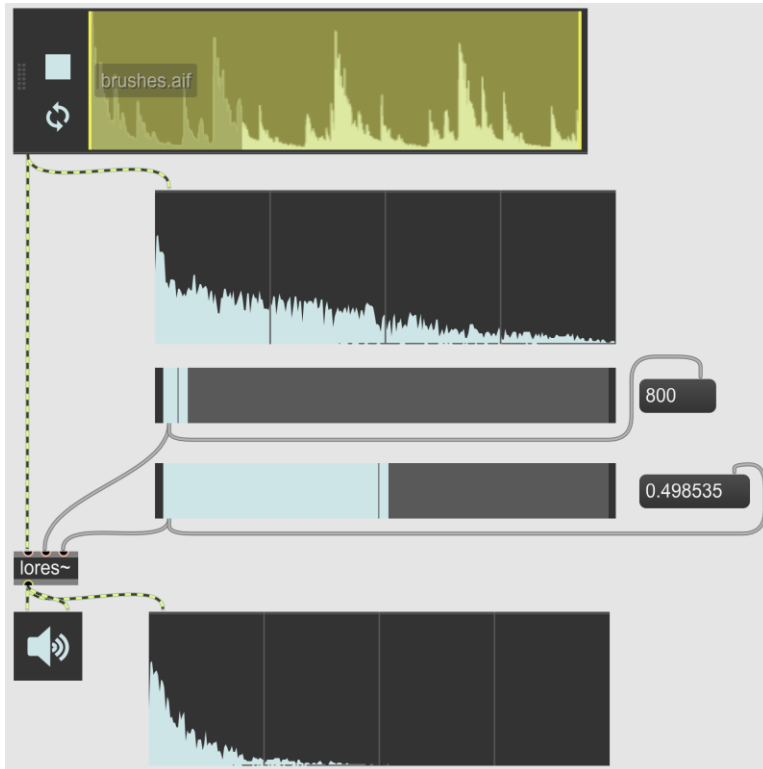


Lecture 19 – Filters

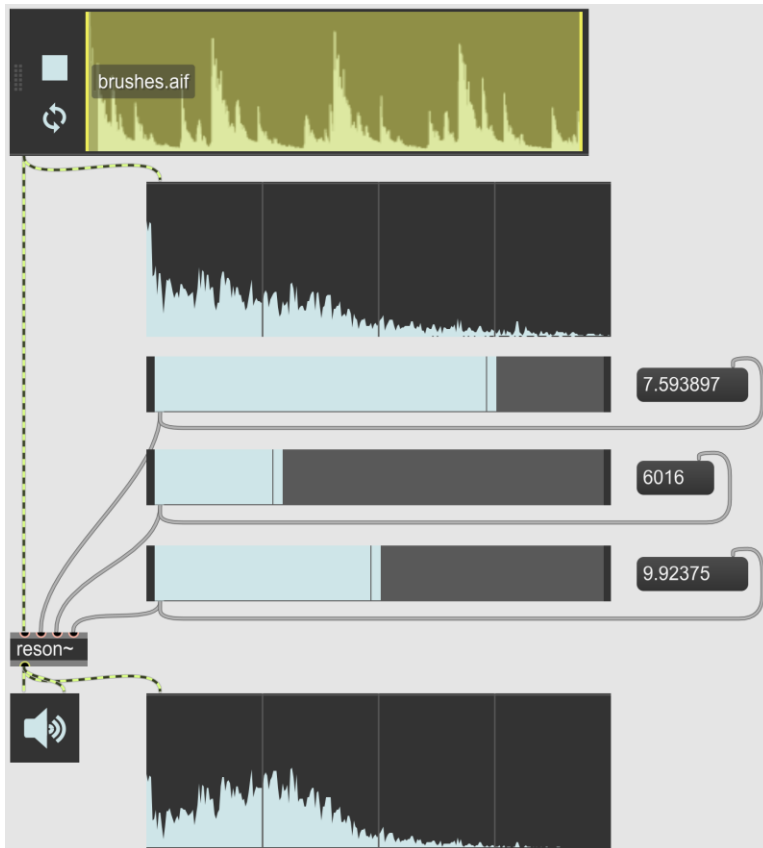
Instructor: Hao-Wen Dong

Example 1: Filters (“1_filters.maxpat”)

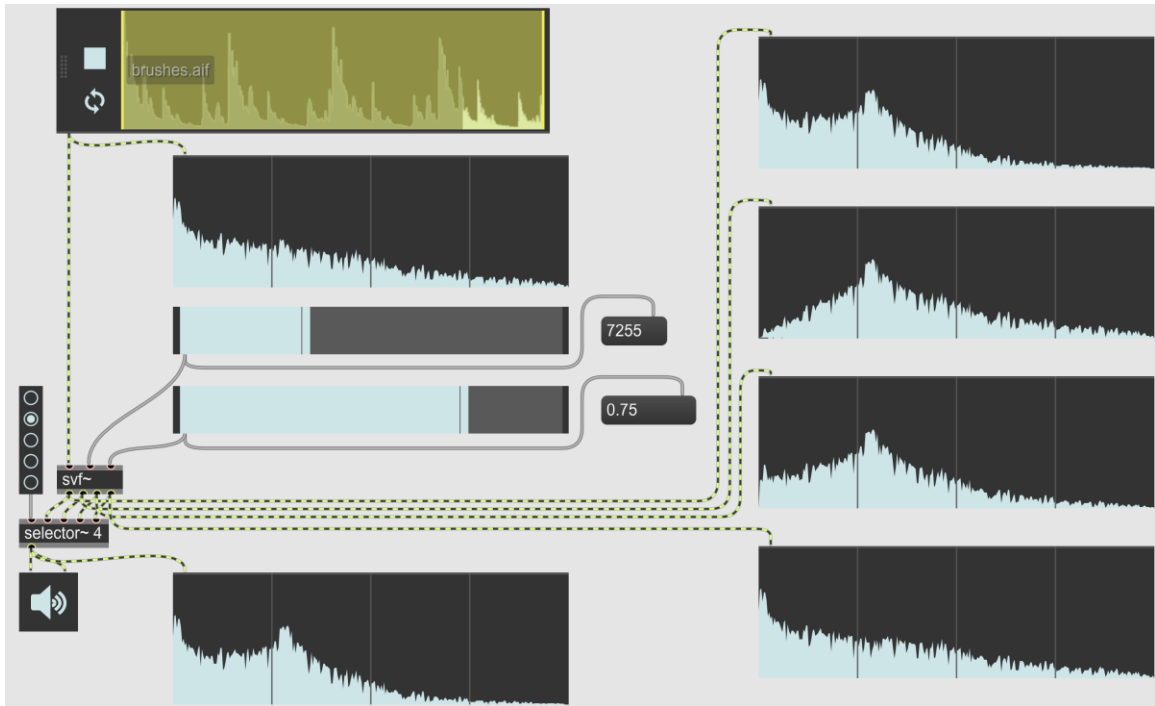
- Use the “lores~” object to create a simple *lowpass* filter that allows only “low” frequency components to “pass” through (i.e., filter our higher frequencies)



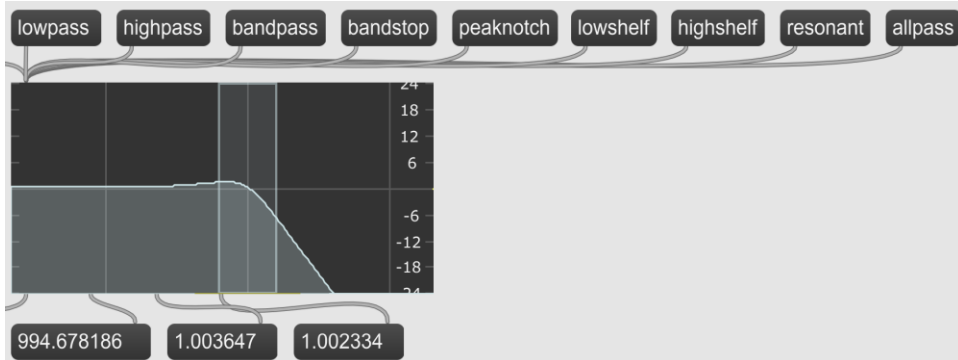
- Use the “reson~” object to create a simple *bandpass* filter that allows only a frequency “band” to “pass” through (i.e., filter out lower and higher frequencies)



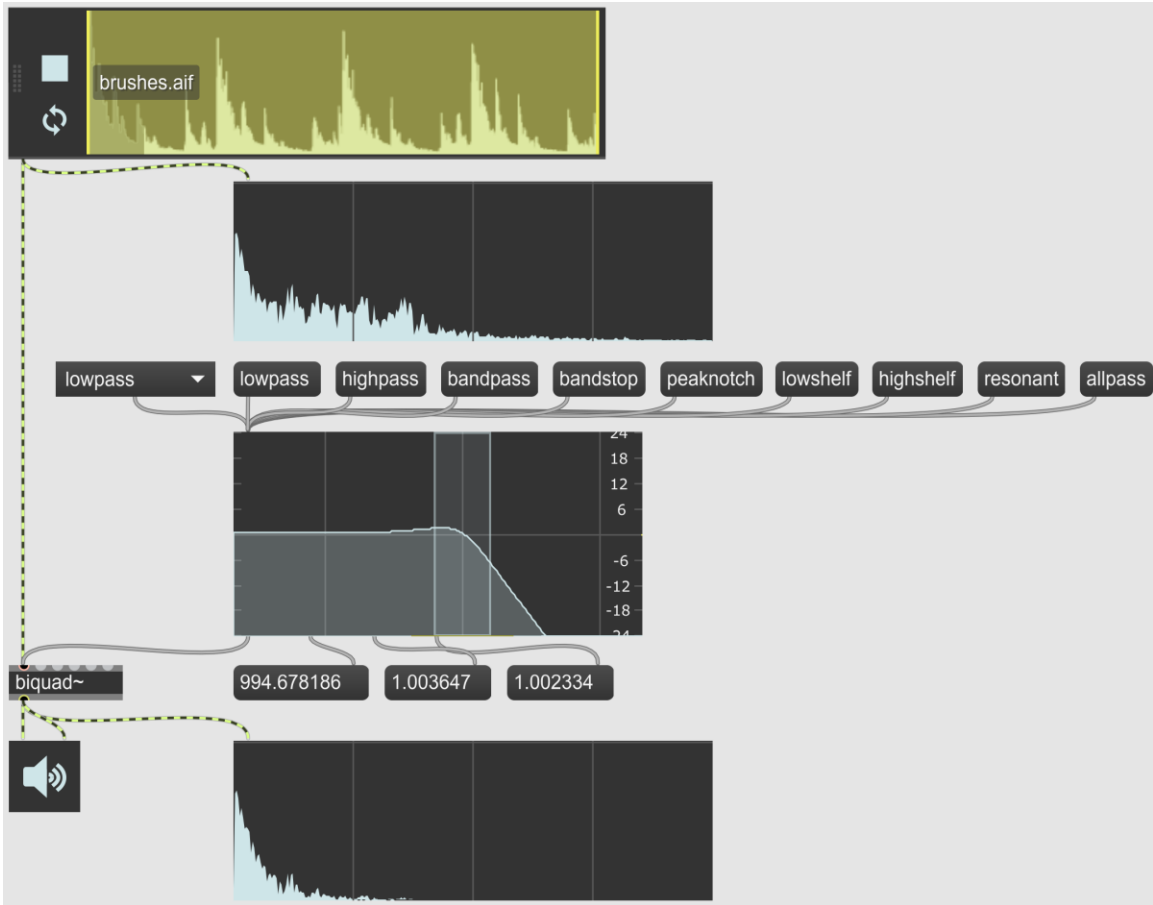
- Use the "svf~" object to create a state-variable filter that has four outlets that correspond to *lowpass*, *highpass*, *bandpass* and *notch* (bandreject) filters



- Use a "filtergraph~" object to create a custom filter from nine filter types

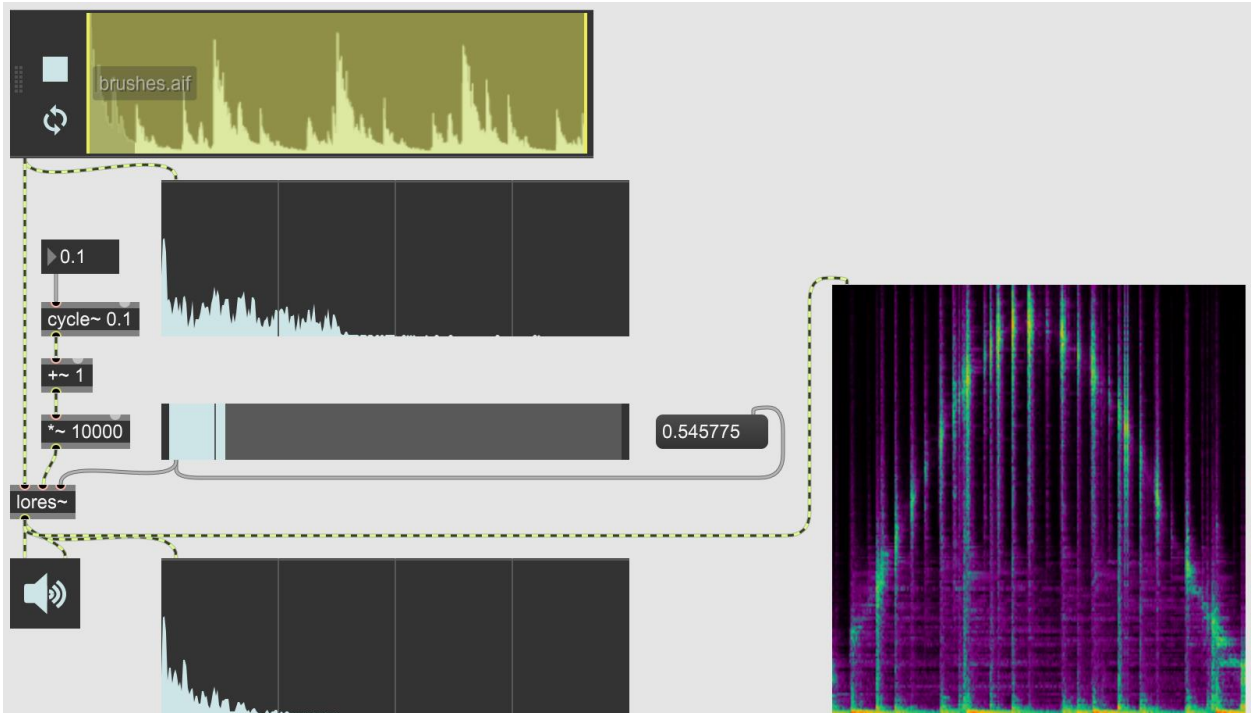


- Use the “biquad~” to create a filter based on the output of the “filtergraph~” object

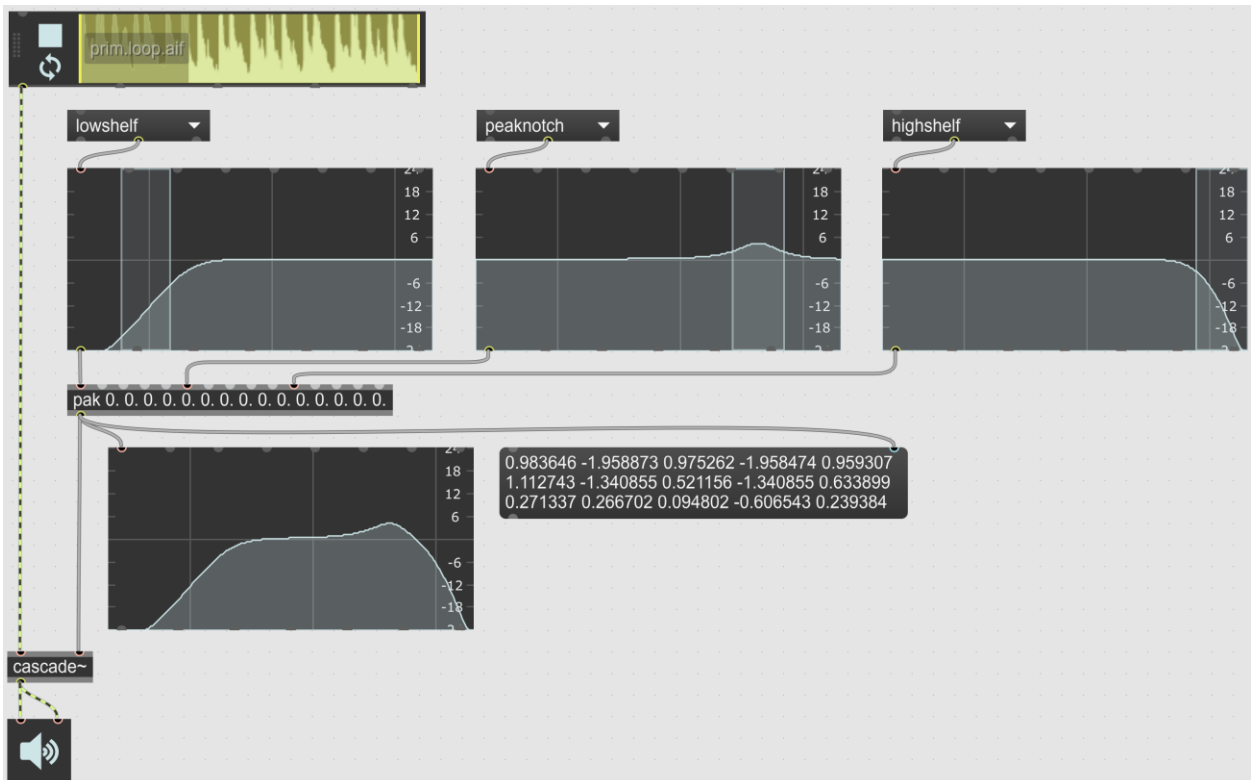


Example 2: Advanced Filters ("2_advanced_filters.maxpat")

- We can send a signal to control the center frequency of the lowpass filter of a "lores~" object dynamically



- Use the "cascade~" object to create a filter that combines a number of simple filters



- Note the difference between a “pack” object and a “pak” object. A “pack” object has only one hot inlet at the very left, while a “pak” object has all the inlets as hot inlets.
- We can also use the “filtergraph~” object to achieve this by setting an “Active Filter(s)” to a value larger than 1

