

Unsupervised Machine Learning

via FluCoMa In Max

Music Hackspace



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

some useful precursor knowledge...

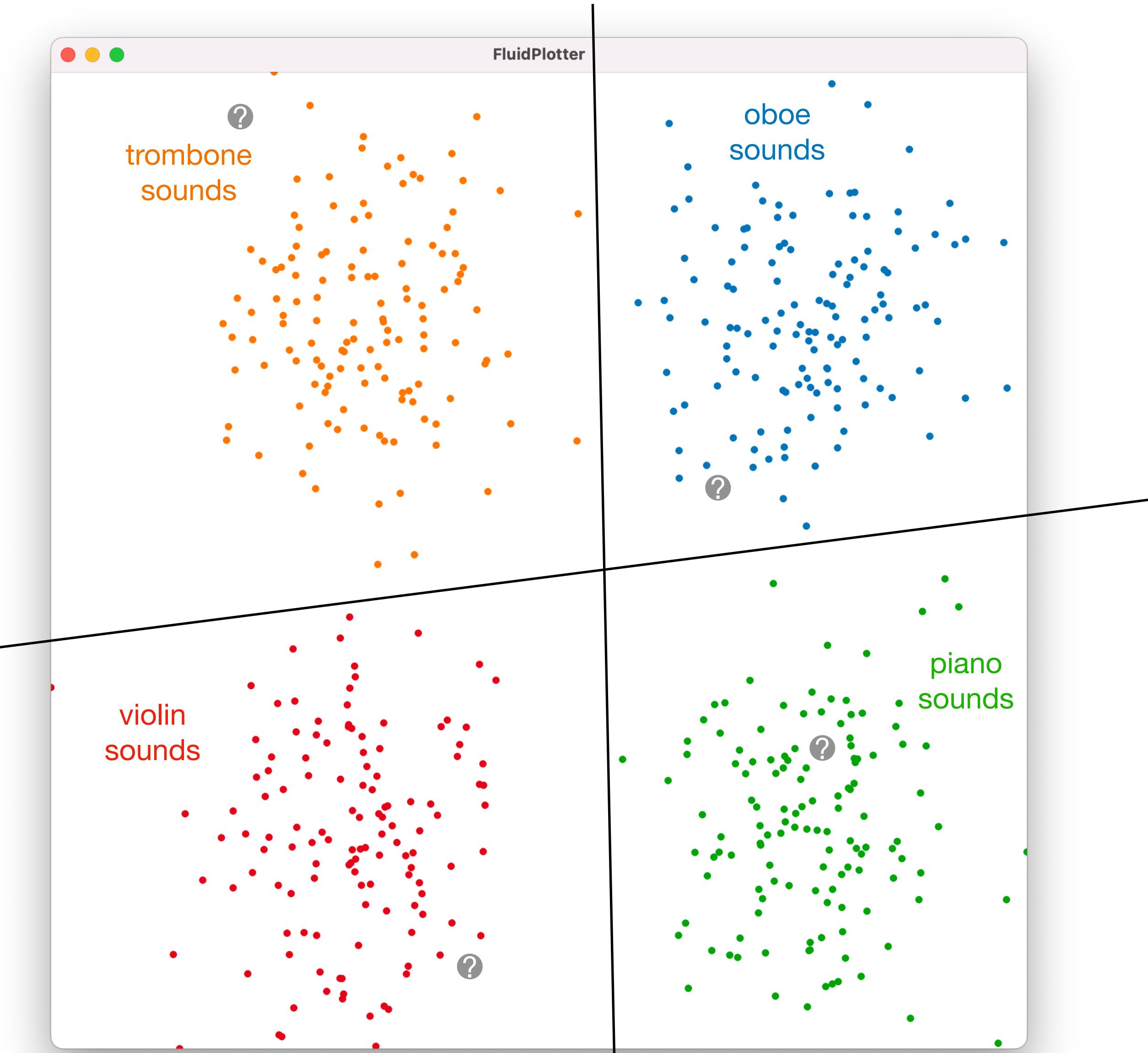
- working with FluCoMa:
 - slicing audio
 - audio analyses
 - fluid.dataset~
 - fluid.labelset~
 - fluidplotter

Plan for Today

- What is Unsupervised Learning?
- Working with Data & Scaling Data
- Dimensionality Reduction with
 - Principal Component Analysis (PCA)
 - Uniform Manifold Approximation and Projection (UMAP)
- Clustering with KMeans
- Comparing Scalers
- Linear Assignment with the Jonker-Volgenant algorithm (Gridification)

Supervised Learning

Learning patterns, associations, or relationships from data that is pre-labeled

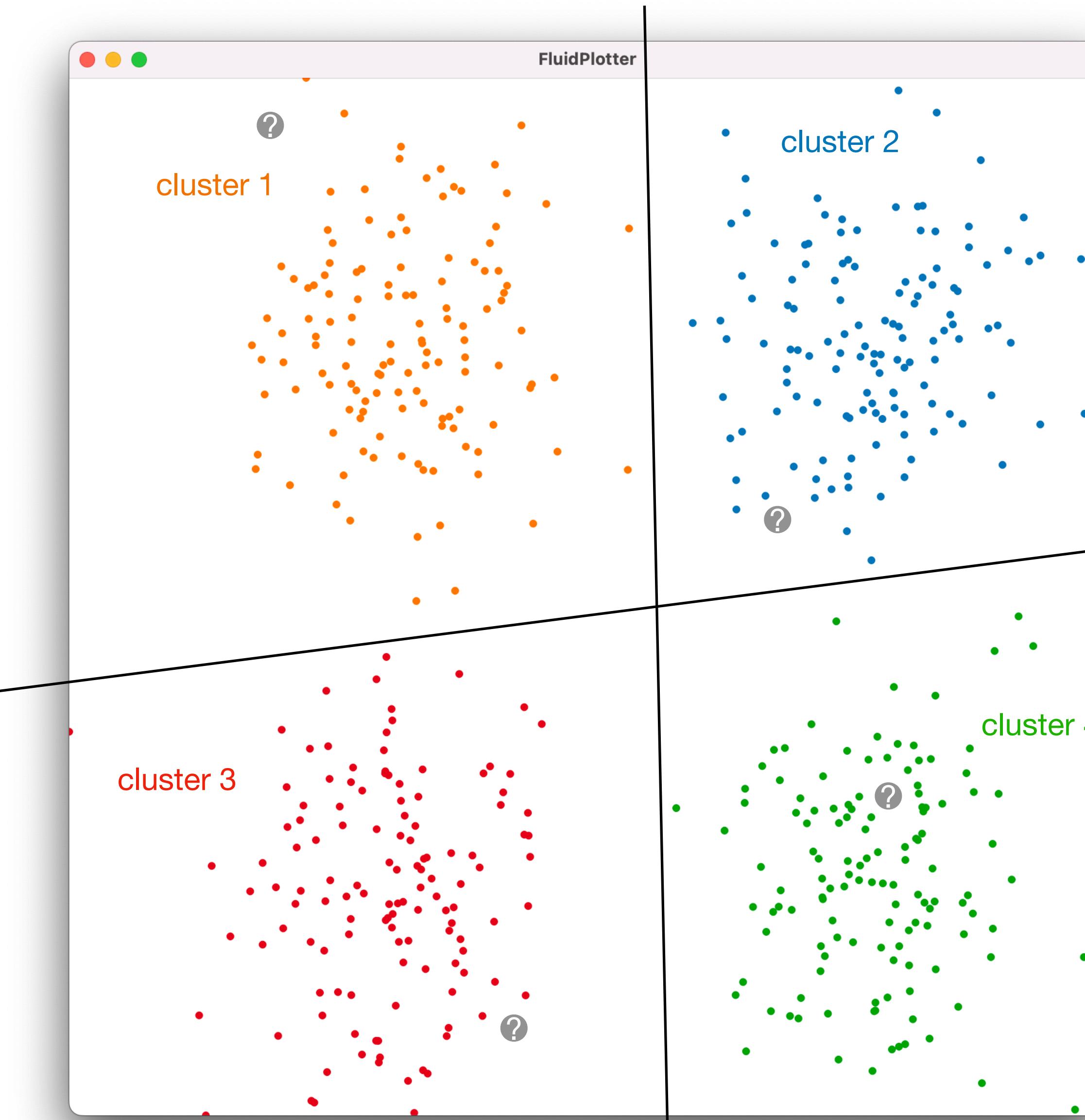


Classification
Regression

Neural Networks
KNN

Unsupervised Learning

Learning/finding patterns and relationships in data that is not labeled



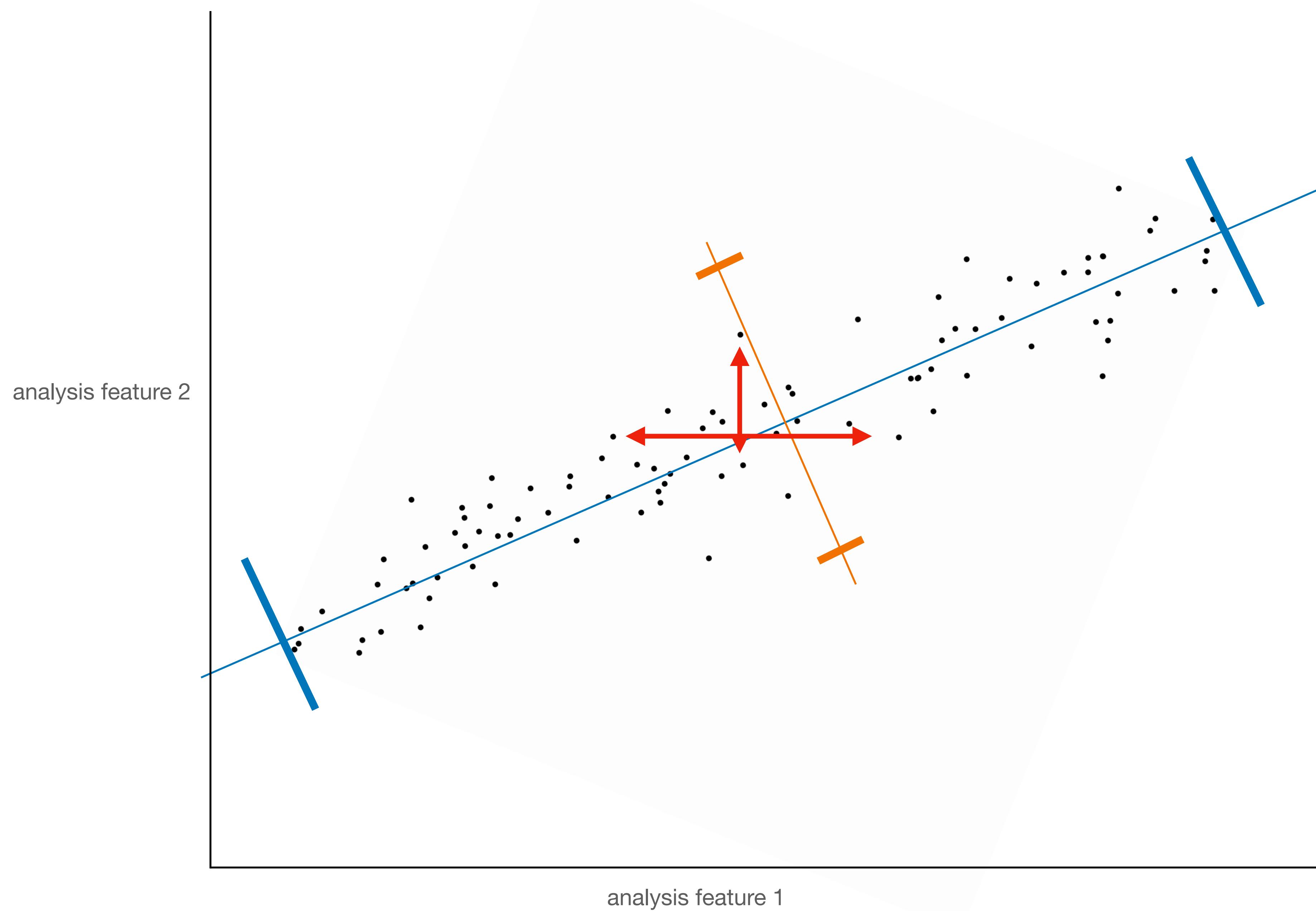
Dimensionality Reduction

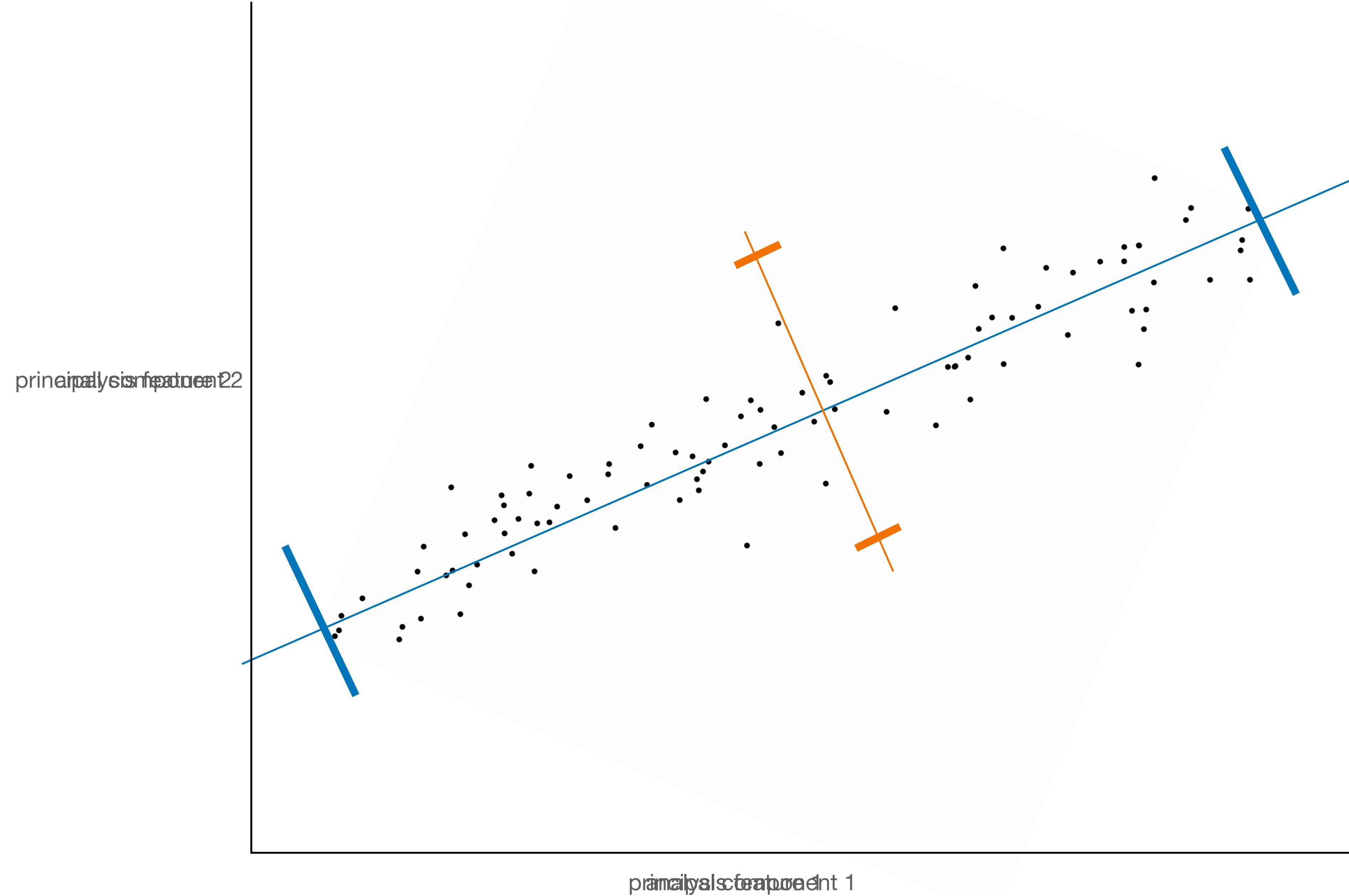
Clustering

Feature Learning

Principal Component Analysis (PCA)

1. analyze (or “fit to”) a dataset
2. use that analysis to transform the data to show more of the “variance” (or “differences”) between data points
 - can help remove redundancy
 - can help remove noise
 - can offer dimensionality reduction





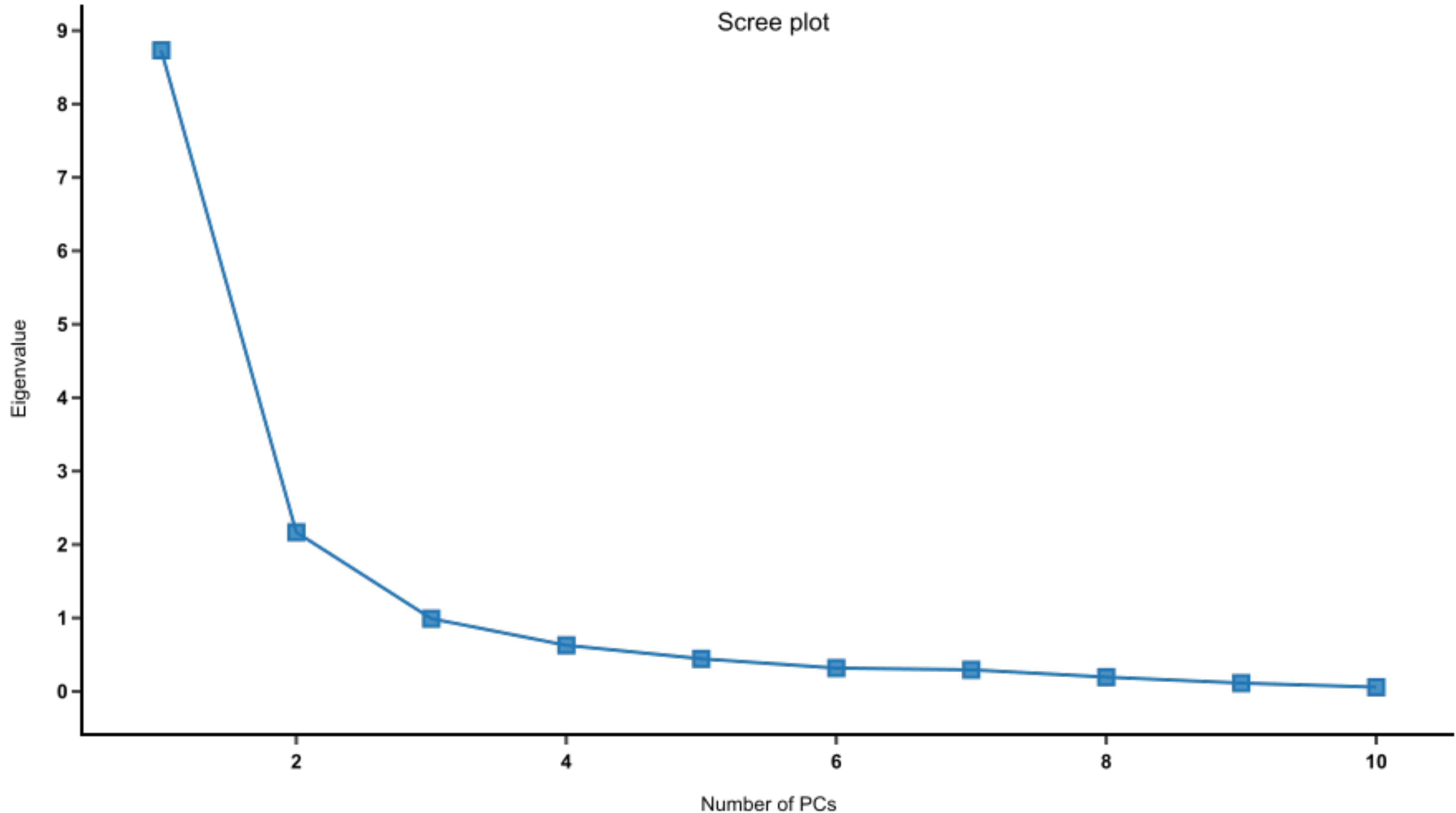
13 dimensions in → 13 PCs out

dimensions

13 dimensions in → 13 PCs out

principal components

Scree plot



13 dimensions in → 13 PCs out

principal components

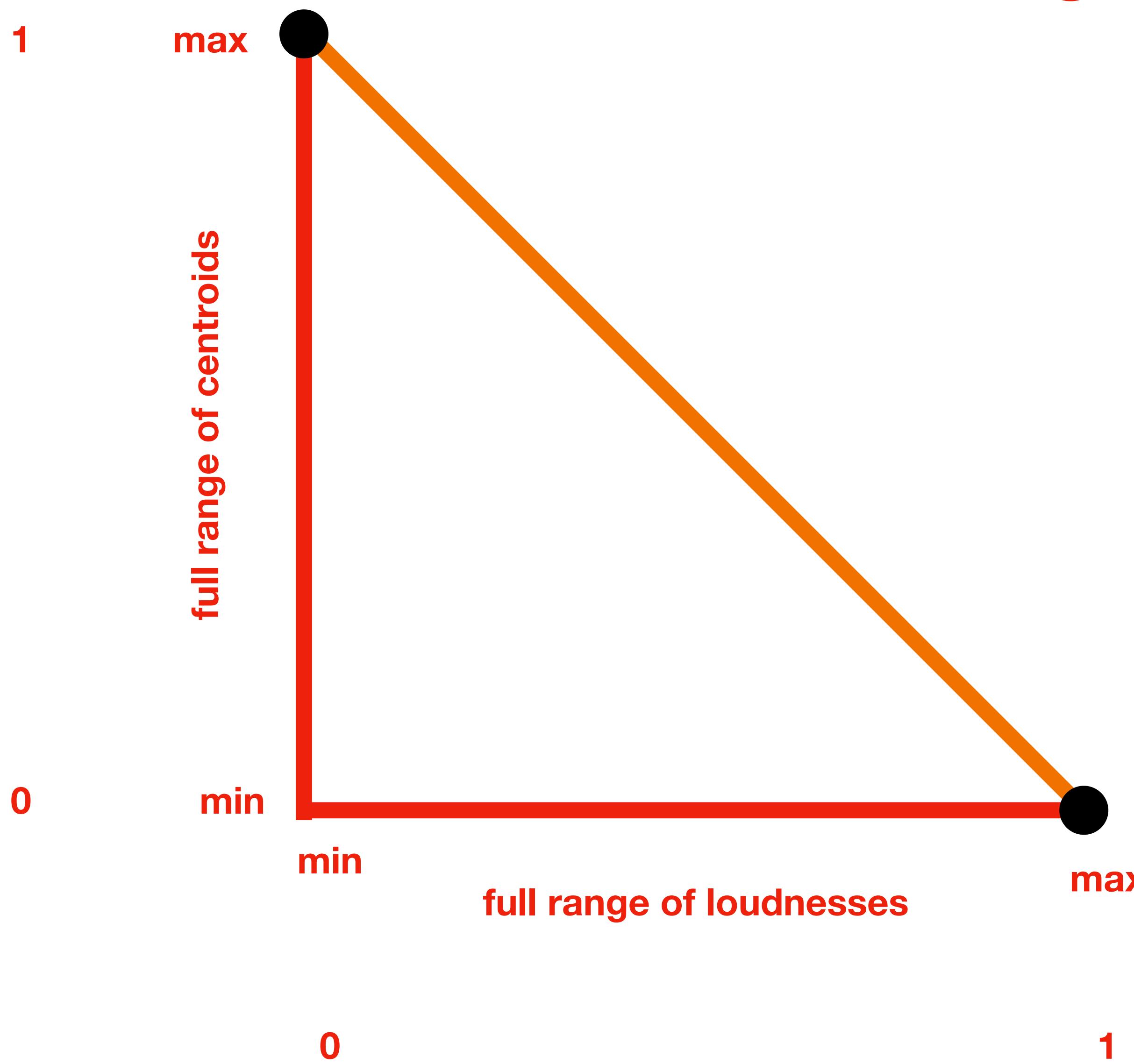
Scaling



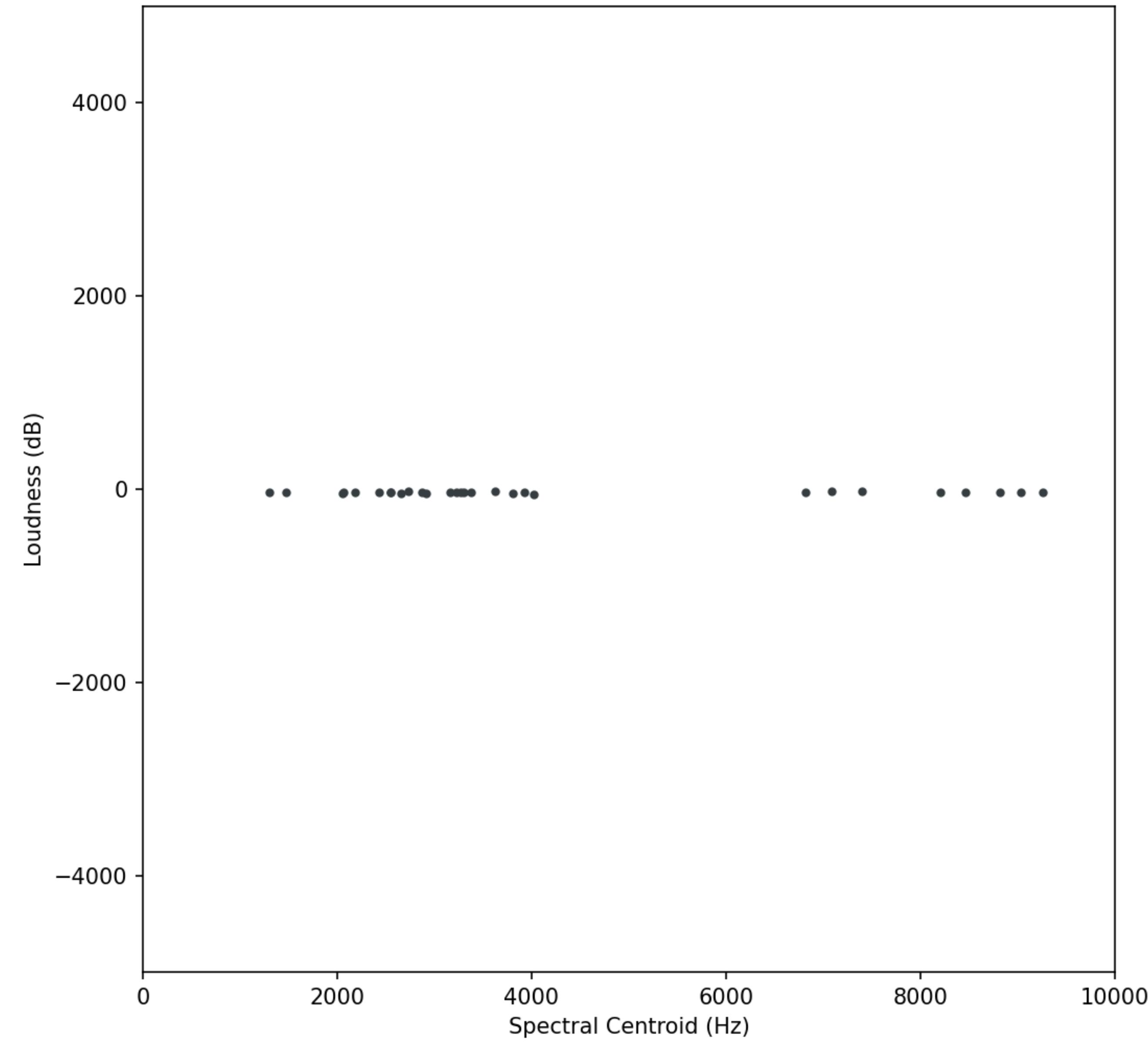
If 1 Hz = 1dB



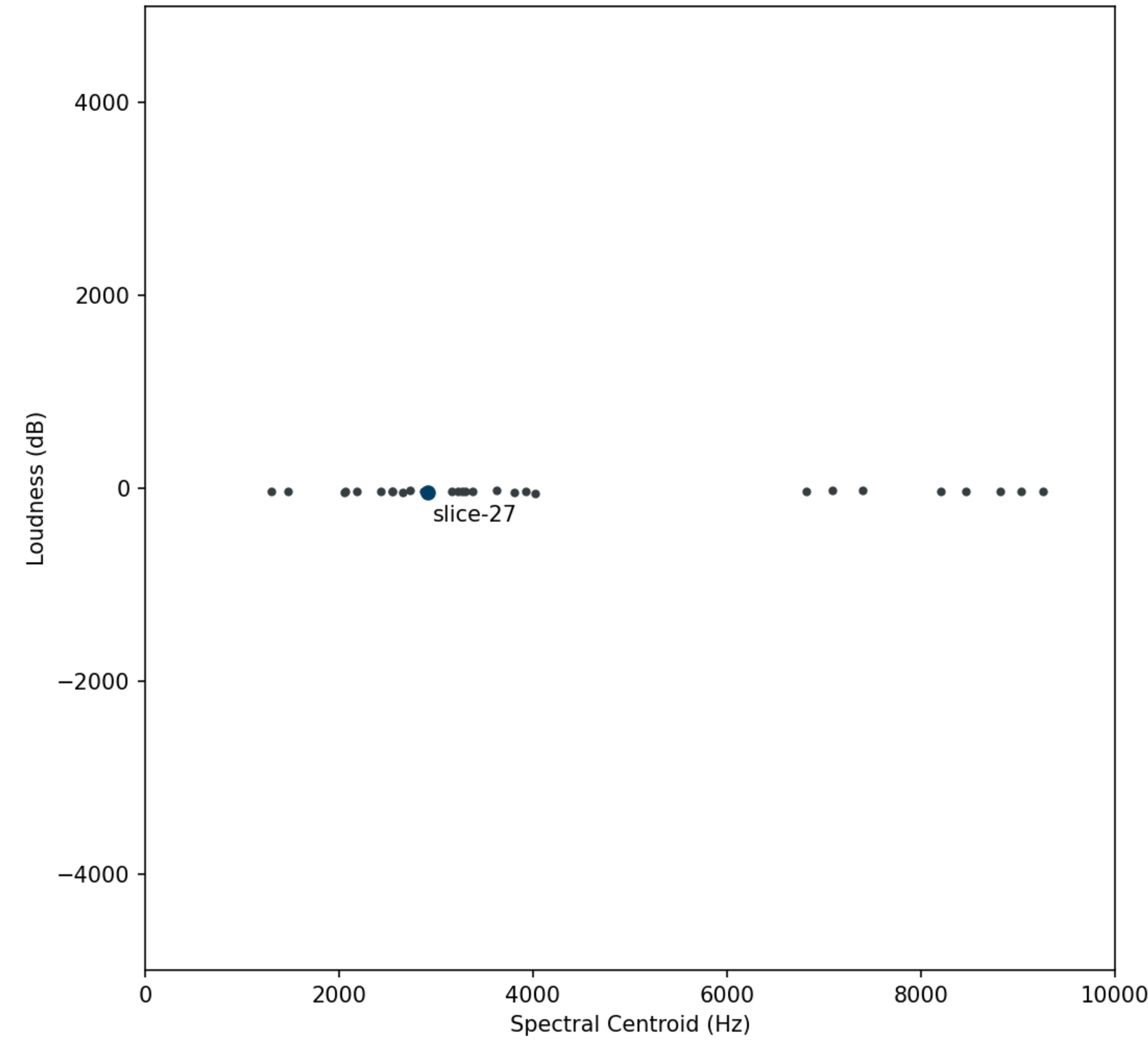
Normalization



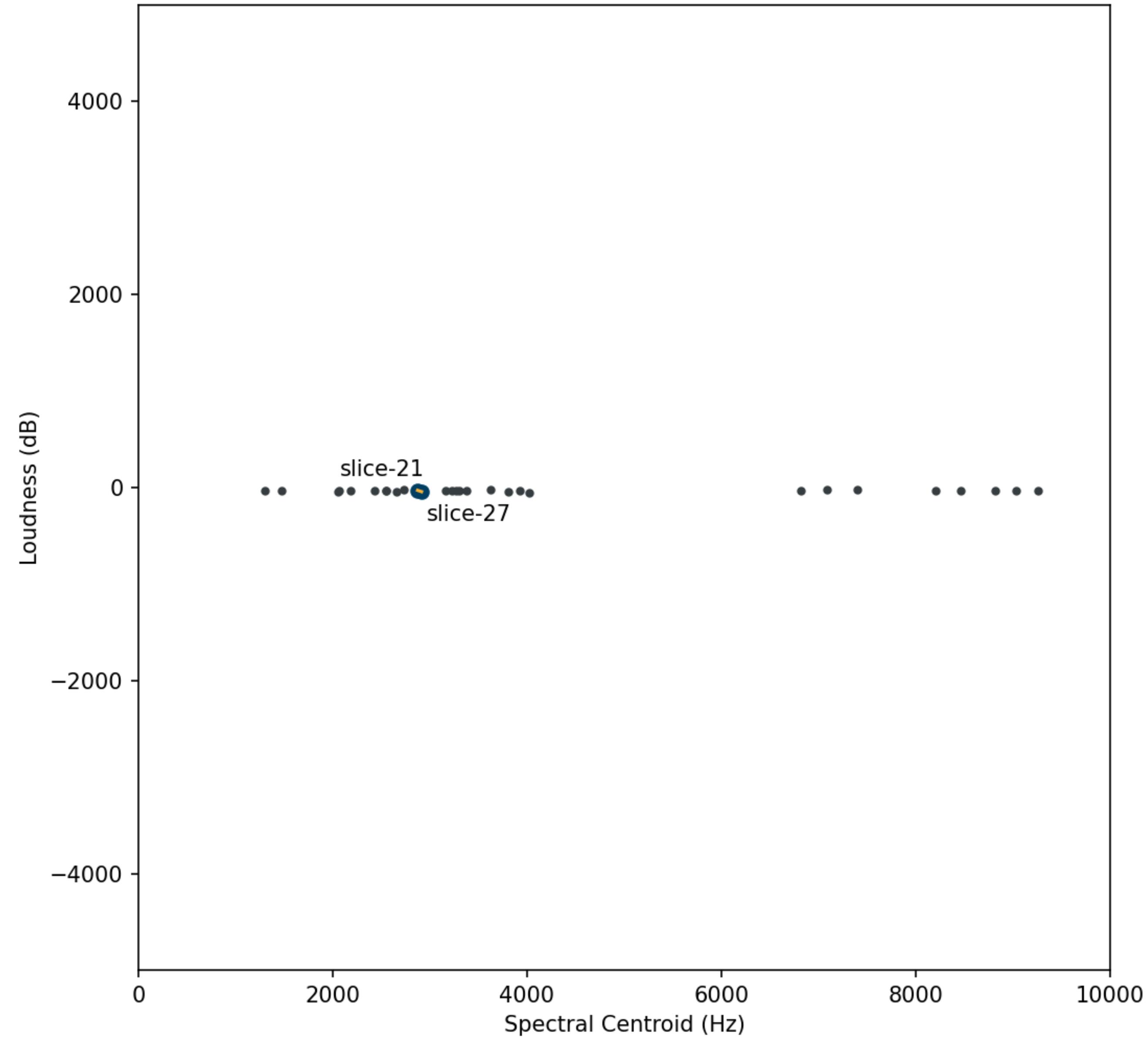
Raw scaling: 1 Hz = 1 dB



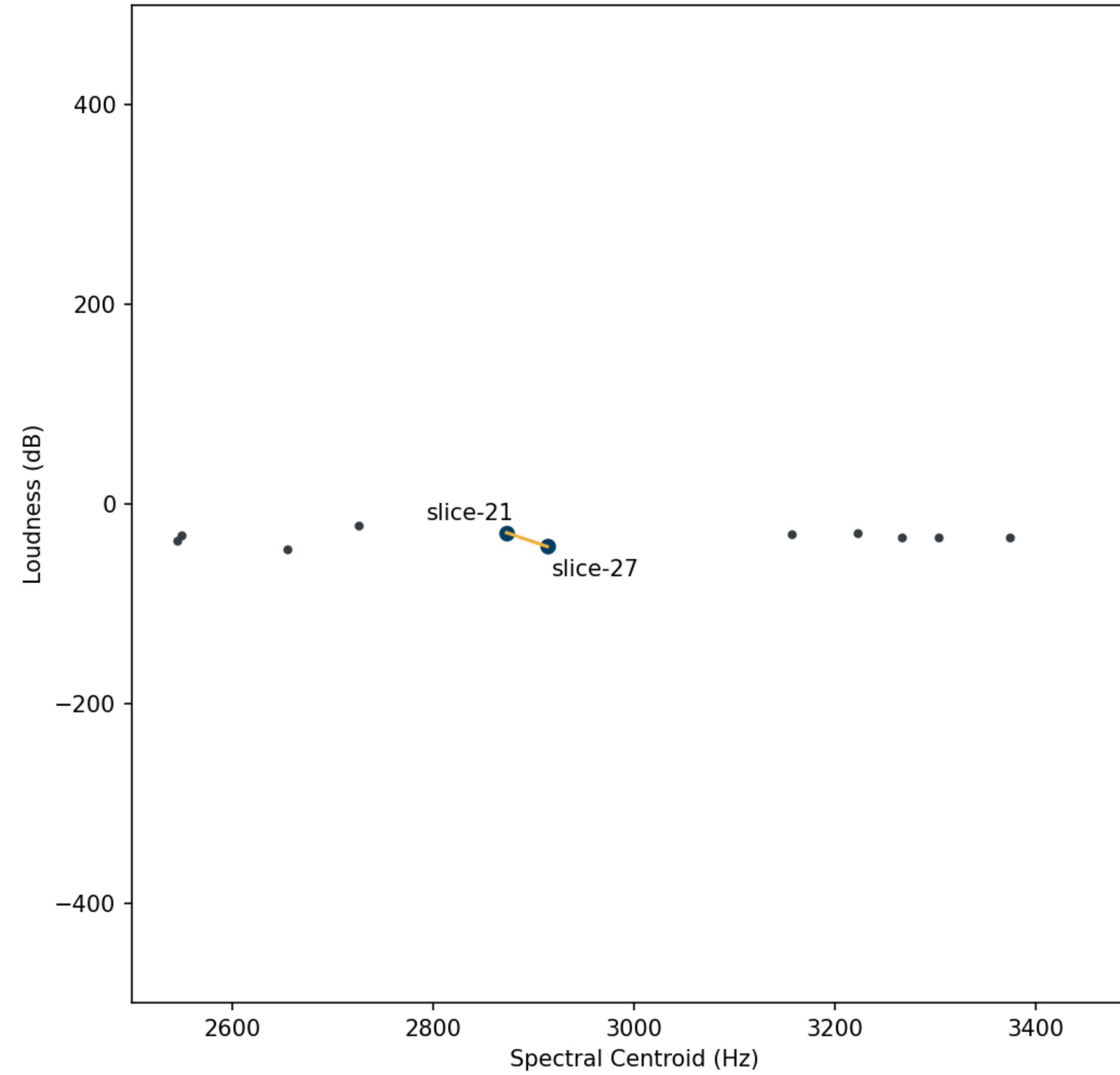
Raw scaling: 1 Hz = 1 dB



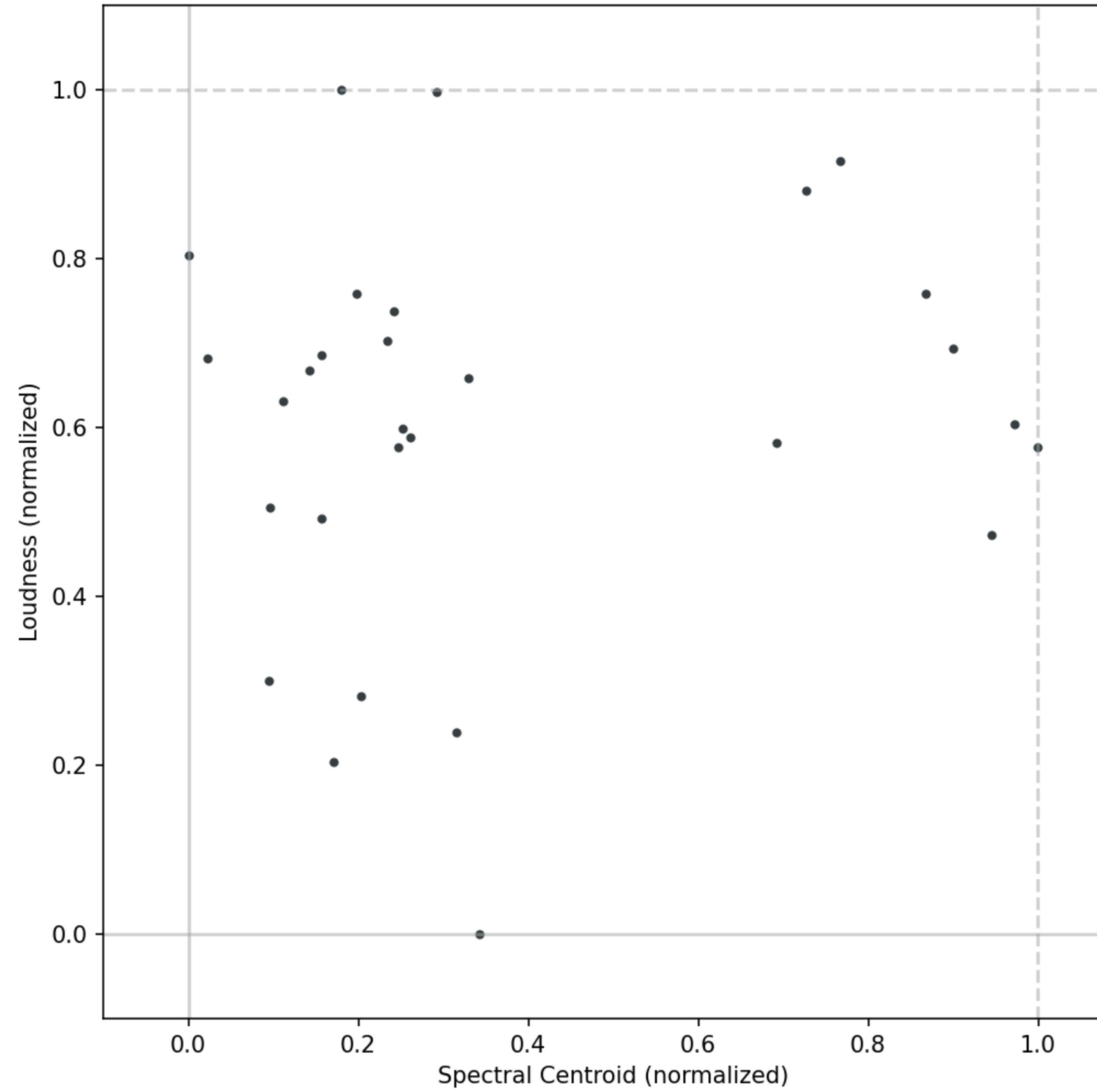
Raw scaling: 1 Hz = 1 dB



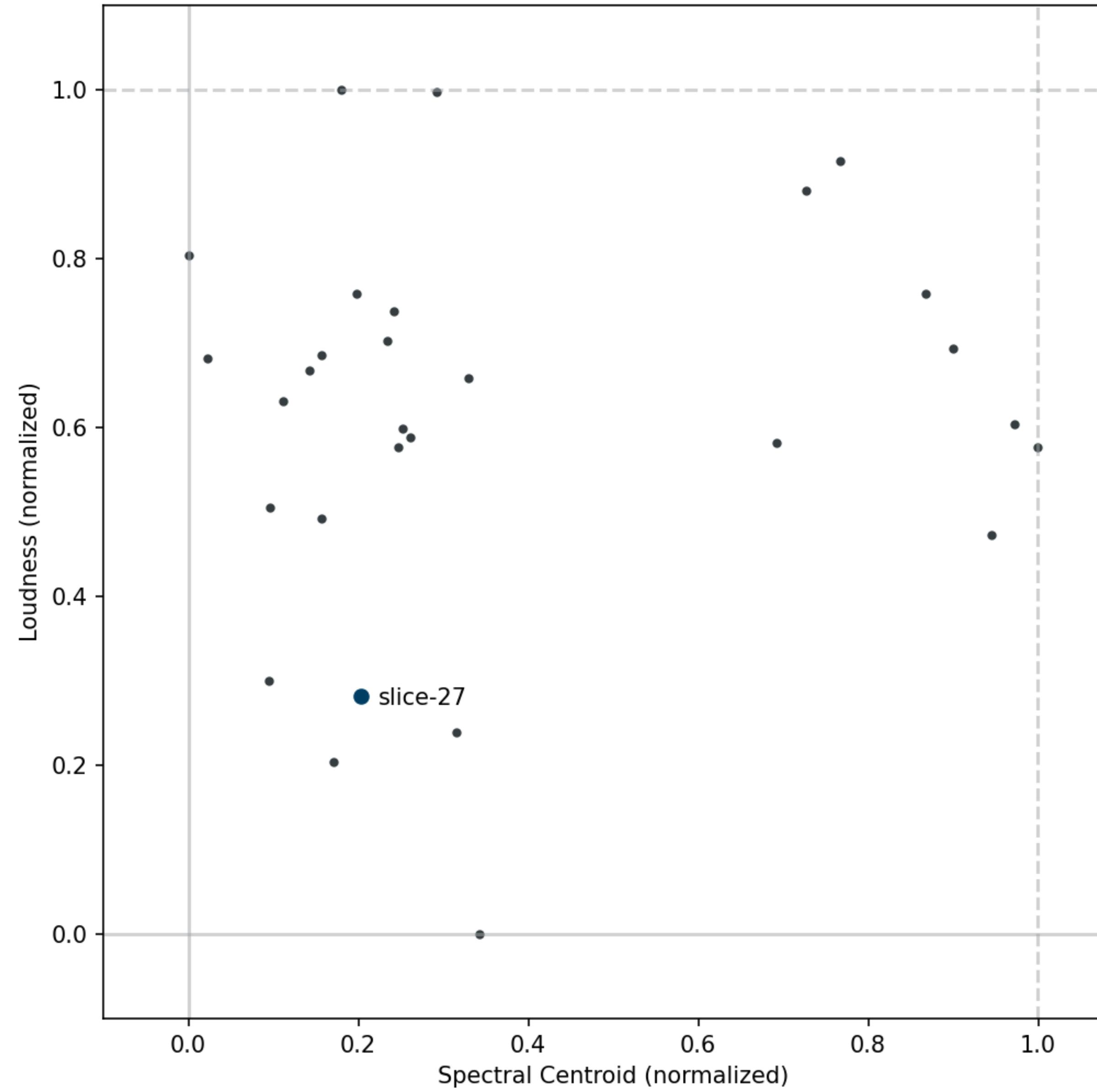
Raw scaling: 1 Hz = 1 dB



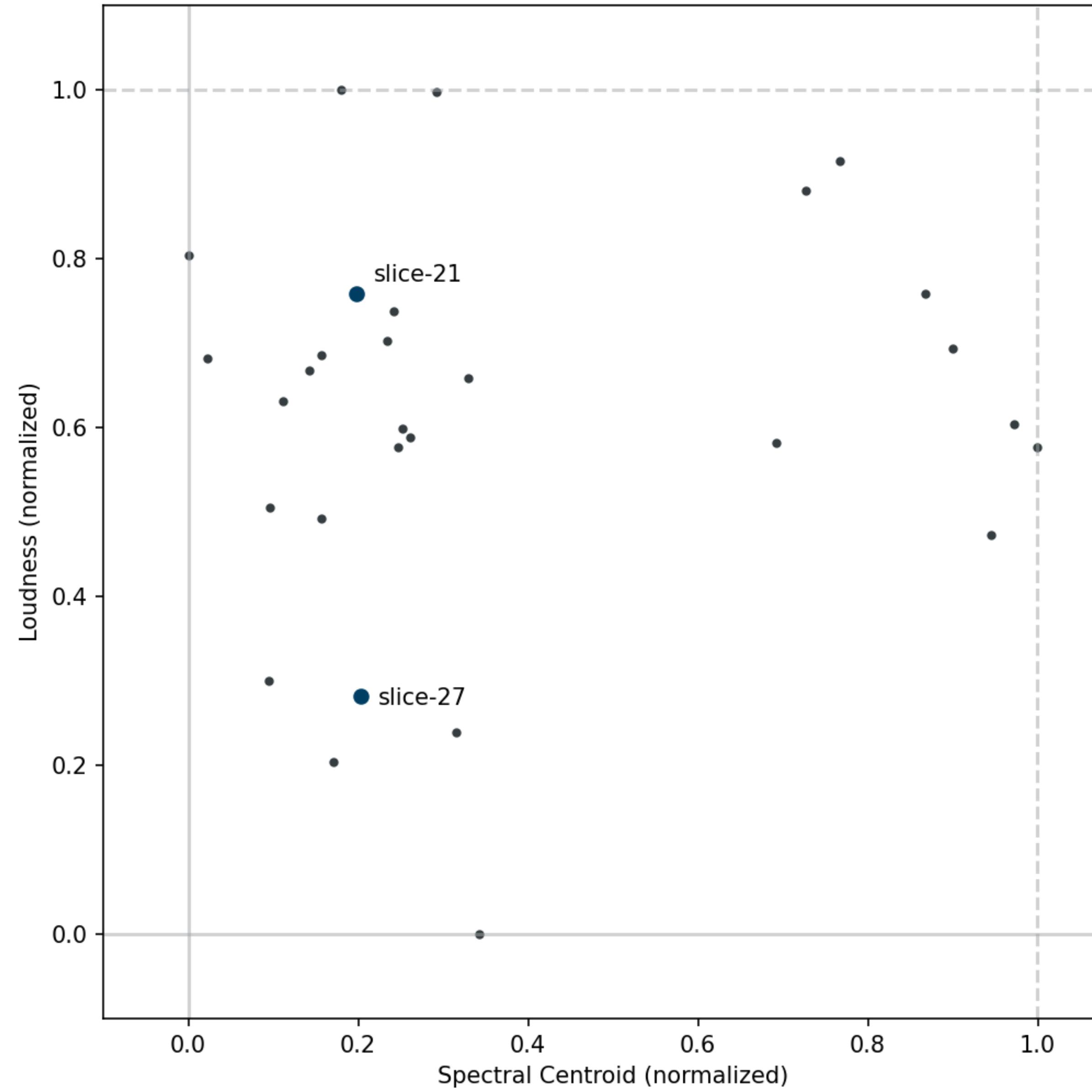
Normalized (min = 0, max = 1)



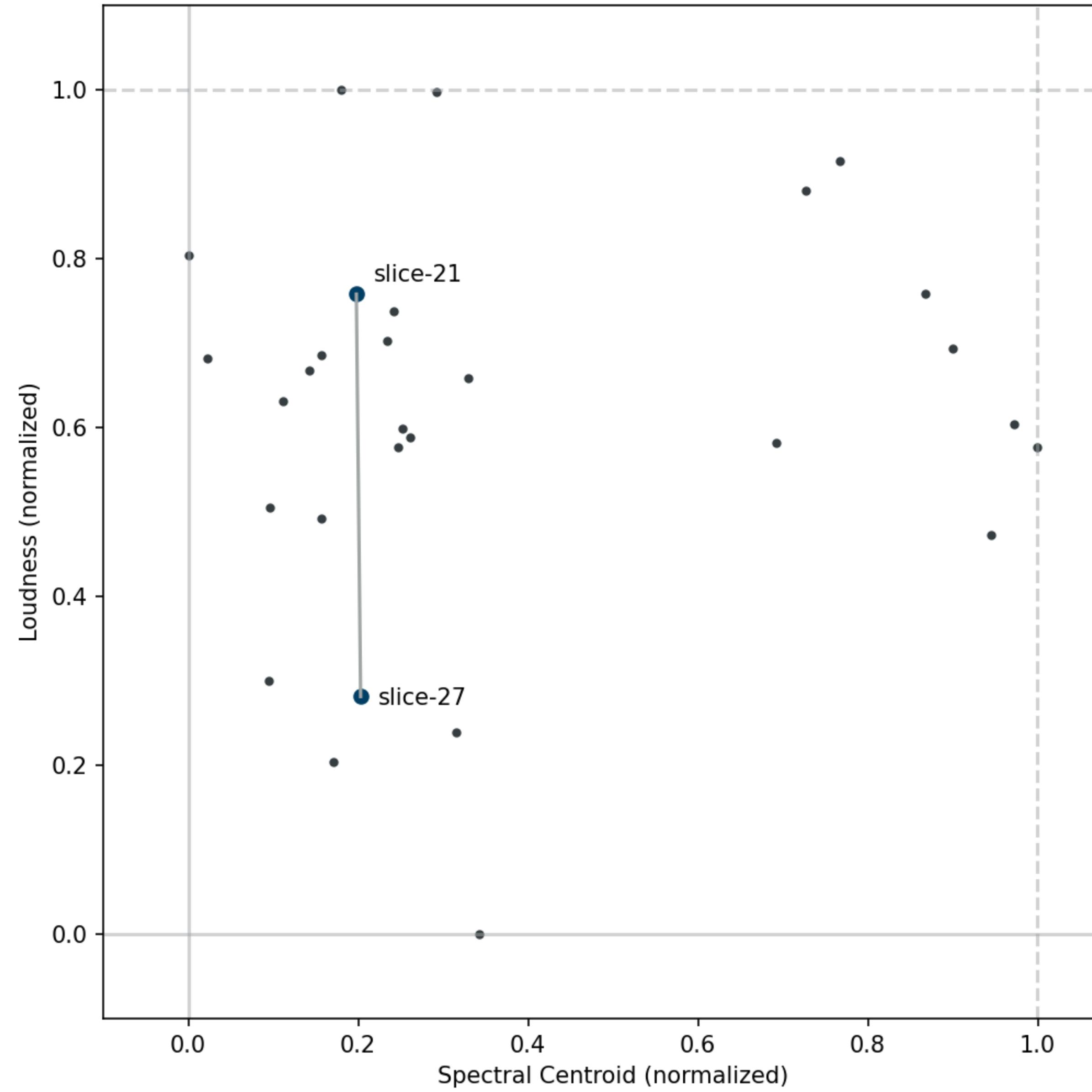
Normalized (min = 0, max = 1)



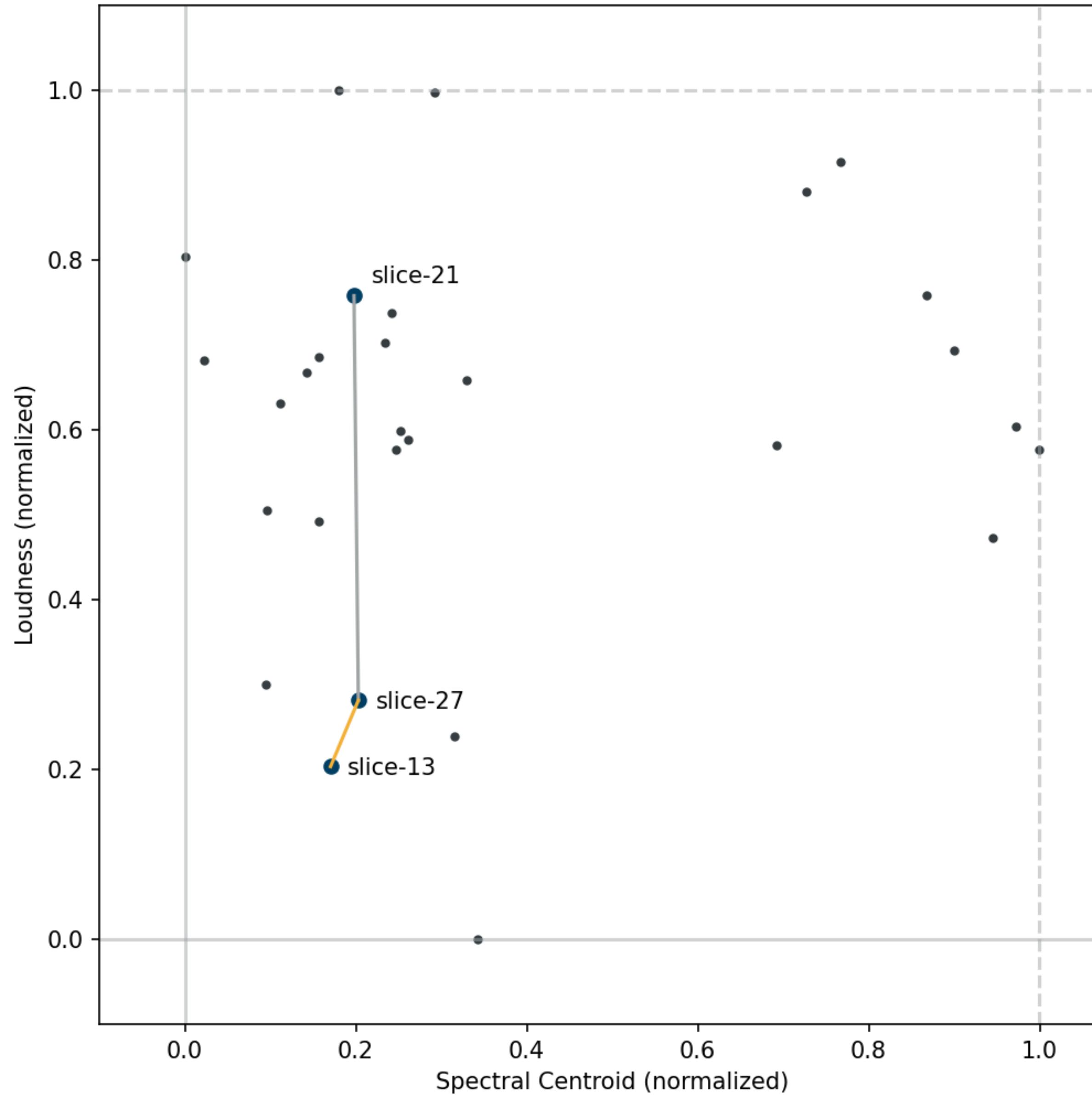
Normalized (min = 0, max = 1)



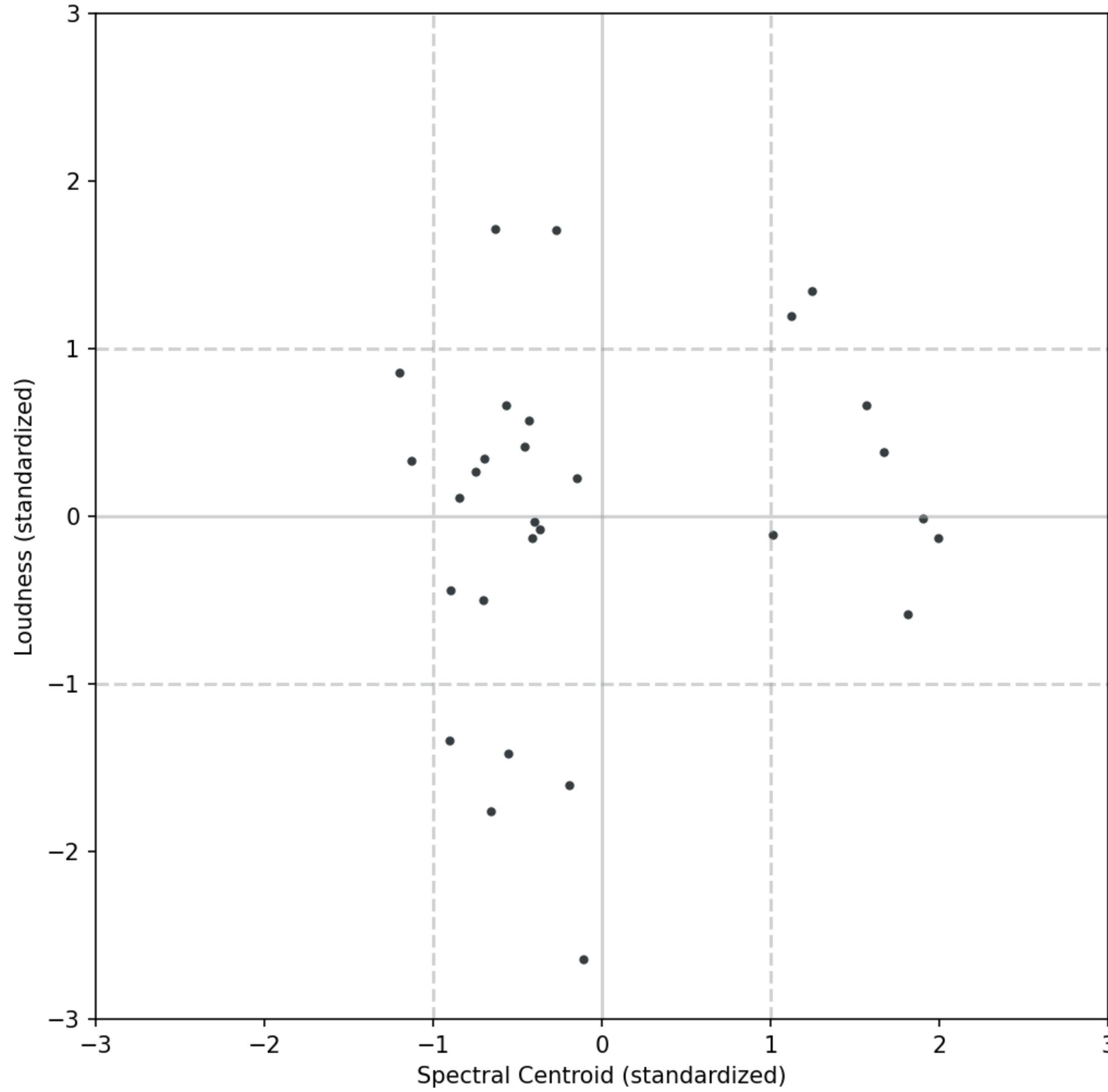
Normalized (min = 0, max = 1)



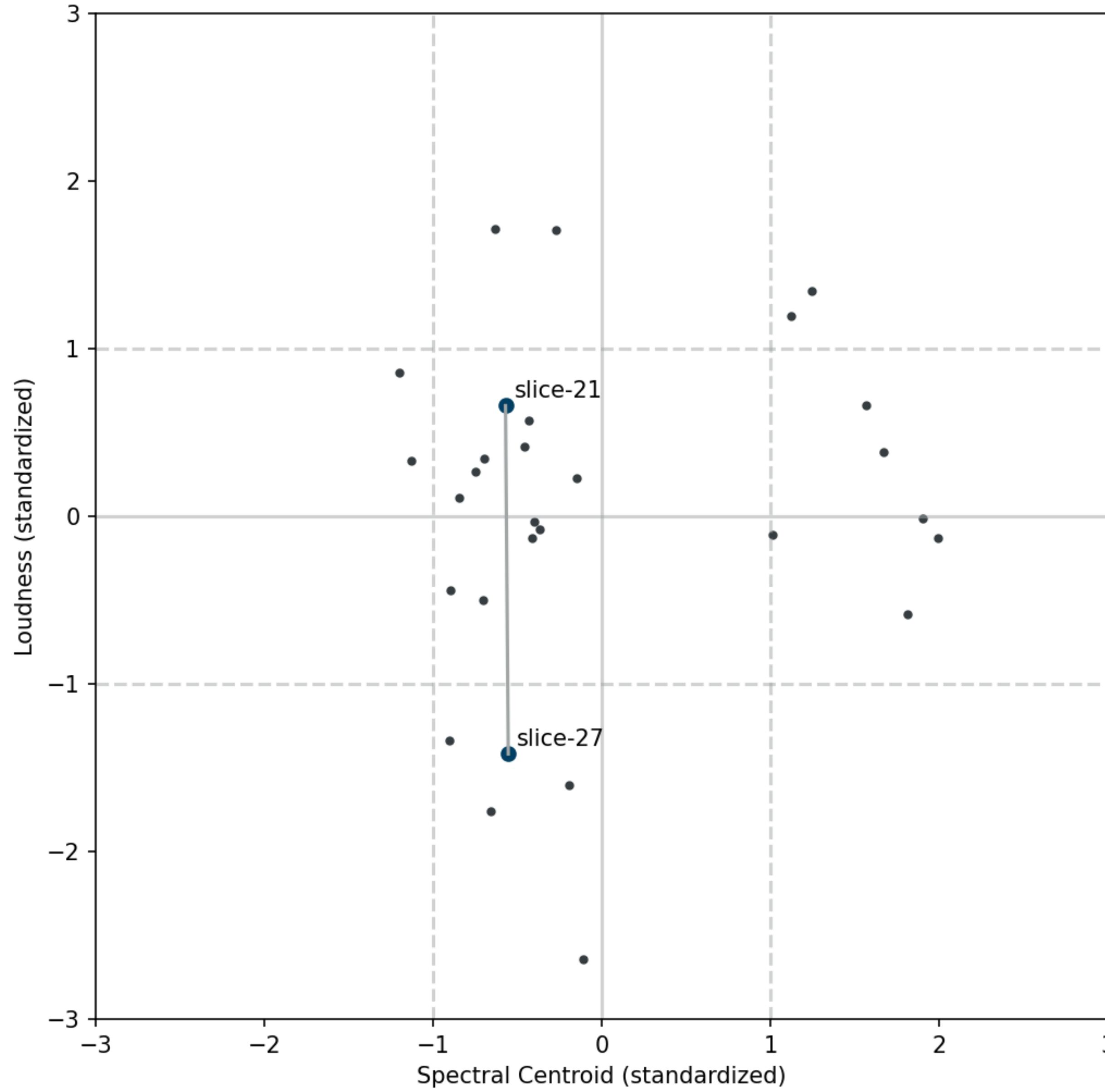
Normalized (min = 0, max = 1)



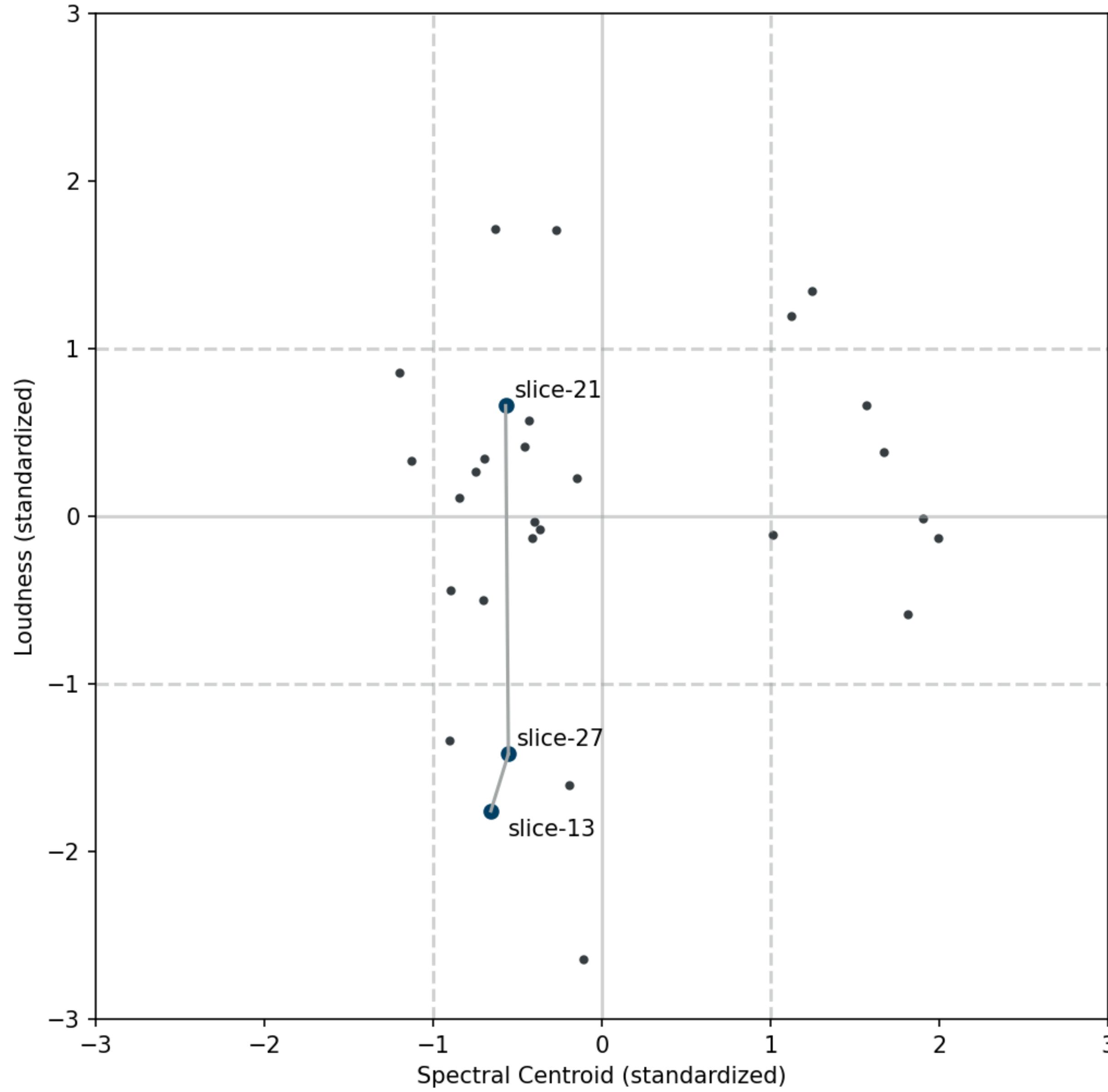
Standardized (mean = 0, standard deviation = 1)



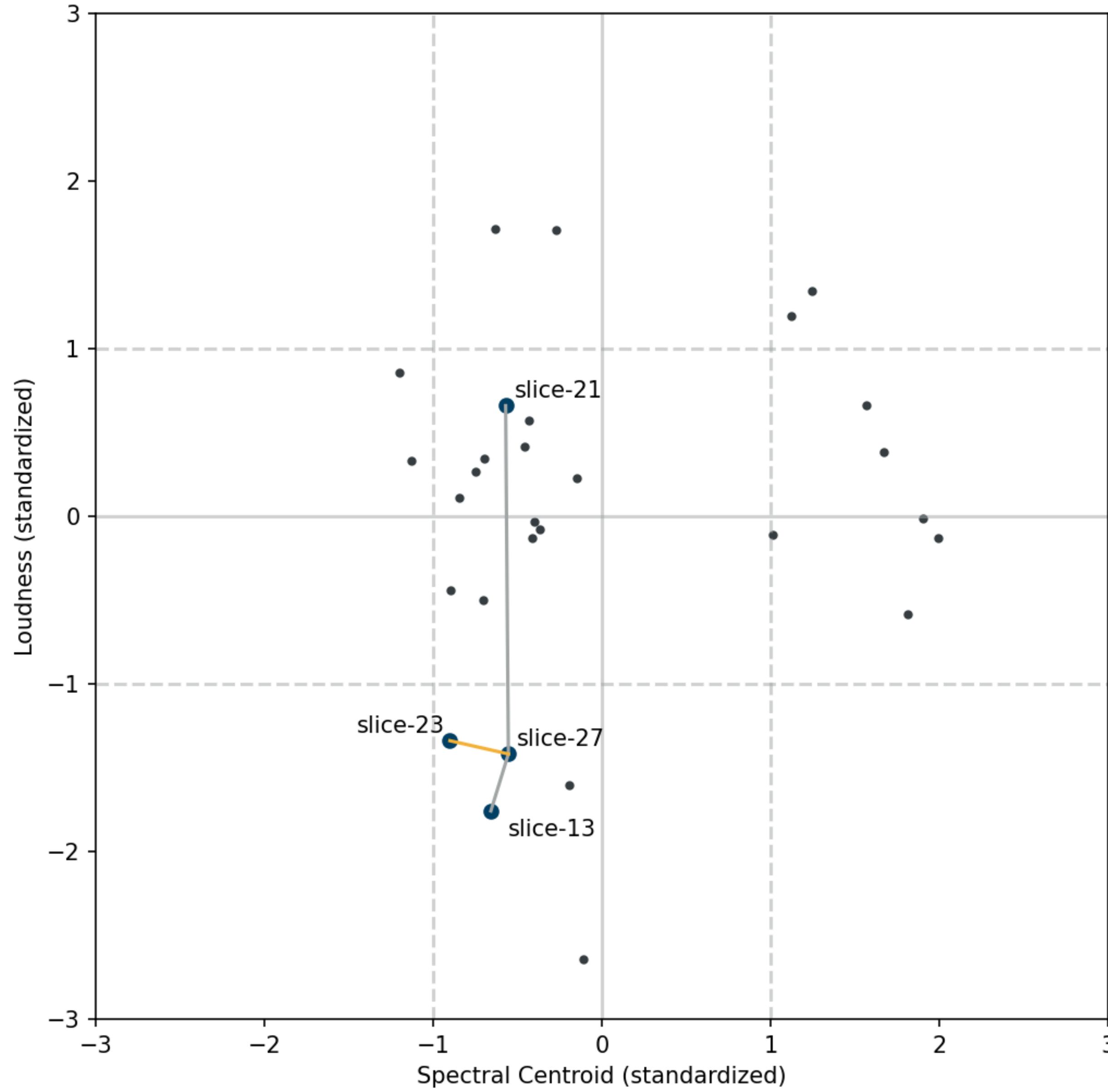
Standardized (mean = 0, standard deviation = 1)



Standardized (mean = 0, standard deviation = 1)



Standardized (mean = 0, standard deviation = 1)

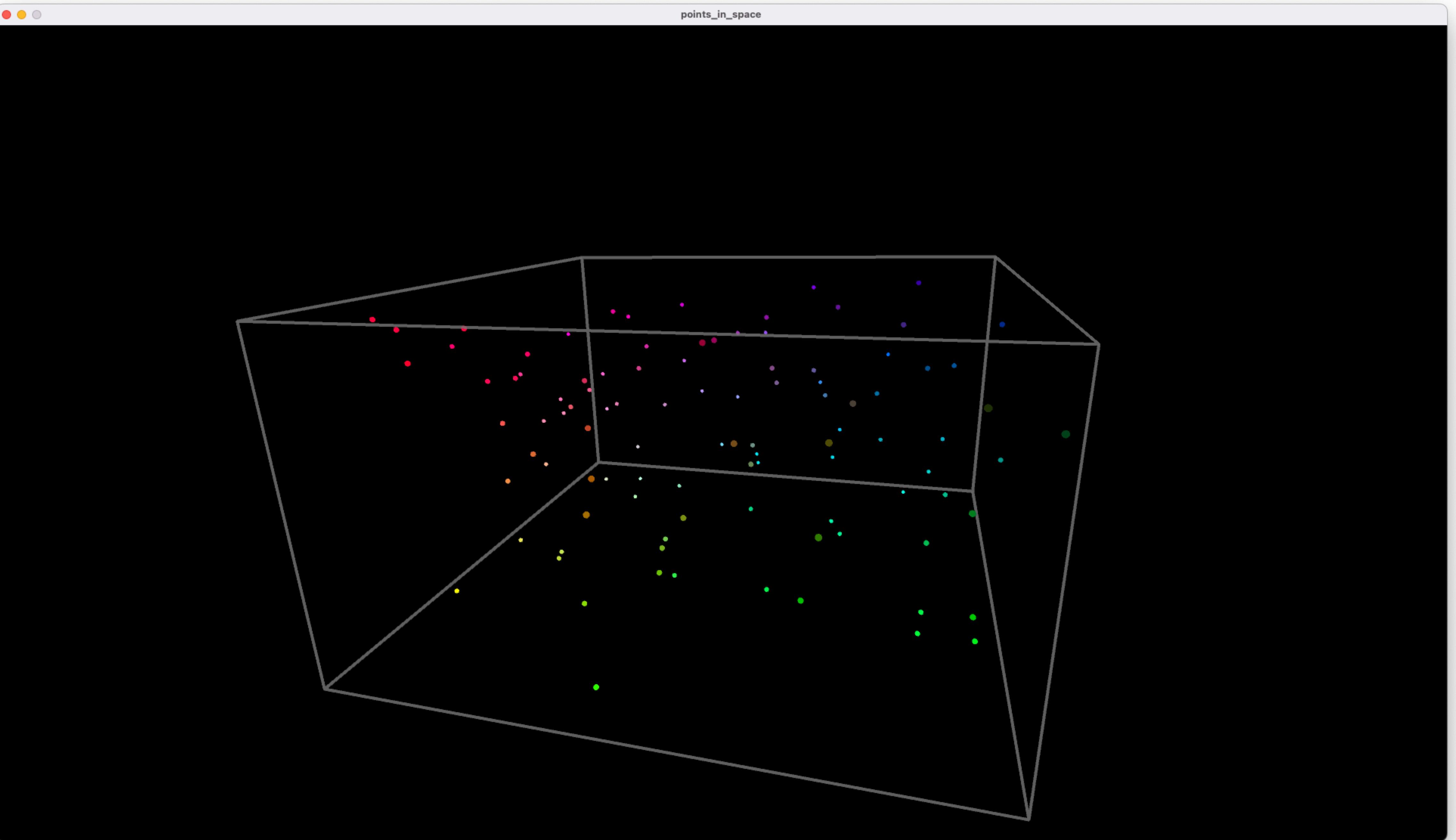


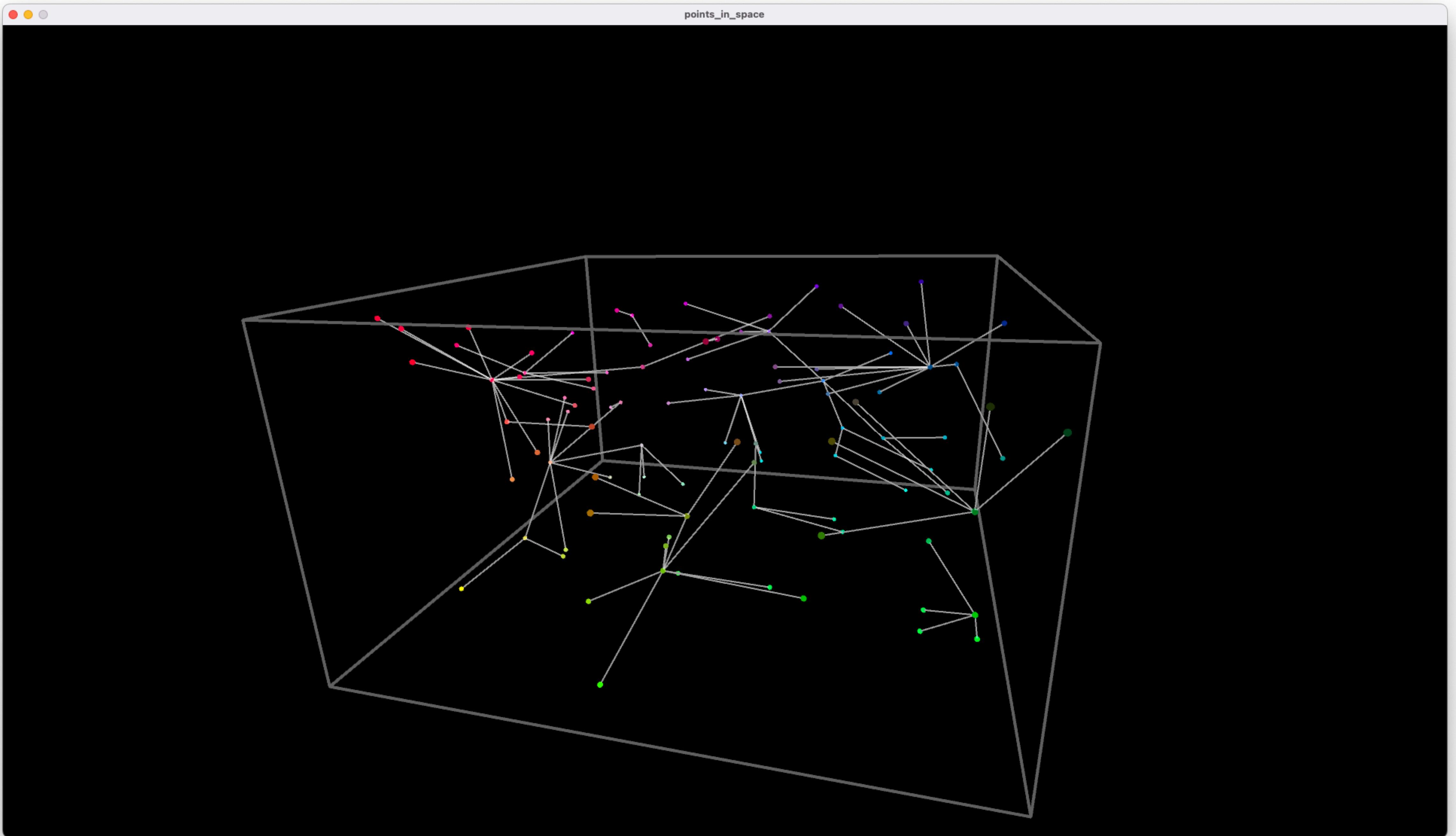
learn.flucoma.org/reference/kdtree



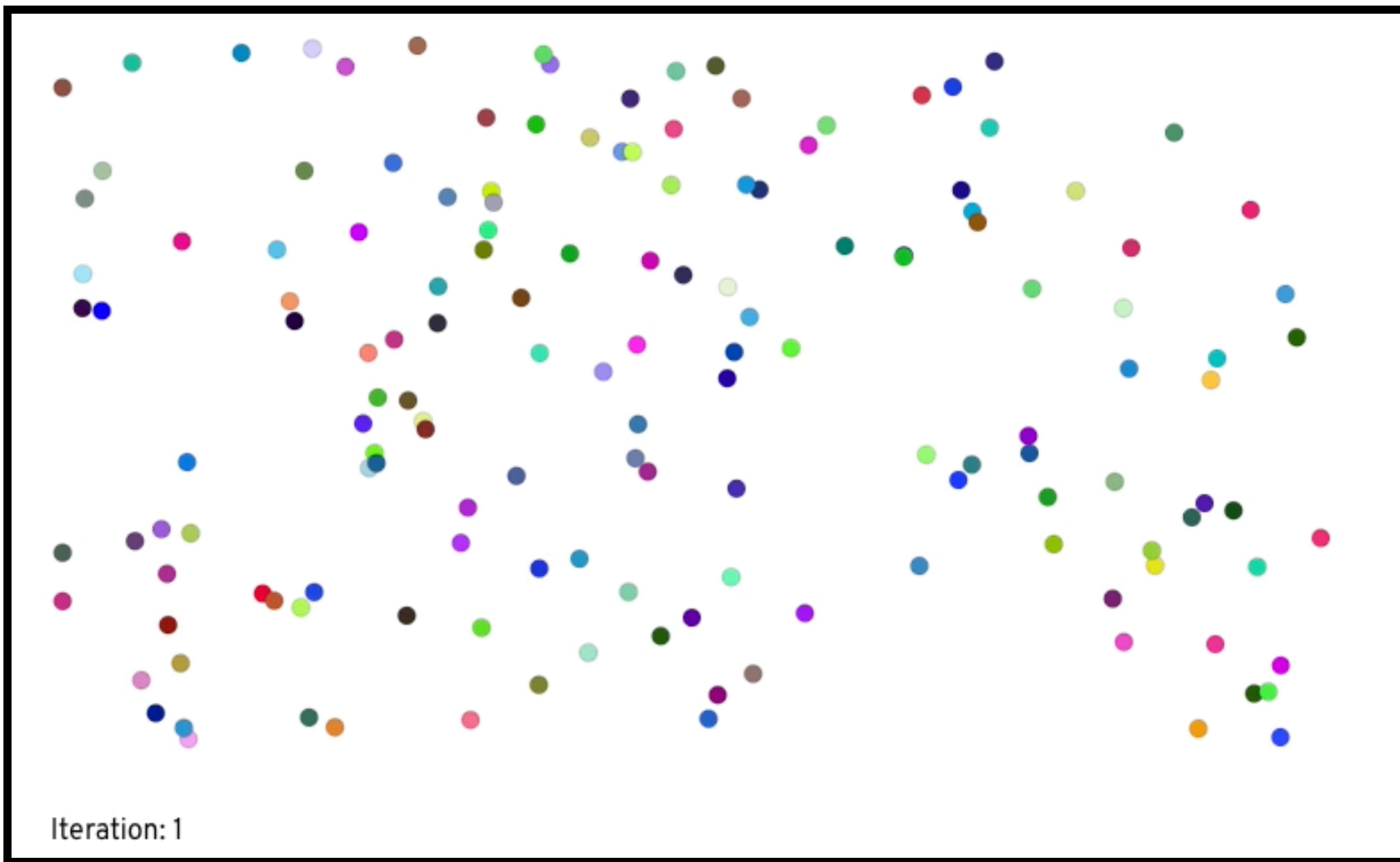
Uniform Manifold Approximation and Projection (UMAP)

1. in the high dimensional space, create a tree-like structure of neighbor relations (which points are close, which points are far)
2. randomly toss the points into the lower dimensional space (tree relations will be...poor)
3. iterate on the points, shifting them around in the lower dimensional space until the tree relations are more similar to how they were in higher dimensions



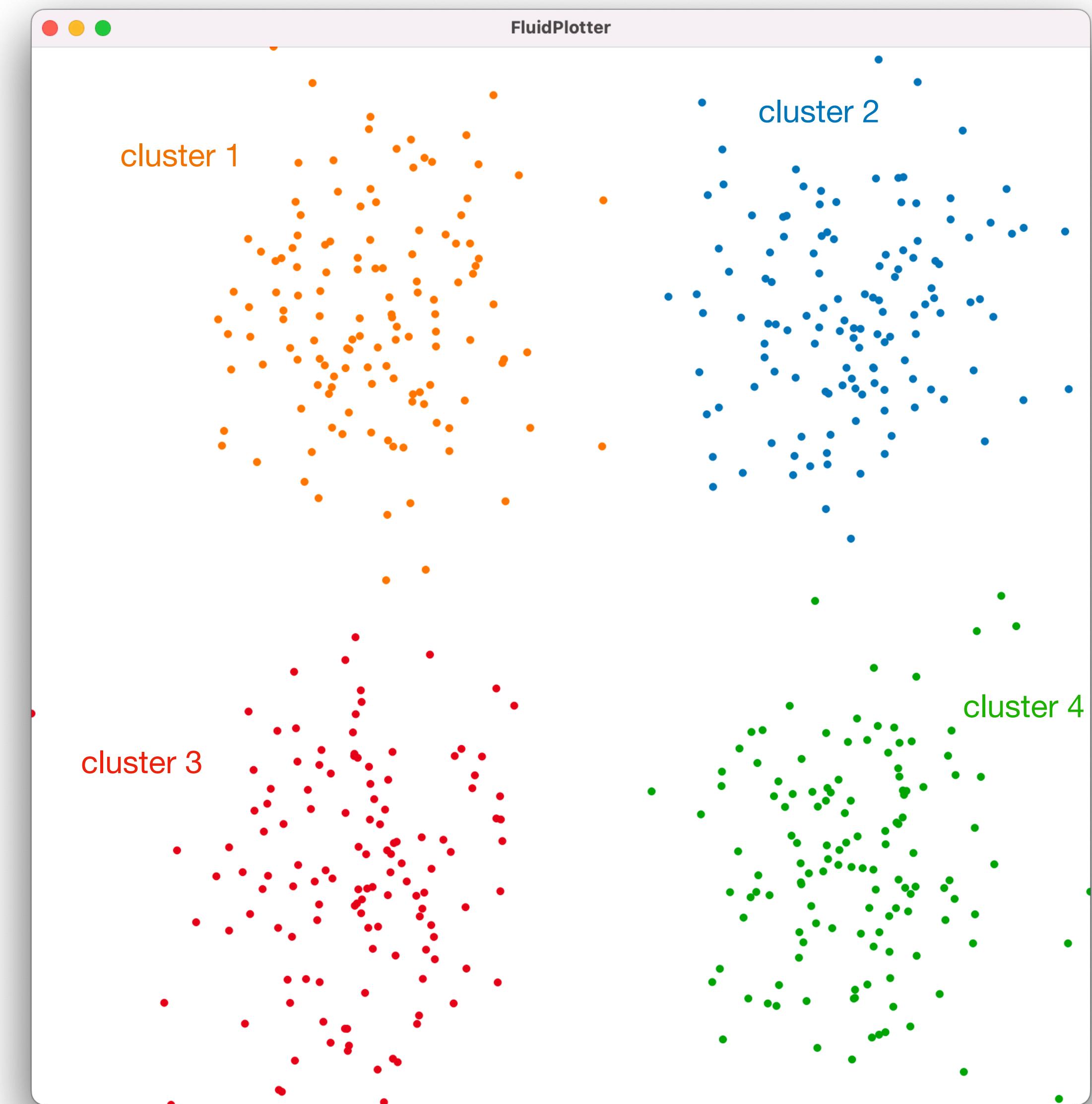


Uniform Manifold Approximation and Projection (UMAP)

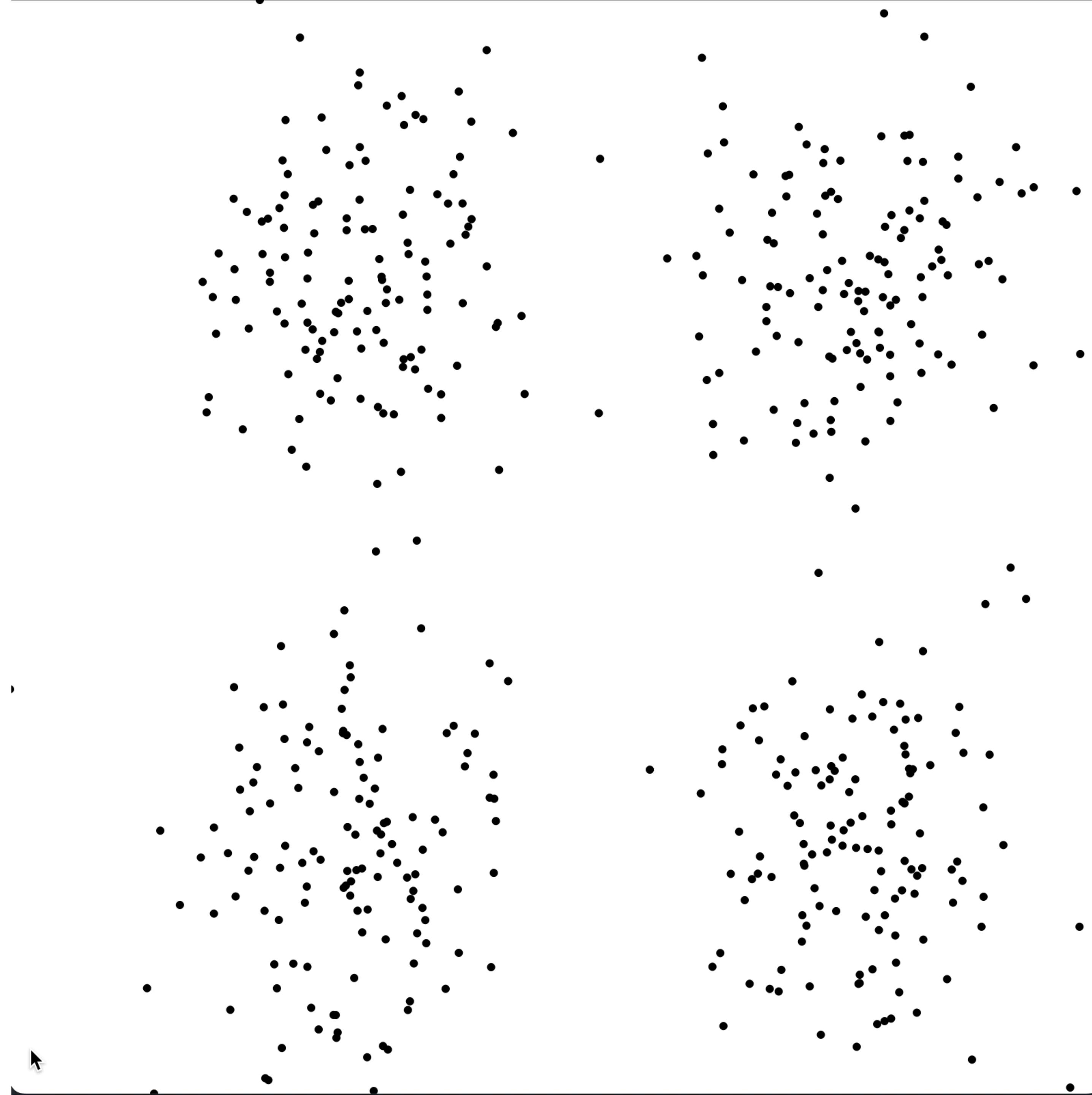


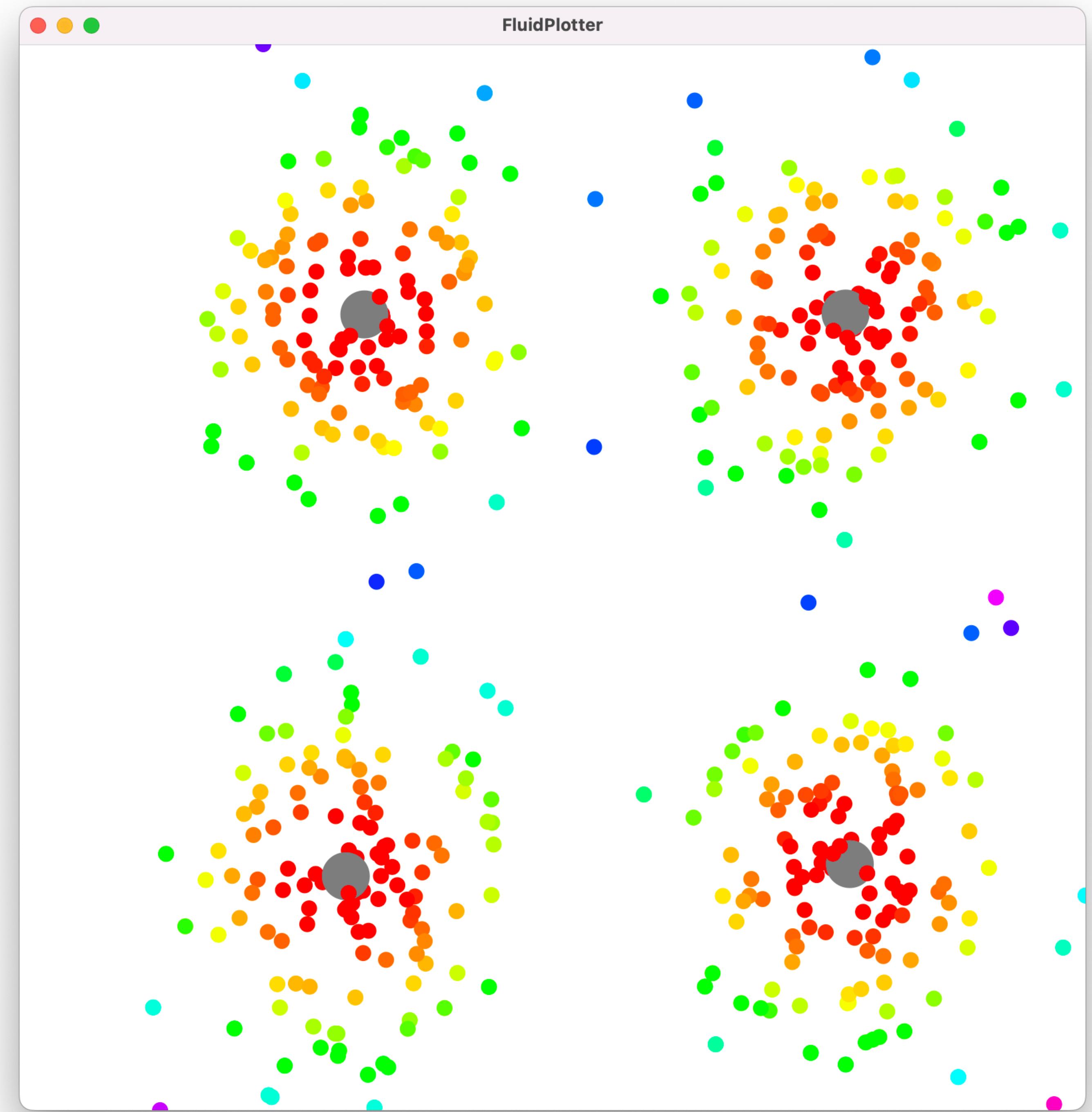
KMeans Clustering

find “clusters” of data points



Watching the iterations



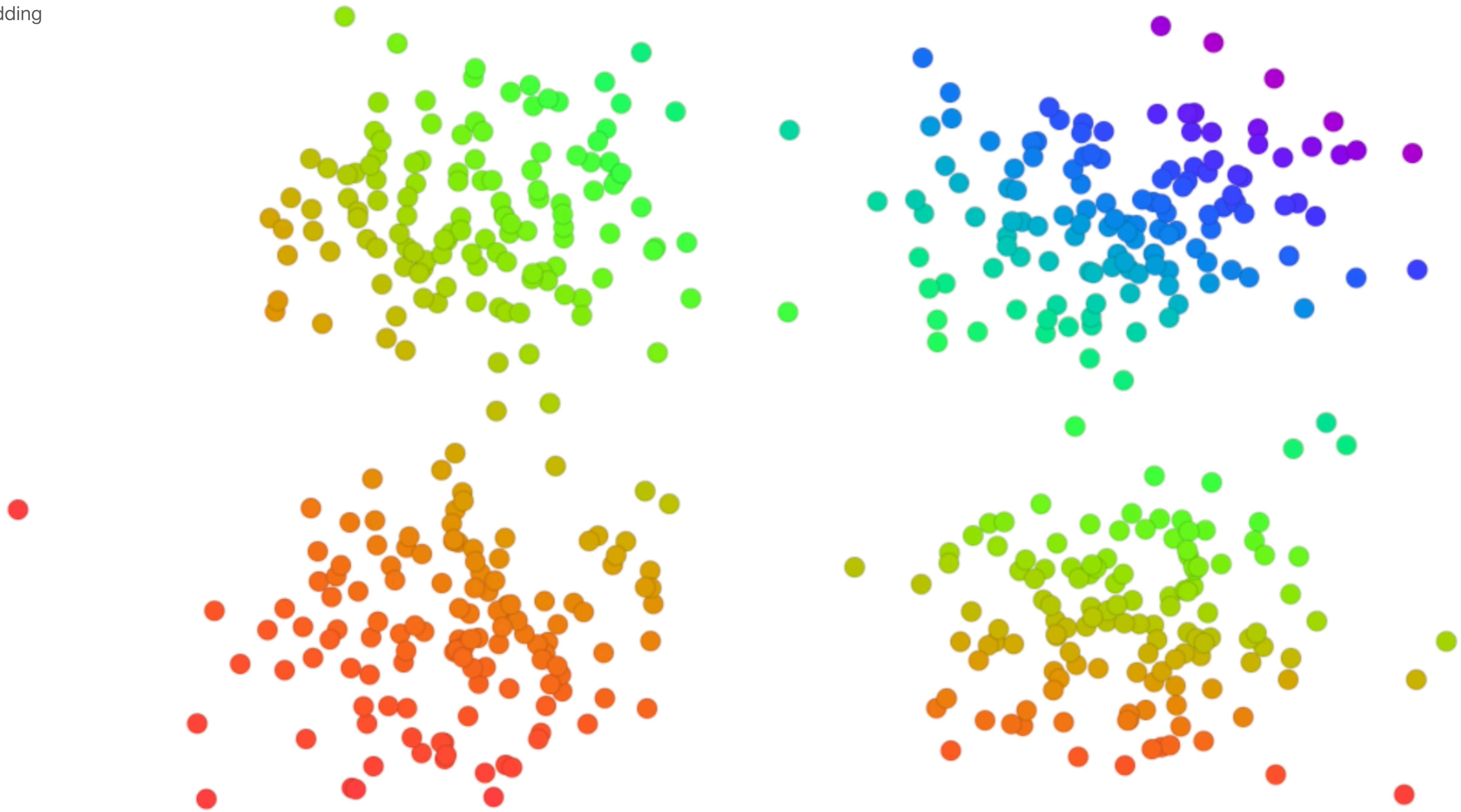


Grid (Jonker-Volgenant algorithm)

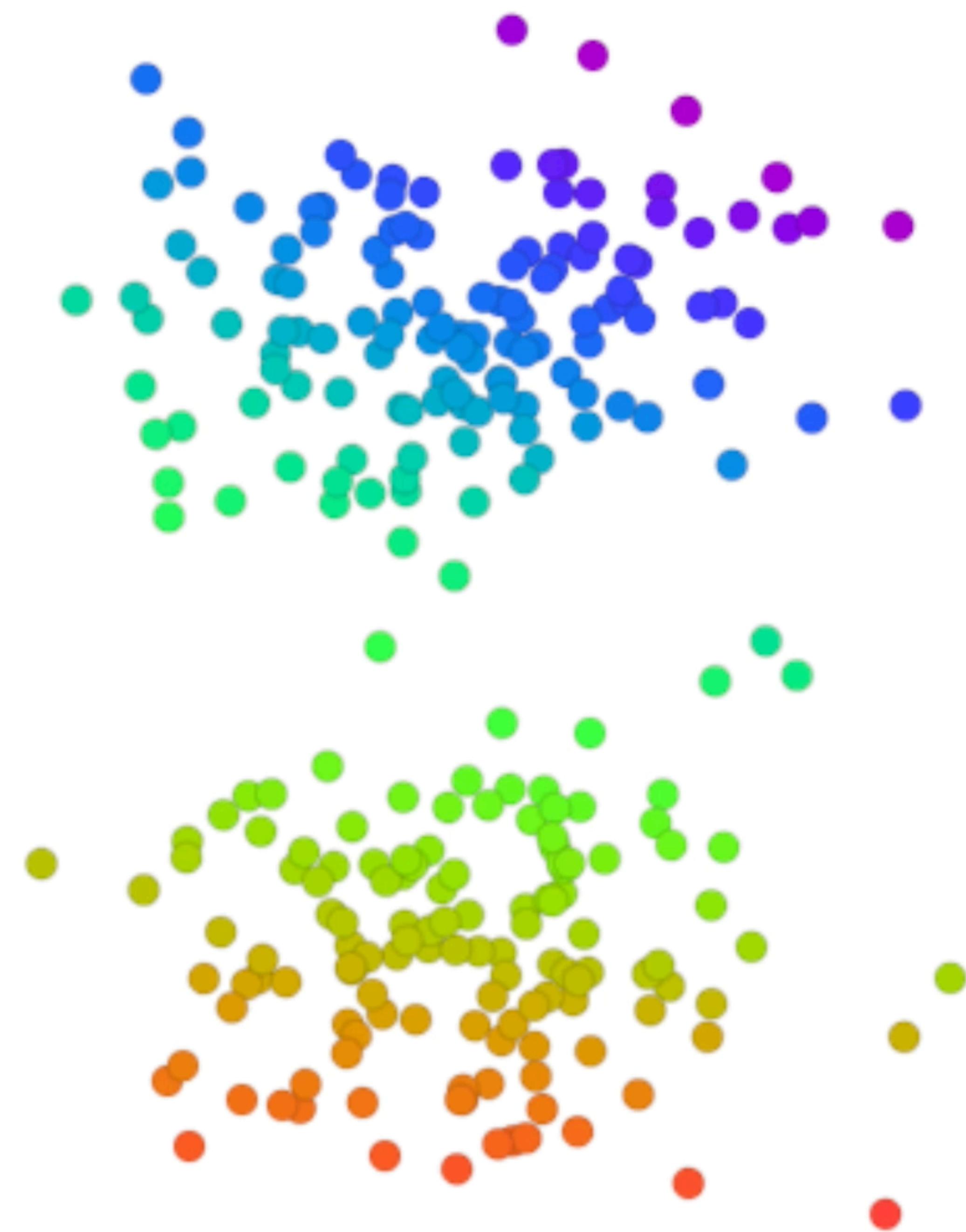
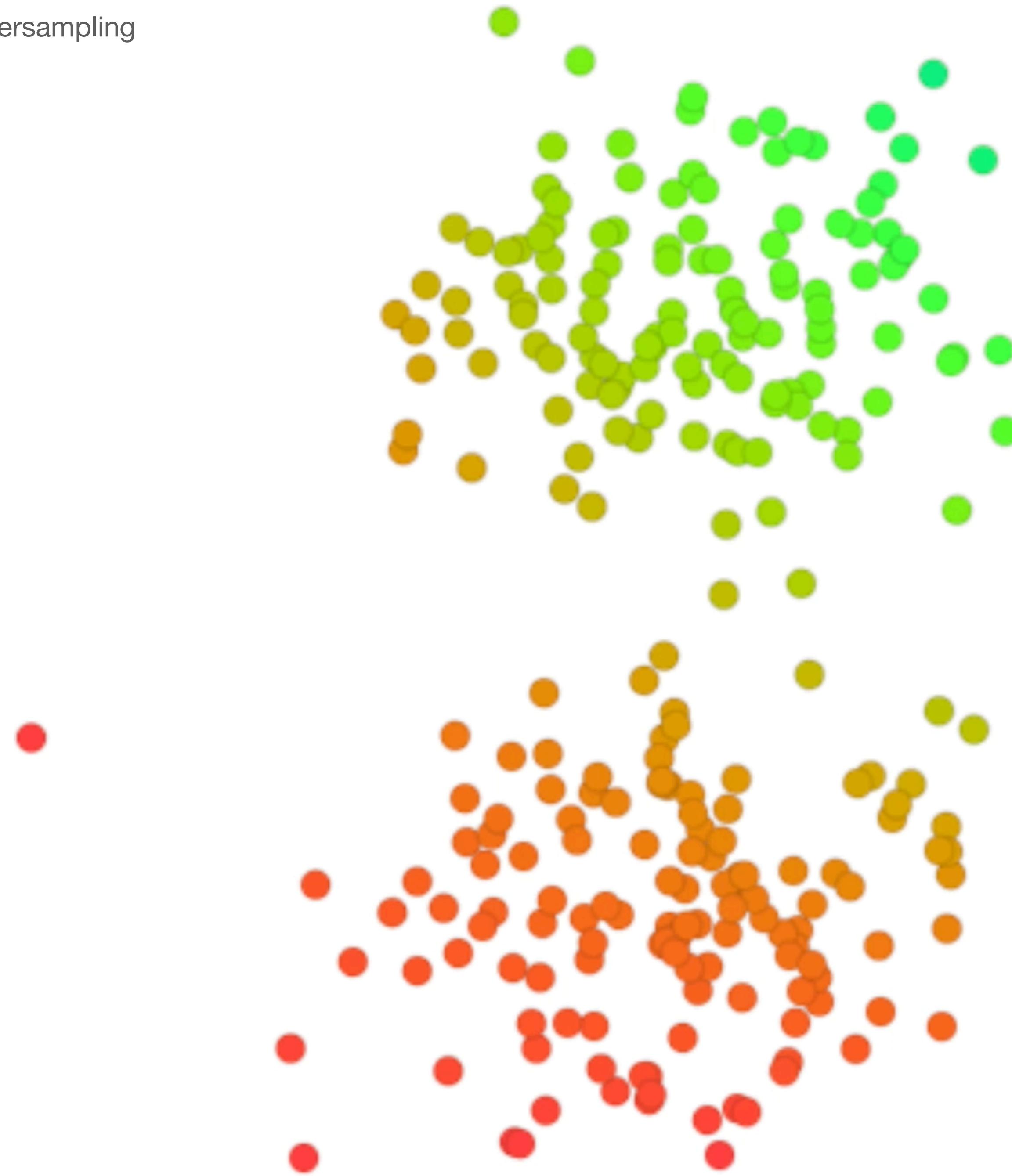
assign each point in 2D space to a point on a 2D grid

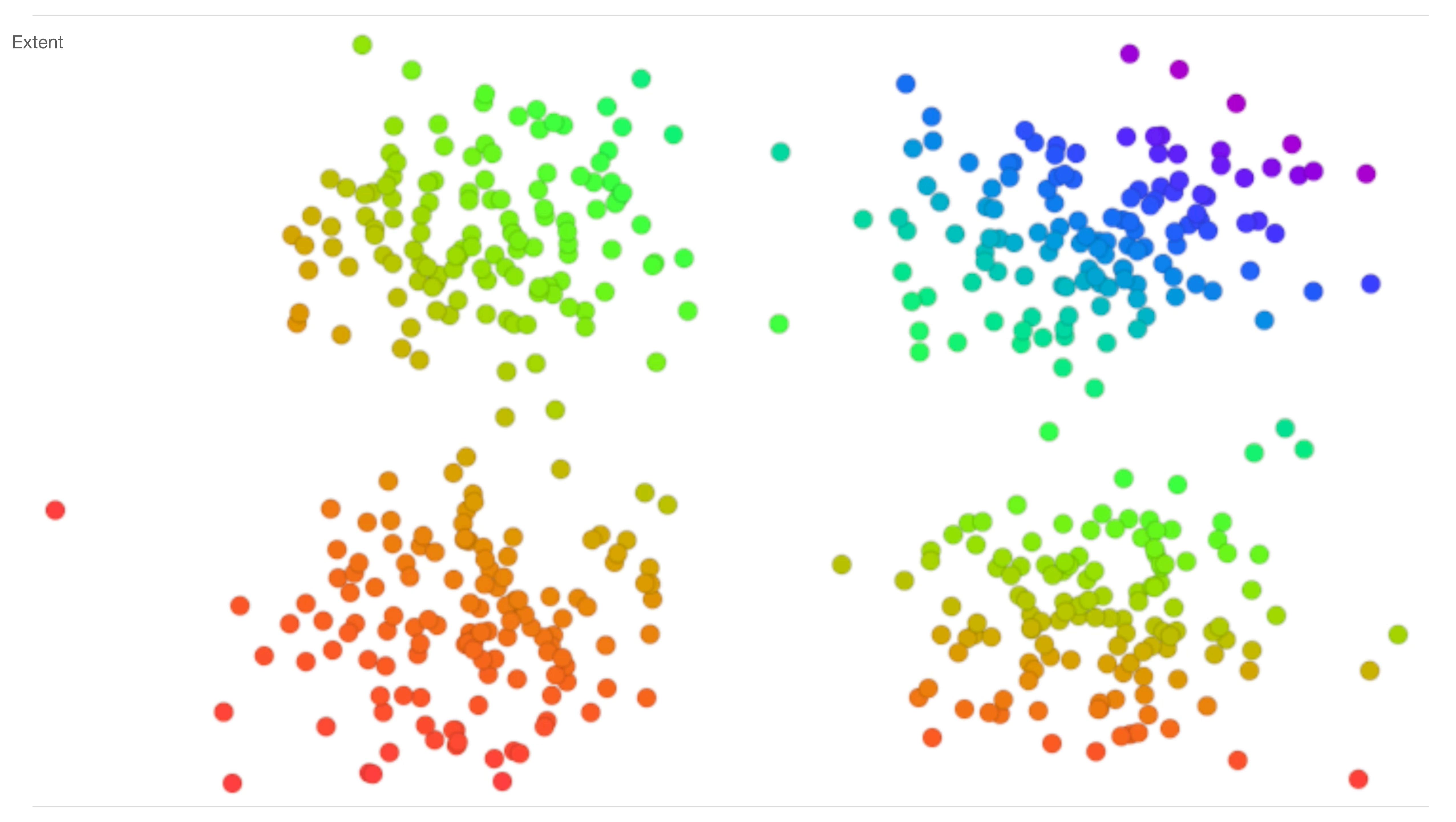
point-0:	[0.75, 0.65]	-->	[1.0, 1.0]
point-1:	[0.50, 0.80]	-->	[0.5, 1.0]
point-2:	[0.35, 0.68]	-->	[0.0, 0.5]
point-3:	[0.64, 0.41]	-->	[1.0, 0.5]
point-4:	[0.36, 0.36]	-->	[0.0, 0.0]
point-5:	[0.37, 0.68]	-->	[0.5, 0.5]
point-6:	[0.34, 0.73]	-->	[0.0, 1.0]
point-7:	[0.45, 0.09]	-->	[0.5, 0.0]
point-8:	[0.55, 0.49]	-->	[1.0, 0.0]

Gridding



Oversampling





flucoma.org
learn.flucoma.org



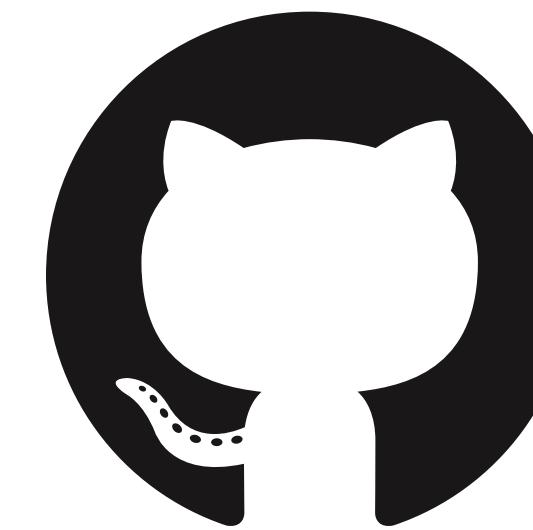
download FluCoMa package
toolkit reference
learn from tutorials
hear & see works made with FluCoMa

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

FluidBufSpectralShape 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
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

FluidBufMFCC writes the analysis to a buffer

FFT frames (time): →

