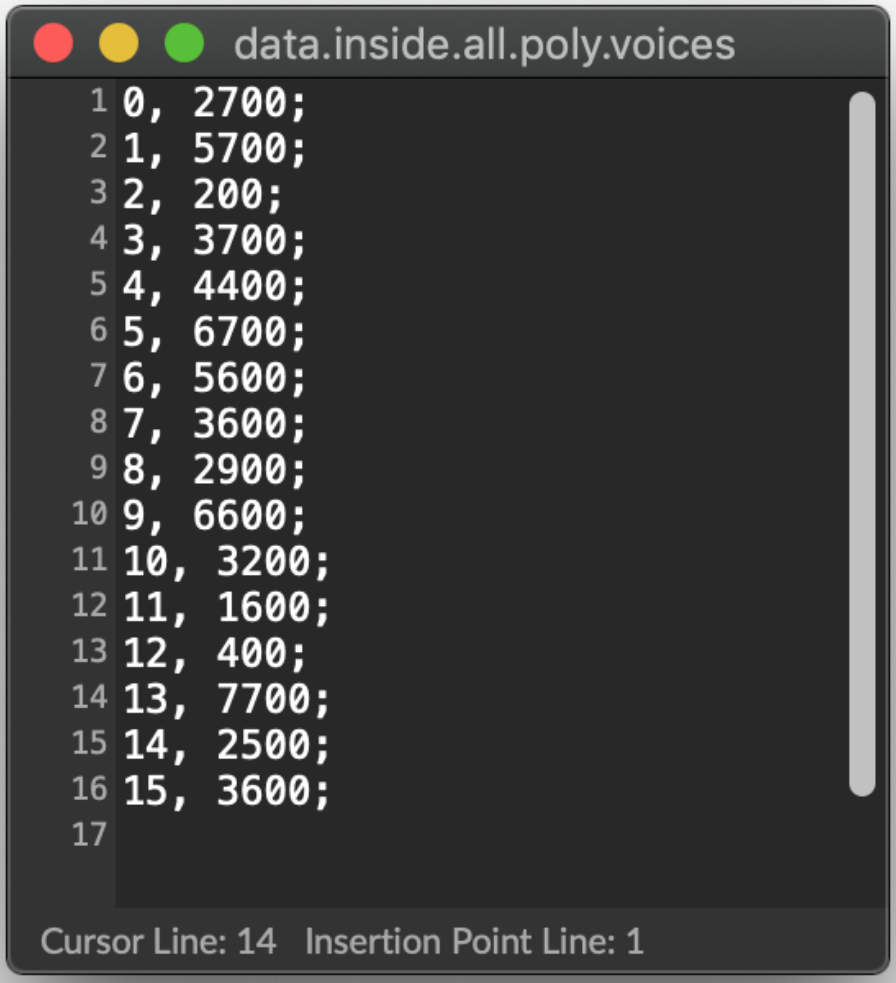


3.3 Data Stream Simulation

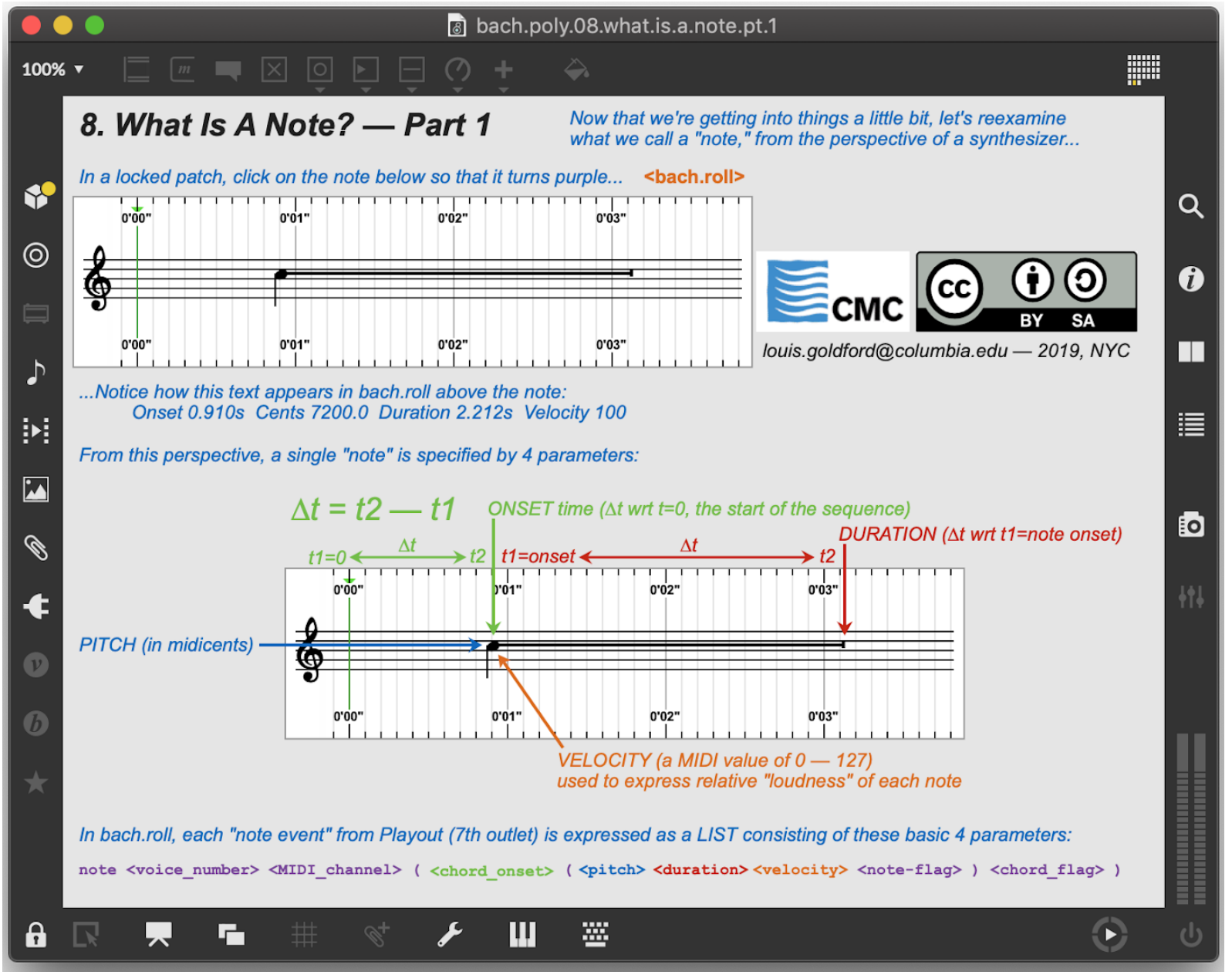
This could be a stream of data updating itself live from a webpage, or it could be algorithmically-generated from within your main patch somewhere. The idea is to shuttle a lot of numbers at the same time into your poly~ and to be able to use them all, or individually based on the voice number (as in this example, see 3.1 above).



3.4 Example Data Inside Coll



4. What Is A Note? — Part 1



5. What Is A Chord?

bach.poly.09.what.is.a.chord

100%

9. What Is A Chord?

So far we've mostly been working with notes that appear one after another. At least, until we added POLYPHONY in the last patch. Chords, as it turns out, are a special case of polyphony...

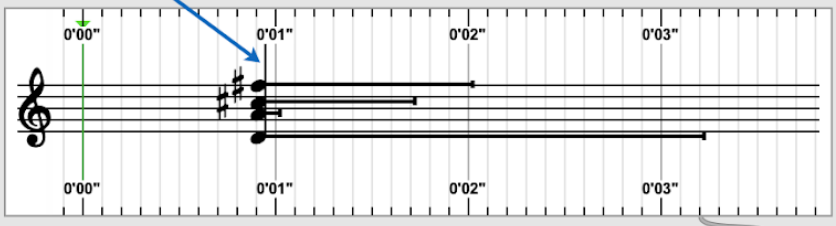
A chord is just a collection of notes that all sound at the same time... In other words, they all share the same ONSET value.

In traditional music notation, we show this by placing all members of the same chord on one "STEM," i.e. the line that points up here:

CMC

louis.goldford@columbia.edu — 2019, NYC

In bach, all notes in a chord can have their own duration:



<bach.roll>

These 4 notes all have the same ONSET, and bach interprets this by placing them all on the same stem, just like in music.

bach.ezmidiplay

<= Listen to the chord. Another way to play: With the patch locked, click on the bach.roll object and press the space bar to toggle playback on and off.

You can listen to the whole chord or to each note of the chord individually. With the patch locked, click on bach.roll and press the spacebar to hear the whole chord. Now, click + drag only over the notes you want to hear, hiighlighting only those notes in purple. Now, shift + spacebar to hear only your selection.

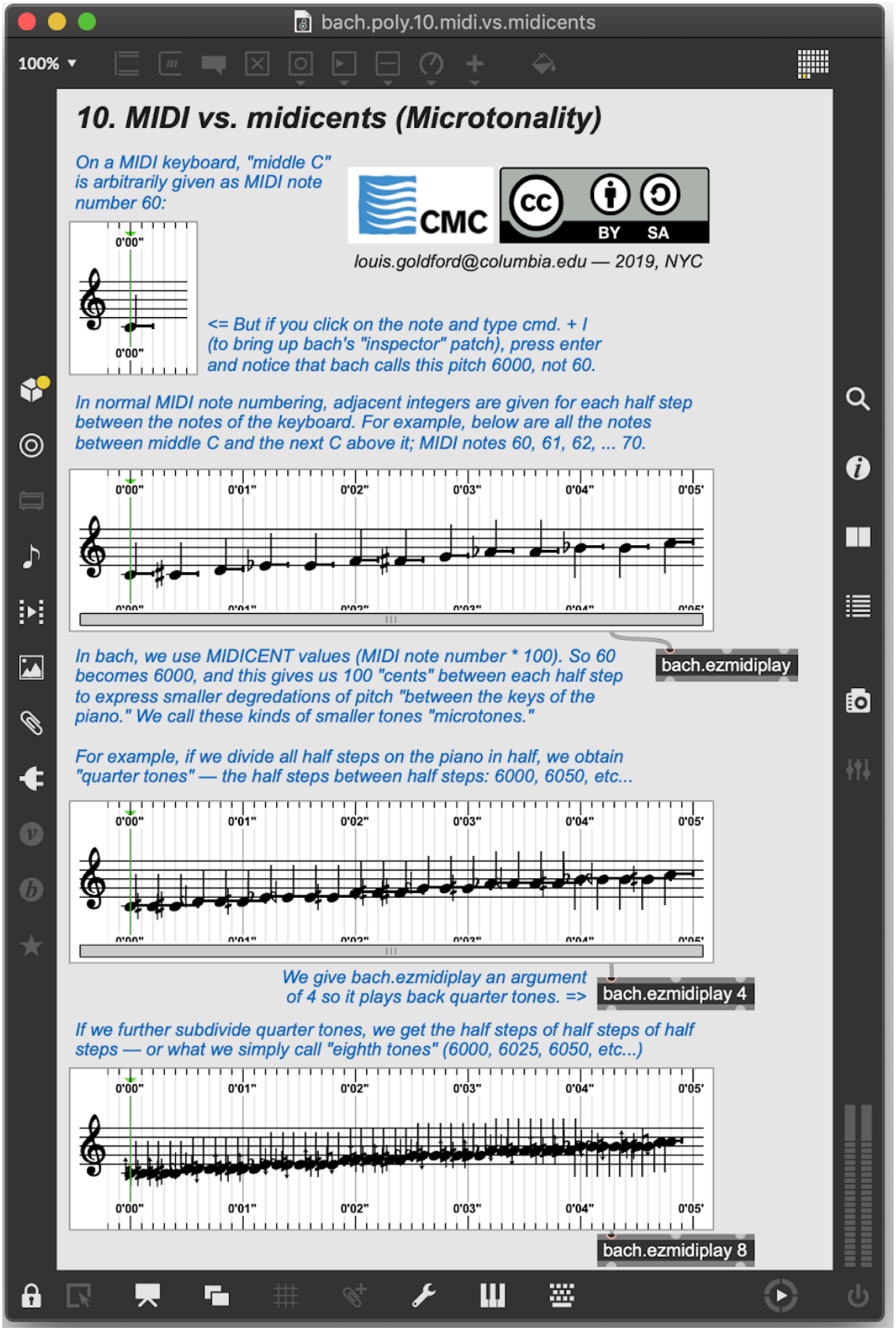
ADD NOTES TO THIS CHORD:
With the patch locked, hover over a note in the chord until you see the cross.
Hold down option + shift. The green + (plus) sign appears.
With these keys held down and the green + sign visible, click on a note.
It turns purple as usual, but...
Now DRAG and it creates a COPY of the note, still on the same stem.
Drag up or down to the pitch you want, and then release everything.

Congrats... now you have 5 notes in your chord!
Try dragging the notes up or down, rearranging them as you please.
Listen to them (see the text next to bach.ezmidiplay), and make changes in pitch and in the number of notes until you find a chord that you like.

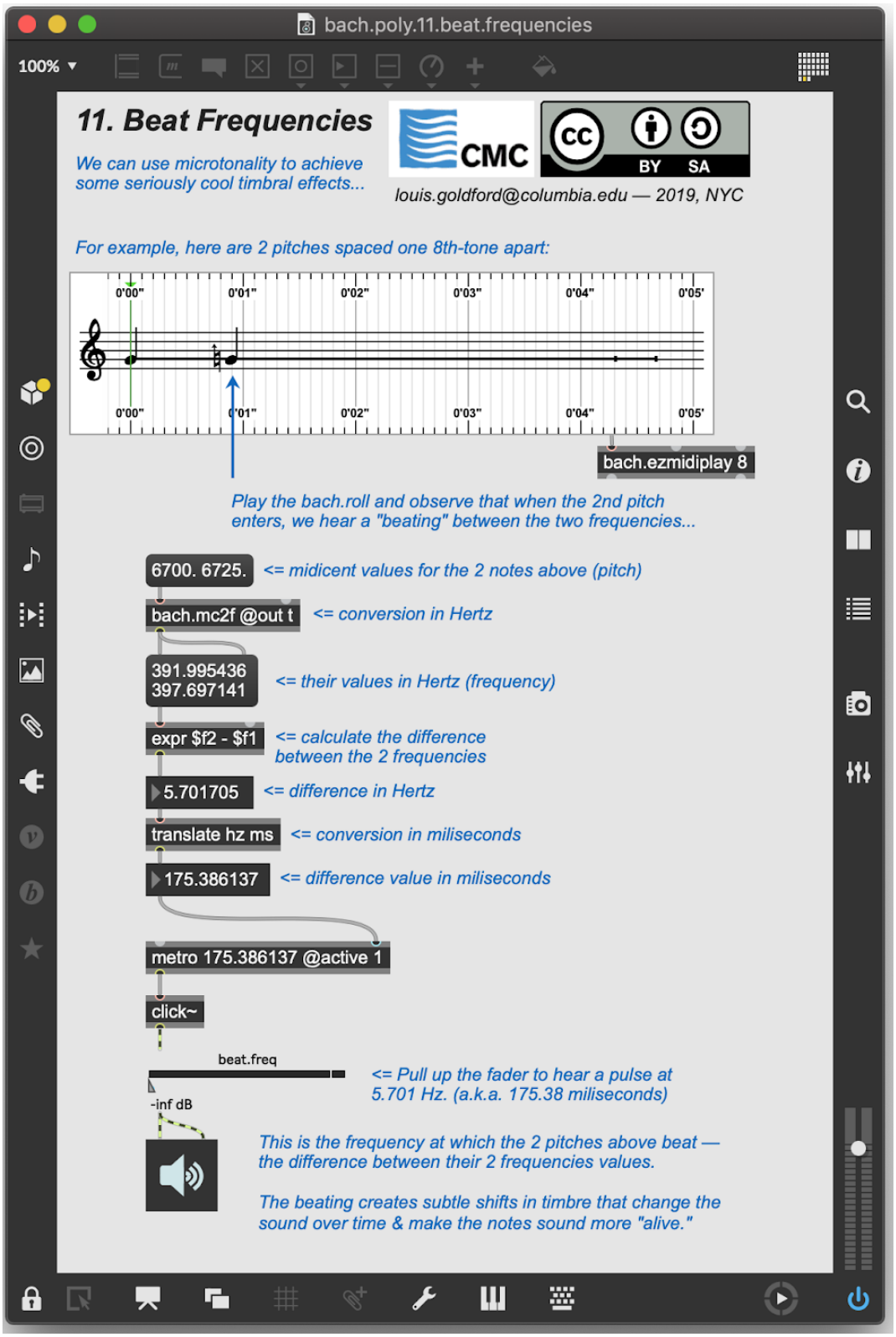
In bach.roll, each "chord" from Playout (7th outlet) is expressed as a LIST of LISTS consisting of the different notes. Each list is wrapped in parentheses or brackets. Here is a chord with 3 notes in it:

```
chord <voice_number> <MIDI_channel> (<chord_onset> ( <pitch> <duration>
<velocity> <note-flag> )( <pitch> <duration> <velocity> <note-flag> ) ( <pitch>
<duration> <velocity> <note-flag> ) <chord_flag> )
```

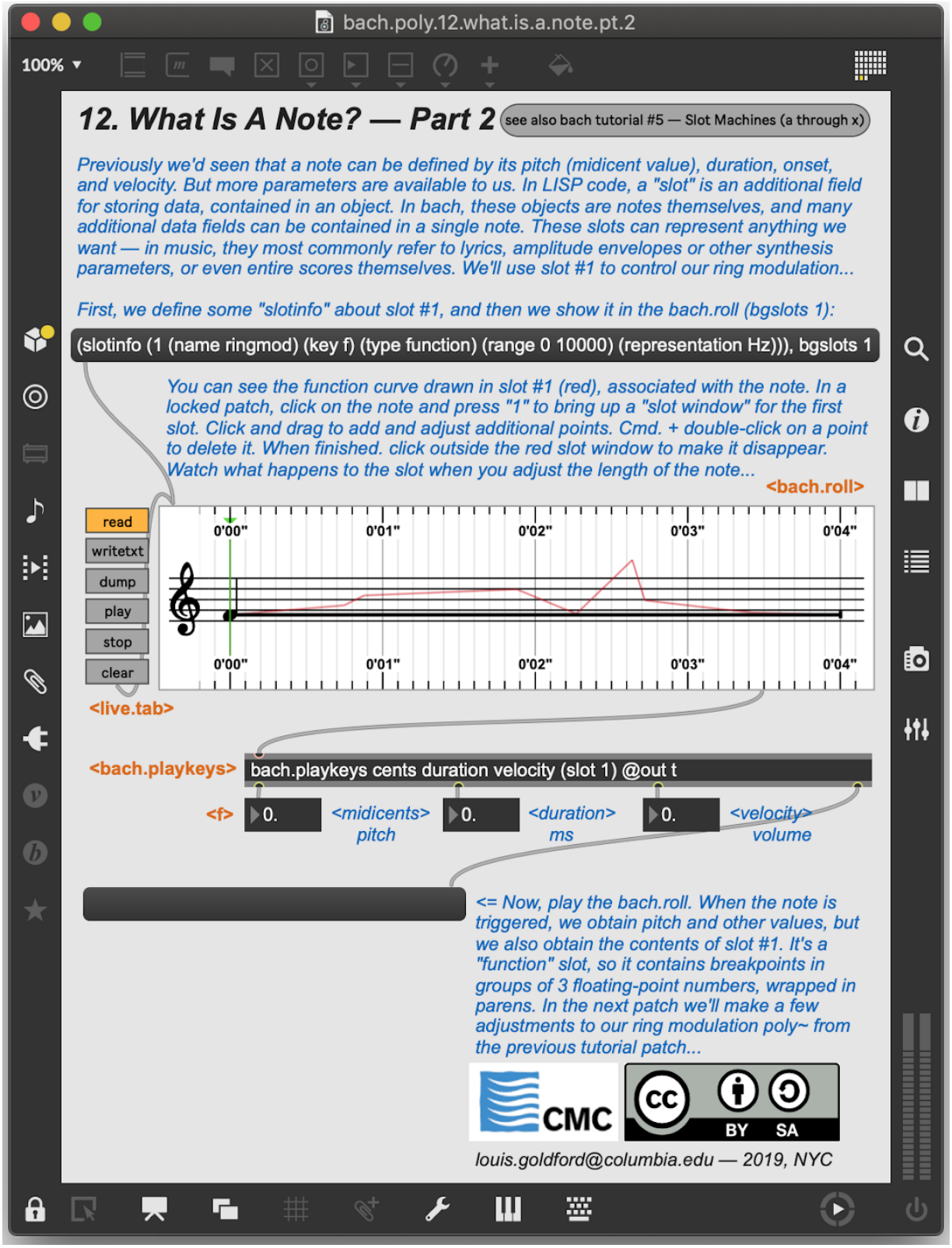
6. MIDI vs. Midicents (Microtonality)



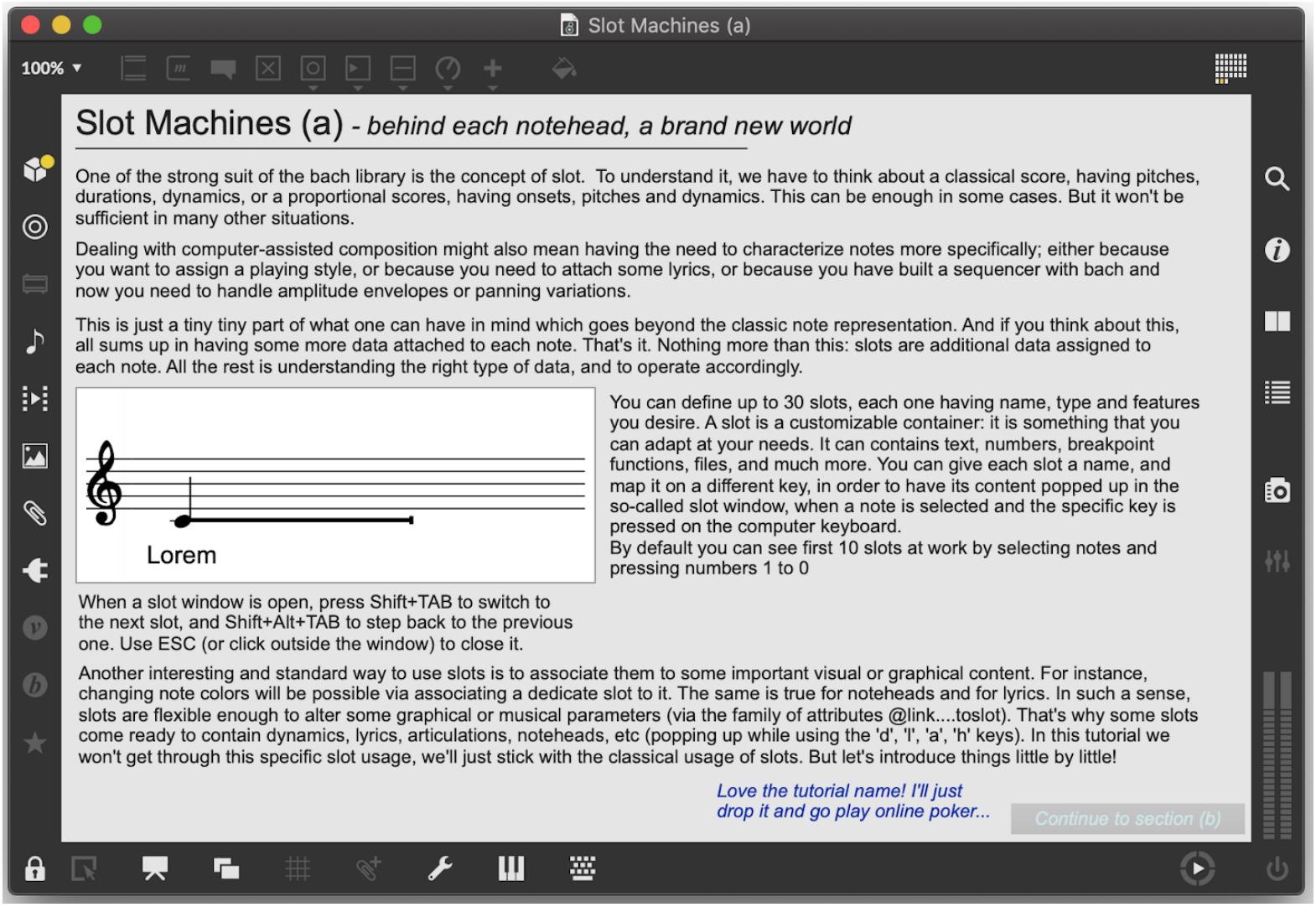
7. Playing With Beat Frequencies



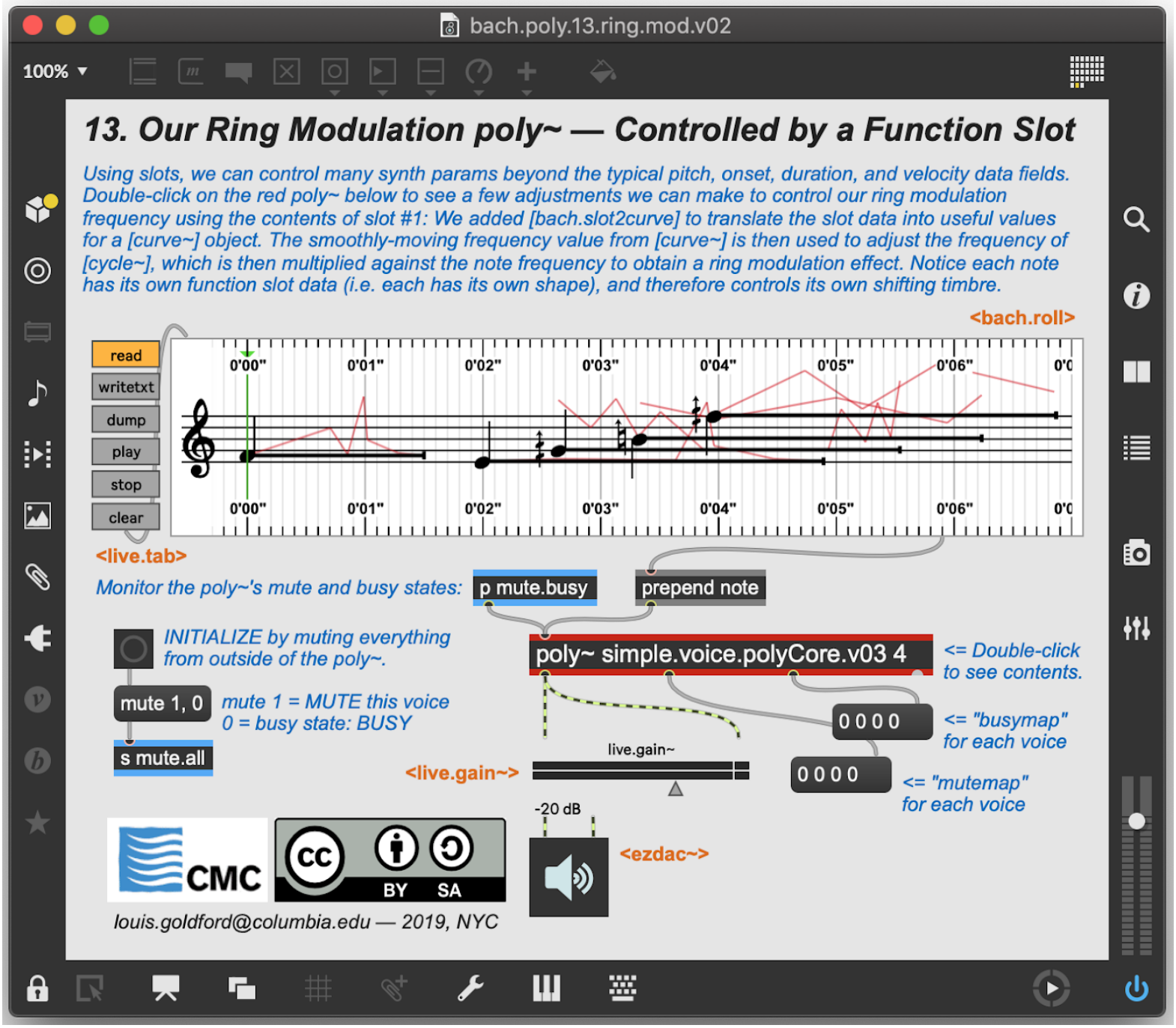
8. What Is a Note — Part 2



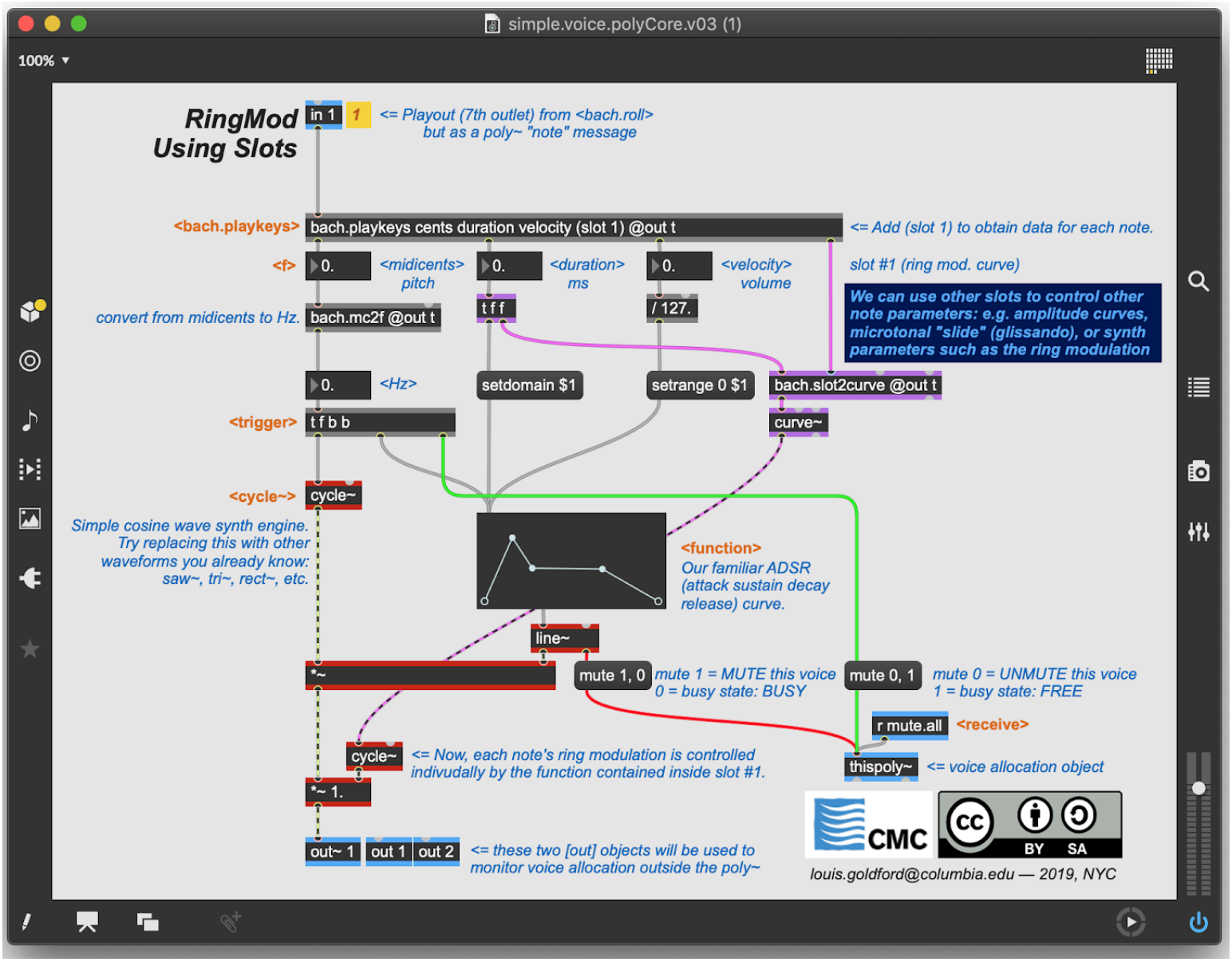
8.1 bach “Slot Machines” Tutorial — Part A



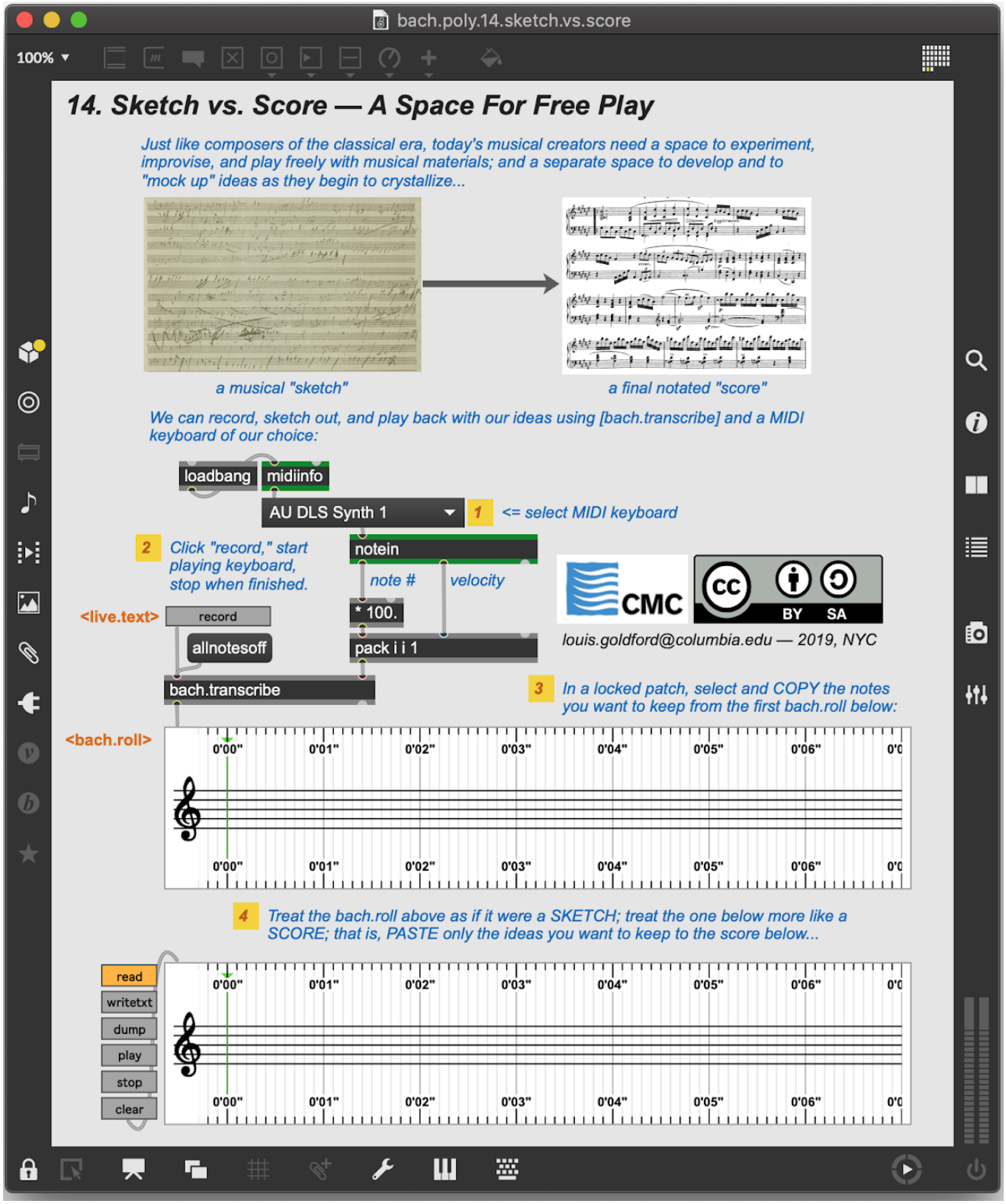
9. Controlling poly~ With bach Slots



9.1 poly~ Subpatch



10. Sketch Vs. Score



11. How to Record Audio from Max

You can quickly record any signal generated in Max.
*Any sound coming into a dac~ object can be captured using the **Quickrecord** subpatch:*