

FluCoMa



Ted Moore

tedmooremusic.com

ted@tedmooremusic.com

FluCoMa: Fluid Corpus Manipulation



- Toolbox for Programmatic Data Mining of Sound Banks
- Integrating Machine Listening and Machine Learning in...
- Max, SuperCollider & Pure Data
- Learning Resources (learn.flucoma.org)
- Discourse Community (discourse.flucoma.org)



Ted Moore, (James Bradbury, Owen Green, Jacob Hart, Gerard Roma, & Pierre Alexandre Tremblay)

University of Huddersfield | UK

FluCoMa: Fluid Corpus Manipulation



Slice Audio

- onset slice
- transient slice
- novelty slice
- amplitude slice
- amplitude gate

Decompose Audio

- extract transients
- harmonic/percussive separation
- model as sine waves
- non-negative matrix factorisation

Analyse Audio

- pitch
- loudness
- mel-bands
- mel-frequency cepstral coefficients
- spectral centroid
- spectral flatness
- chromagram

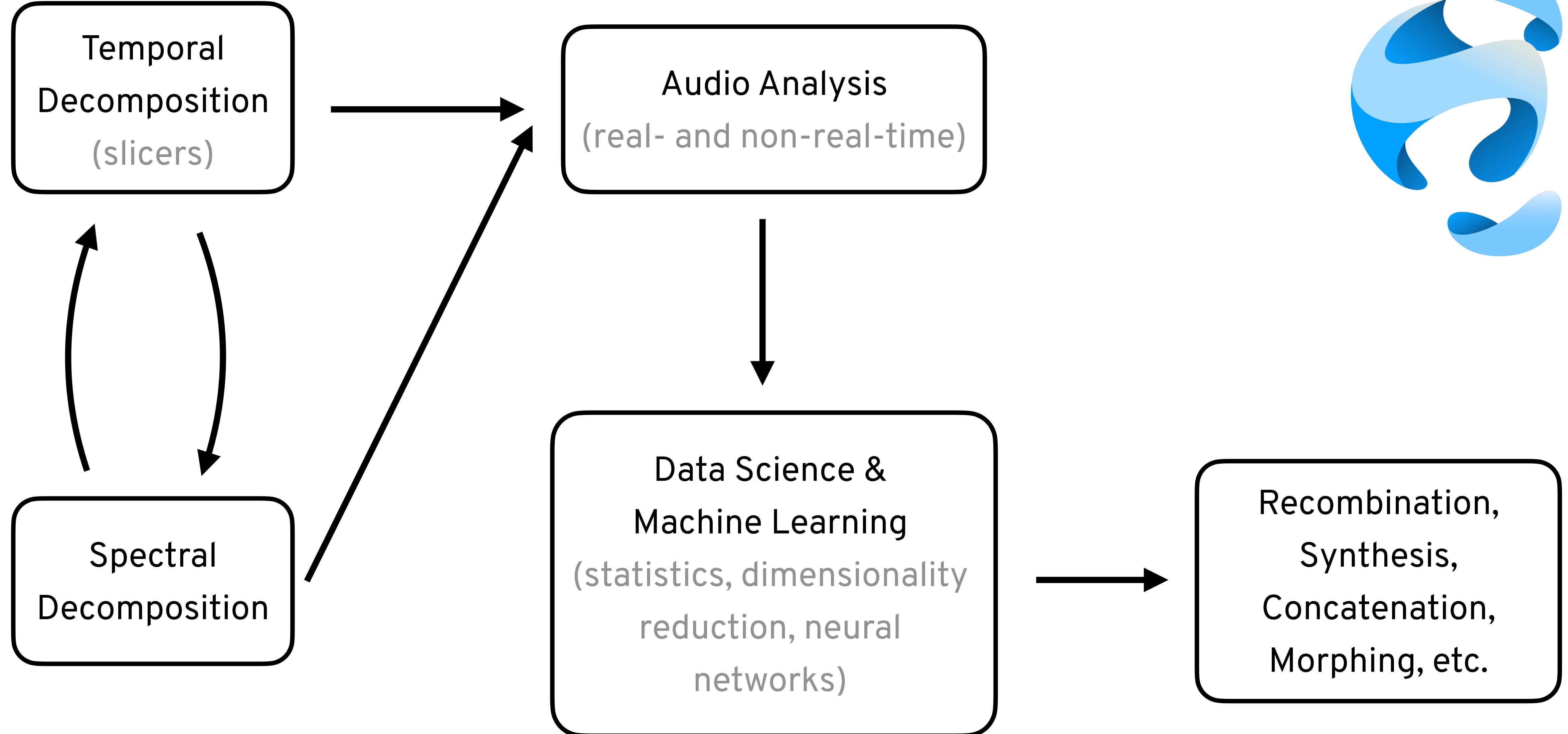
Transform Audio

- audio transport
- non-negative matrix factorisation filters & morphing

Analyse Data

- datasets
- labelsets
- statistical analyses
- normalization
- standardization
- robust scaler
- principal component analysis
- MDS
- KDTree
- K Nearest Neighbours
- neural networks
- SQL-type query
- KMeans
- UMAP
- grid

... and more



Install FluCoMa



- Install FluCoMa
- Access to resources for today via:
 - GitHub: <https://github.com/tedmoore/FluCoMa-SuperCollider-Yale>

MLPRegressor

control *many* synthesizer
parameters from a smaller
control space



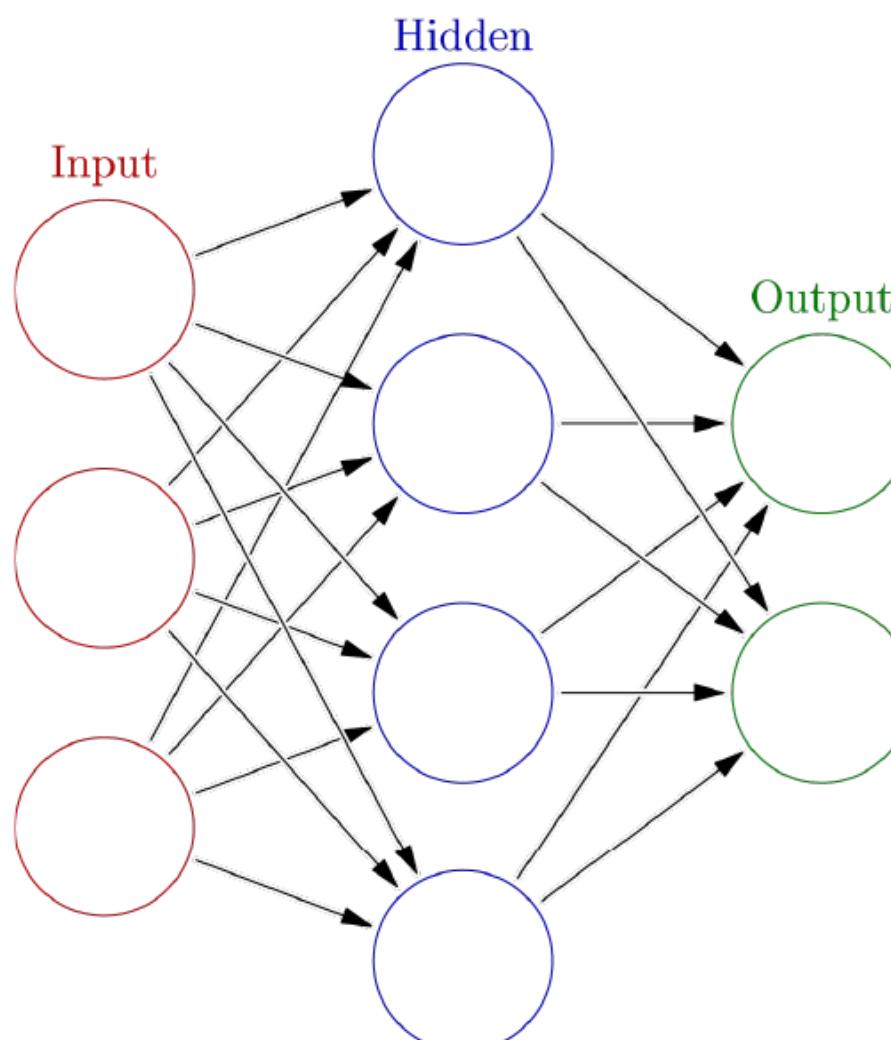
Neural Network Training a Regressor

identifier

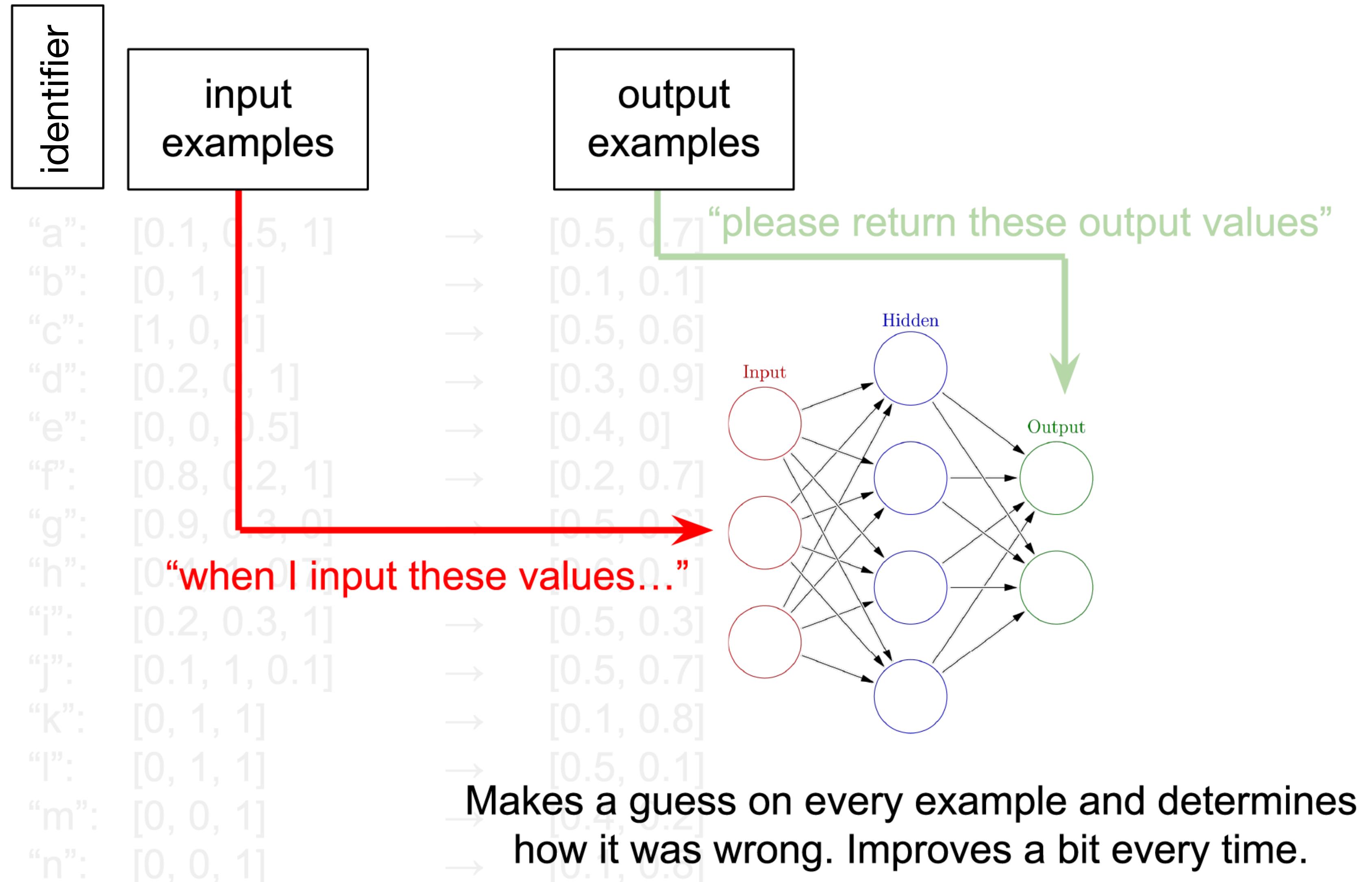
input
examples

“a”:	[0.1, 0.5, 1]	→	[0.5, 0.7]
“b”:	[0, 1, 1]	→	[0.1, 0.1]
“c”:	[1, 0, 1]	→	[0.5, 0.6]
“d”:	[0.2, 0, 1]	→	[0.3, 0.9]
“e”:	[0, 0, 0.5]	→	[0.4, 0]
“f”:	[0.8, 0.2, 1]	→	[0.2, 0.7]
“g”:	[0.9, 0.3, 0]	→	[0.5, 0.6]
“h”:	[0.4, 1, 0.7]	→	[0.6, 0.1]
“i”:	[0.2, 0.3, 1]	→	[0.5, 0.3]
“j”:	[0.1, 1, 0.1]	→	[0.5, 0.7]
“k”:	[0, 1, 1]	→	[0.1, 0.8]
“l”:	[0, 1, 1]	→	[0.5, 0.1]
“m”:	[0, 0, 1]	→	[0.4, 0.2]
“n”:	[0, 0, 1]	→	[0.1, 0.8]

output
examples



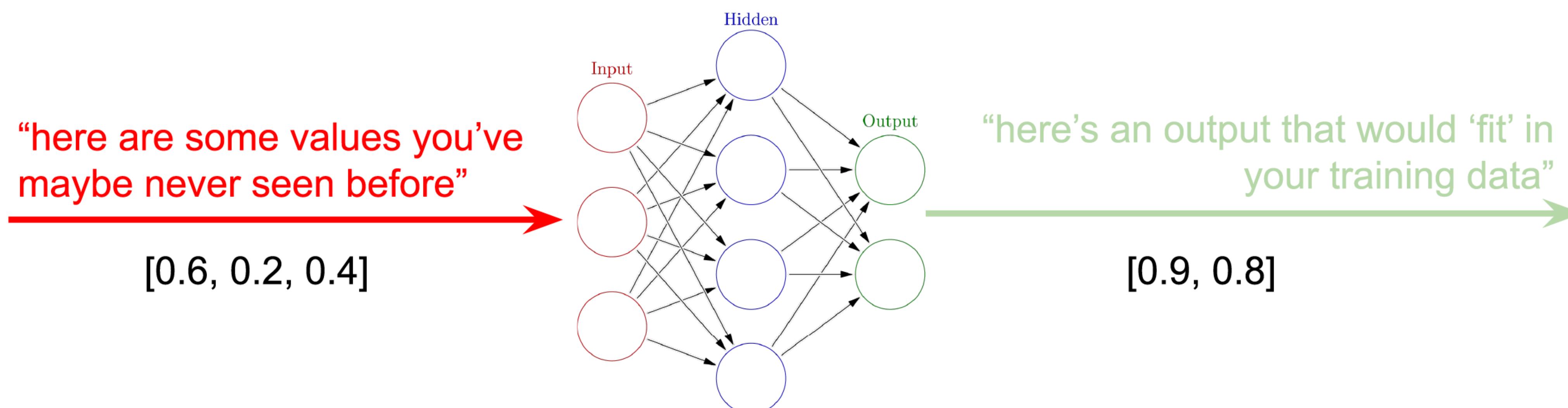
Neural Network *Training* a Regressor



Neural Network *Predicting with Regression*

input
examples

output
examples



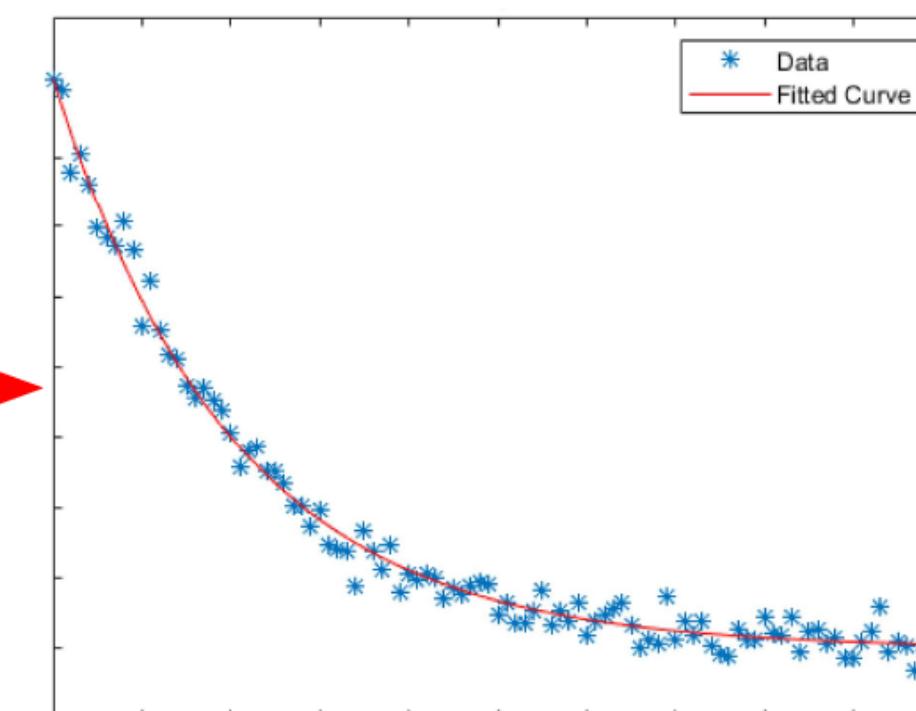
Neural Network *Predicting with Regression*

input
examples

output
examples

“here are some values you’ve
maybe never seen before”

[0.6, 0.2, 0.4]



“here’s an output that would ‘fit’ in
your training data”

[0.9, 0.8]

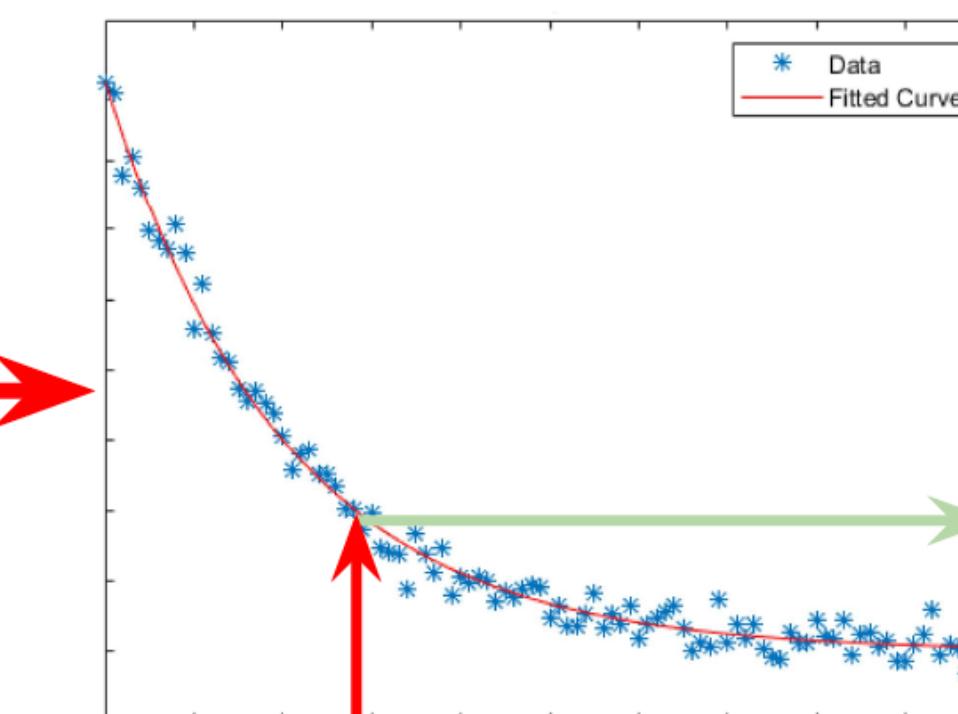
Neural Network *Predicting with Regression*

input
examples

output
examples

“here are some values you’ve
maybe never seen before”

[0.6, 0.2, 0.4]

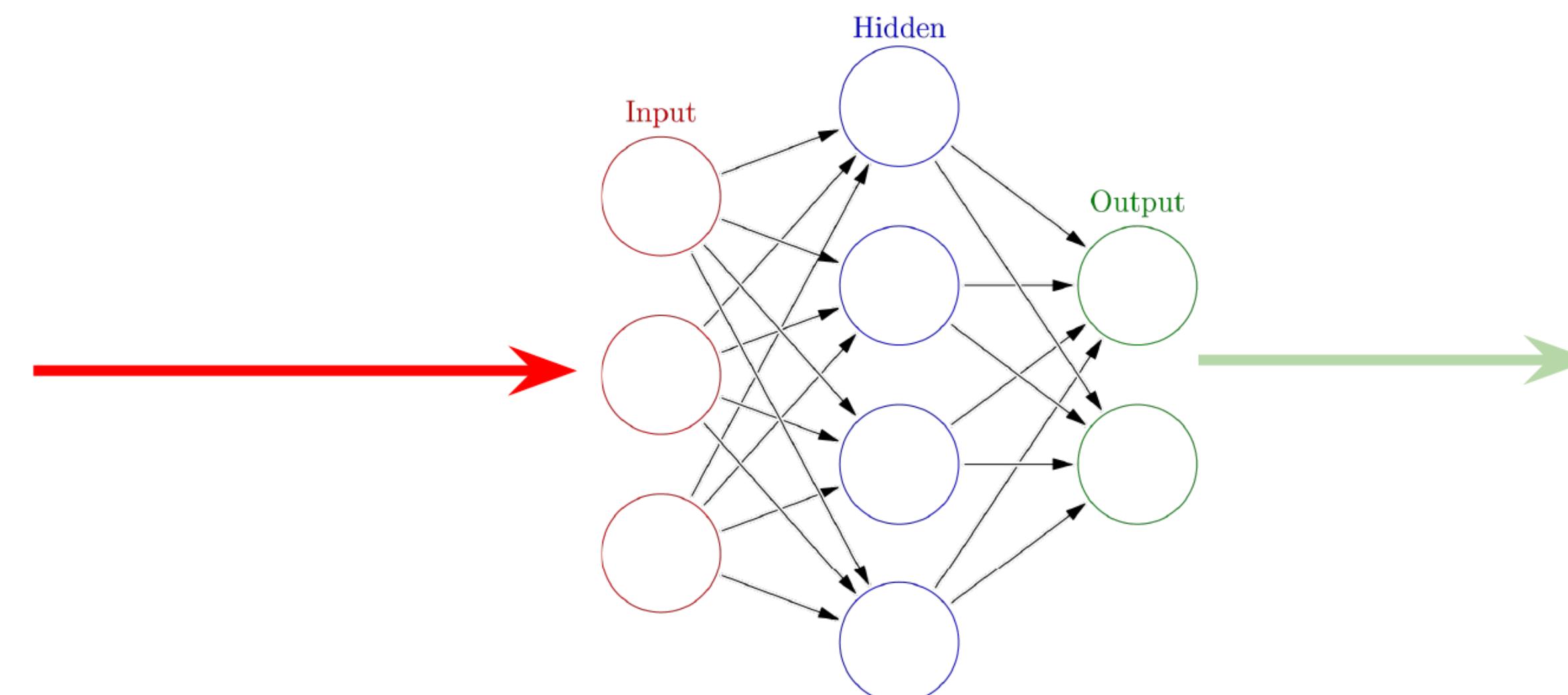


“here’s an output that would ‘fit’ in
your training data”

[0.9, 0.8]

Neural Network Predicting with Regression

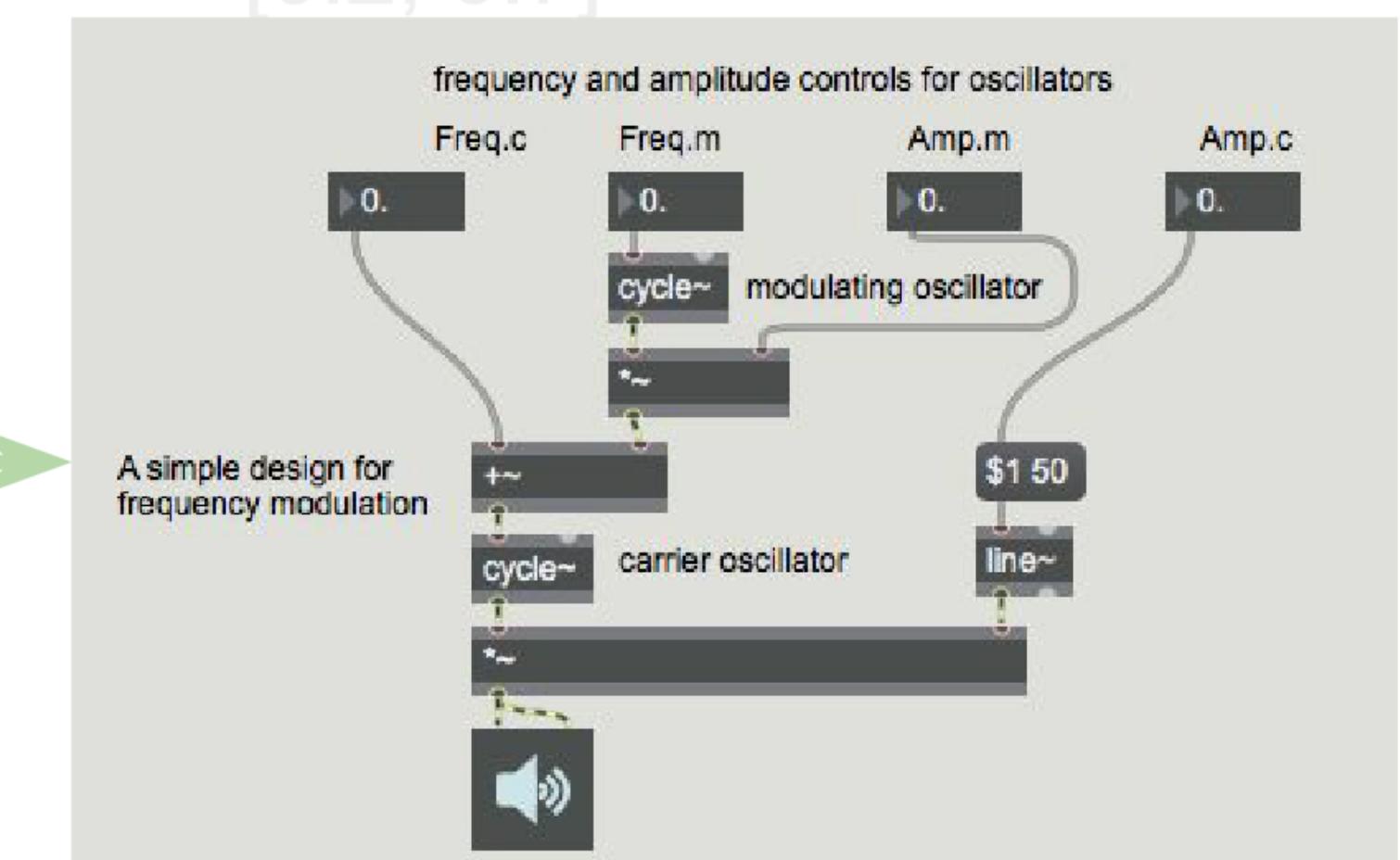
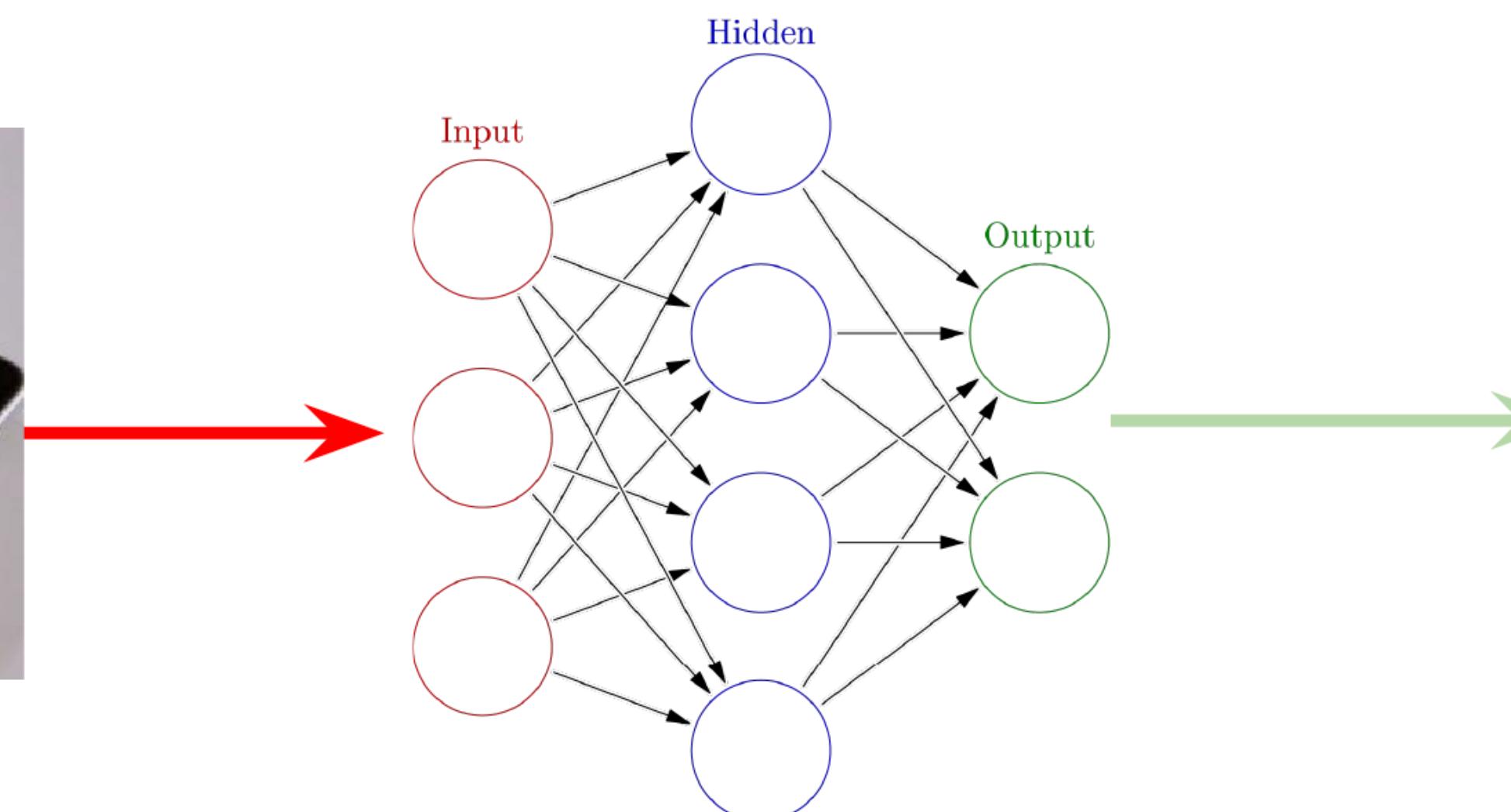
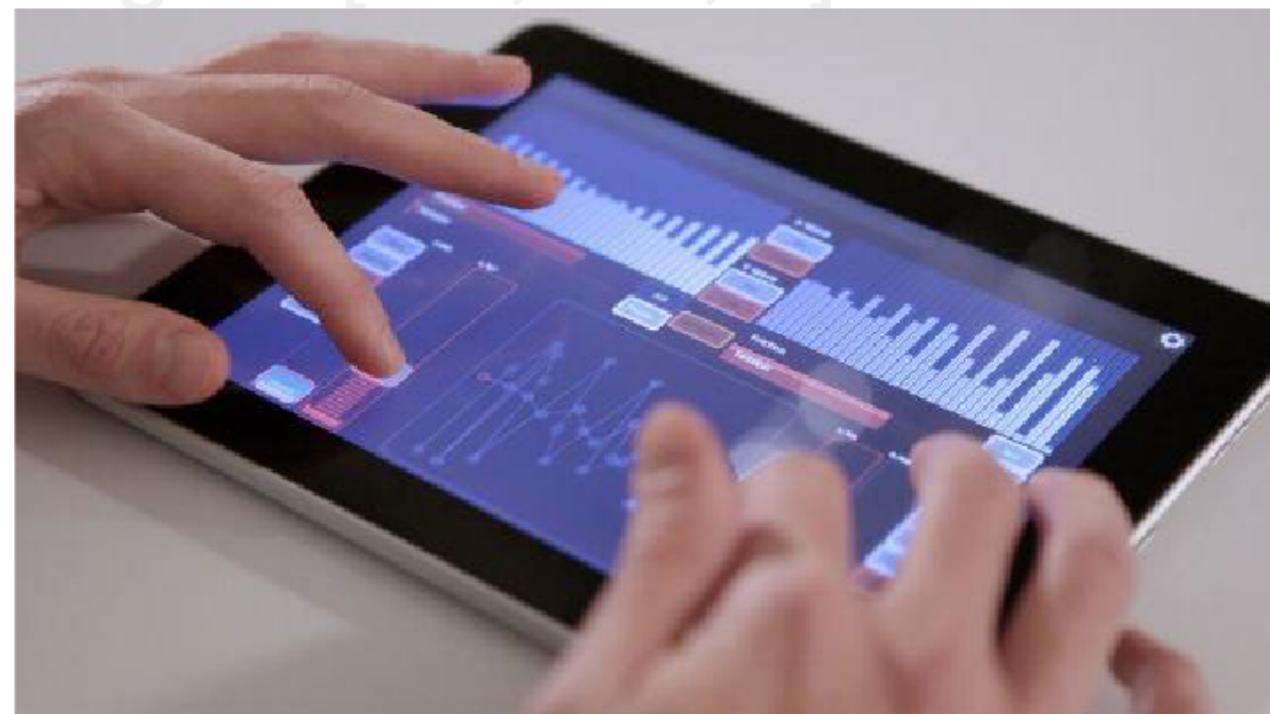
“a”: [0.1, 0.5, 1]
“b”: [0, 1, 1]
“c”: [1, 0, 1]
“d”: [0.2, 0, 1]
“e”: [0, 0, 0.5]
“f”: [0.8, 0.2, 1]
“g”: [0.9, 0.3, 0]
“h”: [0.4, 1, 0.7]
“i”: [0.2, 0.3, 1]
“j”: [0.1, 1, 0.1]
“k”: [0, 1, 1]
“l”: [0, 1, 1]
“m”: [0, 0, 1]
“n”: [0.4, 1, 0.7]
“o”: [0.2, 0.3, 1]
“p”: [0.1, 1, 0.1]
“q”: [0, 1, 1]
“r”: [0, 1, 1]



[0.5, 0.7]
[0.1, 0.1]
[0.5, 0.6]
[0.3, 0.9]
[0.4, 0]
[0.2, 0.7]
[0.5, 0.6]
[0.6, 0.1]
[0.5, 0.3]
[0.5, 0.7]
[0.1, 0.8]
[0.5, 0.1]
[0.4, 0.2]
[0.1, 0.8]
[0.5, 0.6]
[0.3, 0.9]
[0.4, 0.3]
[0.2, 0.7]

Neural Network Predicting with Regression

“a”: [0.1, 0.5, 1]
“b”: [0, 1, 1]
“c”: [1, 0, 1]
“d”: [0.2, 0, 1]
“e”: [0, 0, 0.5]
“f”: [0.8, 0.2, 1]
“g”: [0.9, 0.3, 0]

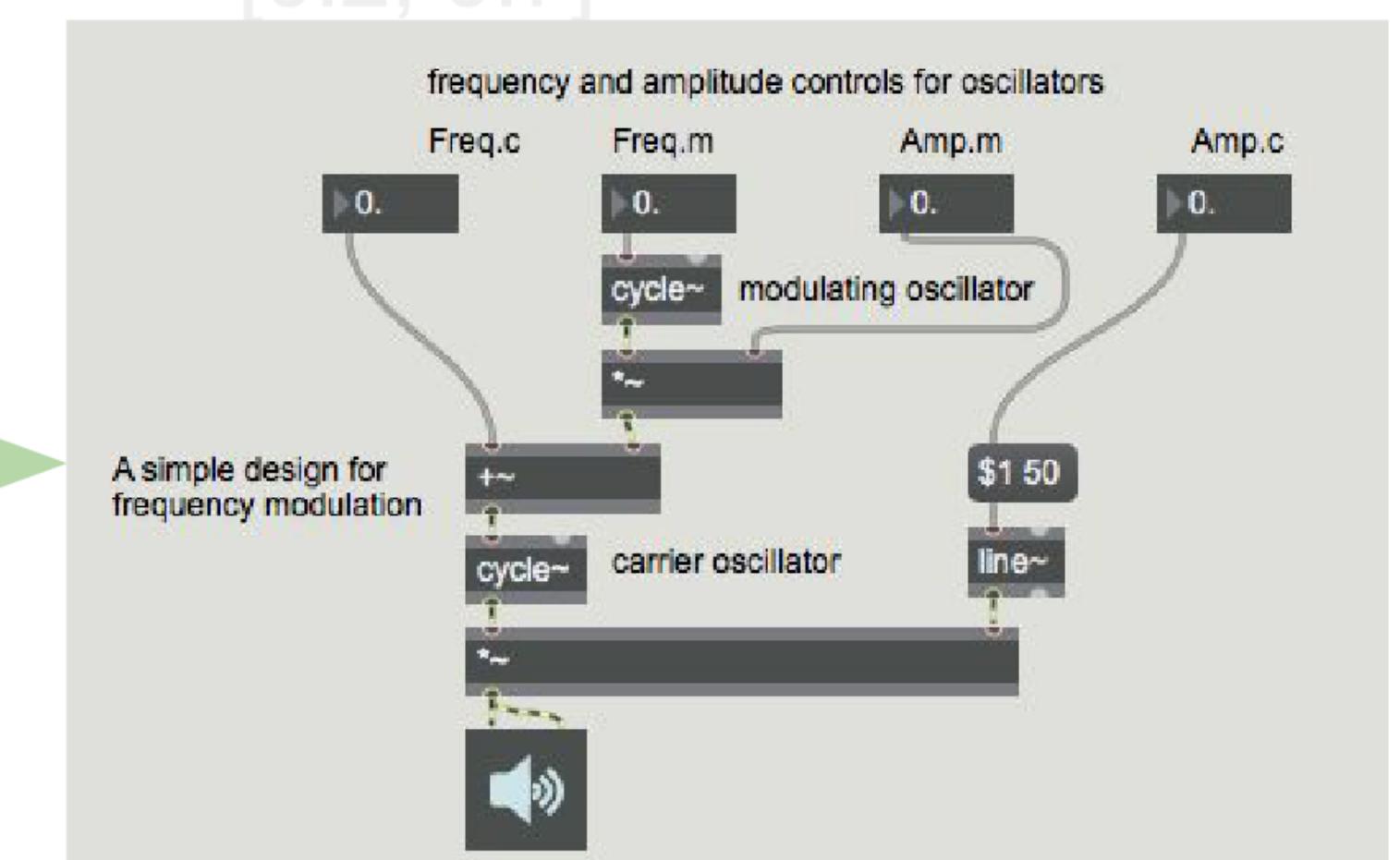
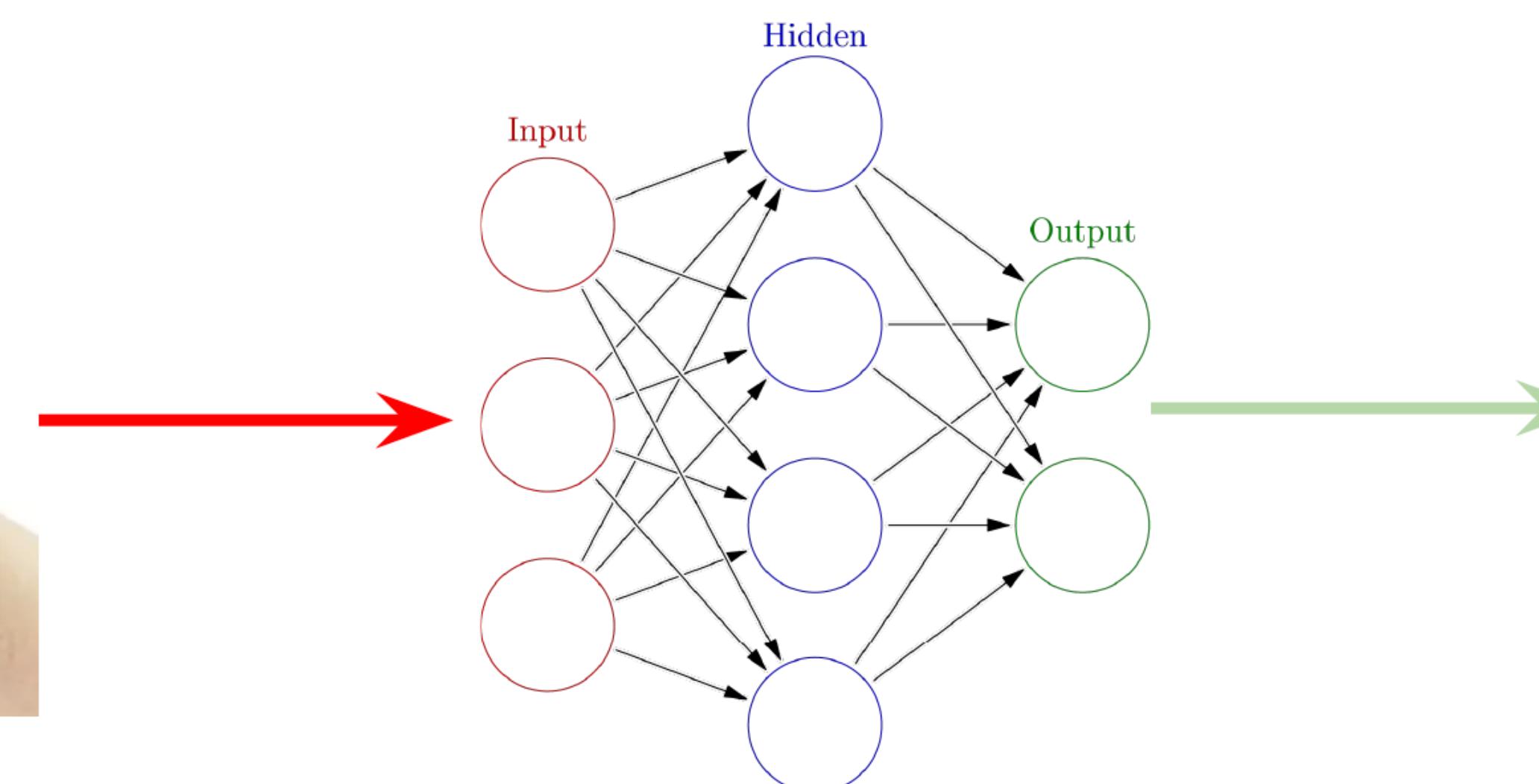
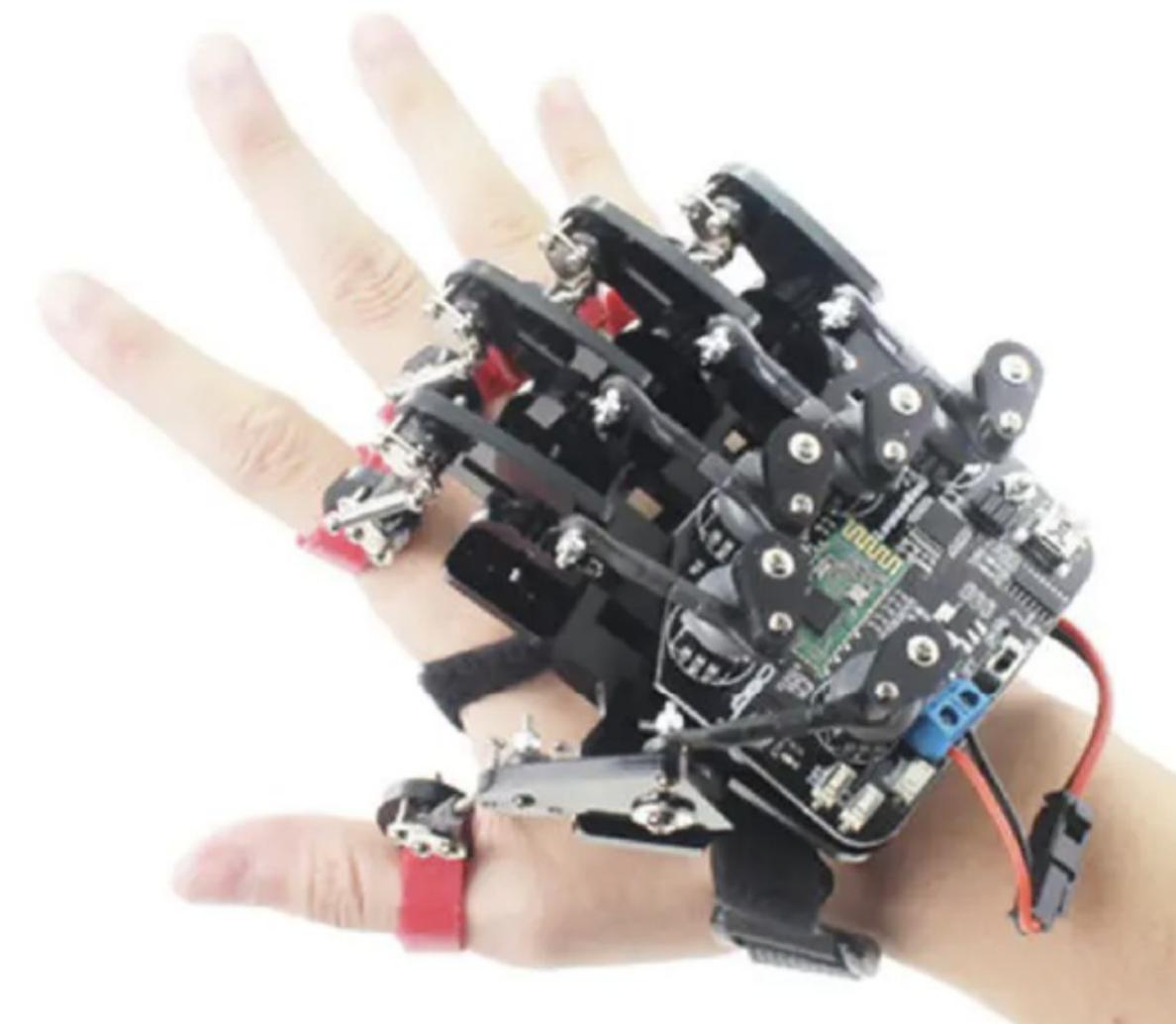


“n”: [0.4, 1, 0.7]
“o”: [0.2, 0.3, 1]
“p”: [0.1, 1, 0.1]
“q”: [0, 1, 1]
“r”: [0, 1, 1]

[0.5, 0.7]
[0.1, 0.1]
[0.5, 0.6]
[0.3, 0.9]
[0.4, 0]
[0.2, 0.7]

[0.5, 0.7]
[0.1, 0.8]
[0.5, 0.1]
[0.4, 0.2]
[0.1, 0.8]

Neural Network Predicting with Regression



Neural Network Predicting with Regression

“a”: [0.1, 0.5, 1]

“b”: [0, 1, 1]

“c”: [1, 0, 1]

“d”: [0.2, 0, 1]

“e”: [0, 0, 0.5]

“f”: [0.8, 0.2, 1]



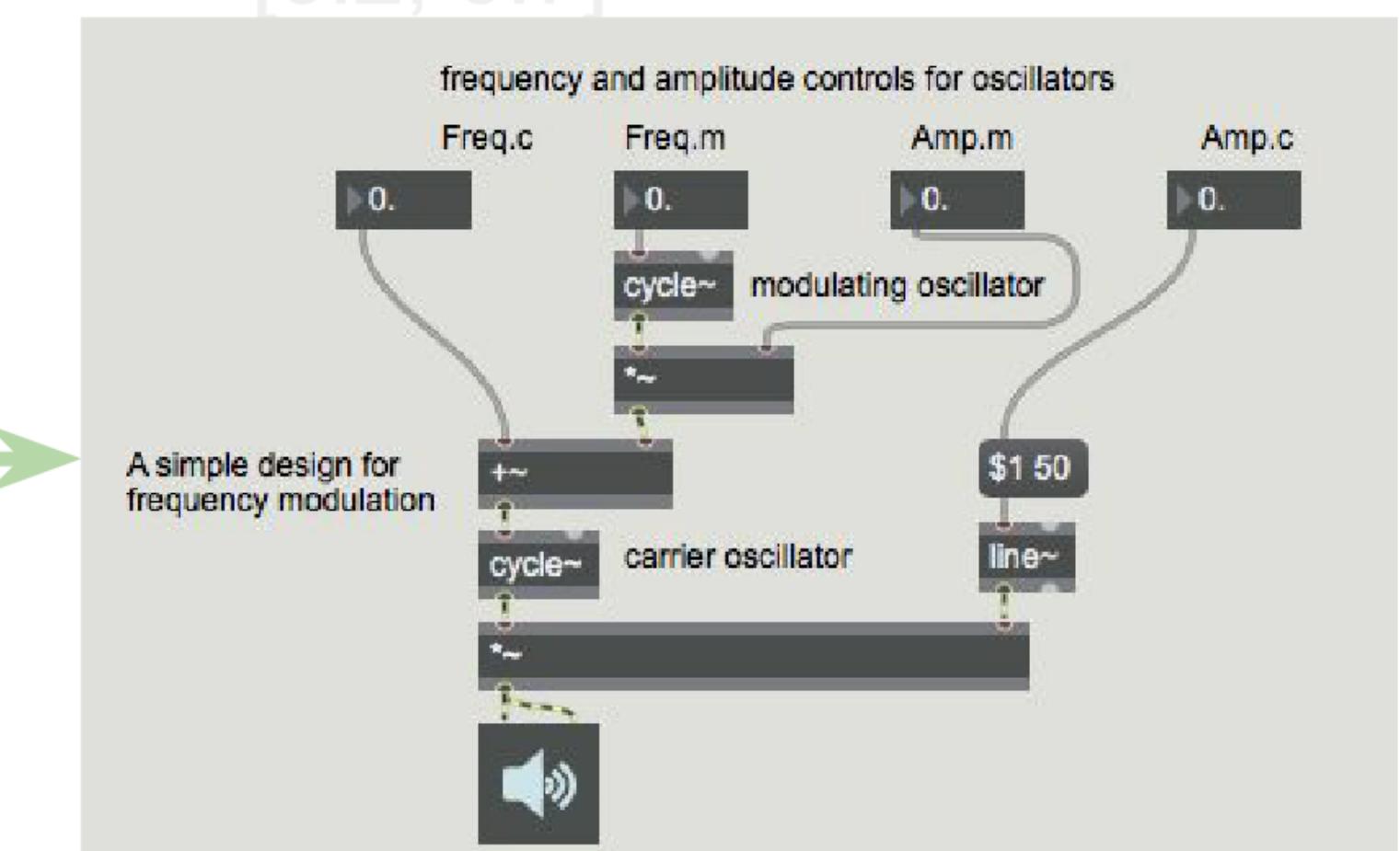
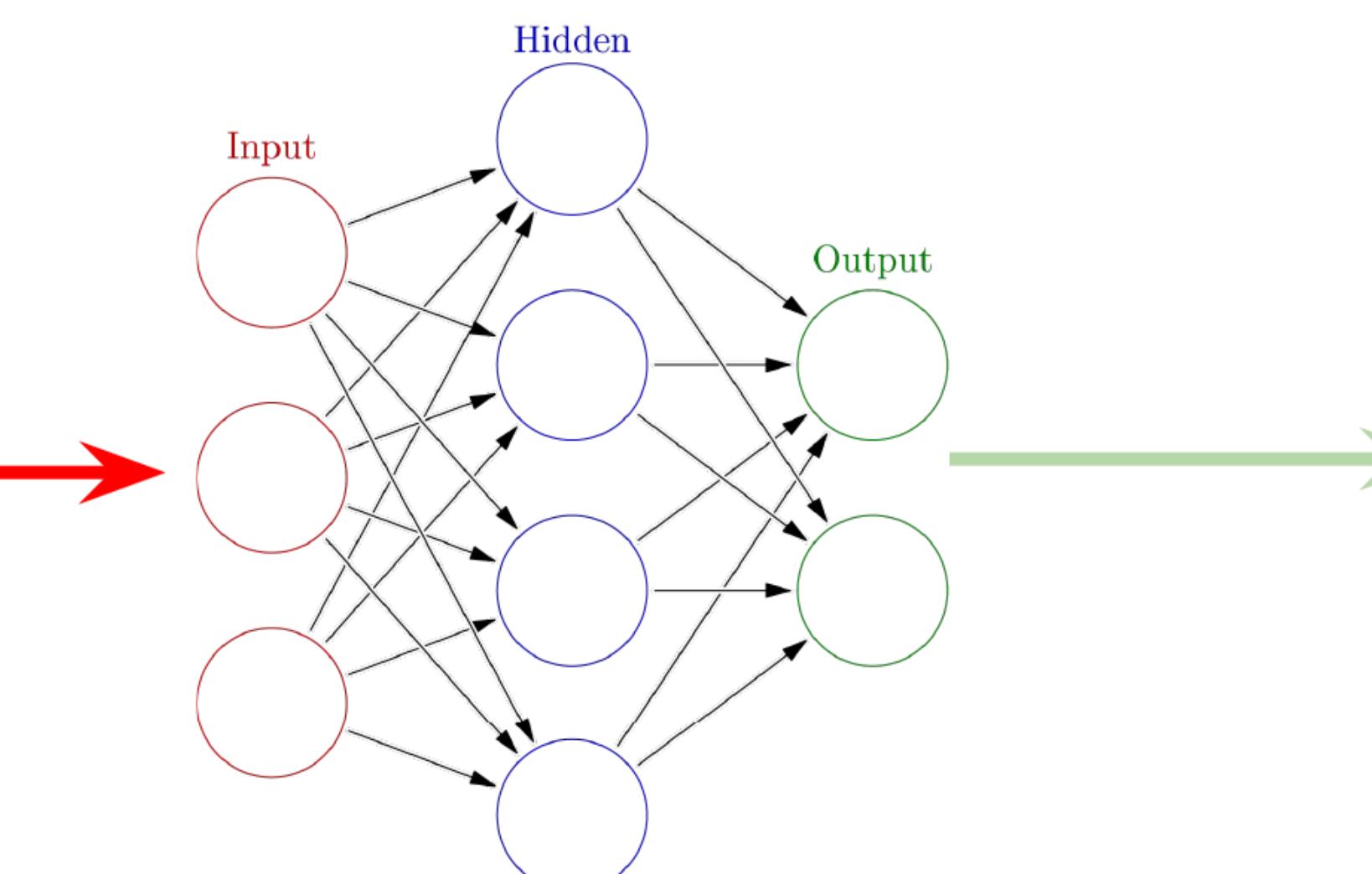
e.g.:

pitch

loudness

spectral centroid

spectral flatness



“g”: [0.1, 0.5, 1]

“h”: [0, 1, 1]

“i”: [1, 0, 1]

“j”: [0.2, 0, 1]

“k”: [0, 0, 0.5]

“l”: [0.8, 0.2, 1]

[0.5, 0.7]

[0.1, 0.1]

[0.5, 0.6]

[0.3, 0.9]

[0.4, 0]

[0.2, 0.7]

[0.5, 0.7]

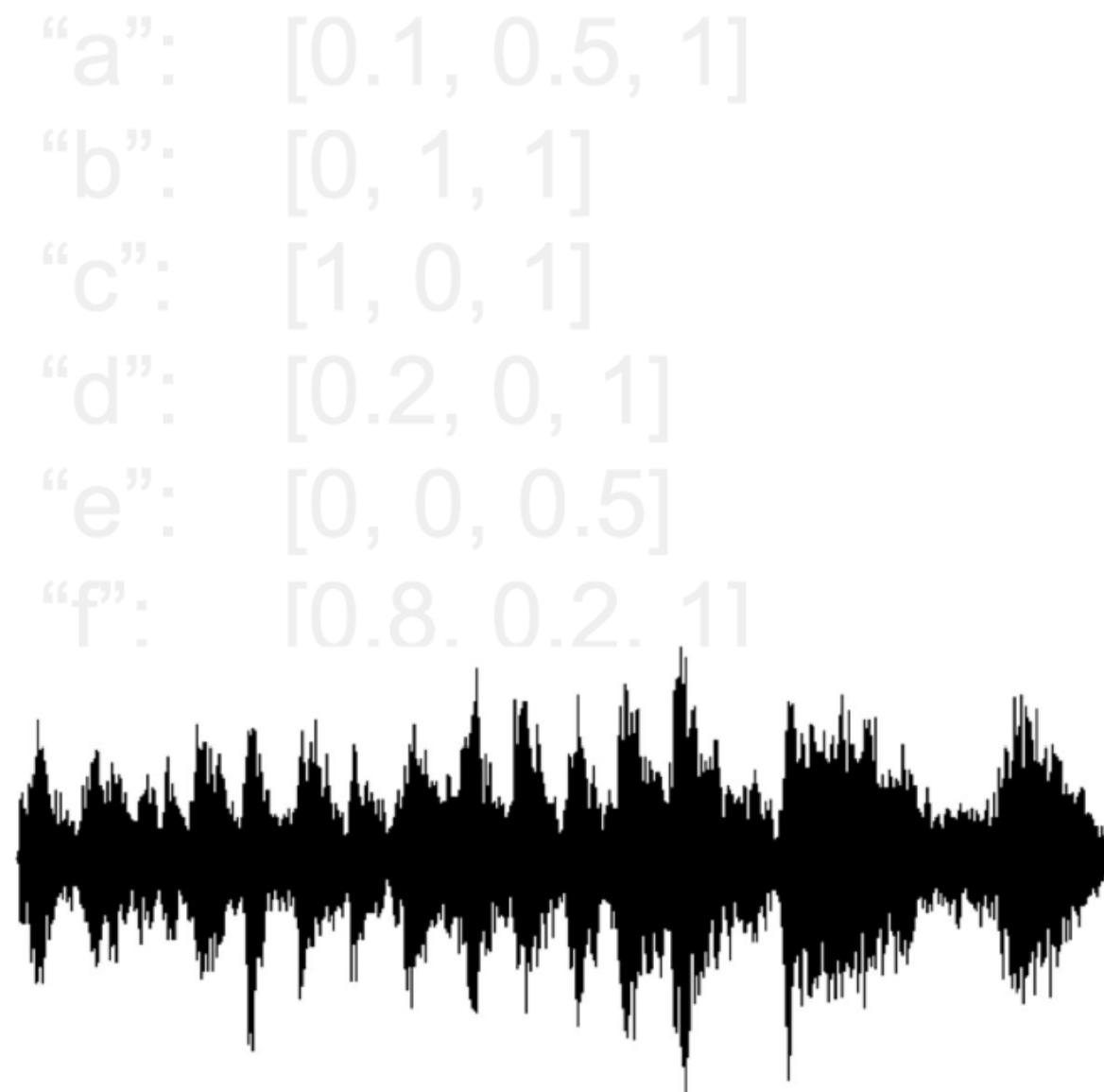
[0.1, 0.8]

[0.5, 0.1]

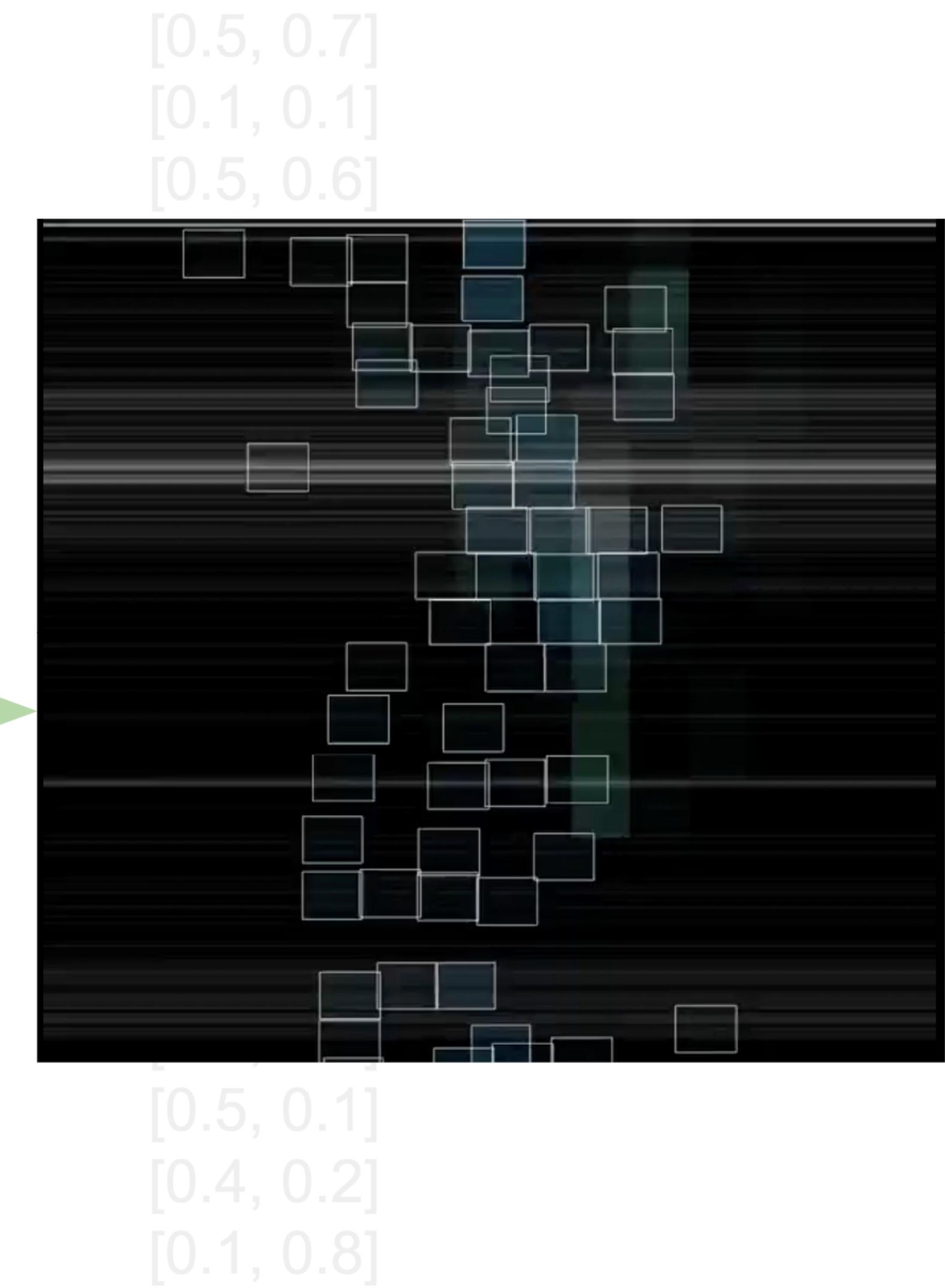
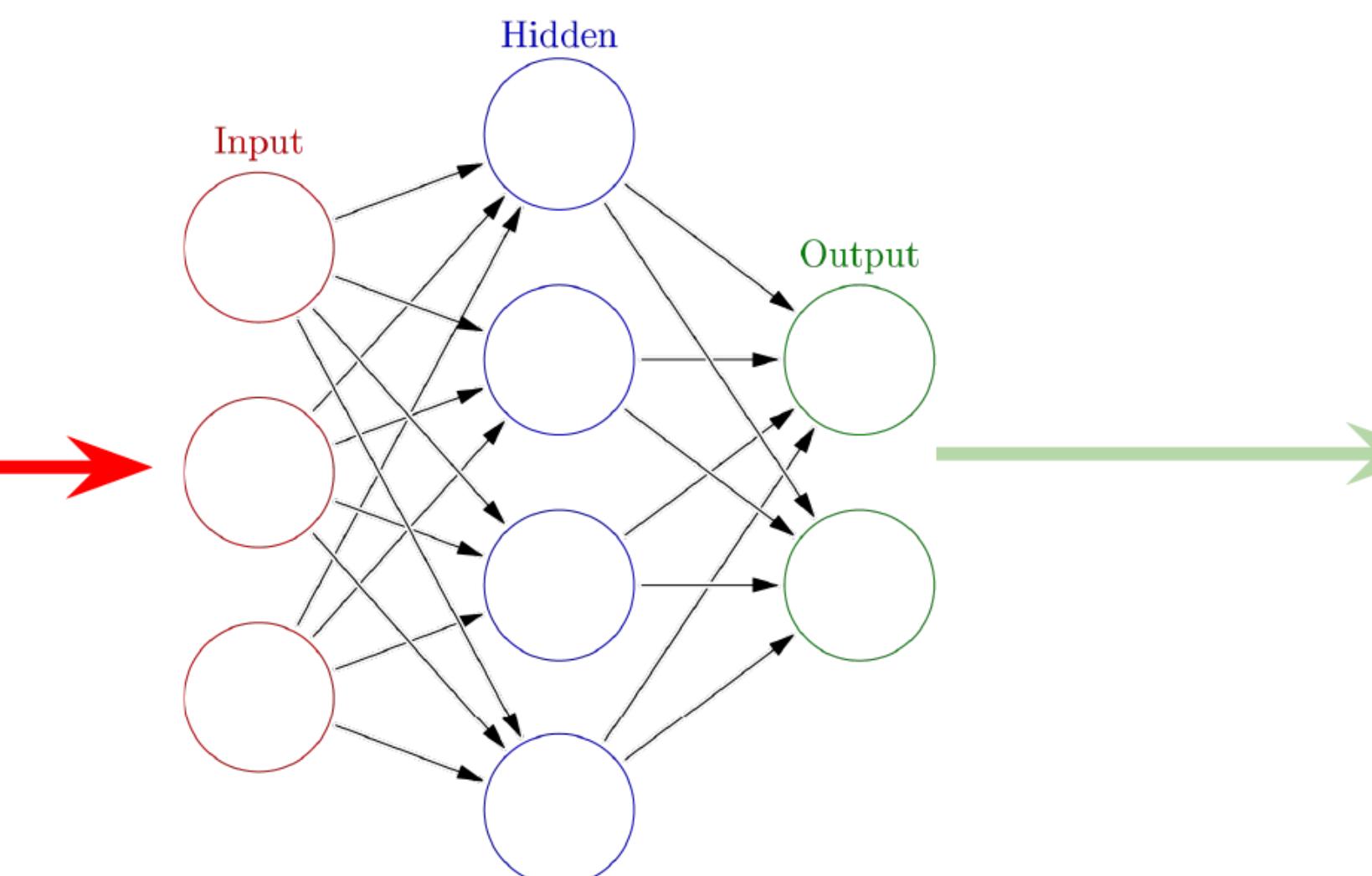
[0.4, 0.2]

[0.1, 0.8]

Neural Network Predicting with Regression

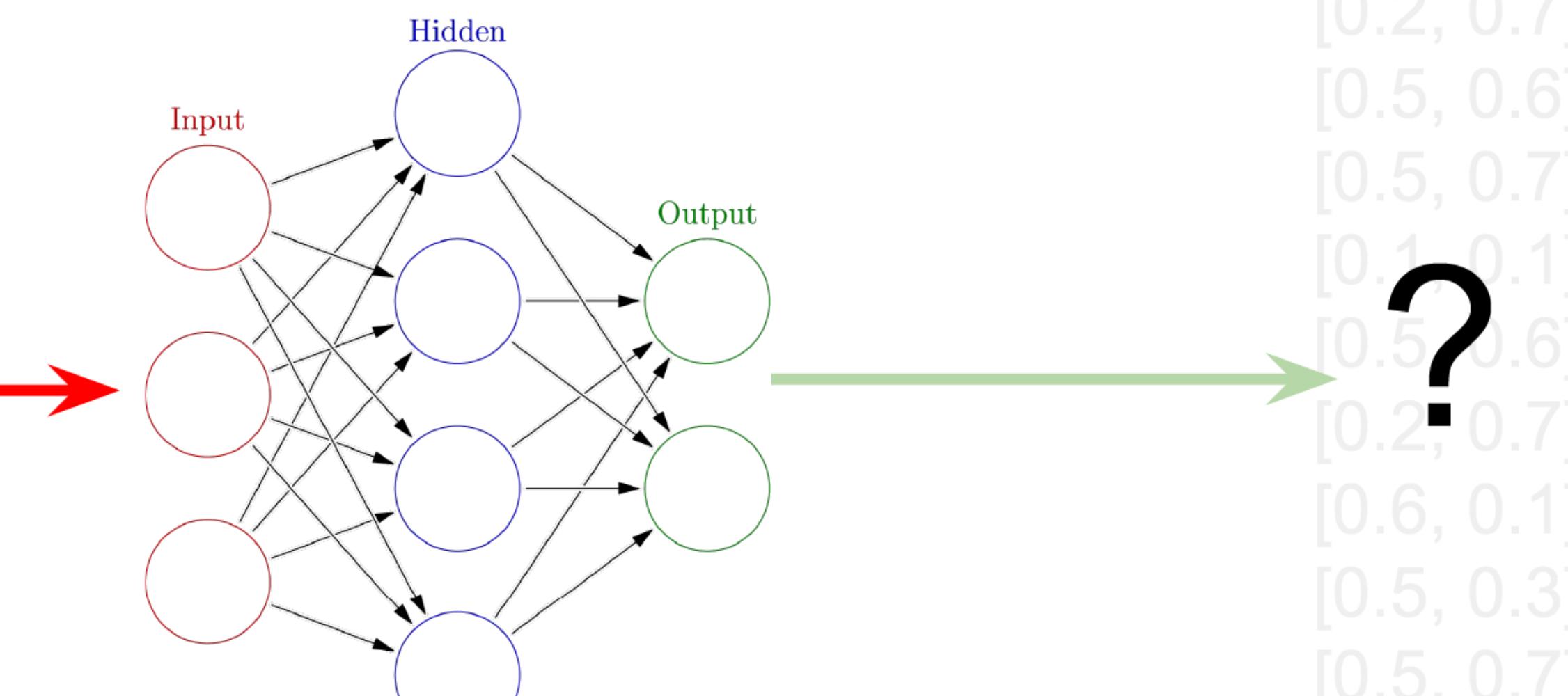


“a”: [0.1, 0.5, 1]
“b”: [0, 1, 1]
“c”: [1, 0, 1]
“d”: [0.2, 0, 1]
“e”: [0, 0, 0.5]
“f”: [0.8, 0.2, 1]
“j”: [0, 0, 0.5]
“k”: [0, 1, 1]
“l”: [0, 1, 1]
“m”: [0, 0, 1]
“n”: [0.4, 1, 0.7]
“o”: [0.2, 0.3, 1]
“p”: [0.1, 1, 0.1]
“q”: [0, 1, 1]
“r”: [0, 1, 1]



Neural Network Predicting with Regression

“a”: [0.1, 0.5, 1]
“b”: [0, 1, 1]
“c”: [1, 0, 1]
“d”: [0.2, 0, 1]
“e”: [0, 0, 0.5]
“f”: [0.8, 0.2, 1]
“g”: [0.9, 0.3, 0]
“h”: [0.4, 1, 0.7]
“i”: [0.2, 0.3, 1]
“j”: [0.1, 0.1, 0.1]
“k”: [0, 1, 1]
“l”: [0, 1, 1]
“m”: [0, 0, 1]
“n”: [0.4, 1, 0.7]
“o”: [0.2, 0.3, 1]
“p”: [0.1, 1, 0.1]
“q”: [0, 1, 1]
“r”: [0, 1, 1]



[0.5, 0.7]
[0.1, 0.1]
[0.5, 0.6]
[0.3, 0.9]
[0.4, 0]
[0.2, 0.7]
[0.5, 0.6]
[0.5, 0.7]
[0.1, 0.1]
[0.5, 0.6]
[0.2, 0.7]
[0.6, 0.1]
[0.5, 0.3]
[0.5, 0.7]
[0.1, 0.8]
[0.5, 0.1]
[0.4, 0.2]
[0.1, 0.8]

There are certain things that we care about, as musicians for example, that are really hard to articulate in code. It's hard for me to talk about what kind of quality of sound I want and then translate that into a set of filter coefficients. It's hard for me to talk about how I want a performer to move on stage and then translate that into some sort of mathematical equation for their trajectory. But it's a lot easier for me to either find examples of sounds that have a particular quality or to give examples of movements or if I'm using other types of modalities, often curating or creating examples are just way easier for us as people. And this relates to the types of tacit knowledge and embodied knowledge we bring to creative practices.

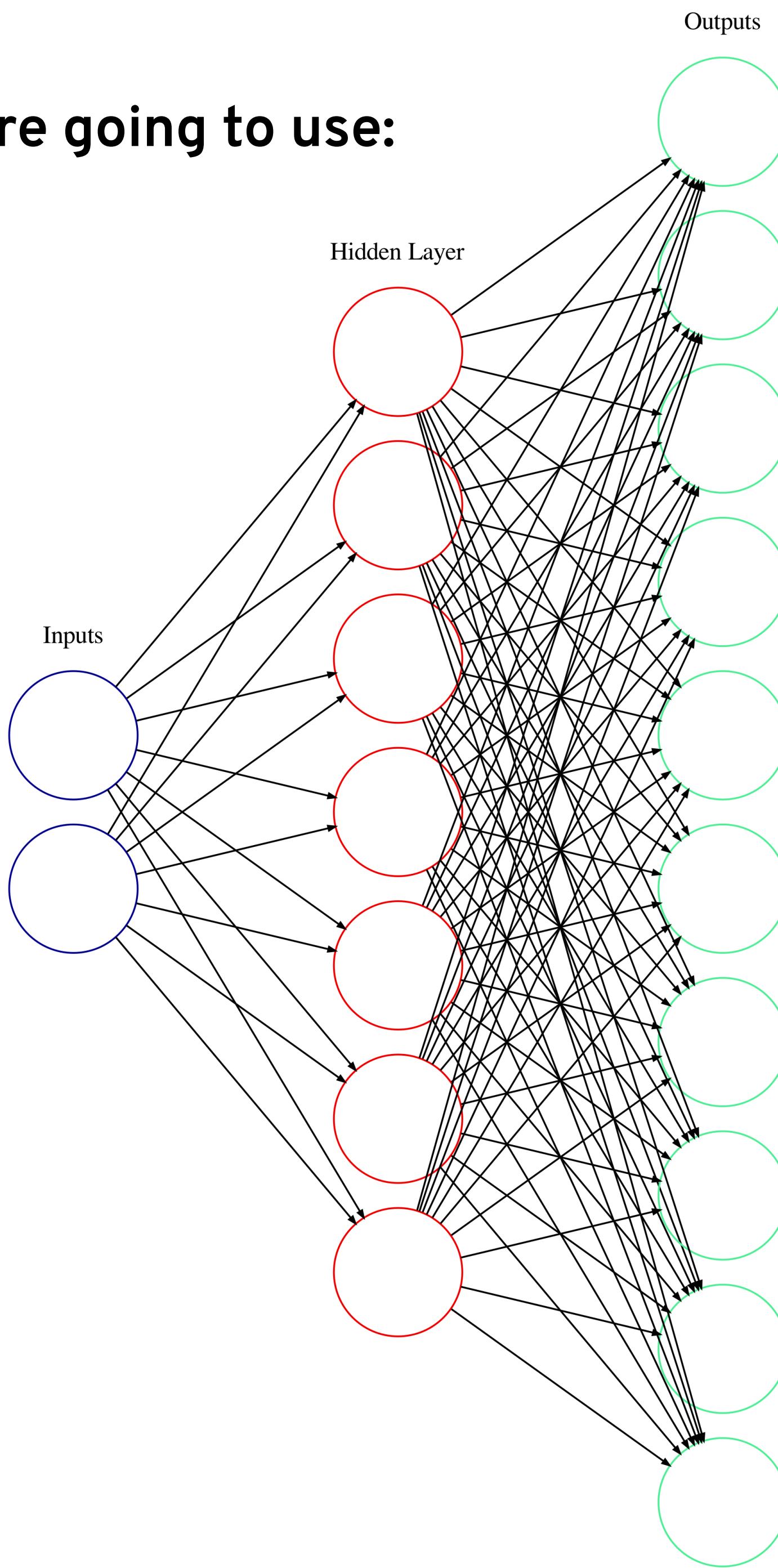
-Rebecca Fiebrink

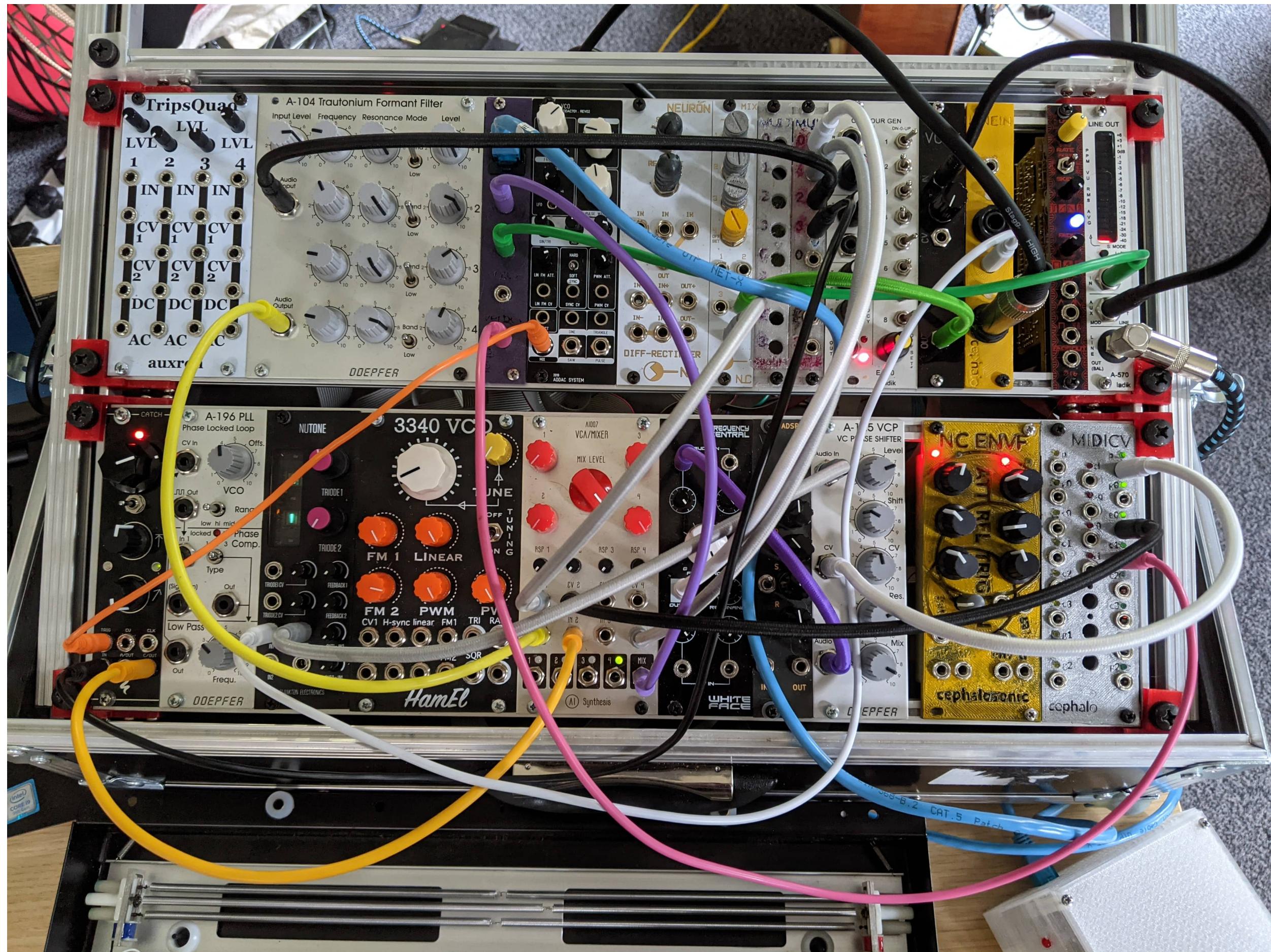
Structure of the neural network we're going to use:

2 inputs

1 hidden layer of 7 nodes

10 outputs





FeedbackFeedforward

by Alice Eldridge and
Chris Kiefer

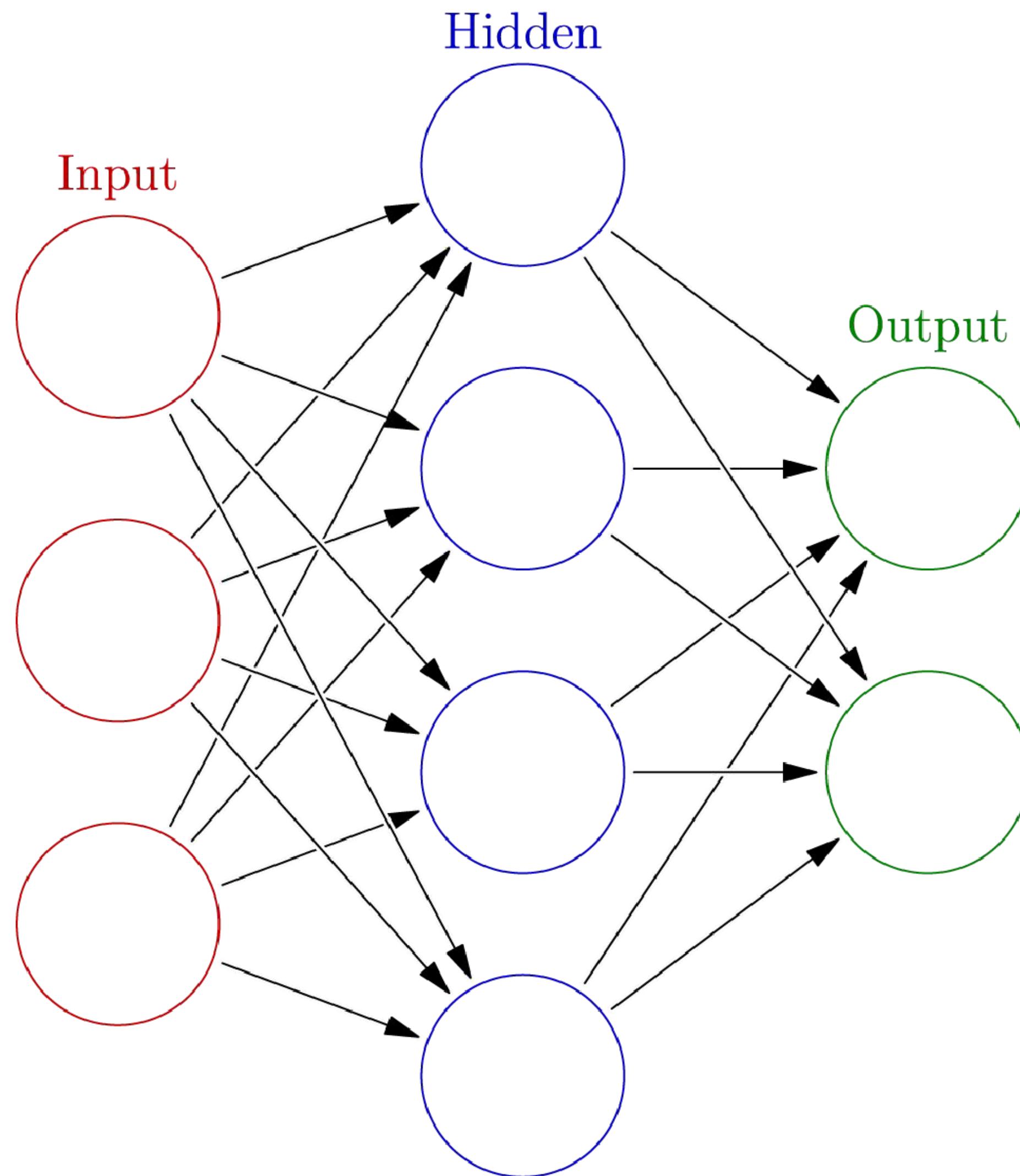
learn.flucoma.org/explore/eldridge-kiefer

FluidMLPClassifier

classify sounds by
timbre in real-time



Neural Network (Multi-Layer Perceptron)

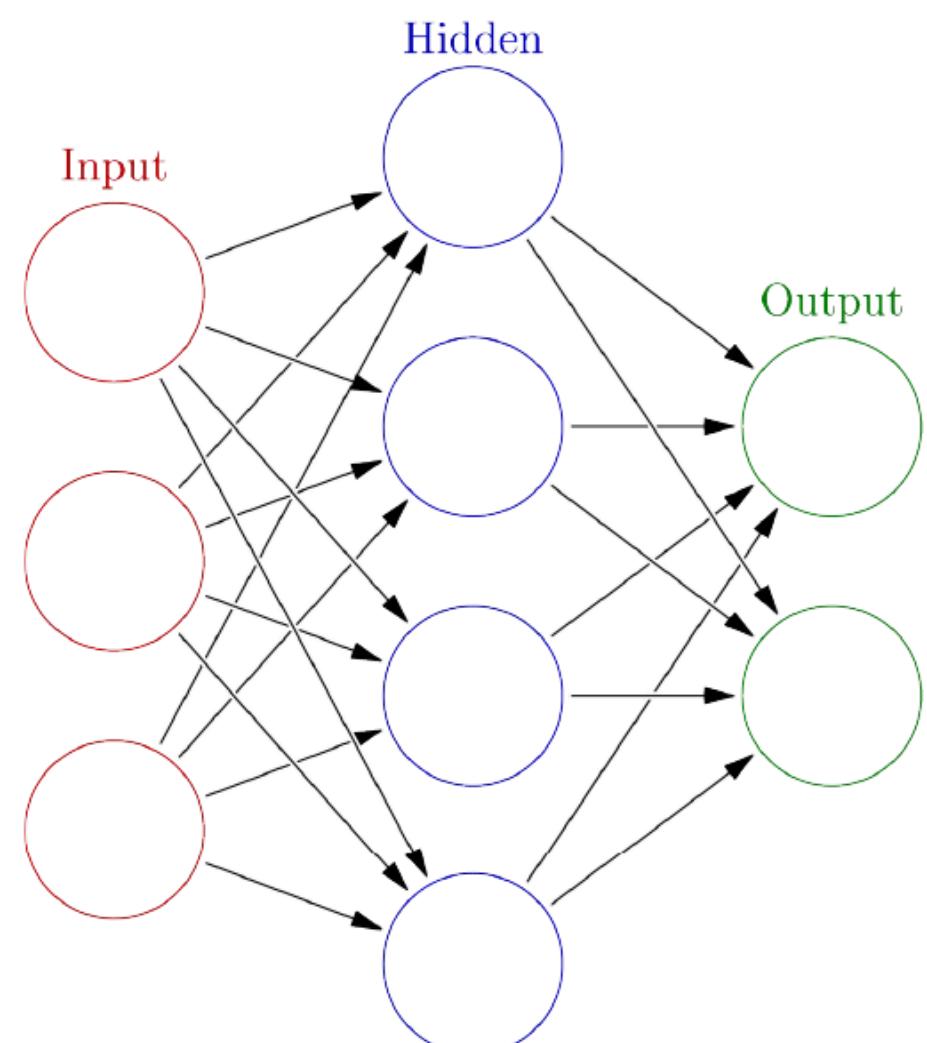


Classification:
a neural network predicts
which category (or “class”)
an input belongs to

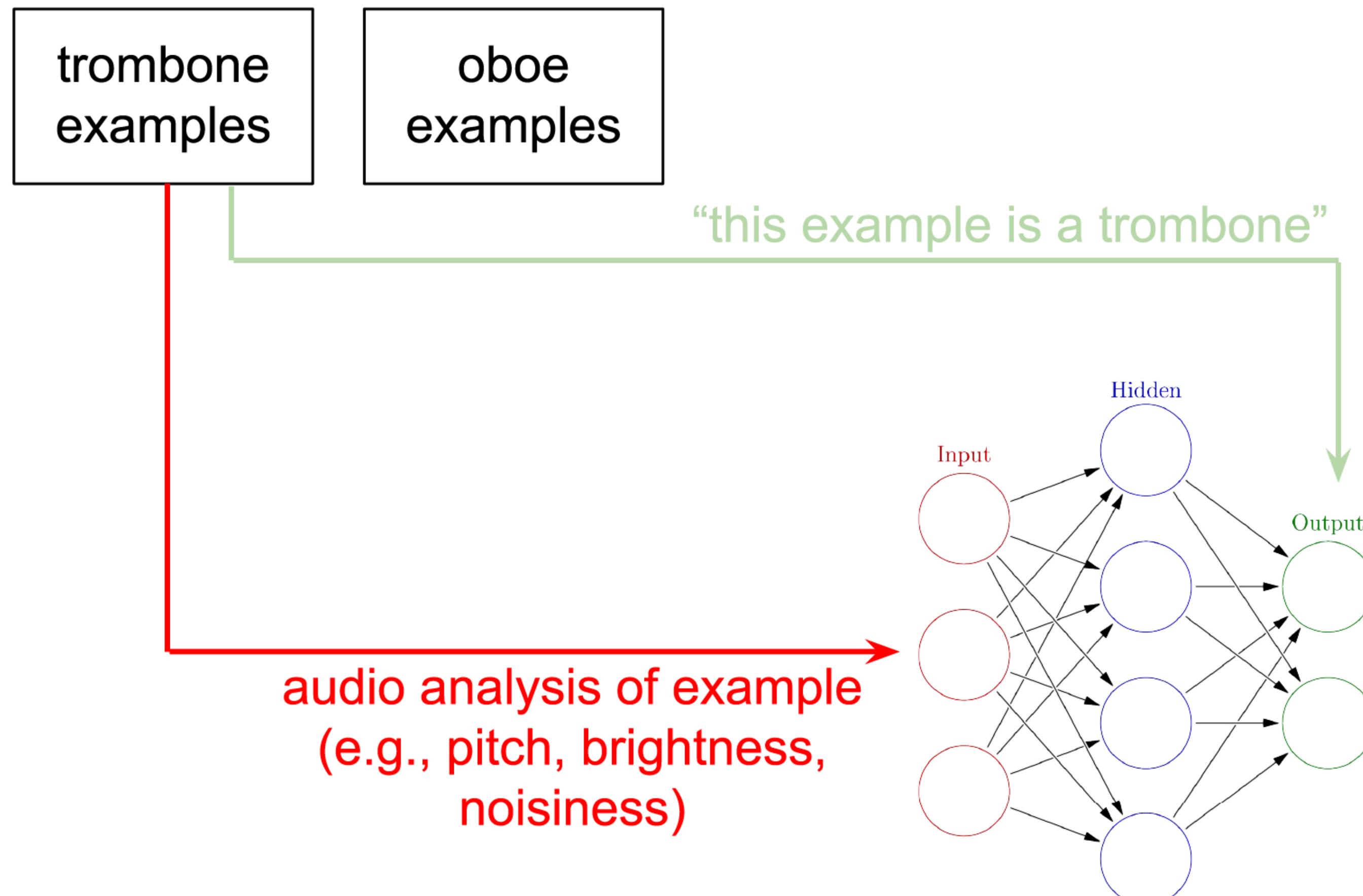
Neural Network ***Training a Classifier***

trombone
examples

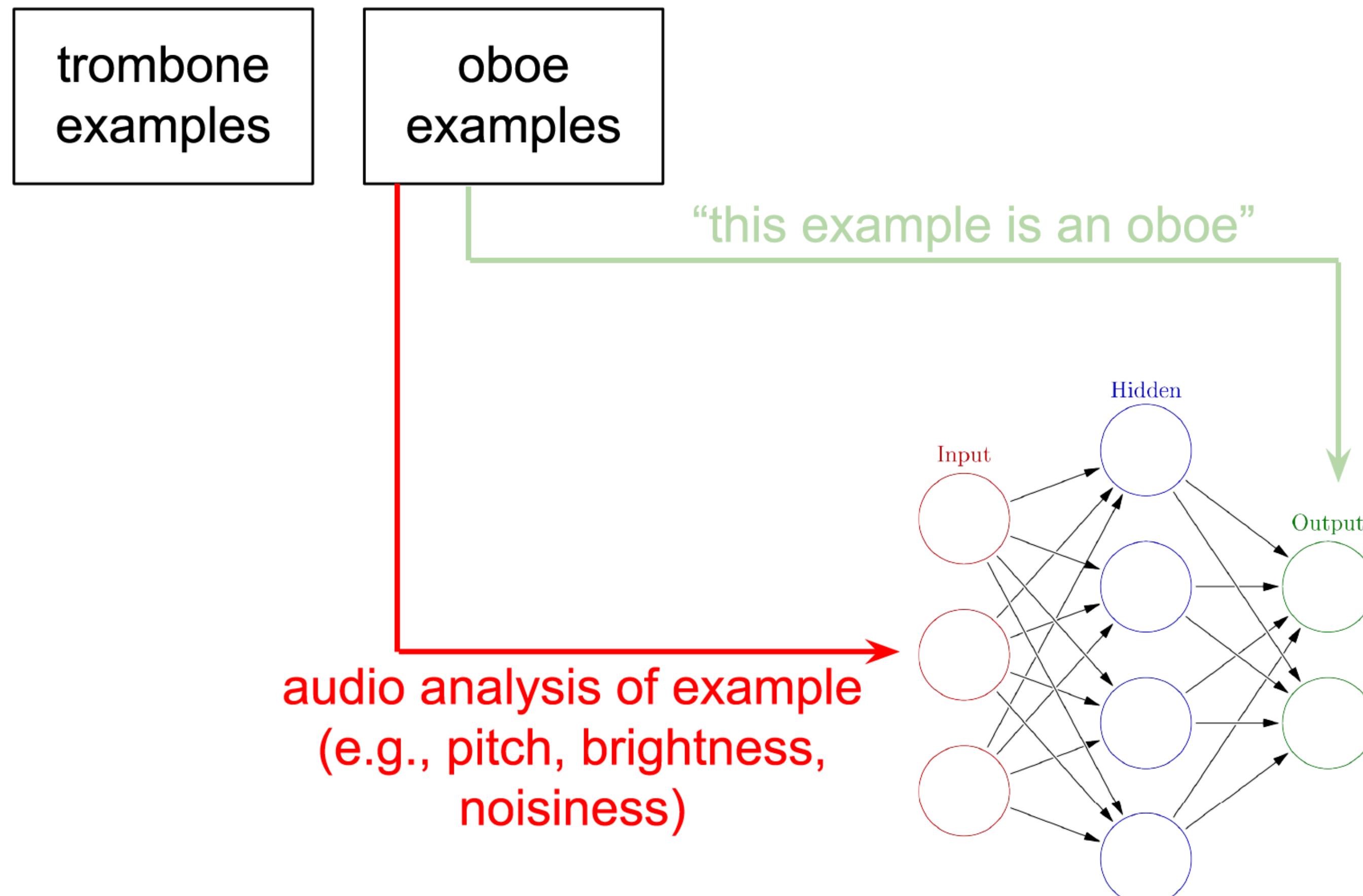
oboe
examples



Neural Network **Training a Classifier**



Neural Network **Training a Classifier**



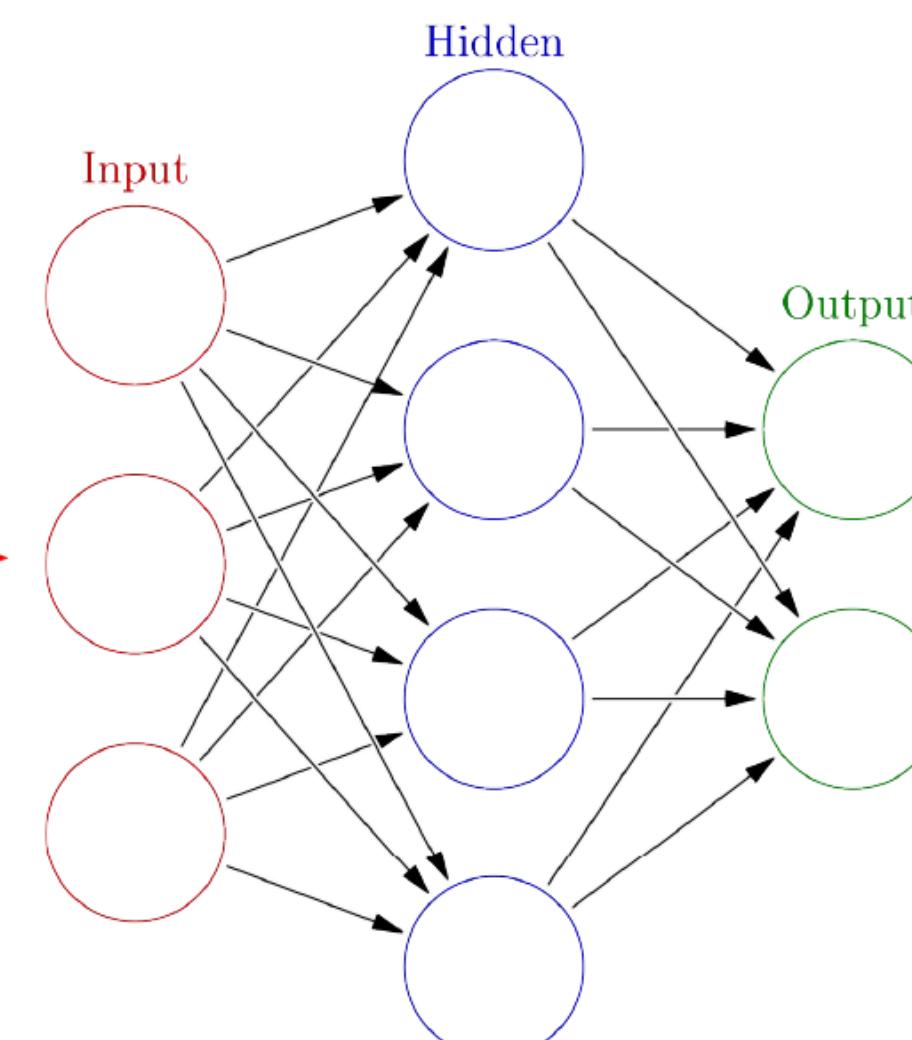
Neural Network Predicting a Classification

trombone
examples

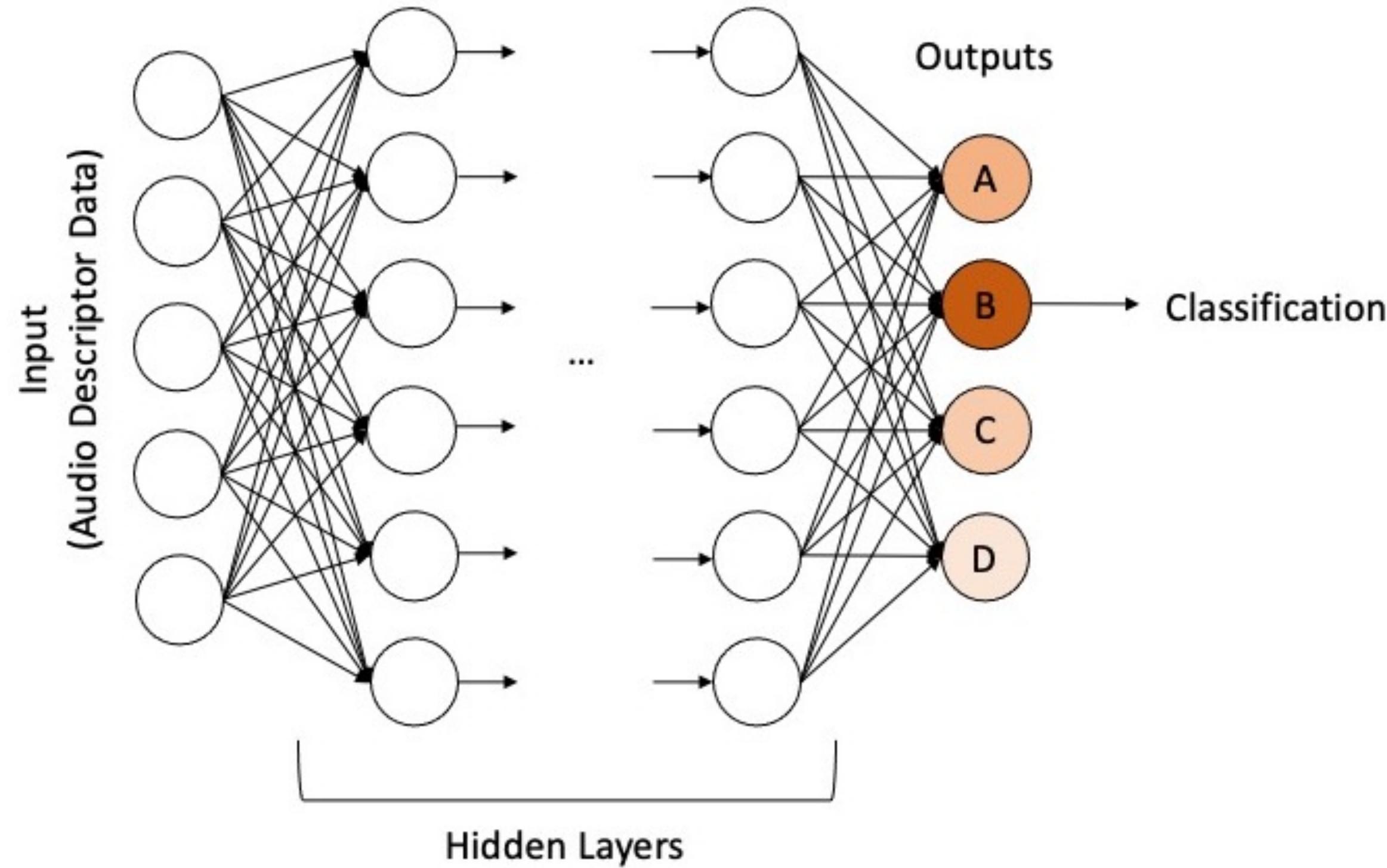
oboe
examples

new example it has
never seen before

audio analysis of example
(e.g., pitch, brightness,
noisiness)



“this new example is most
like the oboe examples you
showed me before”
(or trombone...)



Drift Shadow by Alex Harker

learn.flucoma.org/explore/harker

quartet



Using FluCoMa *a workflow*

Artistic Hypothesis:
reorganize a corpus of sounds
according to their similarity

Using FluCoMa a workflow

1. corpus curation

2. decomposition

3. analysis

4. pattern finding

5. creating

The image shows a musical score with three staves:

- Perc.** (Percussion): This staff contains four measures. Measure 1: Shell chimes (three strokes). Measure 2: Maraca swirl (two strokes), followed by free rebound (one stroke). Measure 3: Free rebound (one stroke), followed by floor tom (three strokes). Measure 4: Floor tom (three strokes).
- Pno.** (Piano): This staff contains five measures. Measure 1: Aggressive (slurs over two octaves). Measure 2: ff < fff. Measure 3: f < fff. Measure 4: f < ff f. Measure 5: fff (pedal on bass notes).
- Vc.** (Cello/Bass): This staff contains six measures. Measure 1: ff mf. Measure 2: f < sffz. Measure 3: mfp (with a dynamic bracket below). Measure 4: m.s.p. (marked with a circled '3'). Measure 5: sfz mp. Measure 6: f. Measure 7: sfzp.

Each measure includes dynamic markings such as *f*, *mp*, *ff*, *fff*, *mf*, *sfz*, *ord. jeté*, and *m.s.p.*. Articulation marks like slurs, dashes, and dots are also present. Measures 1-3 for the piano and 1-4 for the cello/bass include performance instructions like "aggressive" and "pedal on bass notes". Measures 4-5 for the piano and 5-7 for the cello/bass include performance instructions like "ff" and "sfzp". Measures 1-3 for the cello/bass include performance instructions like "ff" and "sfz". Measures 4-5 for the cello/bass include performance instructions like "ord. jeté" and "m.s.p.". Measures 6-7 for the cello/bass include performance instructions like "f" and "sfzp".

Using FluCoMa *a workflow*

1. corpus curation



2. decomposition

3. analysis

4. pattern finding

5. creating



Using FluCoMa a workflow

1. corpus curation



2. decomposition

3. analysis

4. pattern finding

5. creating



Using FluCoMa a workflow

1. corpus curation

2. decomposition

3. analysis

4. pattern finding

5. creating



	spec cent	spec spread	spec skewness	spec kurtosis	spec rolloff	spec flatness	spec crest	loudness	true peak	pitch	pitch conf	mfcc 0	mfcc 1	mfcc 2	mfcc 3	mfcc 4	mfcc 5	mfcc 6
	0.276	0.362	0.891	0.135	0.772	0.843	0.869	0.372	0.523	0.041	0.053	0.270	0.103	0.759	0.370	0.901	0.689	0.698
	0.854	0.266	0.789	0.708	0.889	0.041	0.445	0.236	0.661	0.819	0.704	0.307	0.753	0.177	0.399	0.295	0.789	0.696
	0.566	0.366	0.772	0.133	0.581	0.904	0.203	0.553	0.838	0.697	0.659	0.981	0.072	0.503	0.588	0.964	0.764	0.982
	0.448	0.466	0.153	0.581	0.679	0.332	0.450	0.579	0.479	0.833	0.781	0.077	0.702	0.376	0.749	0.608	0.863	0.348
	0.854	0.520	0.015	0.768	0.278	0.359	0.350	0.221	0.629	0.387	0.047	0.703	0.631	0.438	0.170	0.019	0.097	0.973
	0.858	0.220	0.727	0.604	0.854	0.978	0.976	0.252	0.394	0.858	0.223	0.148	0.729	0.308	0.341	0.265	0.026	0.737
	0.661	0.076	0.065	0.848	0.990	0.976	0.249	0.899	0.557	0.276	0.458	0.186	0.995	0.699	0.421	0.621	0.120	0.635
	0.159	0.331	0.874	0.059	0.834	0.650	0.889	0.635	0.714	0.752	0.182	0.820	0.477	0.559	0.305	0.700	0.744	0.859
	0.057	0.032	0.334	0.208	0.842	0.438	0.010	0.789	0.148	0.471	0.791	0.308	0.381	0.332	0.026	0.379	0.370	0.282
	0.427	0.304	0.637	0.269	0.560	0.810	0.455	0.869	0.828	0.397	0.149	0.055	0.259	0.758	0.203	0.584	0.401	0.770
	0.355	0.180	0.043	0.989	0.167	0.263	0.600	0.114	0.332	0.839	0.121	0.611	0.303	0.180	0.142	0.976	0.117	0.163
	0.400	0.348	0.094	0.837	0.286	0.872	0.850	0.817	0.231	0.567	0.965	0.603	0.669	0.941	0.242	0.859	0.593	0.838
	0.941	0.817	0.018	0.783	0.094	0.661	0.744	0.473	0.449	0.311	0.647	0.861	0.496	0.328	0.340	0.395	0.107	0.439
	0.075	0.817	0.177	0.187	0.096	0.990	0.179	0.605	0.403	0.359	0.828	0.394	0.723	0.456	0.701	0.356	0.496	0.779
	0.669	0.216	0.782	0.567	0.032	0.263	0.951	0.412	0.293	0.415	0.546	0.175	0.094	0.641	0.432	0.166	0.119	0.897

Using FluCoMa *a workflow*

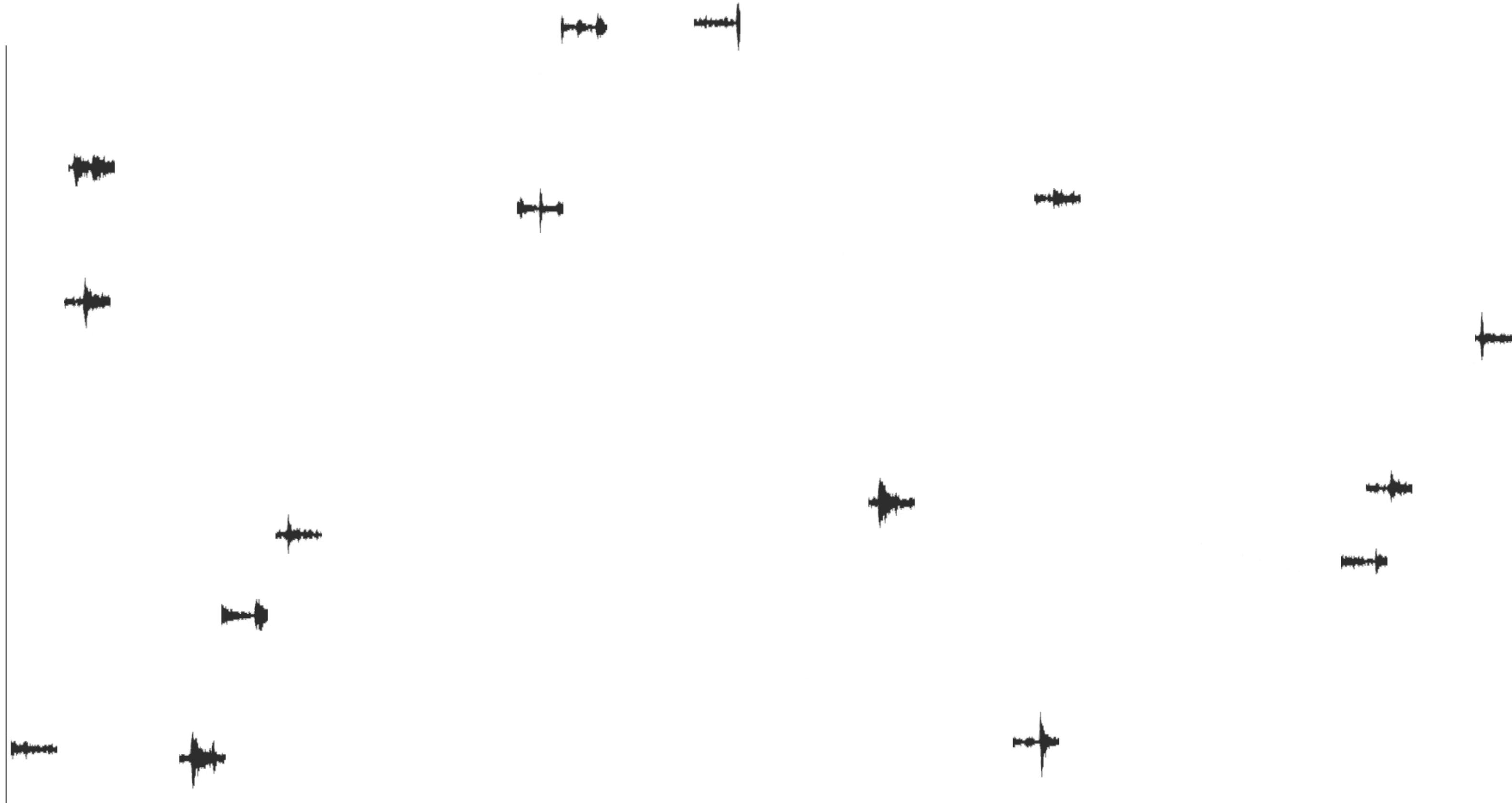
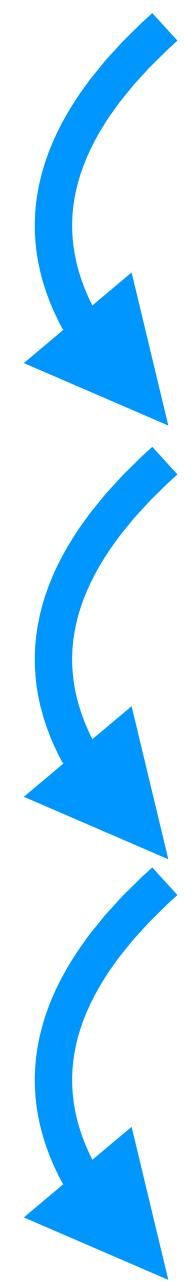
1. corpus curation

2. decomposition

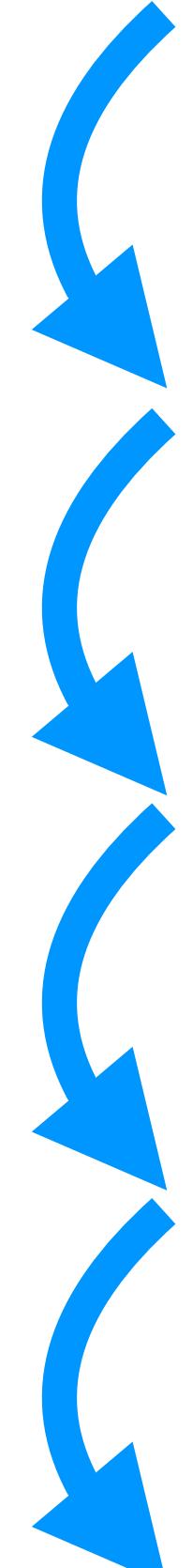
3. analysis

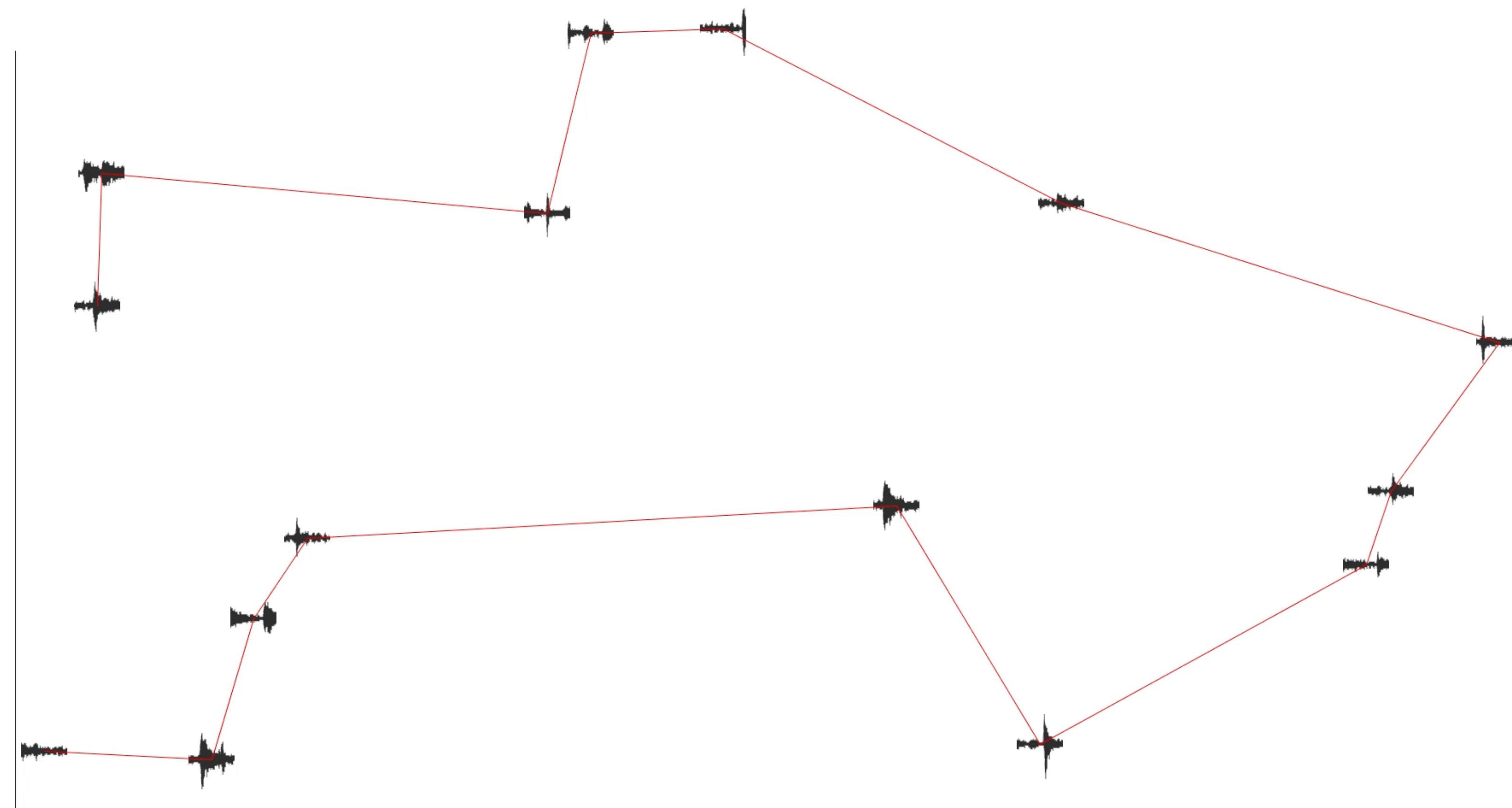
4. pattern finding

5. creating

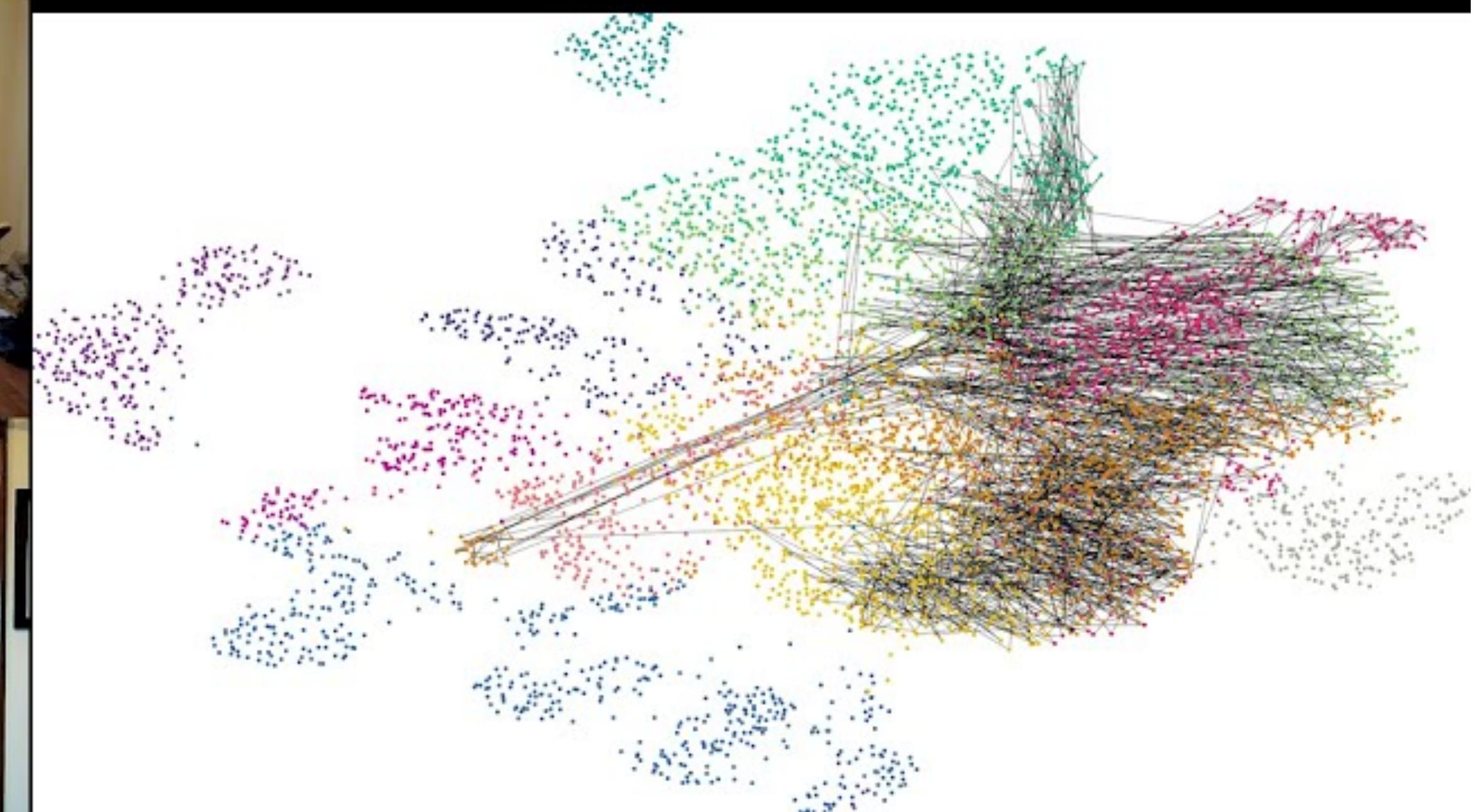


Using FluCoMa *a workflow*

1. corpus curation
 2. decomposition
 3. analysis
 4. pattern finding
 5. creating
- 



Traveling Salesperson Route: UMAP

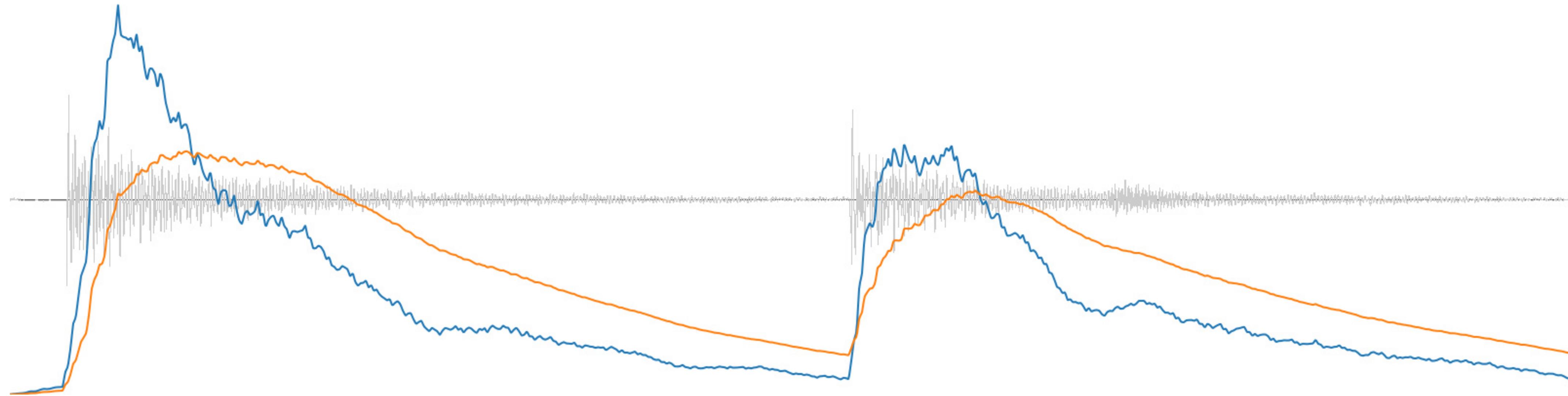


UMAP in 1 dimension

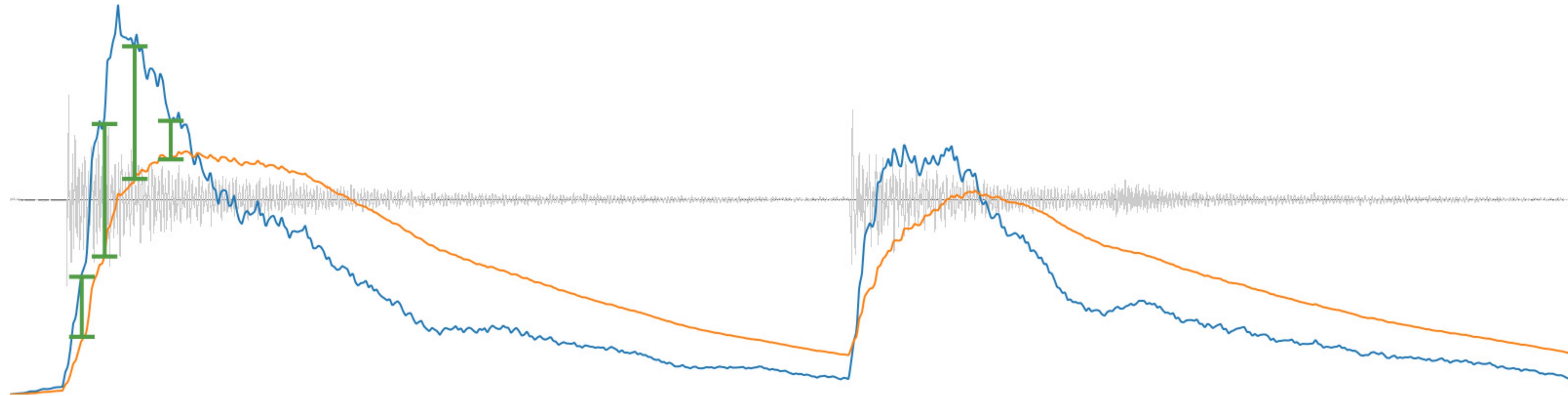
Slicing Audio

slicing up audio temporally

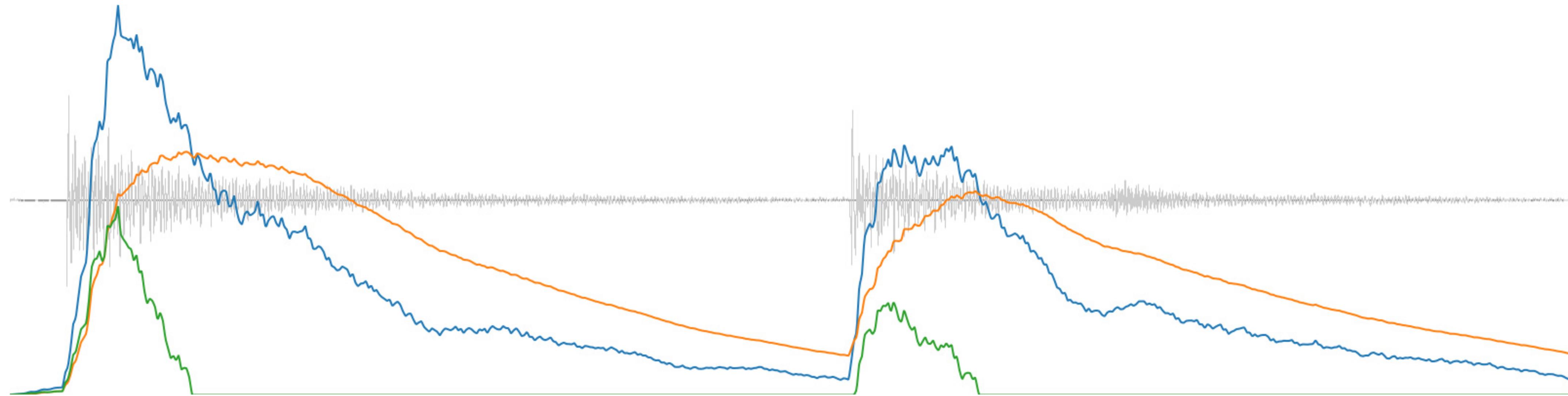




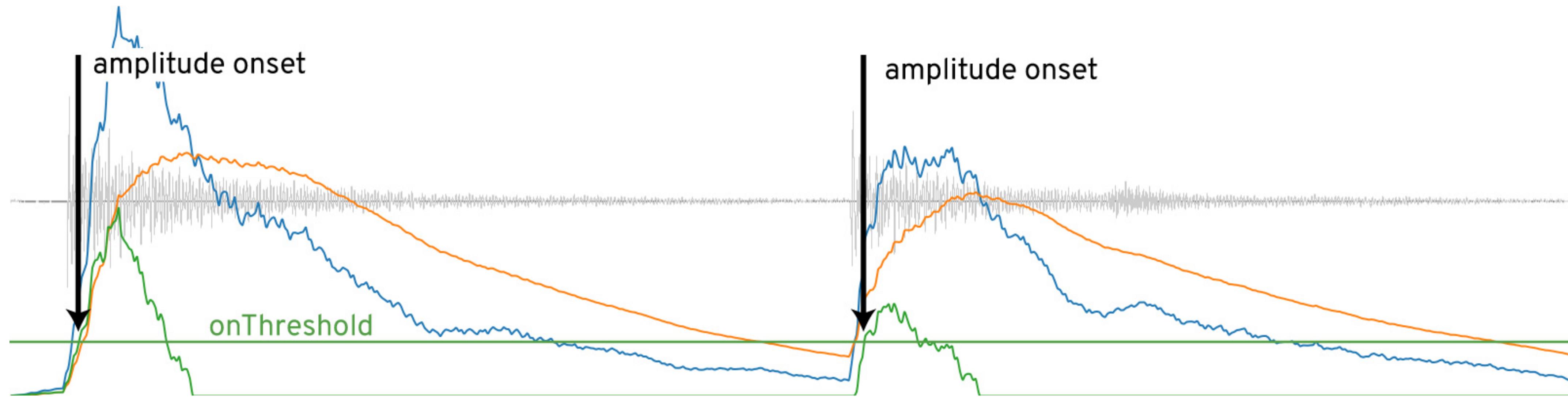
<https://learn.flucoma.org/reference/ampslice/>



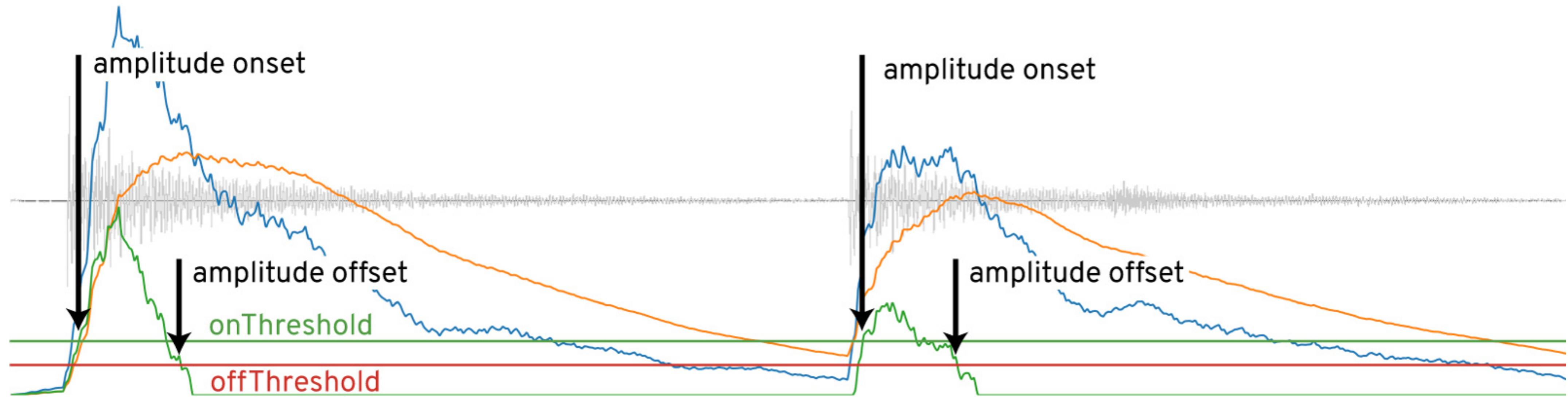
<https://learn.flucoma.org/reference/ampslice/>



<https://learn.flucoma.org/reference/ampslice/>

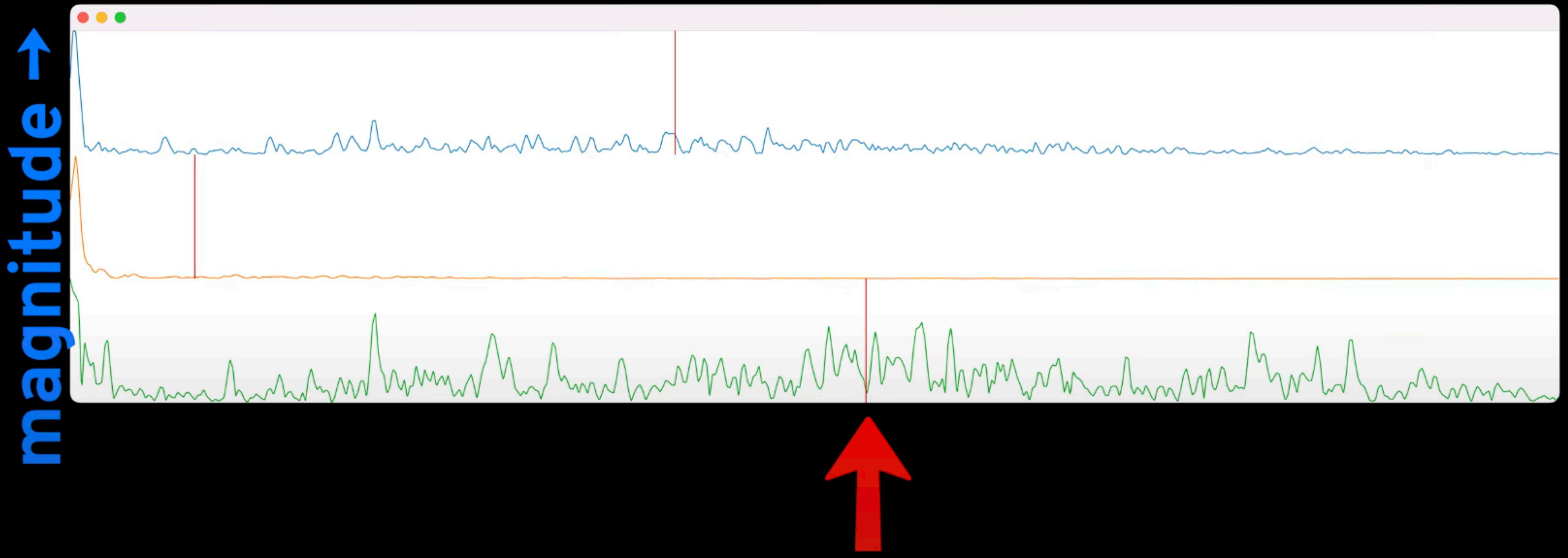


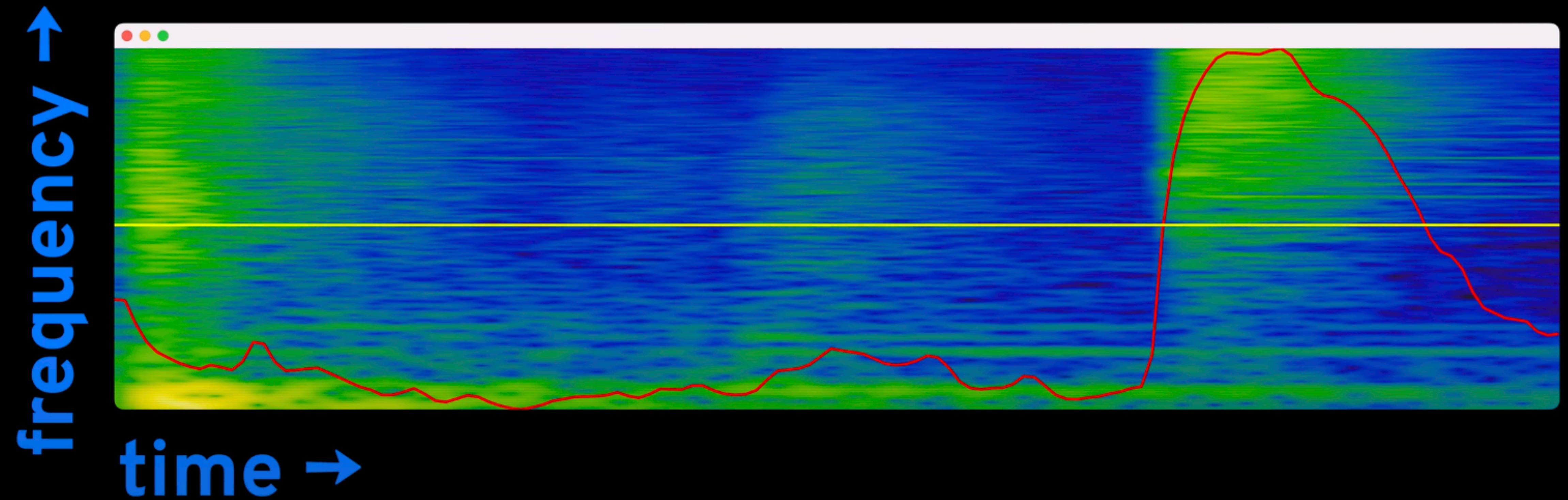
<https://learn.flucoma.org/reference/ampslice/>



<https://learn.flucoma.org/reference/ampslice/>

frequency →







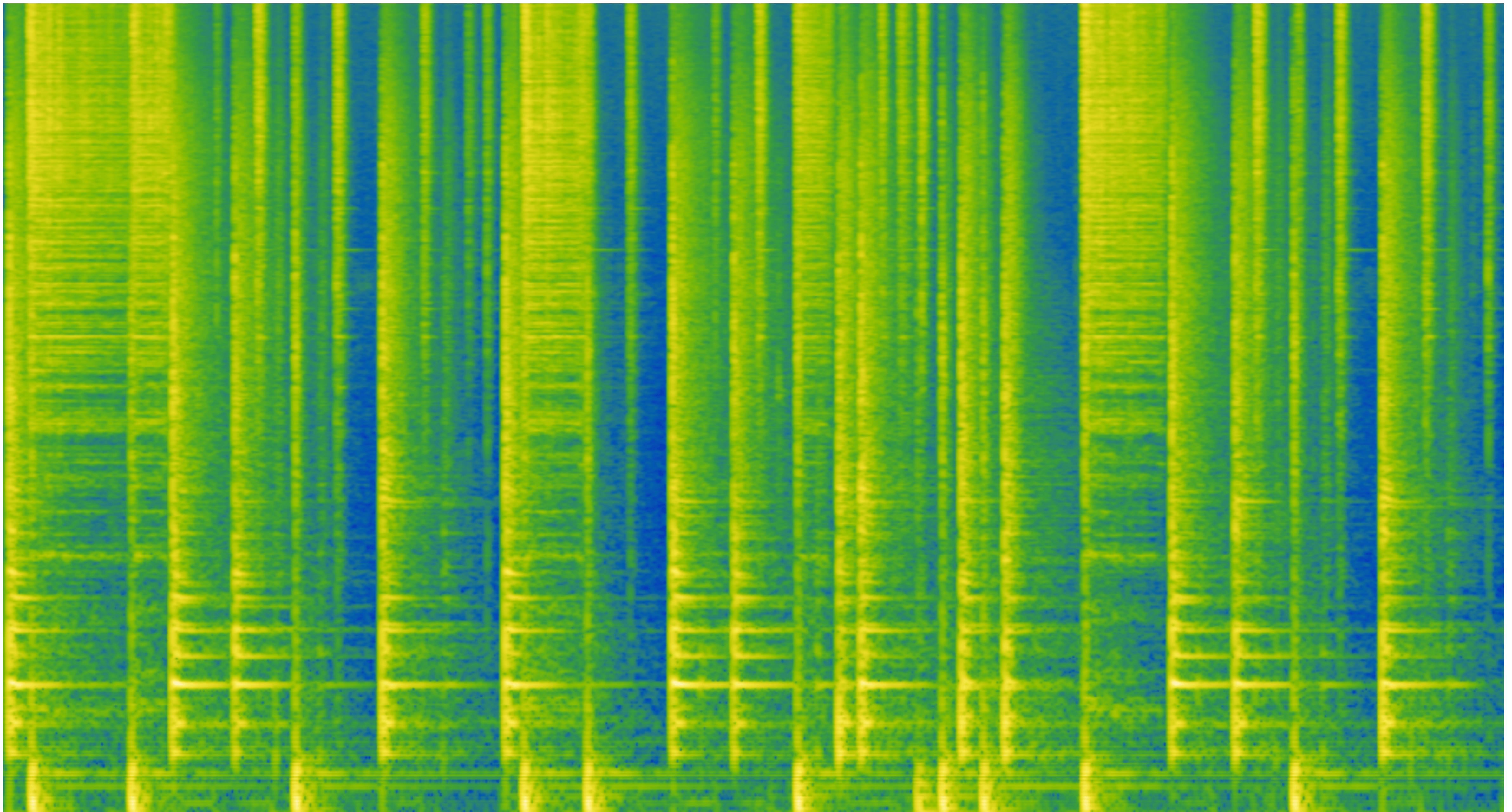
Moon via Spirit
by Lauren Sarah Hayes

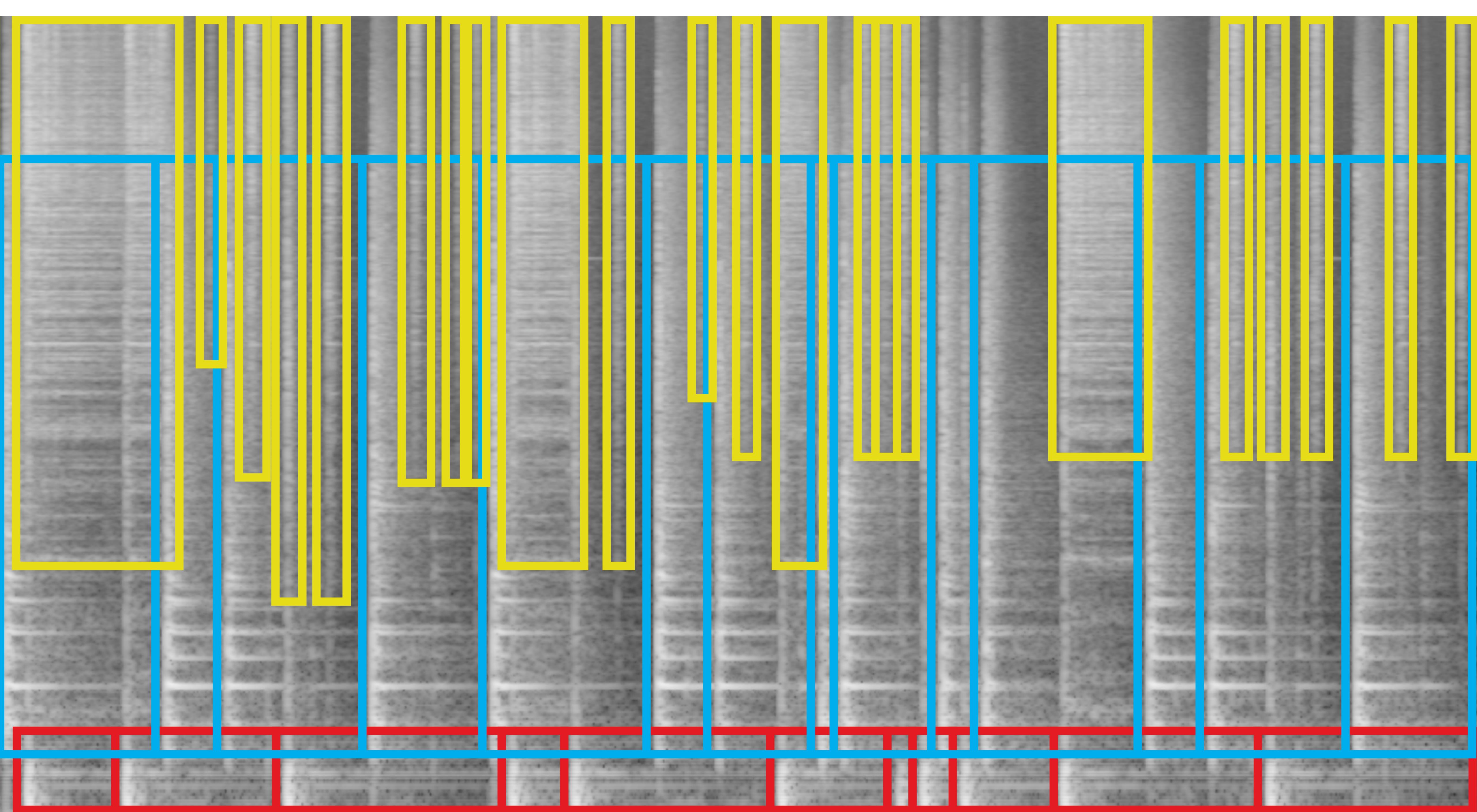
learn.flucoma.org/explore/hayes

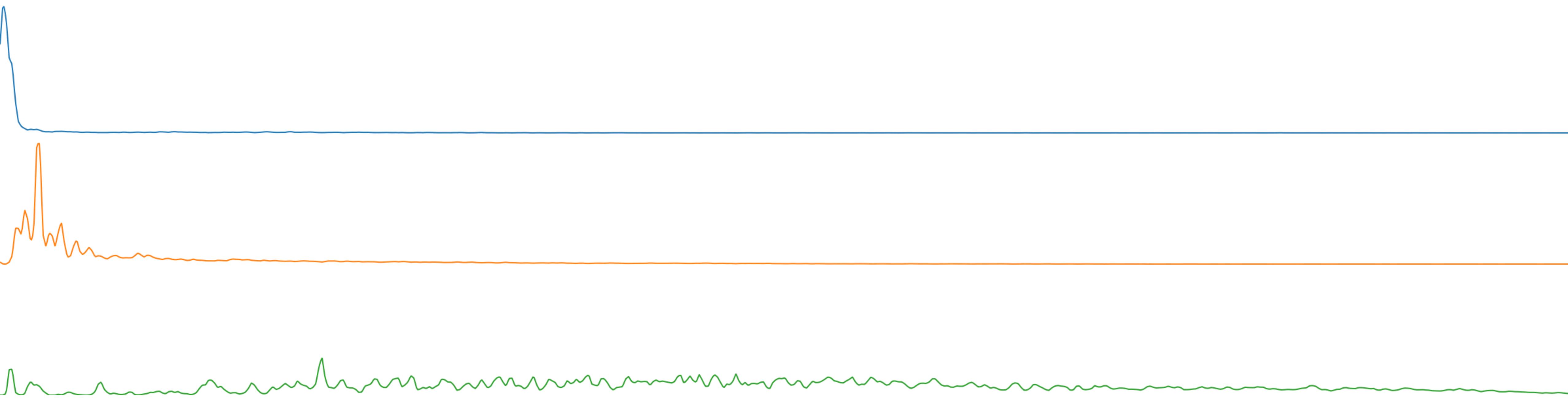
Non-negative Matrix Factorization

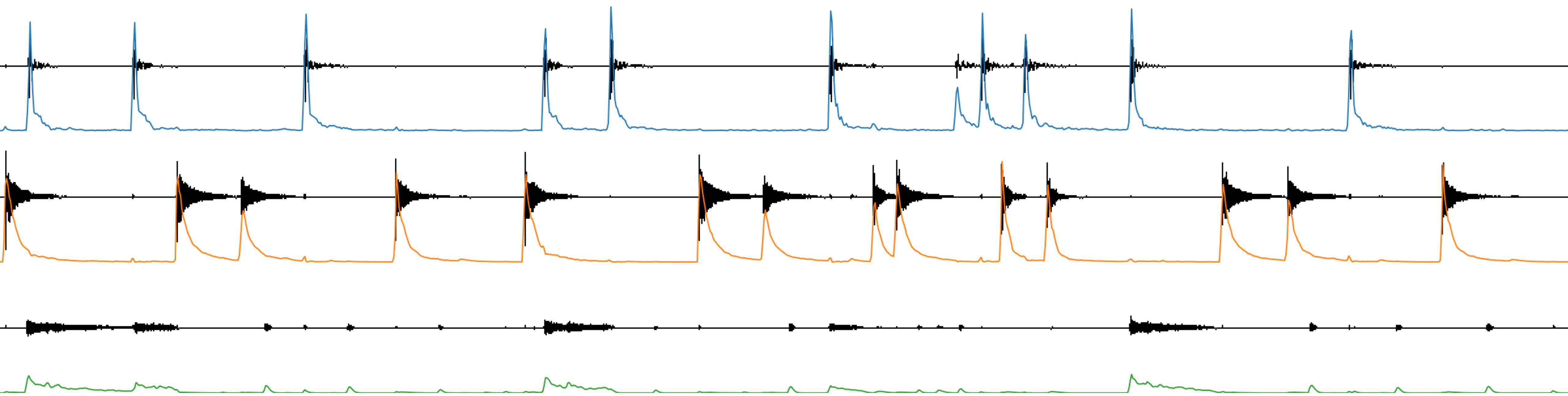
spectral decomposition

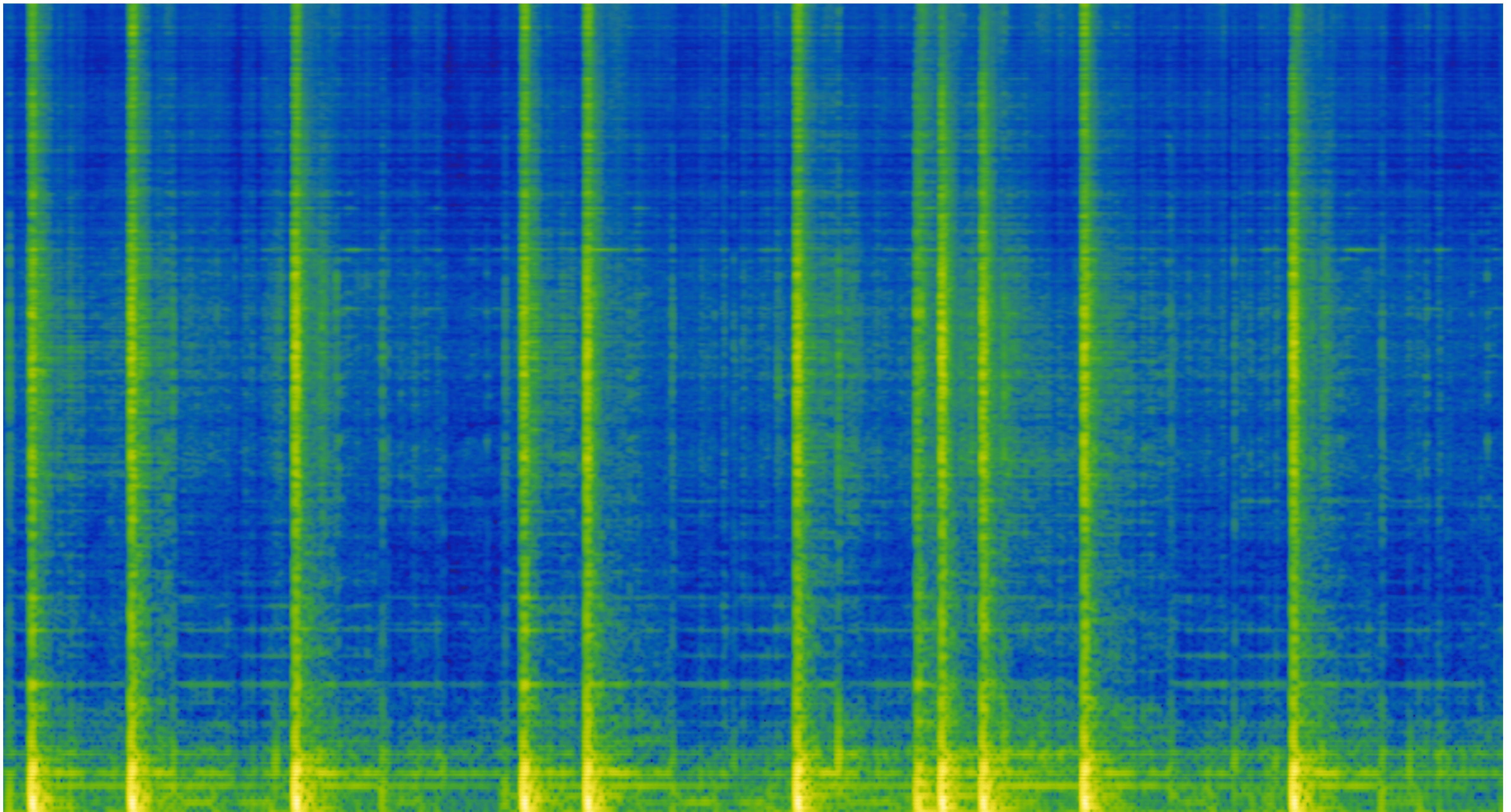


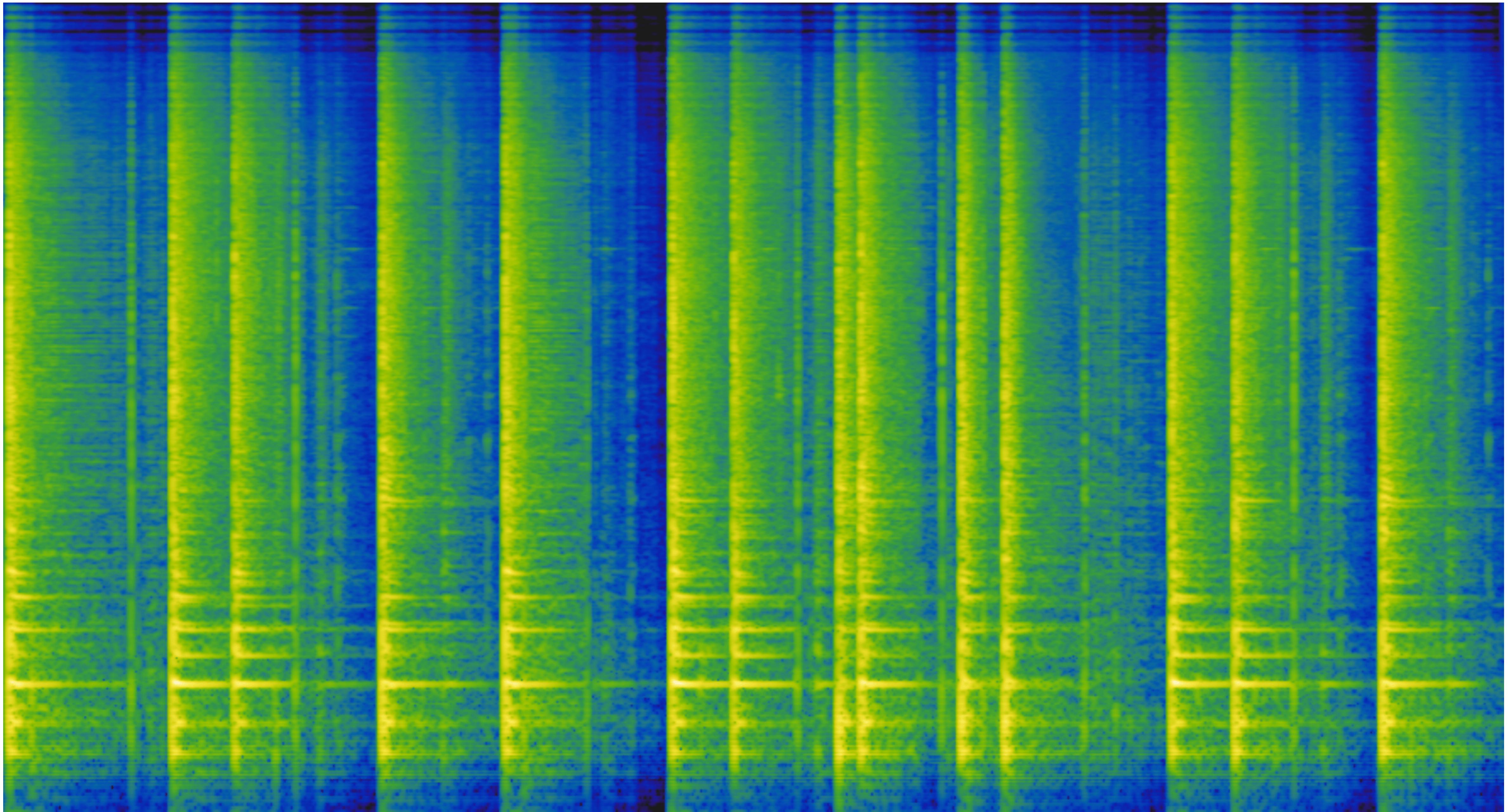


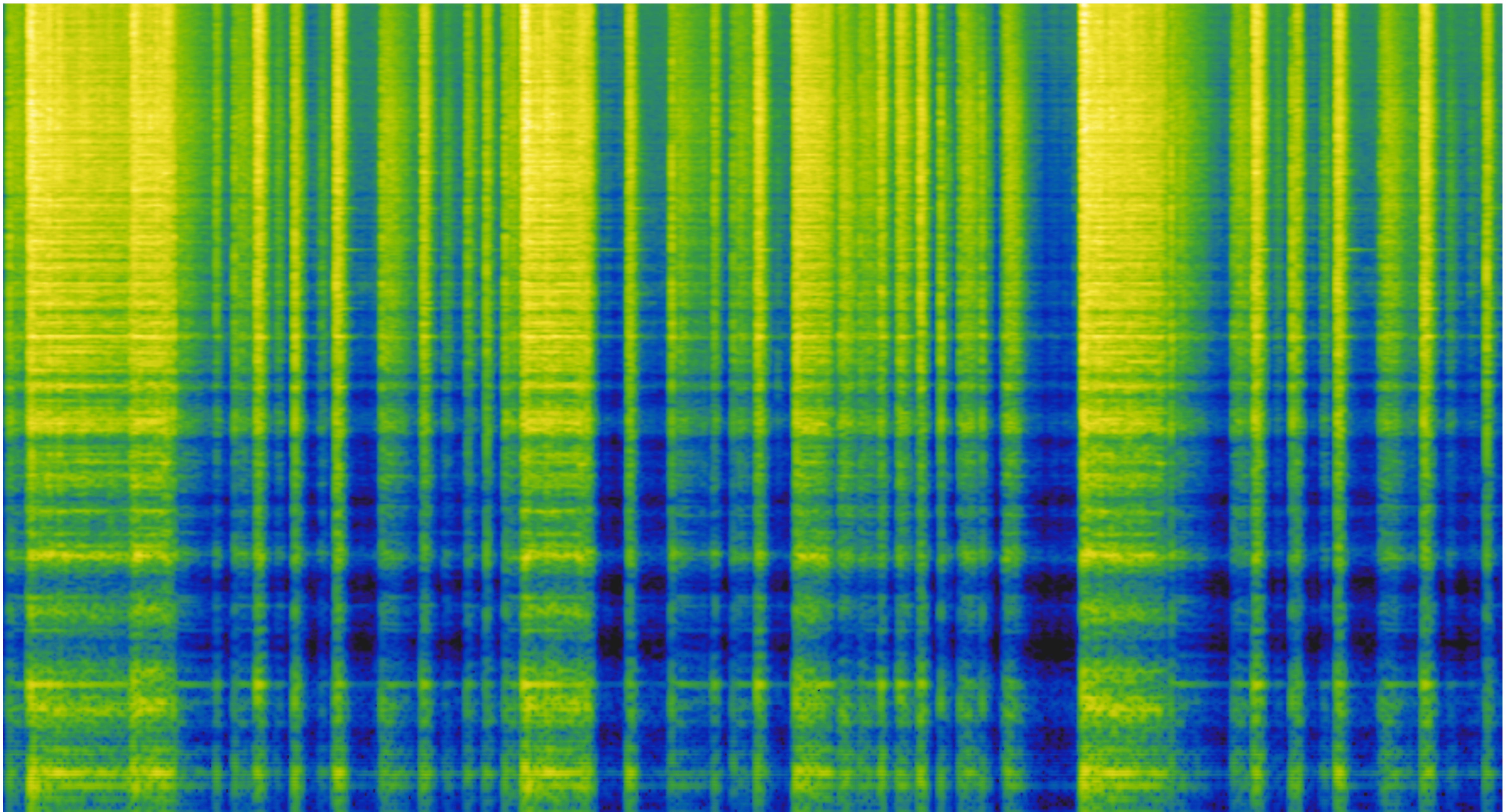














Herbig-Haro
by Olivier Pasquet

learn.flucoma.org/explore/pasquet

Audio Descriptor Analysis

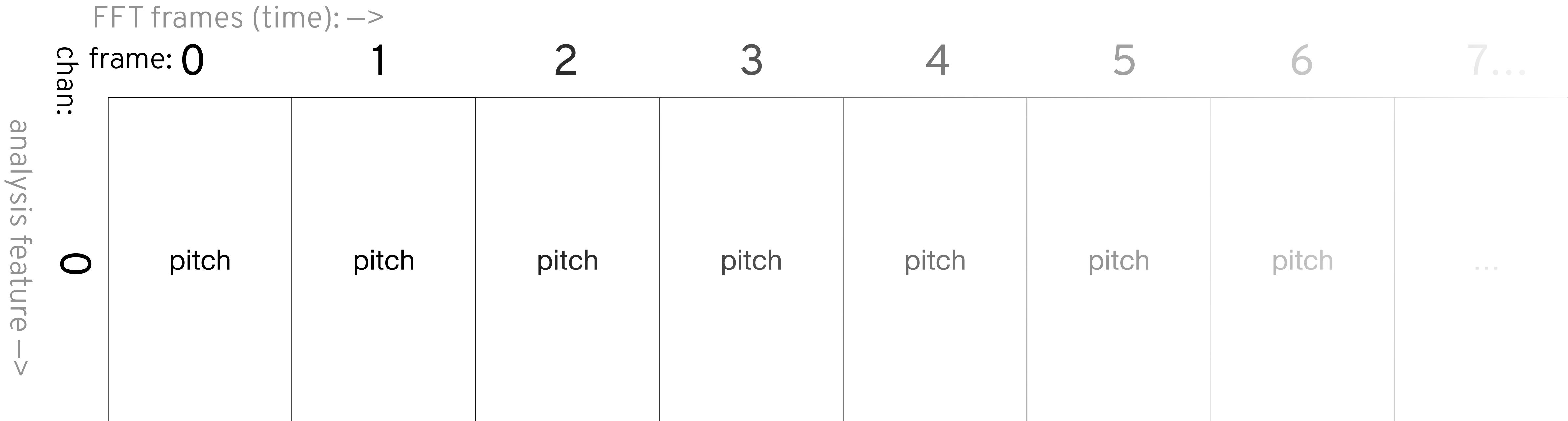
sort grains by pitch



FluidBufPitch writes the analysis to a buffer

FFT frames (time): →

FluidBufPitch writes the analysis to a buffer



FluidBufStats writes the analysis to another buffer

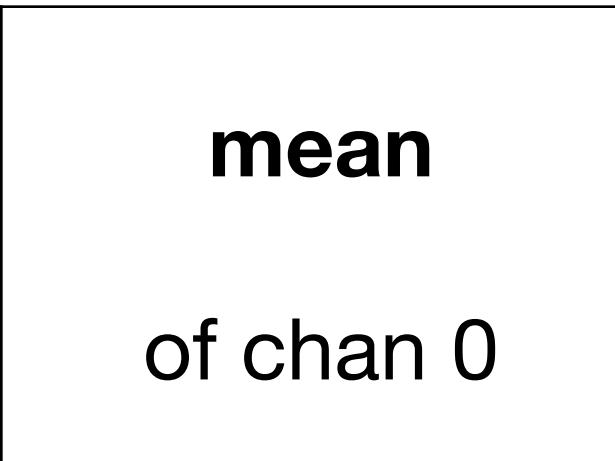
frame:	0	1	2	3	4	5	6
chan:	0	1	2	3	4	5	6
analysis feature →	mean of chan 0	stand. dev. of chan 0	skewness of chan 0	kurtosis of chan 0	low (min) of chan 0	mid (median) of chan 0	high (max) of chan 0
1	mean of chan 1	stand. dev. of chan 1	skewness of chan 1	kurtosis of chan 1	low (min) of chan 1	mid (median) of chan 1	high (max) of chan 1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

FluidBufStats writes the analysis to another buffer

frame:	0	1	2	3	4	5	6
chan:	0						
analysis feature ->	mean of chan 0	stand. dev. of chan 0	skewness of chan 0	kurtosis of chan 0	low (min) of chan 0	mid (median) of chan 0	high (max) of chan 0

FluidBufStats writes the analysis to another buffer

frame: 0
chan: 0
analysis feature ->



mean
of chan 0

FluidPlotter

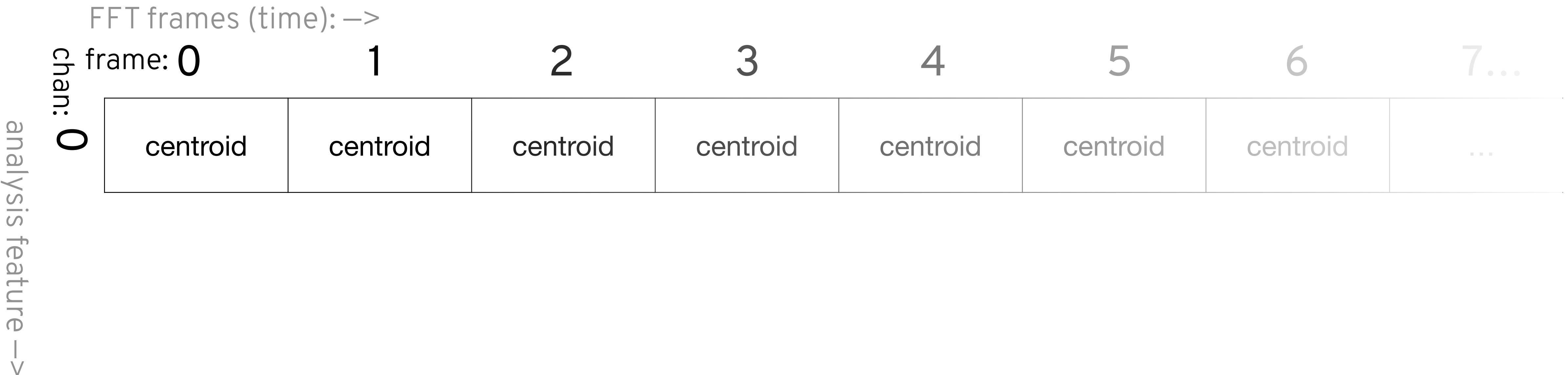
plotting drum hits by spectral
centroid and loudness



FluidBufSpectralShape writes the analysis to a buffer

FFT frames (time): →

FluidBufSpectralShape writes the analysis to a buffer



FluidBufLoudness writes the analysis to a buffer

FFT frames (time): →

This figure displays a grid of data points across 8 frames (x-axis) and 2 channels (y-axis). The columns are labeled 0 through 7, and the rows are labeled chan: 0 and chan: 1.

- chan: 0:** The 'loudness' series is represented by the text "loudness" repeated in each frame under this channel.
- chan: 1:** The 'true peak' series is represented by the text "true peak" repeated in each frame under this channel.

The labels are displayed in a large, bold, black font. The background is white with light gray vertical grid lines corresponding to the frame indices.

FluidBufLoudness writes the analysis to a buffer



FluidBufStats writes the analysis to another buffer

frame:	0	1	2	3	4	5	6
chan:	0	1	2	3	4	5	6
analysis feature →	mean of chan 0	stand. dev. of chan 0	skewness of chan 0	kurtosis of chan 0	low (min) of chan 0	mid (median) of chan 0	high (max) of chan 0
1	mean of chan 1	stand. dev. of chan 1	skewness of chan 1	kurtosis of chan 1	low (min) of chan 1	mid (median) of chan 1	high (max) of chan 1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

FluidBufStats writes the analysis to another buffer

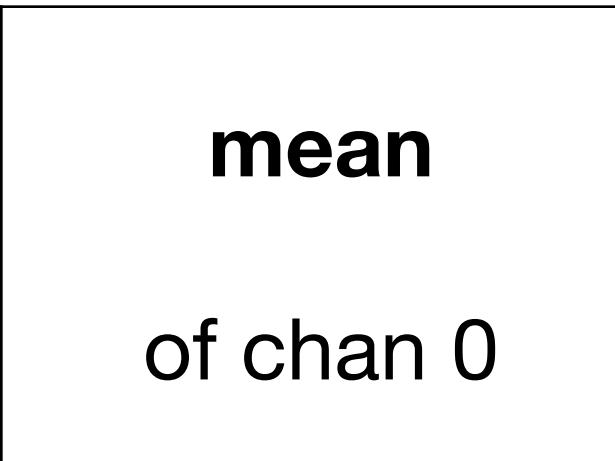
frame:	0	1	2	3	4	5	6
chan:	0						
analysis feature ->	mean of chan 0	stand. dev. of chan 0	skewness of chan 0	kurtosis of chan 0	low (min) of chan 0	mid (median) of chan 0	high (max) of chan 0

FluidBufStats writes the analysis to another buffer

frame: 0

chan: 0

analysis feature ->



mean
of chan 0

Dimensionality Reduction

reducing *many* analysis to be
plotted in 2D space



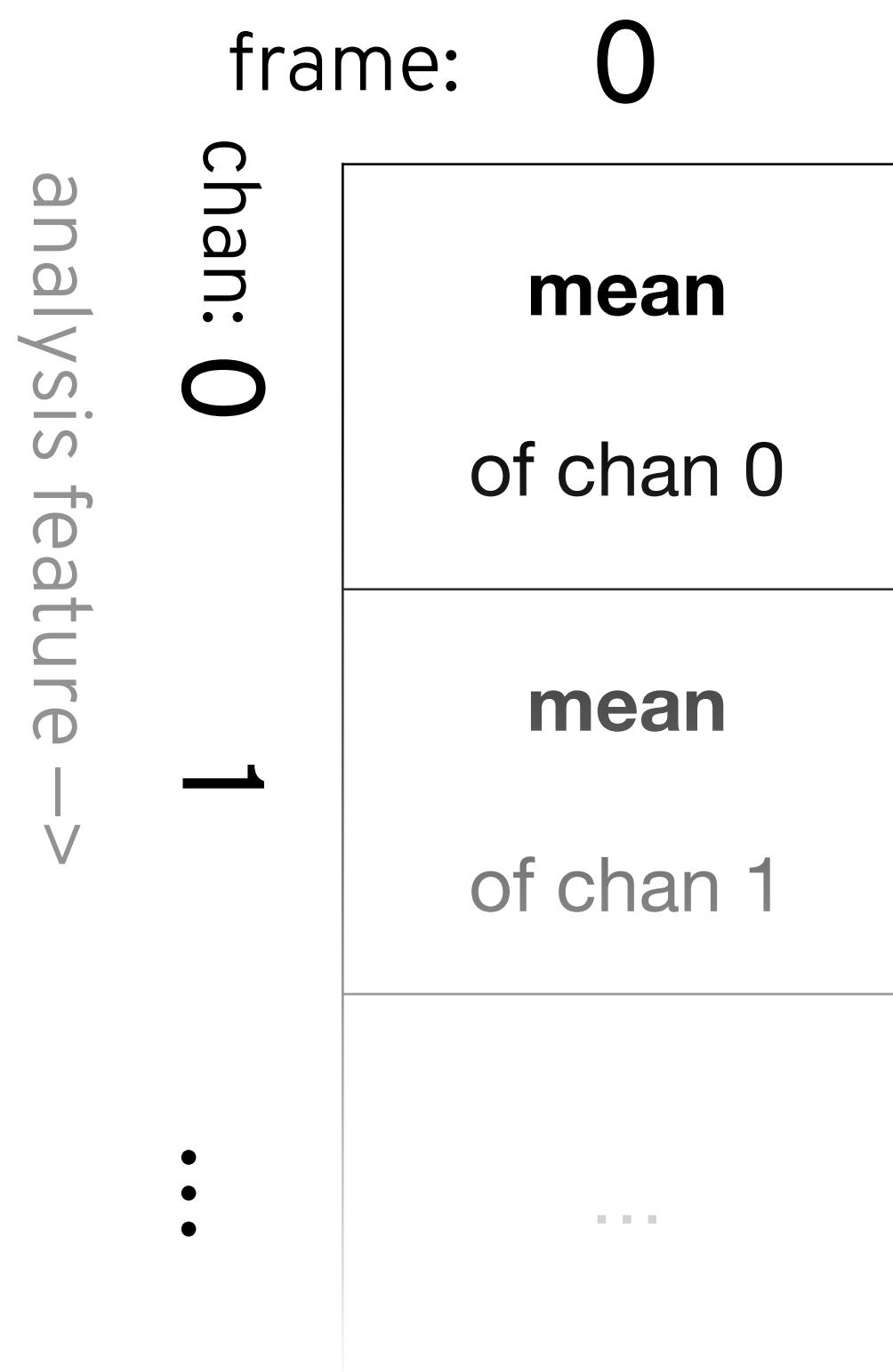
FluidBufMFCC writes the analysis to a buffer

FFT frames (time): →

FluidBufStats writes the analysis to another buffer

frame:	0	1	2	3	4	5	6
chan:	0	1	2	3	4	5	6
analysis feature →	mean of chan 0	stand. dev. of chan 0	skewness of chan 0	kurtosis of chan 0	low (min) of chan 0	mid (median) of chan 0	high (max) of chan 0
1	mean of chan 1	stand. dev. of chan 1	skewness of chan 1	kurtosis of chan 1	low (min) of chan 1	mid (median) of chan 1	high (max) of chan 1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

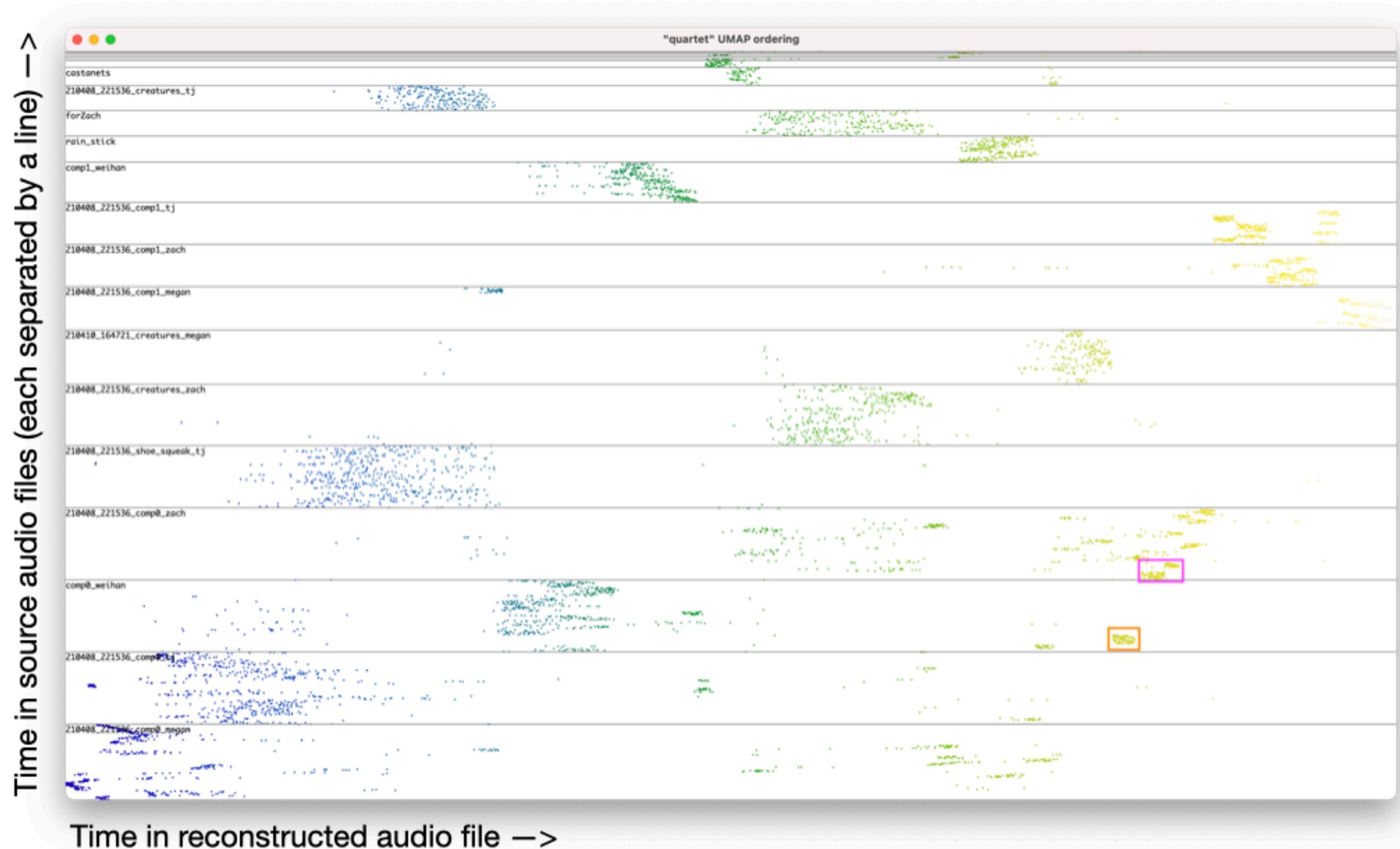
FluidBufStats writes the analysis to another buffer



learn.flucoma.org/reference/uMap



Hamiltonian Path: FluidUMAP



quartet by Ted Moore

(There's a lot more info in the article by Jacob Hart)

learn.flucoma.org/explore/moore

flucoma.org
learn.flucoma.org



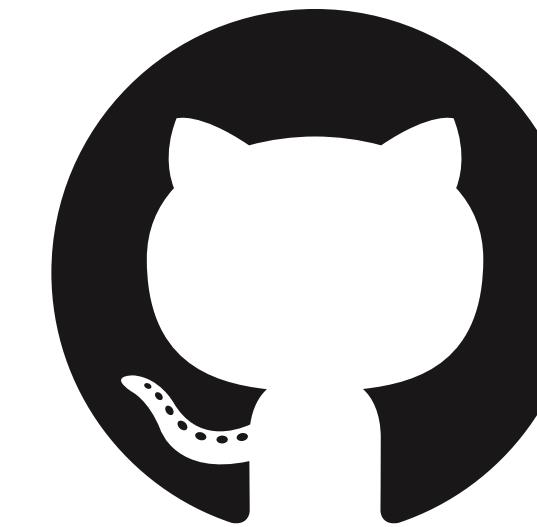
download FluCoMa package
toolkit reference
learn from tutorials
hear & see works made with FluCoMa

discourse.flucoma.org



share works in progress
ask questions
get ideas

<https://github.com/flucoma>



bug reports
pull requests

[https://www.youtube.com/c/
fluidcorpusmanipulation](https://www.youtube.com/c/fluidcorpusmanipulation)



tutorials
performances

Concatenative Synthesis

finding *similar* sounds in high-dimensional space



x Post window Auto Scroll

analyzing slice: 21 / 23
analyzing slice: 22 / 23
analyzing slice: 23 / 23
DataSet 1179:
rows: 23 cols: 13
slice-0 20.27 -8.8067 2.6023 ..
slice-1 -40.249 15.376 21.426 ..
slice-2 -44.279 17.326 22.261 ..
...
slice-20 34.056 -2.3113 8.0918 ..
slice-21 21.741 4.8381 8.8219 ..
slice-22 19.393 23.32 23.99 ..

-> a Routine
kdtree fit
-> a Function
-> a Routine
target slice: 0
source nearest neighbour slice: 296

target slice: 1
source nearest neighbour slice: 191

target slice: 2
source nearest neighbour slice: 261

target slice: 3
source nearest neighbour slice: 331

target slice: 4
source nearest neighbour slice: 331

target slice: 5
source nearest neighbour slice: 372

target slice: 6
source nearest neighbour slice: 332

target slice: 7
source nearest neighbour slice: 296

target slice: 8
source nearest neighbour slice: 286

target slice: 9
source nearest neighbour slice: 372

target slice: 10
source nearest neighbour slice: 331

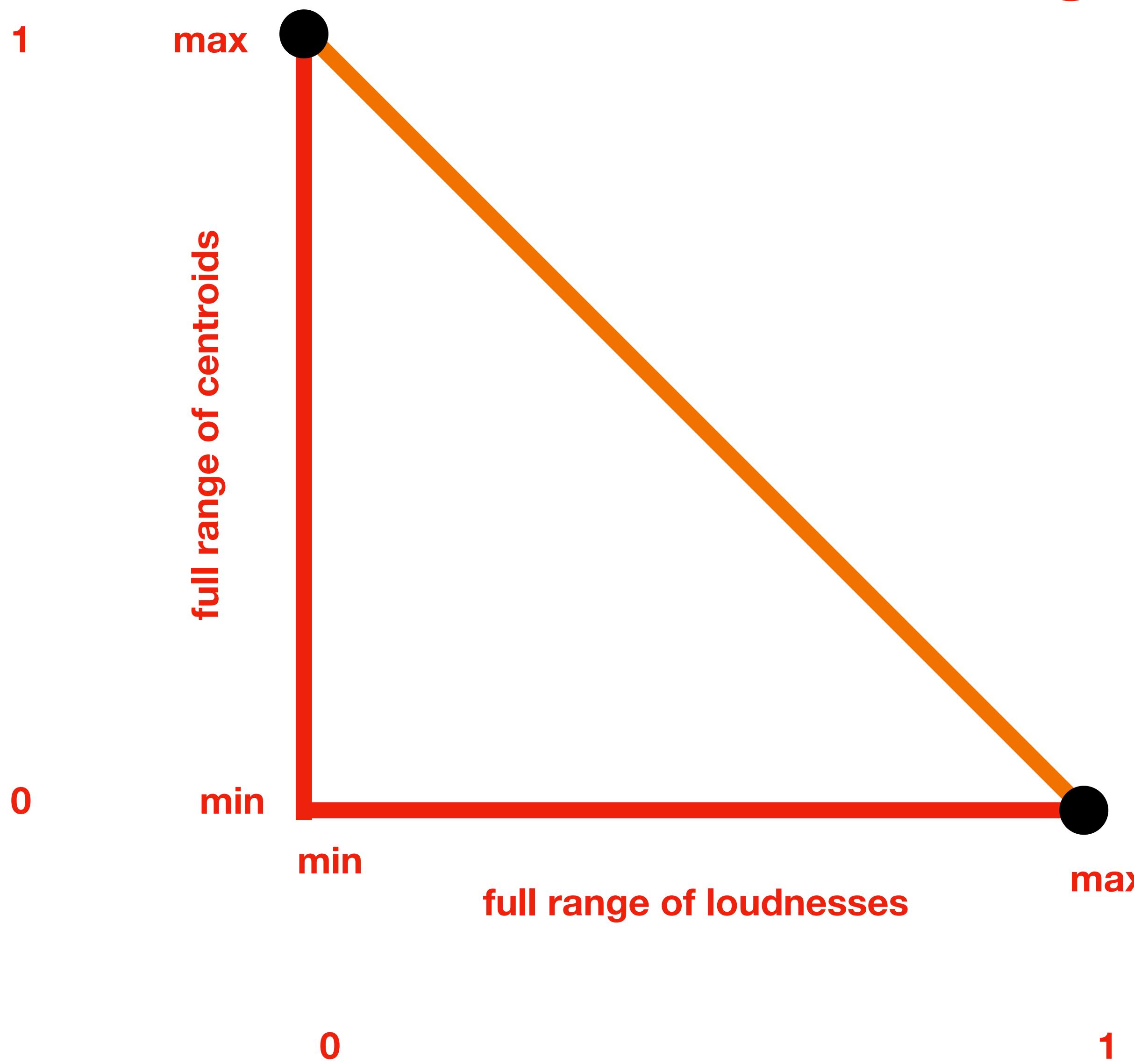
learn.flucoma.org/reference/kdtree

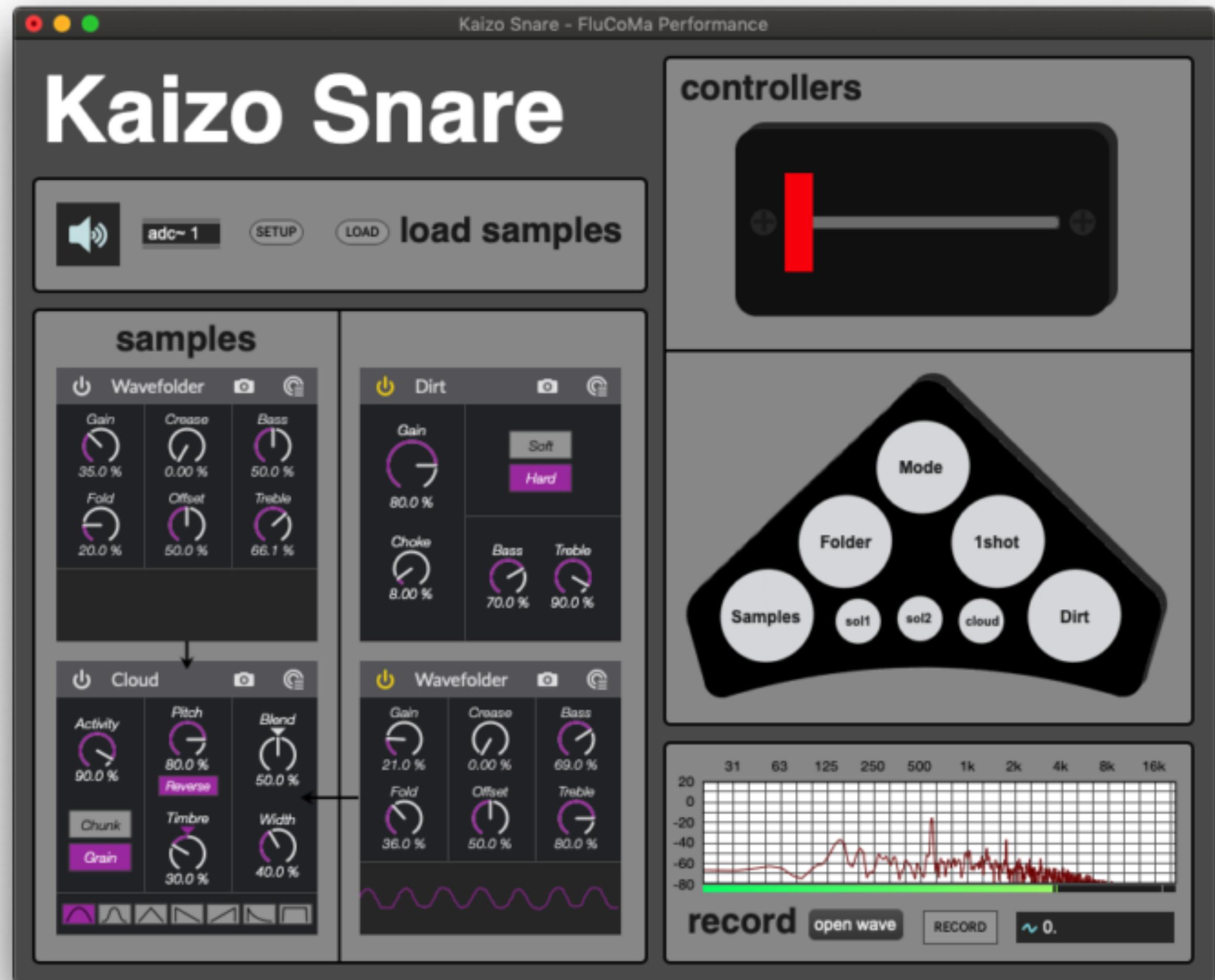


If 1 Hz = 1dB



Normalization





Kaizo Snare

by Rodrigo Constanzo

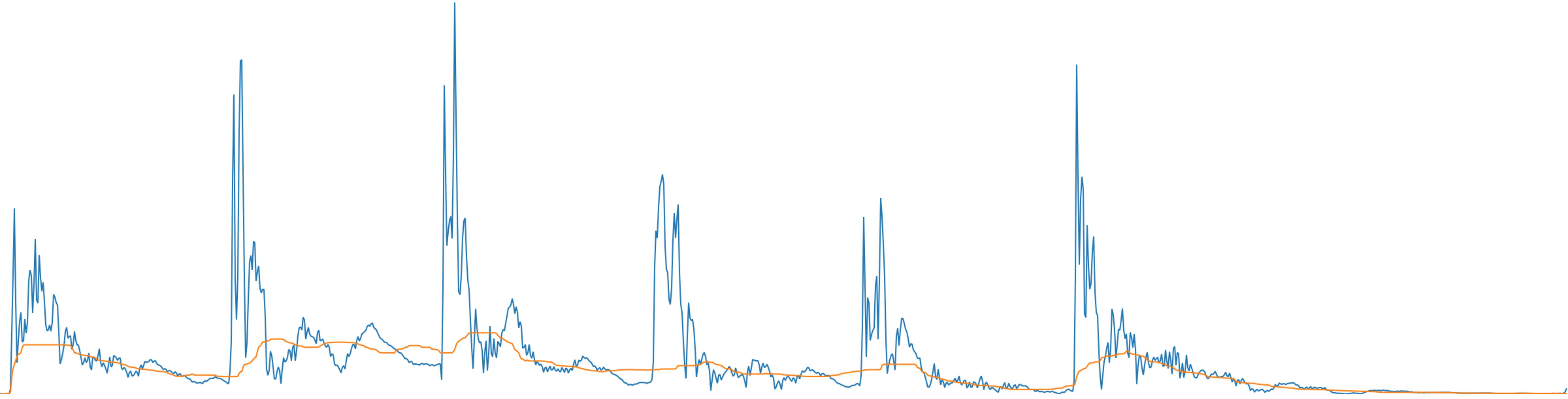
learn.flucoma.org/explore/constanzo

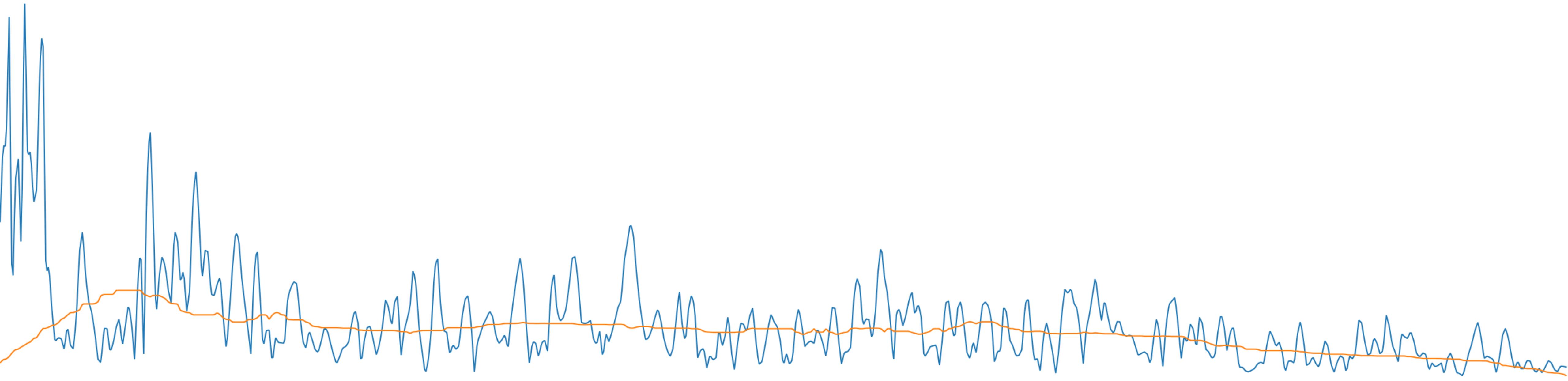
Harmonic-Percussive

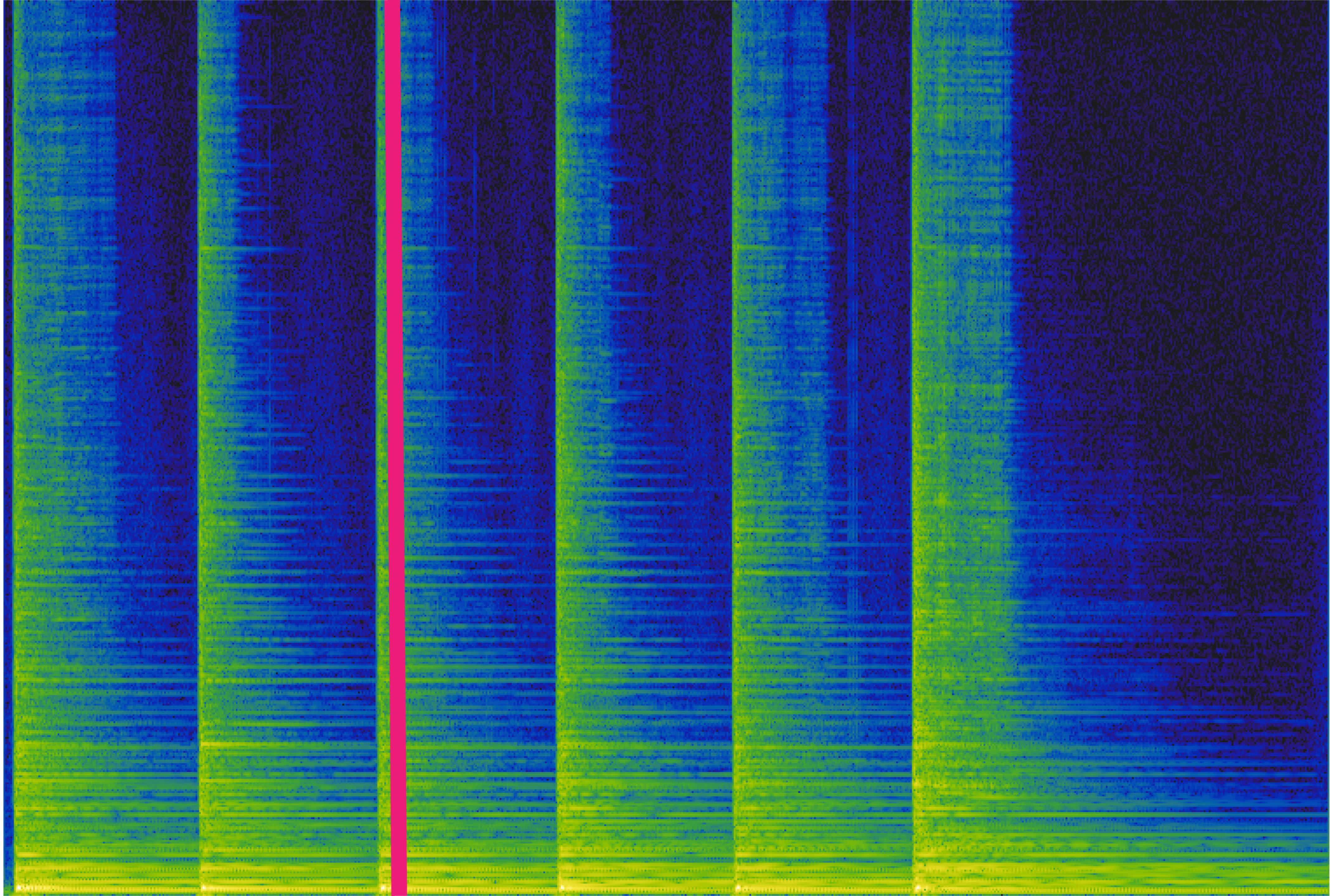
Source Separation

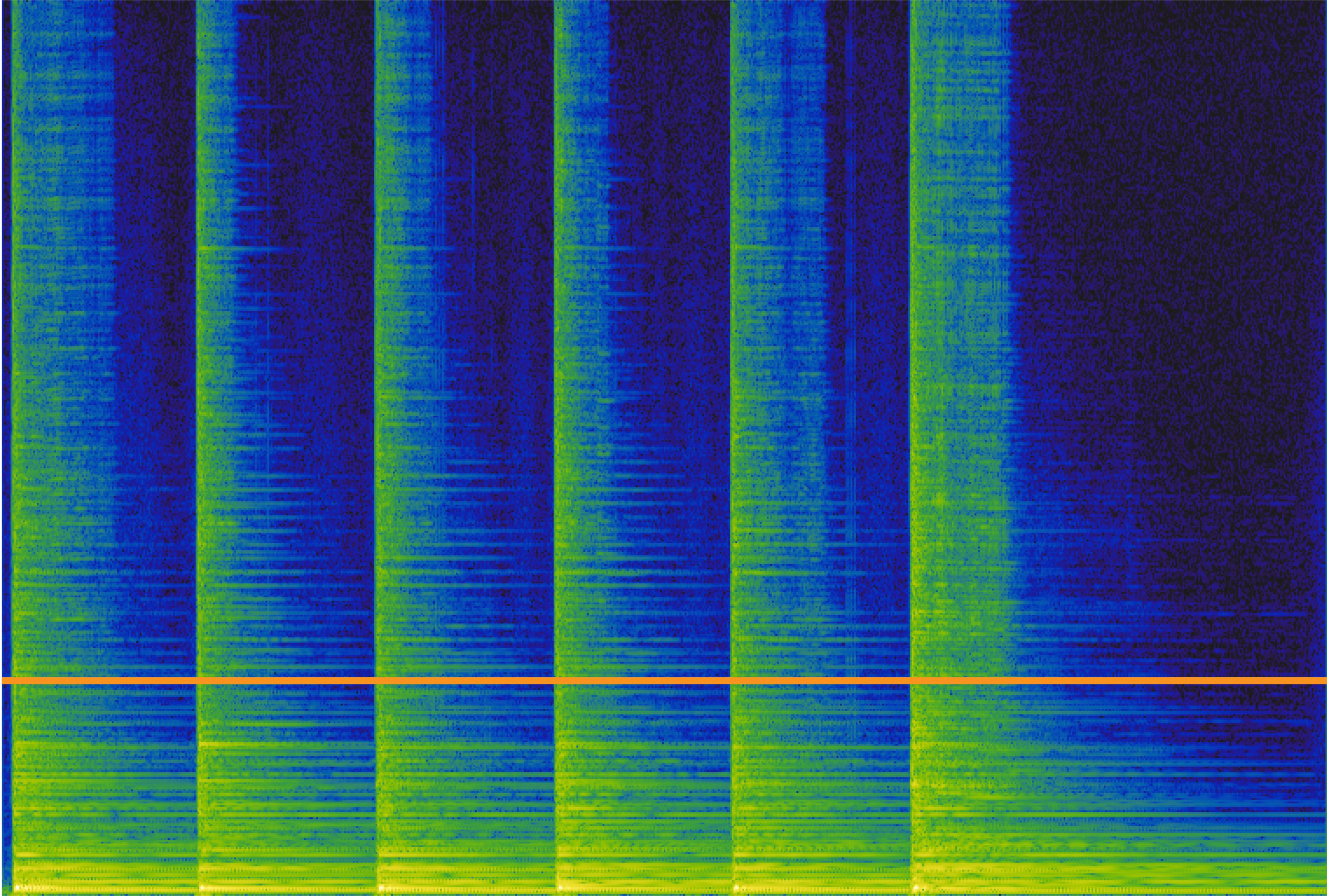
spectral decomposition

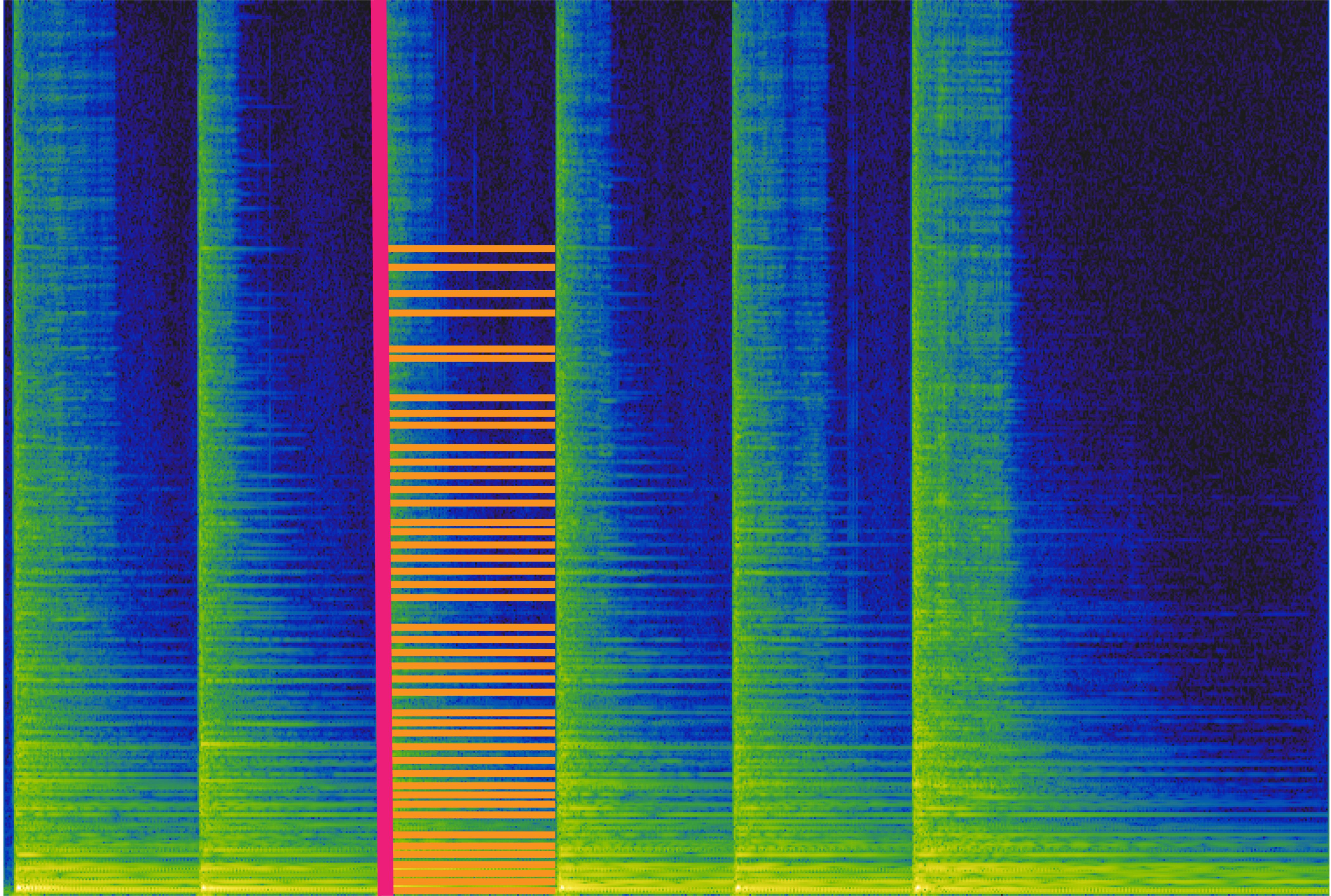


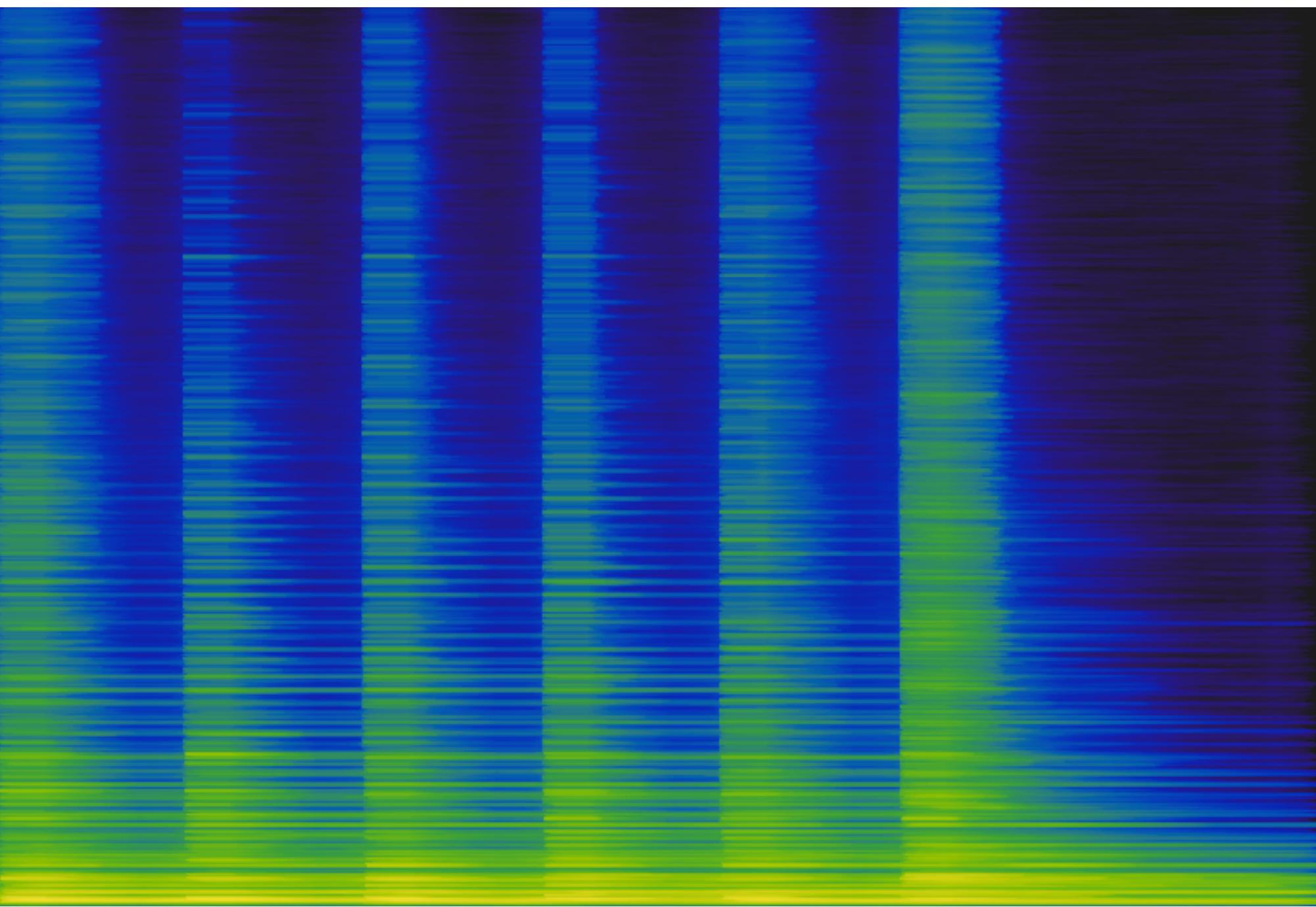


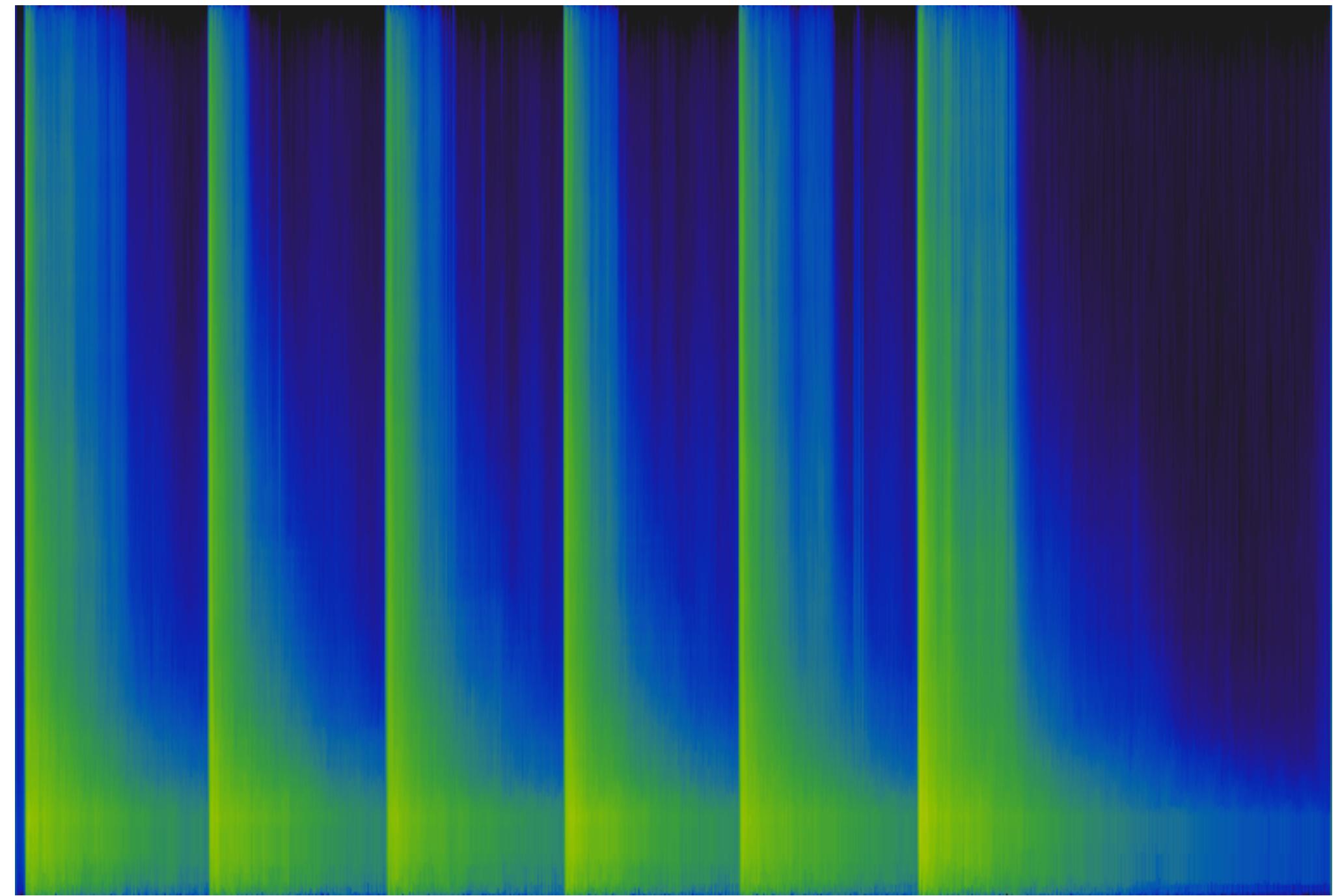


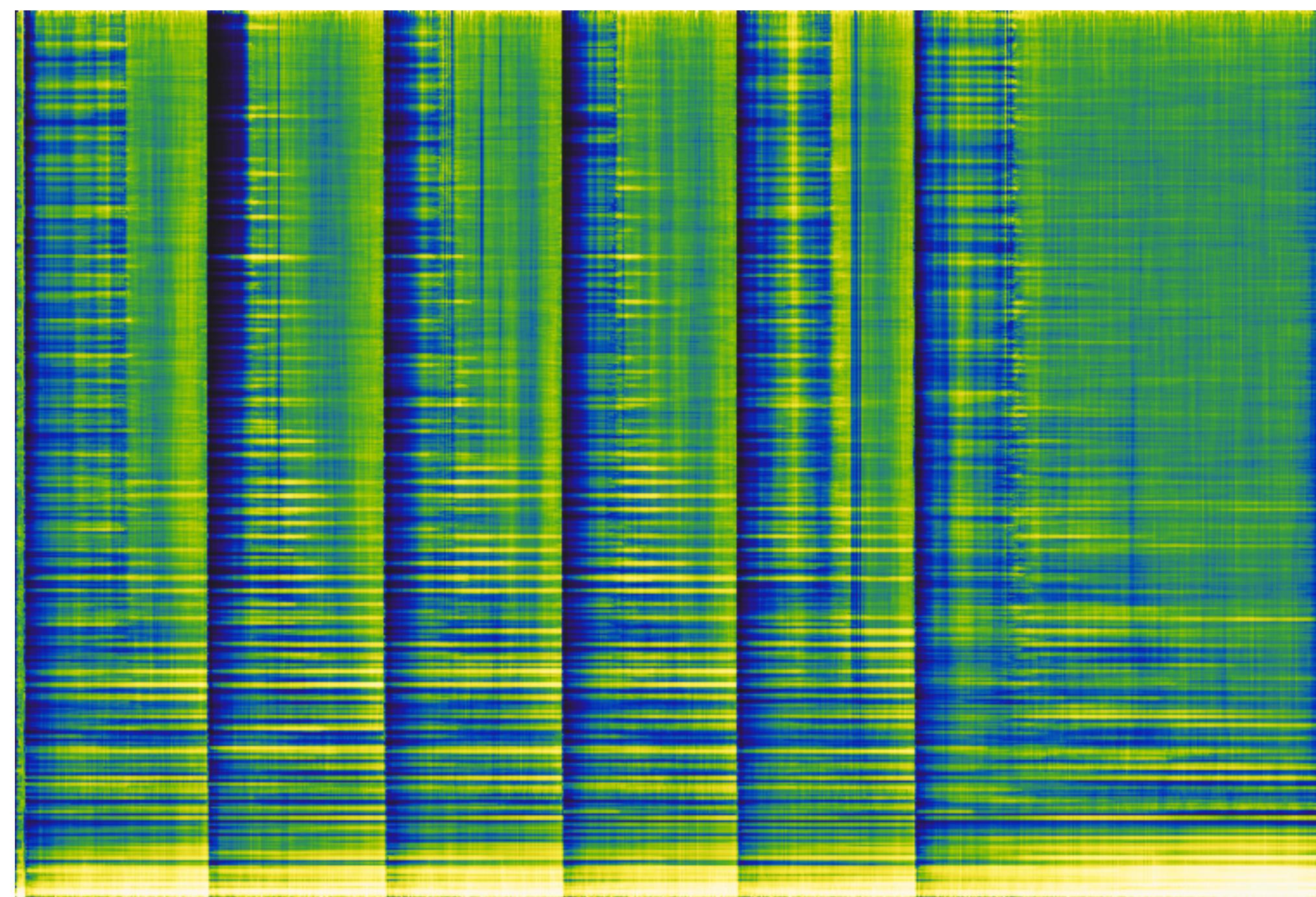


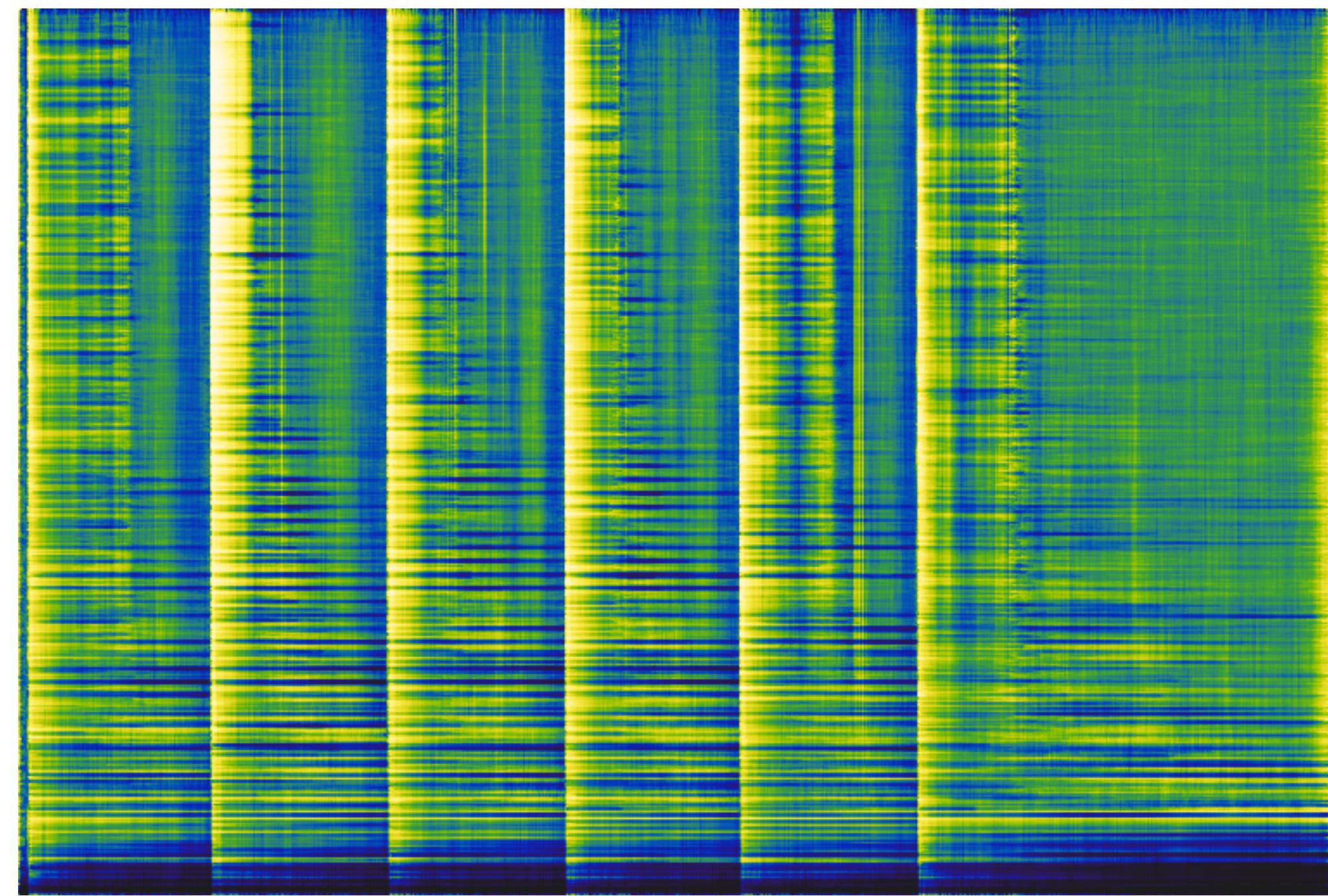


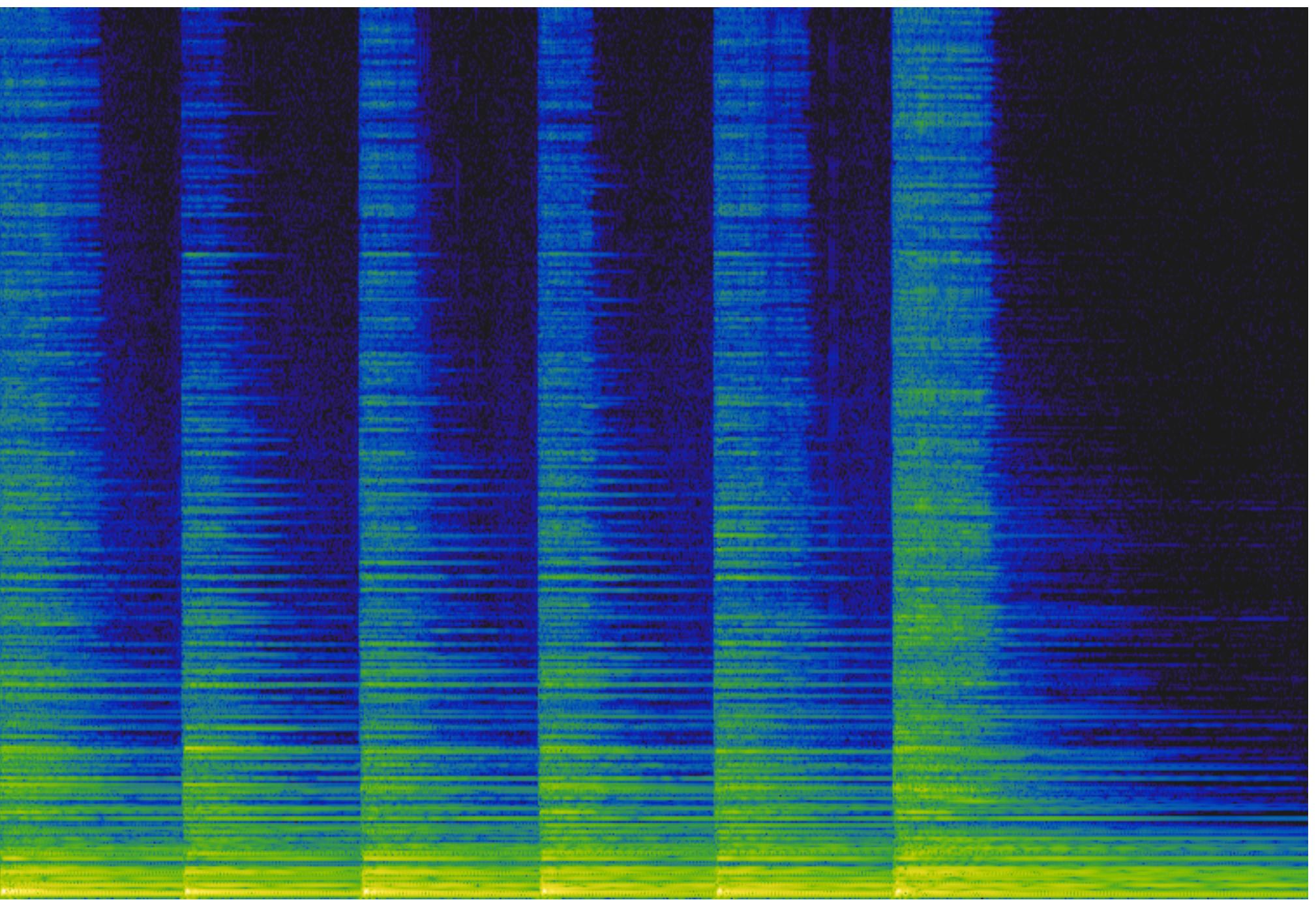


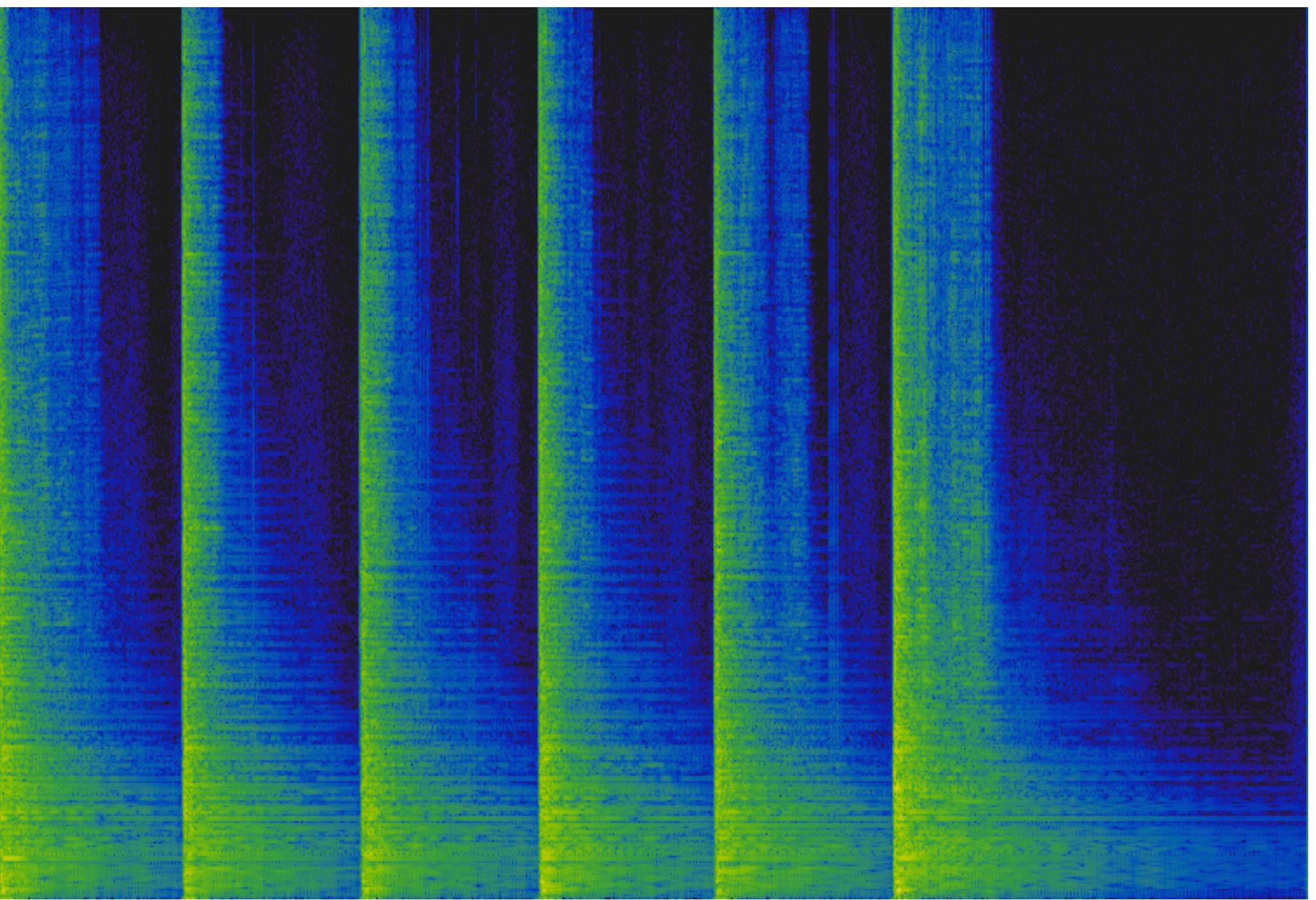


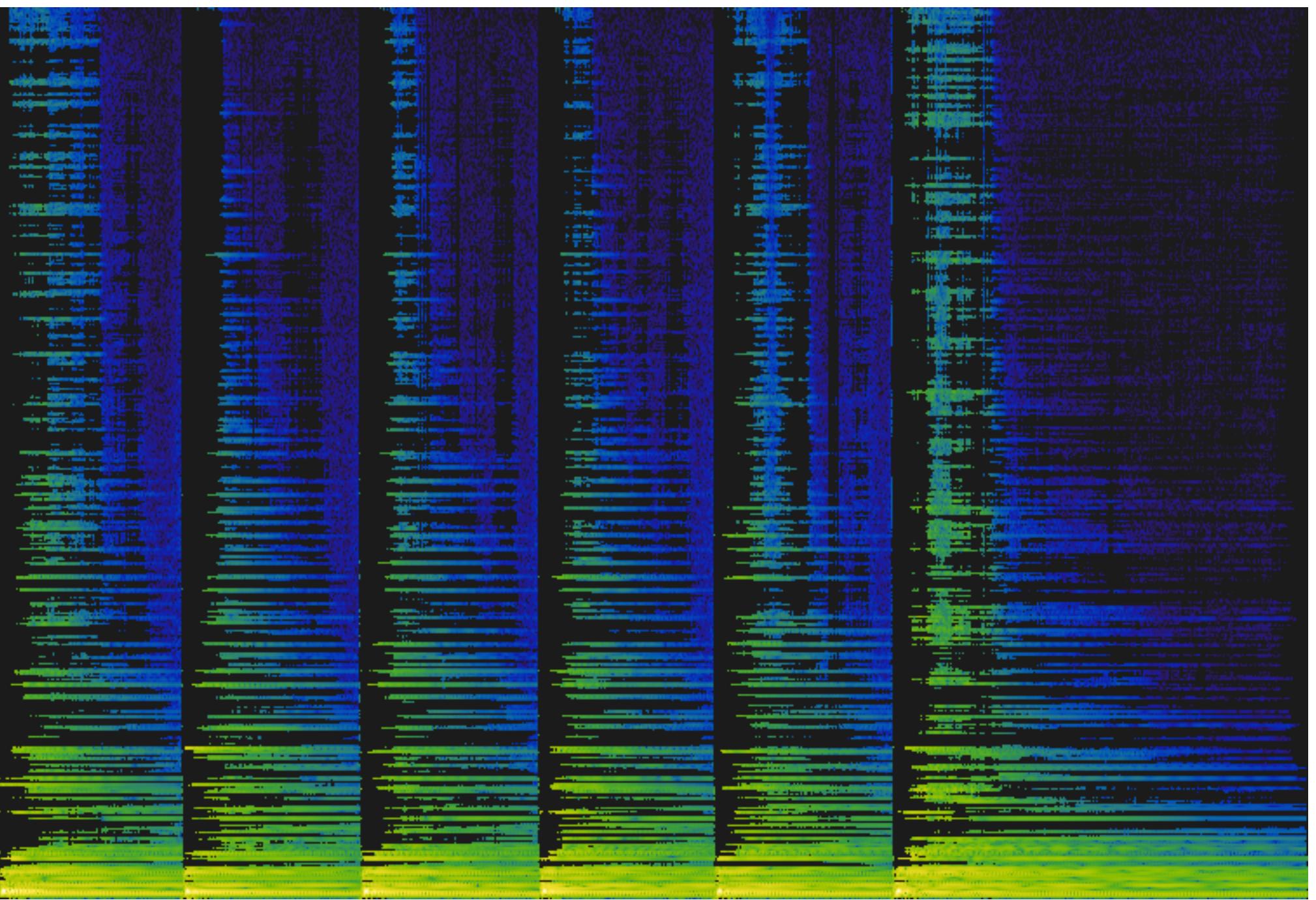


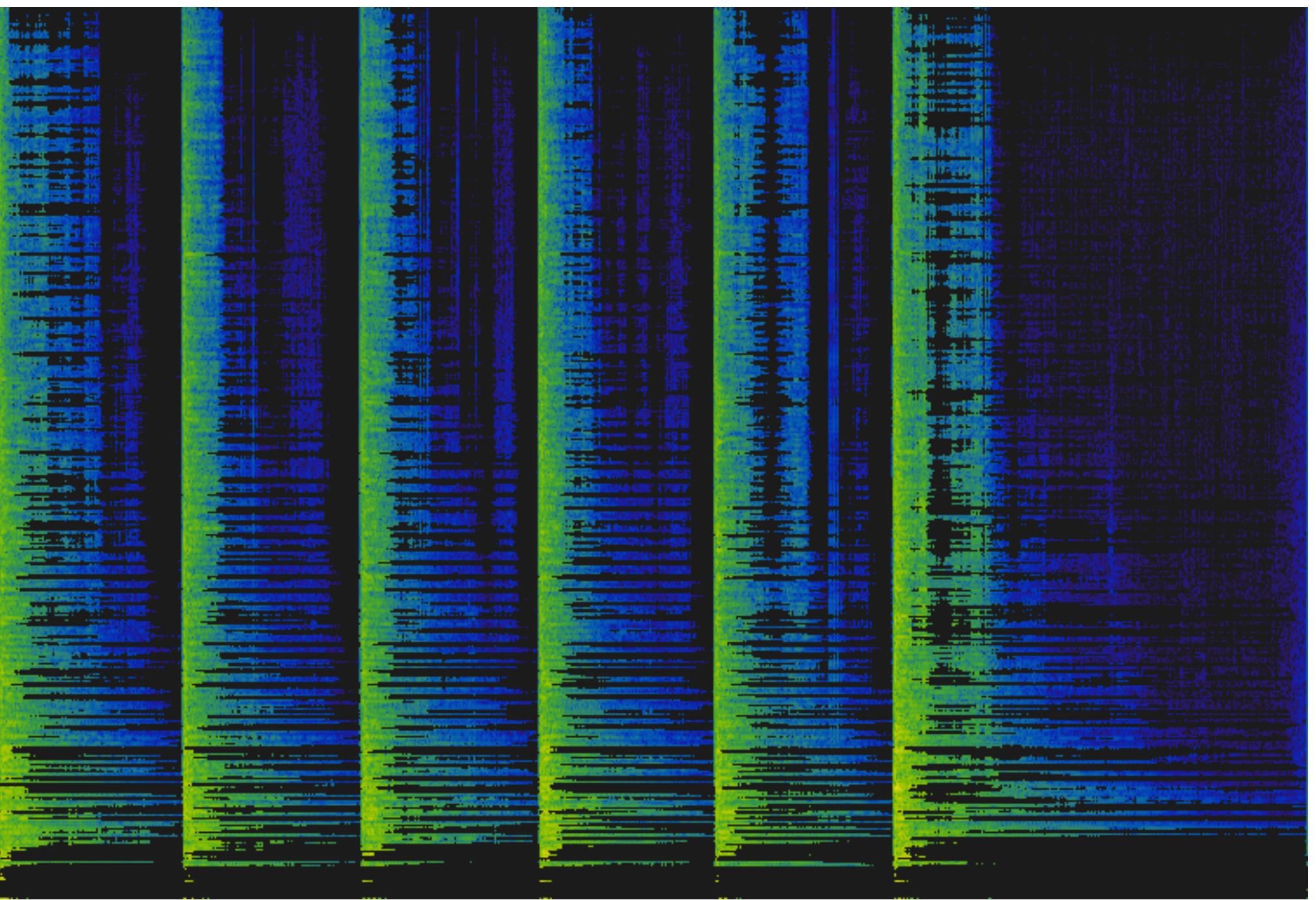




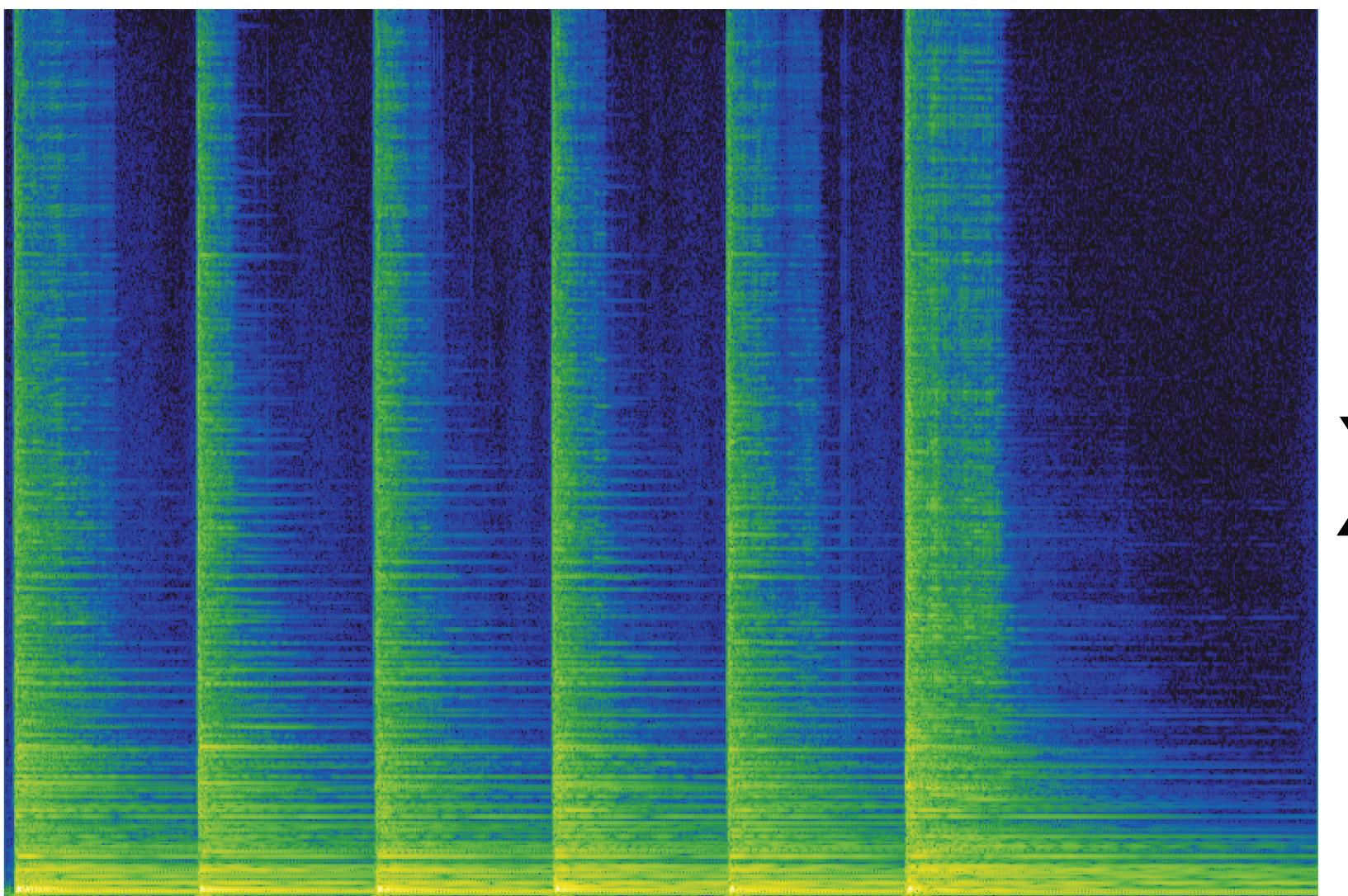




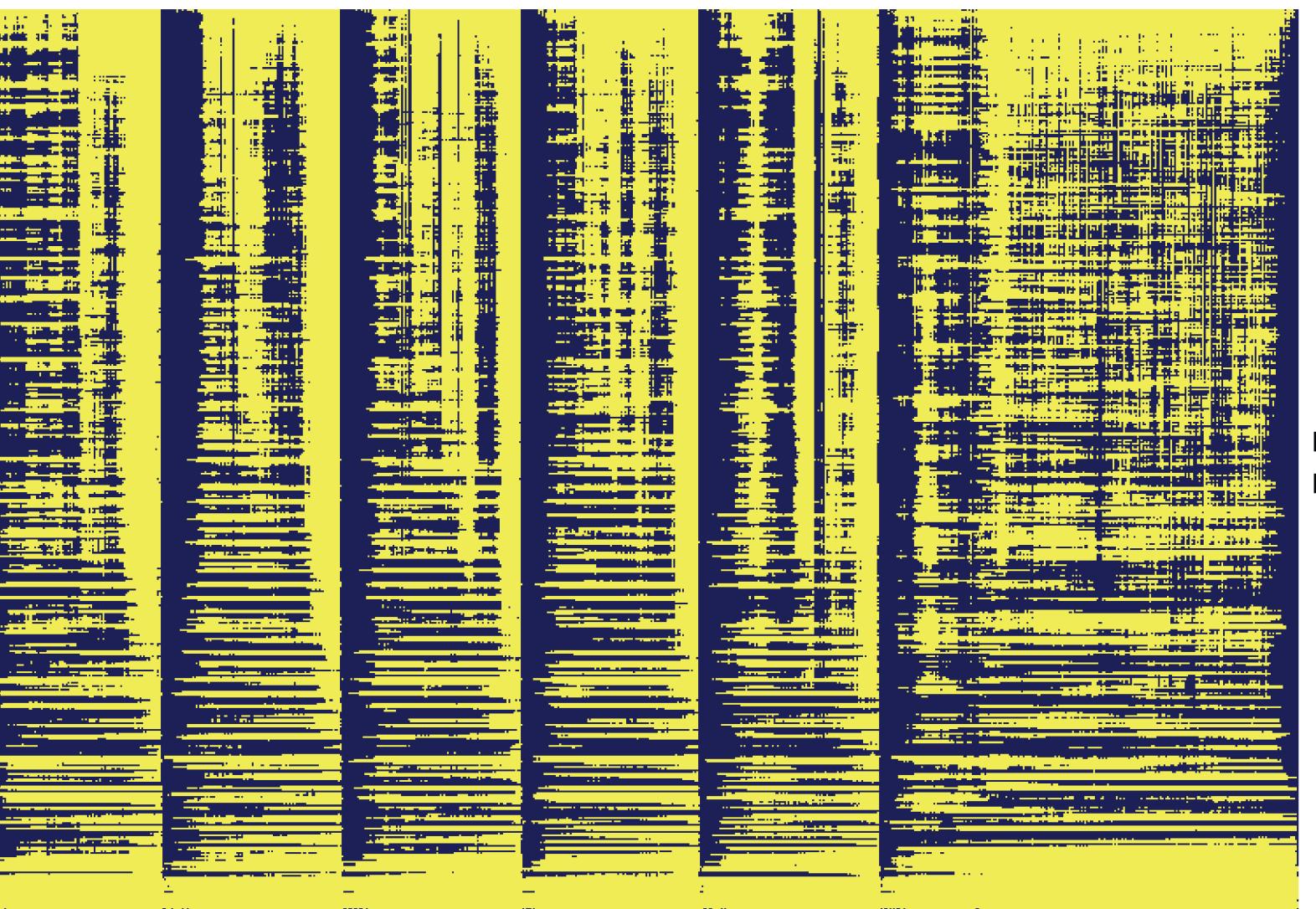




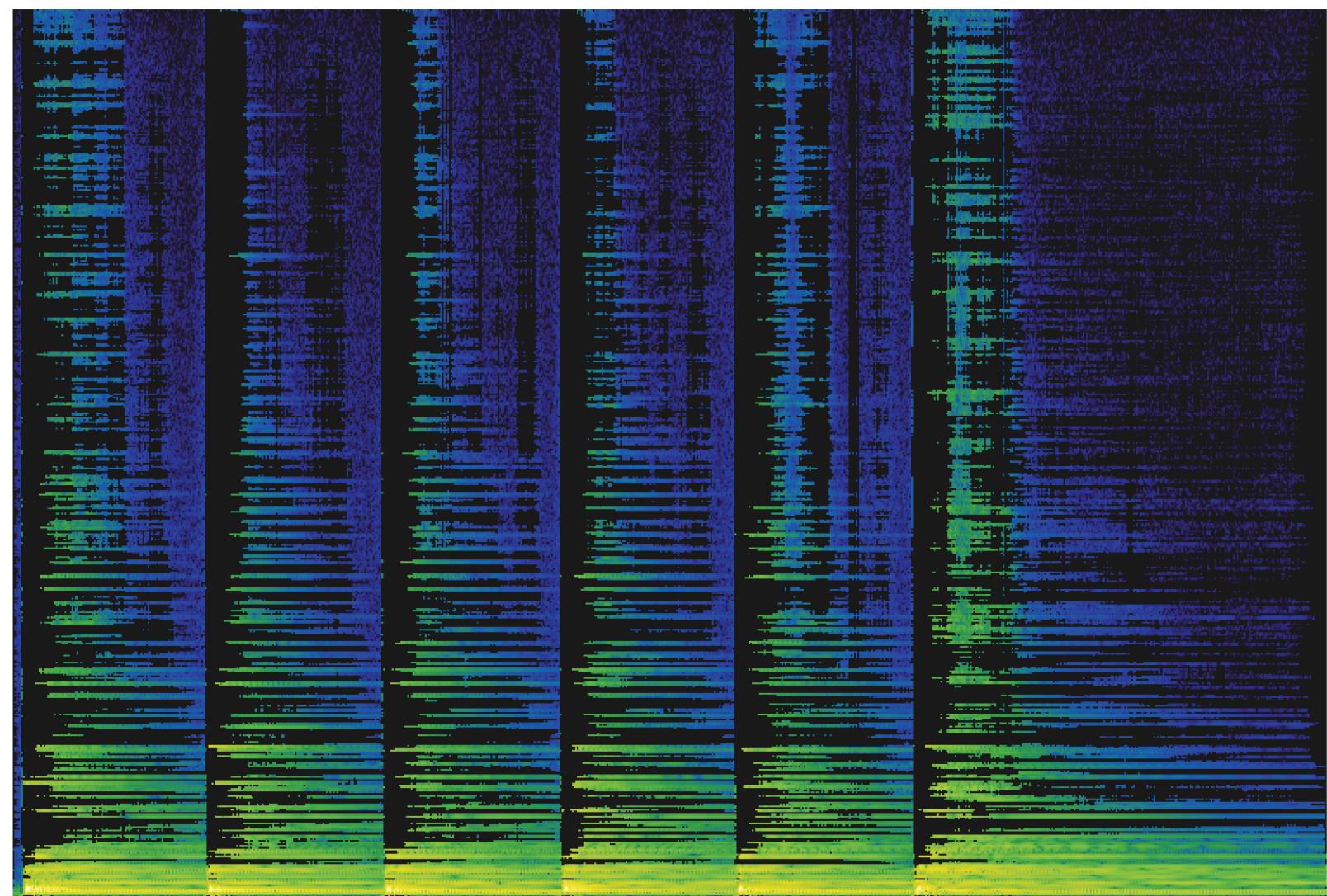
original spectrogram



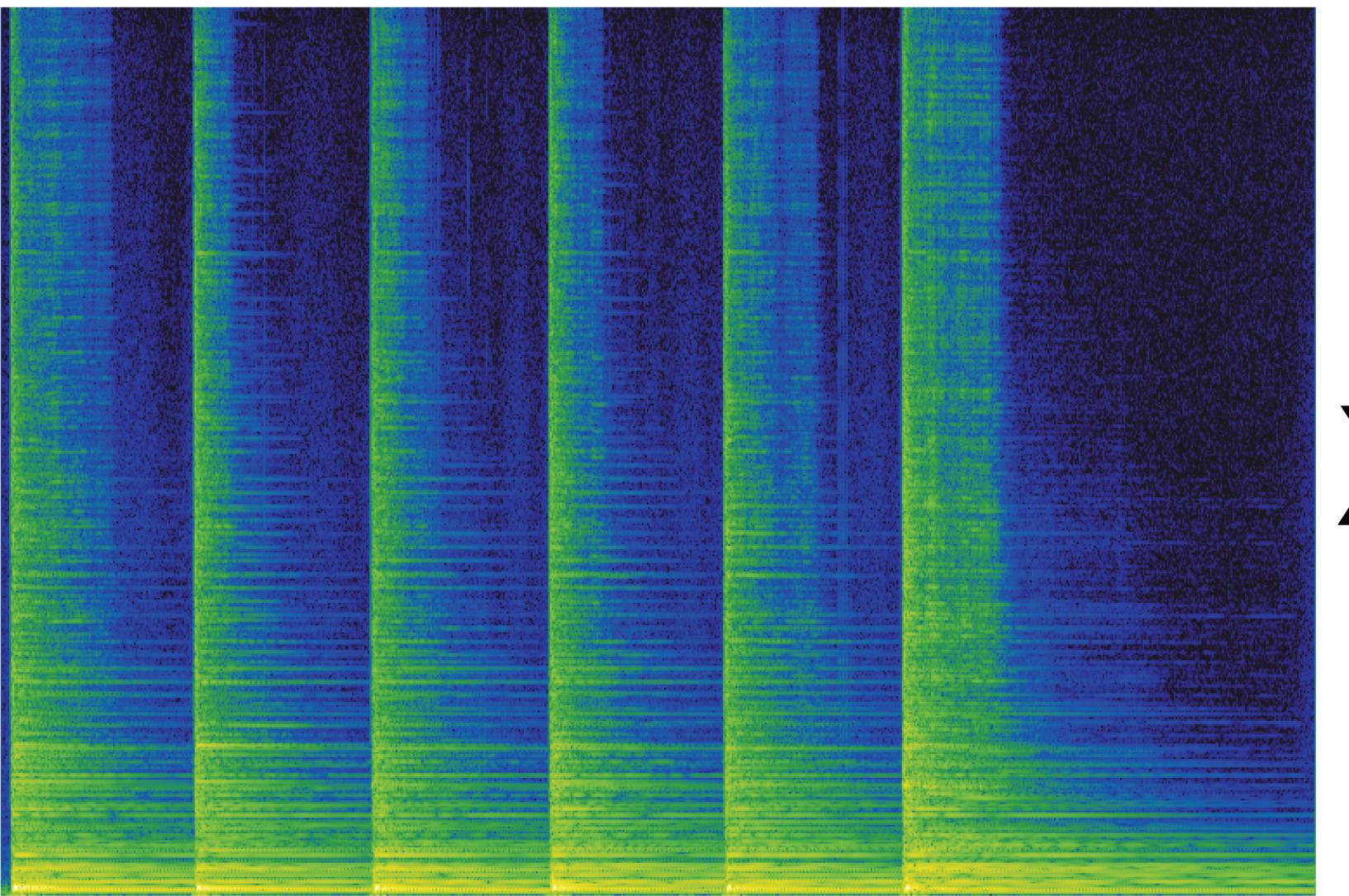
harmonic binary mask



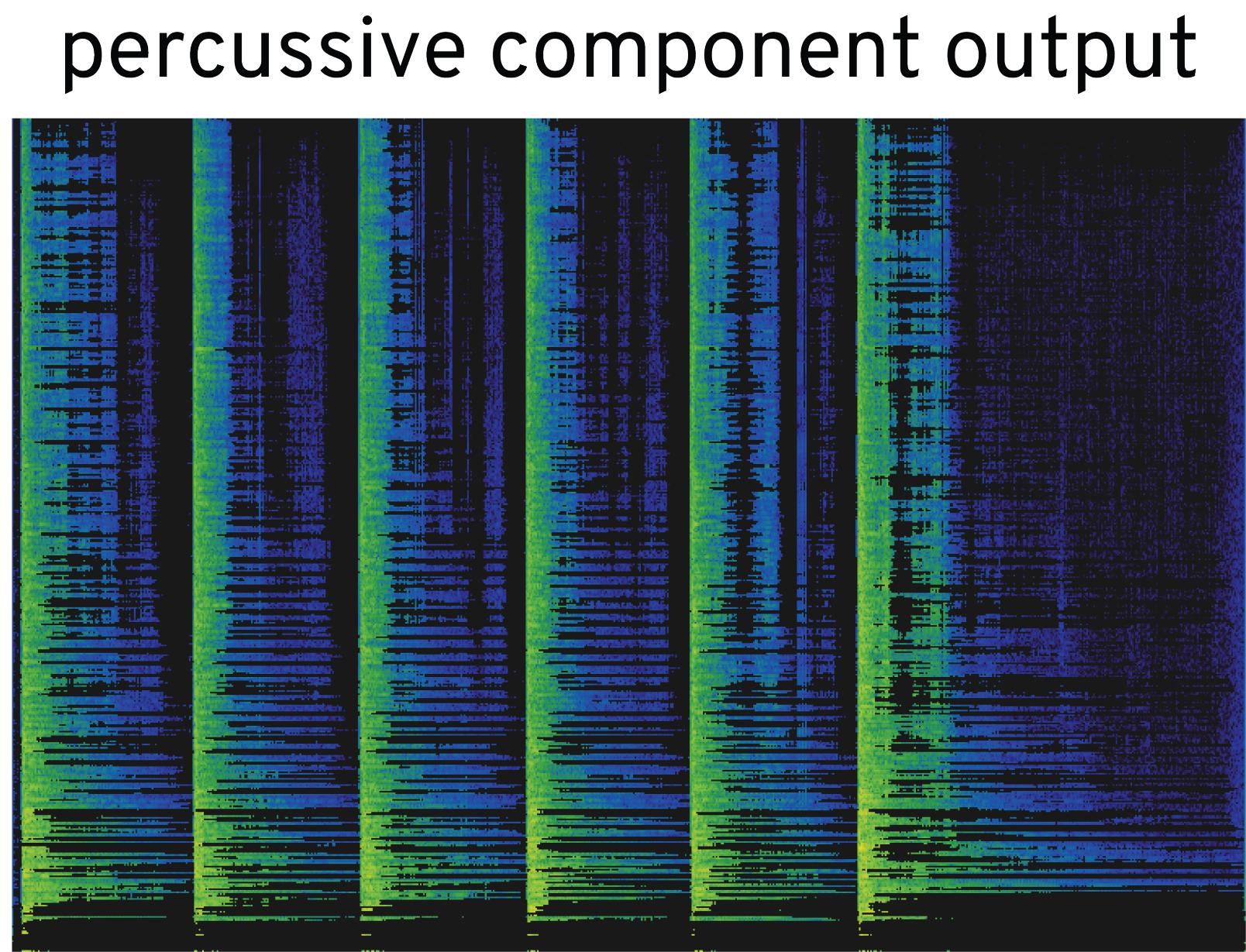
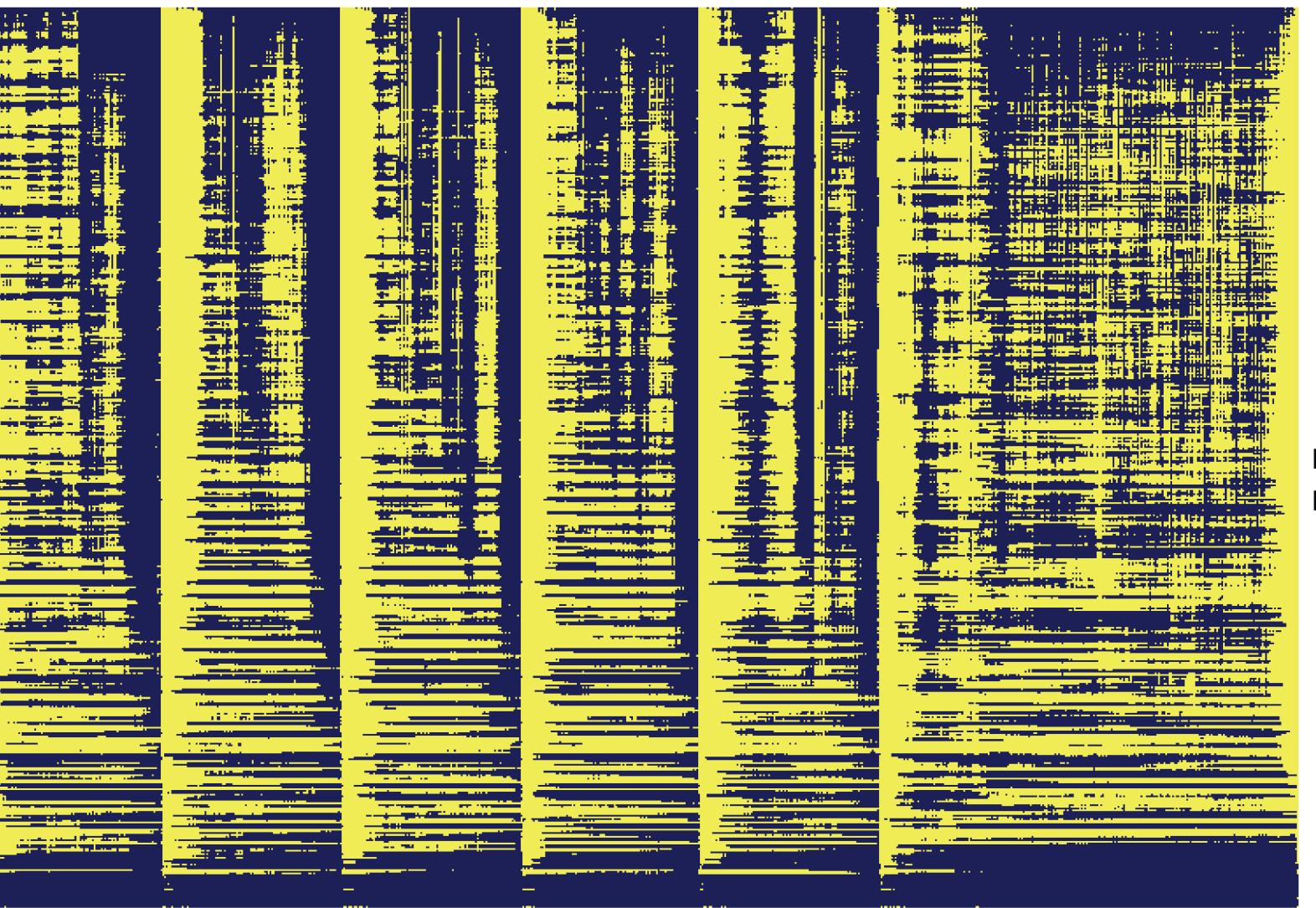
harmonic component output



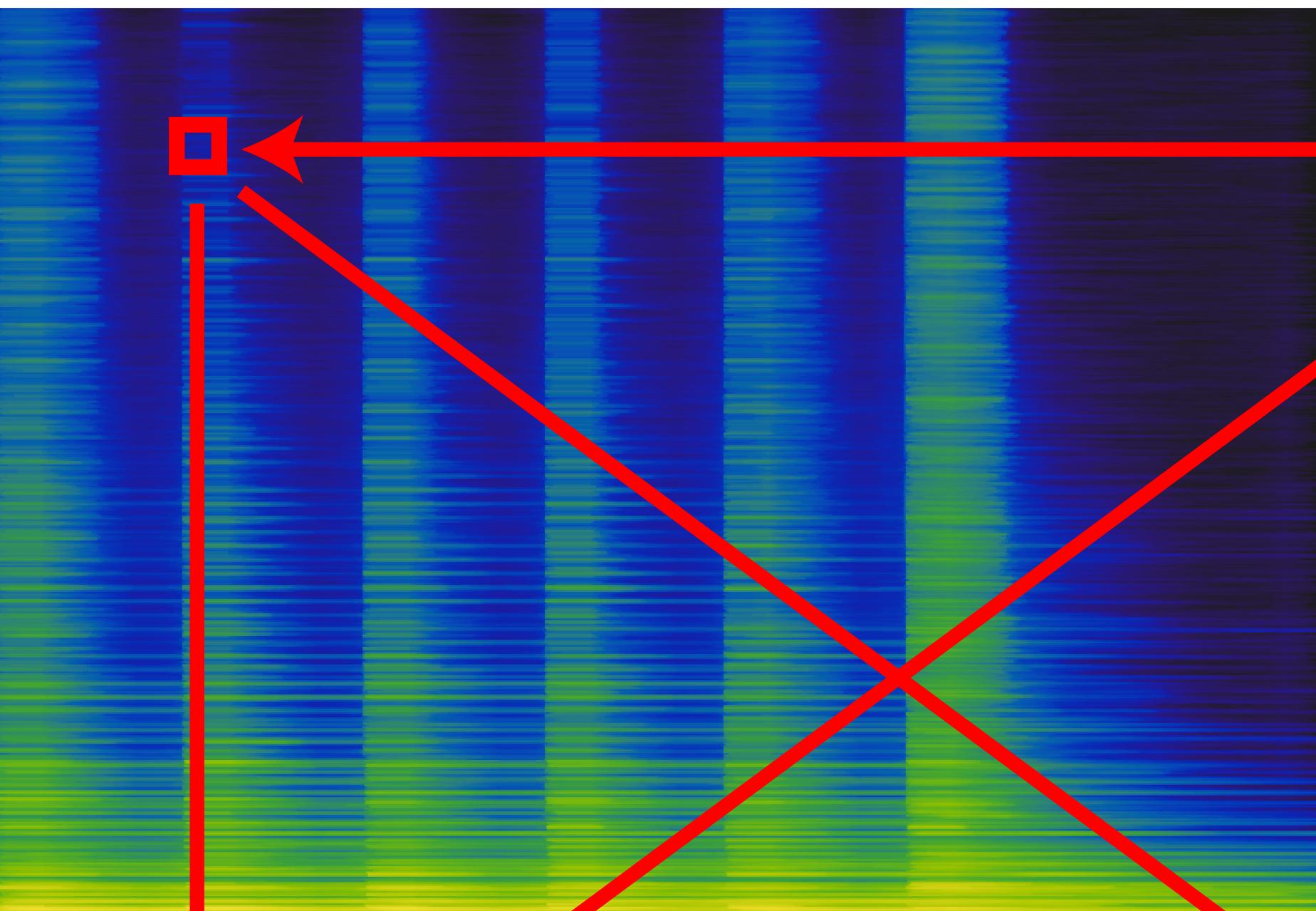
original spectrogram



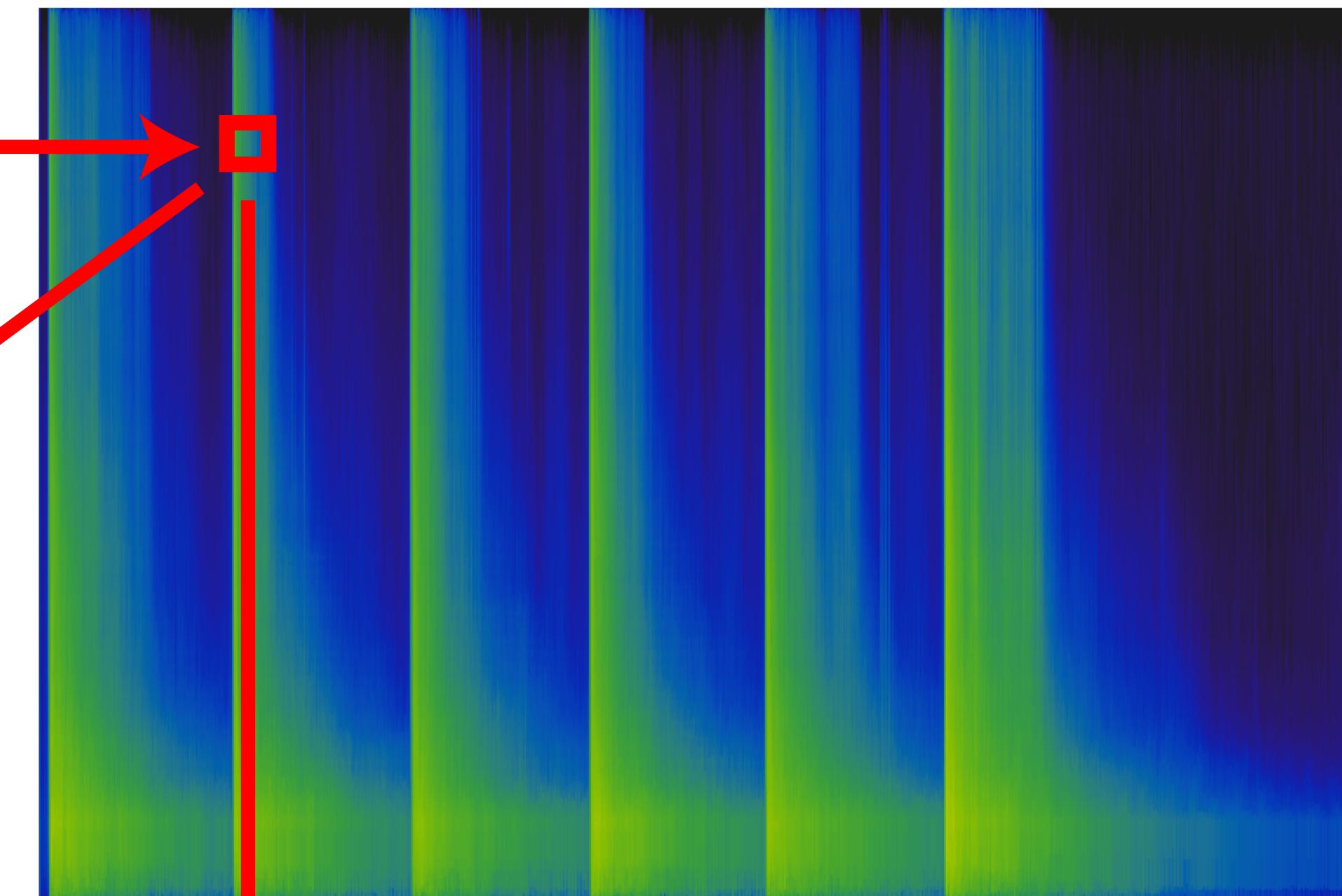
percussive binary mask



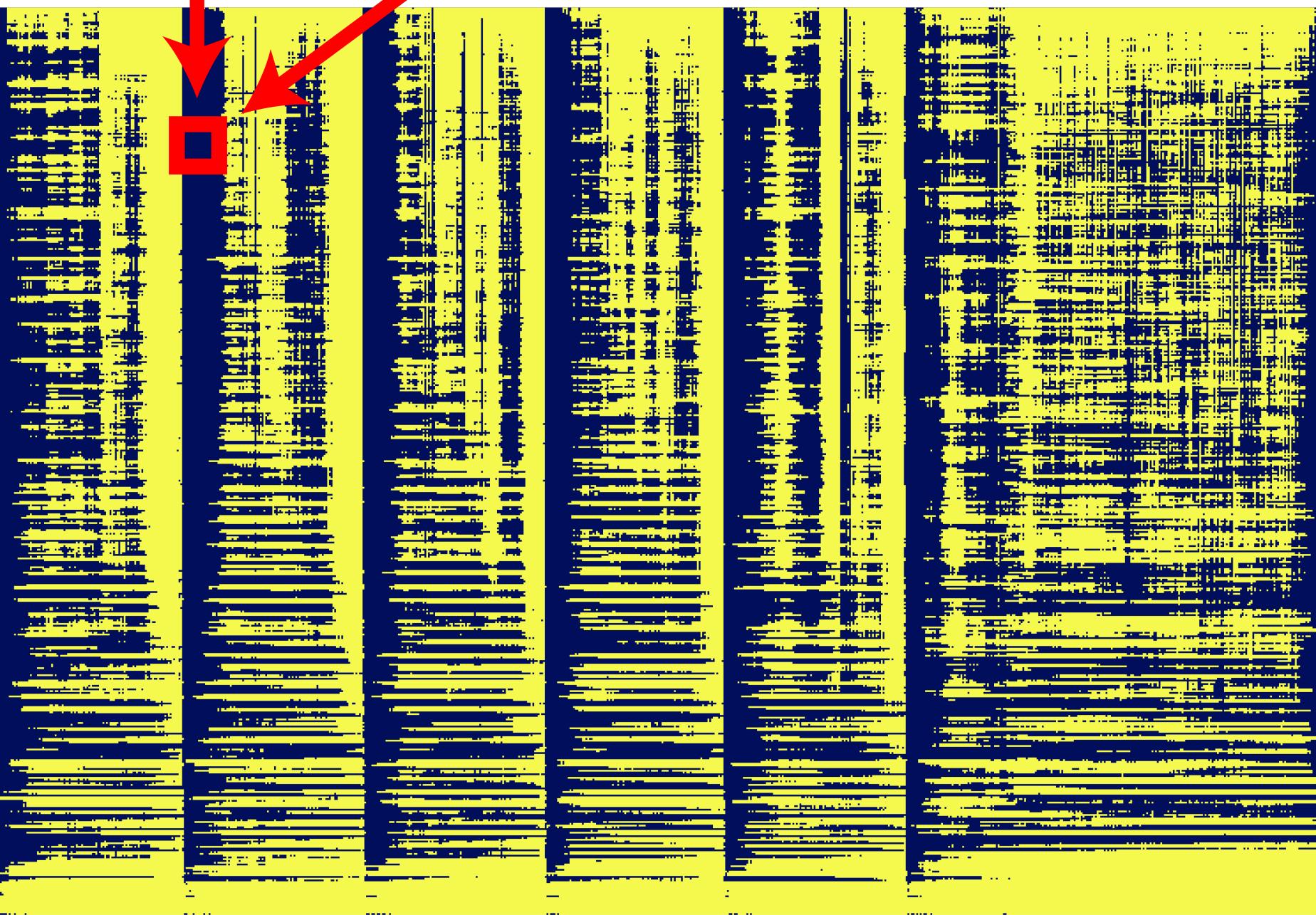
harmonic enhanced spectrogram



percussive enhanced spectrogram



harmonic binary mask



percussive binary mask

