



Online Workshop: "Performing with a virtual agent: machine learning for live coding"

Day 3

Anna Xambó

Music, Technology and Innovation - Institute for Sonic Creativity (MTI²)
De Montfort University

Leicester Hackspace 29.1.2021



MIRLCAuto: A Virtual Agent for Music Information Retrieval in Live Coding

Partners: IKLECTIK, Leicester Hackspace,
L'Ull Cec, Phonos, MTI²

Collaborators: TOPLAP Barcelona, FluCoMa, Freesound

Awarded with an EPSRC HDI Network Plus Grant

Partners

IKLECTIK [off-site]



phonos



Collaborators

toplaphbcn



freesound

Online Workshop Performing with a virtual agent: machine learning for live coding

London (IKLECTIK)

7/9/11.12.2020 - 19:00-21:00 (GMT)

Barcelona (L'Ull Cec)

11/13/15.1.2021 - 19:00-21:00 (CET)

Leicester (Leicester Hackspace)

25/27/29.1.2021 - 19:00-21.00 (GMT)

More info at:

mirlca.dmu.ac.uk/workshops

Materials / Working Tools

- **SuperCollider Extensions:** <http://tiny.cc/LHS-SC-extensions>
- **Working doc:** <https://pad.riseup.net/p/online-workshop-lhs-keep>
- **Zoom / Zoom chat**
- **GitHub repo (tutorials, tickets):** <https://github.com/mirlca/leicesterhackspace-workshop>
- **Tutorials:** <http://tiny.cc/LHS-tutorials> (or under the Github link)
- **Project website:** <https://mirlca.dmu.ac.uk>

Schedule

- **Recap (all) - 10'**
- **Different Similar Sounds by Hernani Villaseñor and Ramon Casamajó - 10'**
- **Project presentation 3rd part: task 1 & task 2 (Anna) - 15'**
- **Tutorial MIRLCA training (Anna) - 10'**
- **Hands-on MIRLCA training (breakout rooms) - 15'**
- **Comfort break - 10'**
- **Breakout rooms: Reflections on machine learning tasks and live coding (All) - 25'**
- **Presentations by groups - 15'**
- **Final discussion, feedback and closing (all) - 10'**

Learning Outcomes

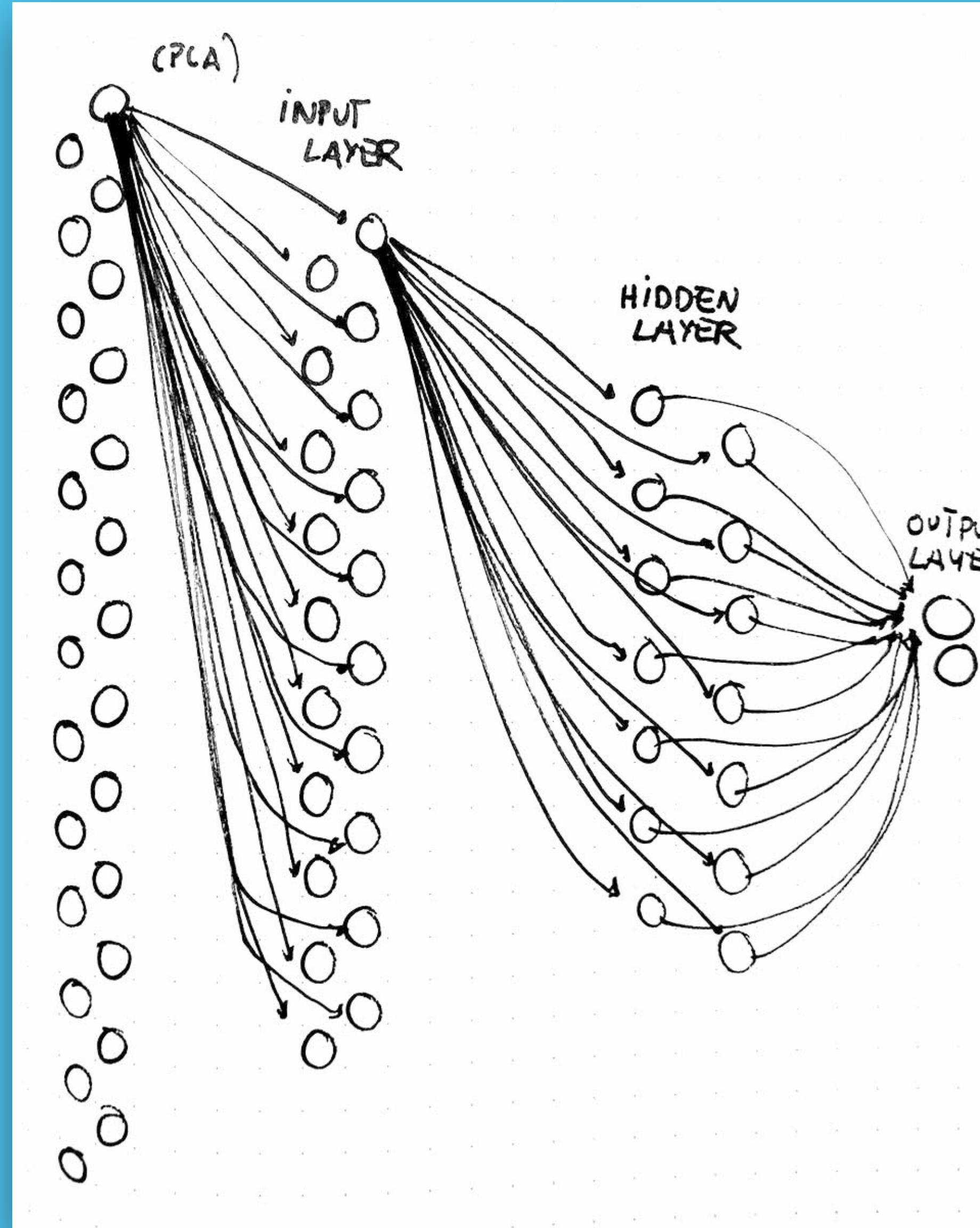
- Get a sense of the practice of live coding (music live performance using code) by manipulating online crowdsourced sounds and the automatic use of feature descriptors obtained from Freesound.org.
- Get familiar with the application of neural networks, in particular a multilayer perceptron used as a classifier, to improve the practice of live coding with crowdsourced sounds.
- **Be exposed to the main steps to solve a problem using machine learning techniques: the creation of a dataset, training a model, testing the model, and performing with / evaluating the model in an iterative cycle.**
- **Understand how to combine different technologies in SuperCollider to build a prototype for live coding performance.**
- **Get insight on a participatory design approach to designing a prototype for live coding performance.**

At the end of the workshop you will be able to...

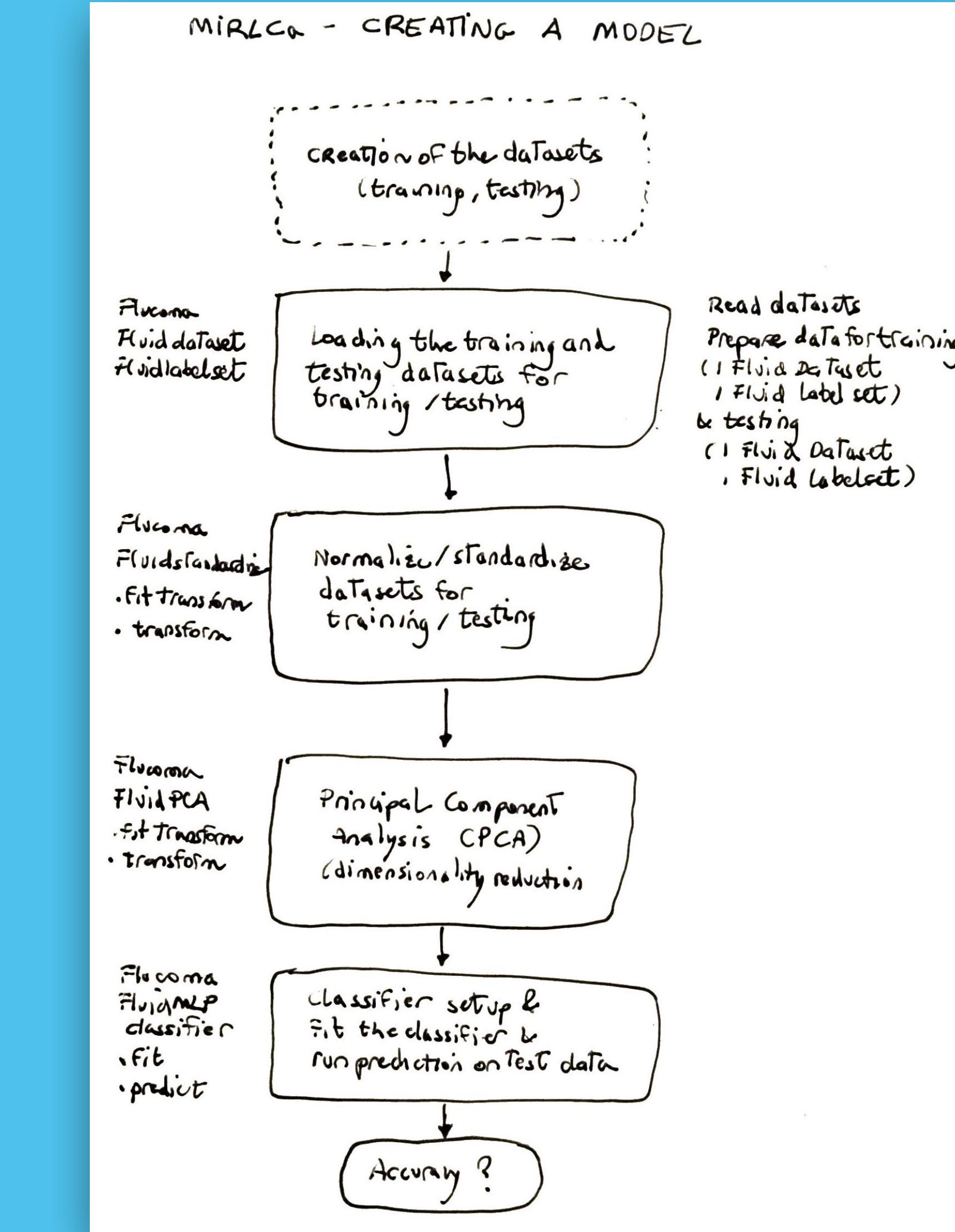
- Use SuperCollider and the MIRLC2 library to retrieve sounds from Freesound.org based on a live coding approach.
- Use a trained model using the FluCoMa library to retrieve sounds that are based on personal musical taste.
- **Train your own model using the FluCoMa library to retrieve sounds that are based on your personal musical taste.**
- **Explore creative strategies to perform with a virtual agent using machine learning for live coding.**
- Analyse how to define a virtual agent that can react to the live coder inputs using the FluCoMa library.

**A follow-up introduction to machine learning
in live coding and task 1**
Train an agent to assist meaningful querying to
Freesound.org ("*situated musical actions*")

Musical taste classifier architecture



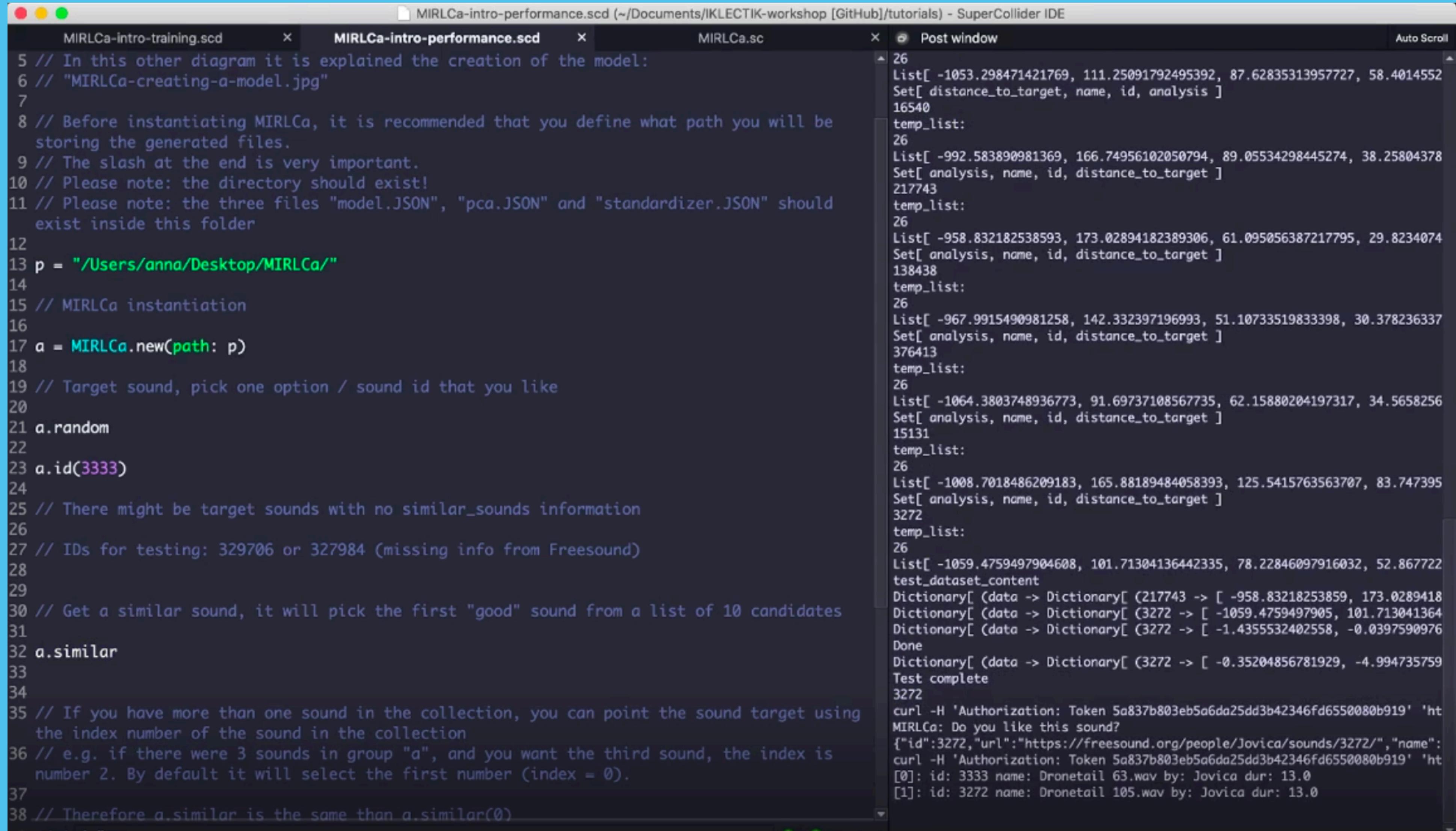
Creating a model



<https://github.com/mirlca/code/blob/main/tutorials/manual-training-musical-taste-classifier.scd>

<https://github.com/mirlca/code/tree/main/datasets>

Performance Mode



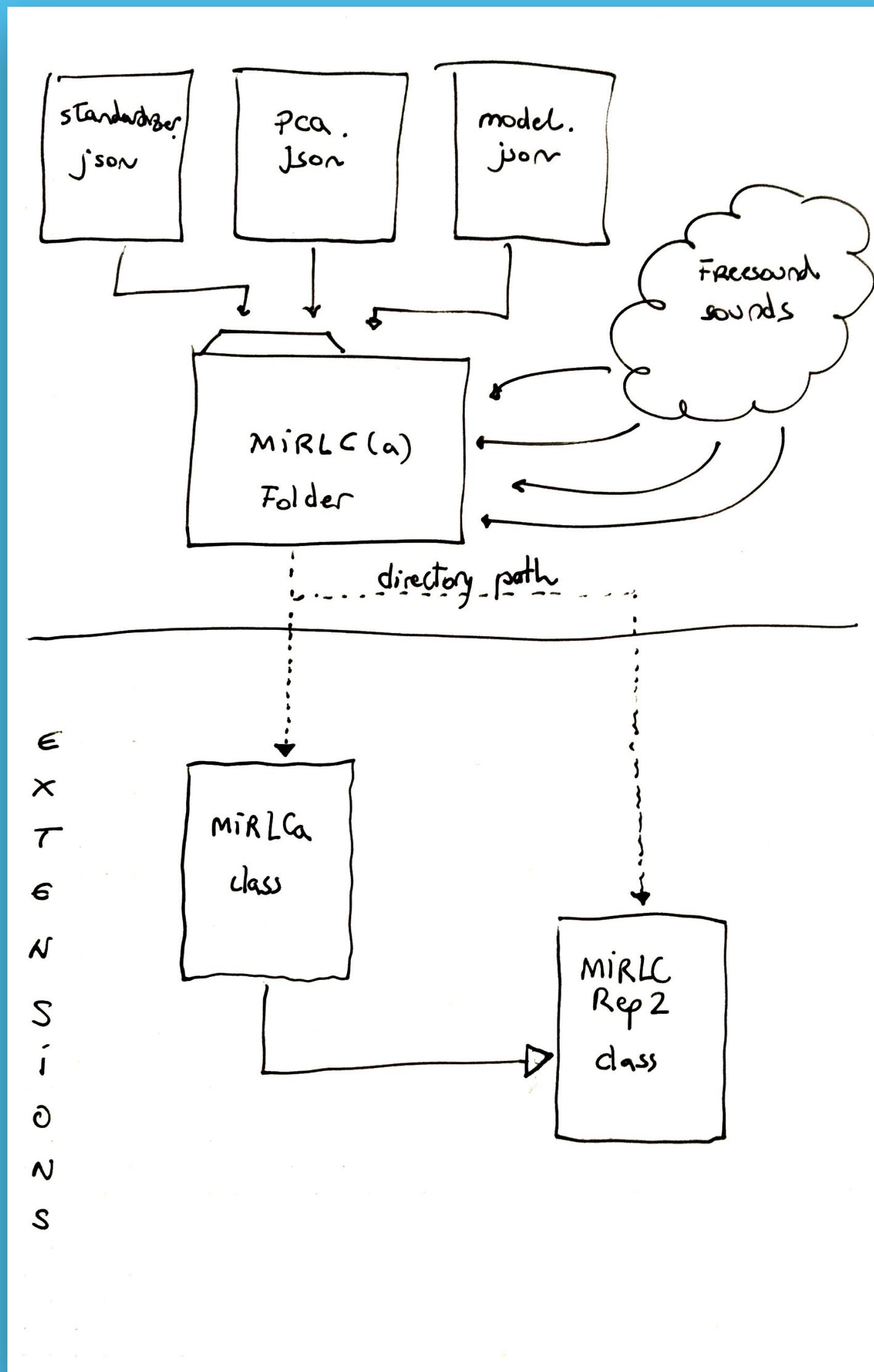
The screenshot shows the SuperCollider IDE interface with three tabs: "MIRLCa-intro-training.sc", "MIRLCa-intro-performance.sc", and "MIRLCa.sc". The "MIRLCa-intro-performance.sc" tab is active, displaying the following code:

```
5 // In this other diagram it is explained the creation of the model:  
6 // "MIRLCa-creating-a-model.jpg"  
7  
8 // Before instantiating MIRLCa, it is recommended that you define what path you will be  
storing the generated files.  
9 // The slash at the end is very important.  
10 // Please note: the directory should exist!  
11 // Please note: the three files "model.JSON", "pca.JSON" and "standardizer.JSON" should  
exist inside this folder  
12  
13 p = "/Users/anna/Desktop/MIRLCa/"  
14  
15 // MIRLCa instantiation  
16  
17 a = MIRLCa.new(path: p)  
18  
19 // Target sound, pick one option / sound id that you like  
20  
21 a.random  
22  
23 a.id(3333)  
24  
25 // There might be target sounds with no similar_sounds information  
26  
27 // IDs for testing: 329706 or 327984 (missing info from Freesound)  
28  
29  
30 // Get a similar sound, it will pick the first "good" sound from a list of 10 candidates  
31  
32 a.similar  
33  
34  
35 // If you have more than one sound in the collection, you can point the sound target using  
the index number of the sound in the collection  
36 // e.g. if there were 3 sounds in group "a", and you want the third sound, the index is  
number 2. By default it will select the first number (index = 0).  
37  
38 // Therefore a.similar is the same than a.similar(0)
```

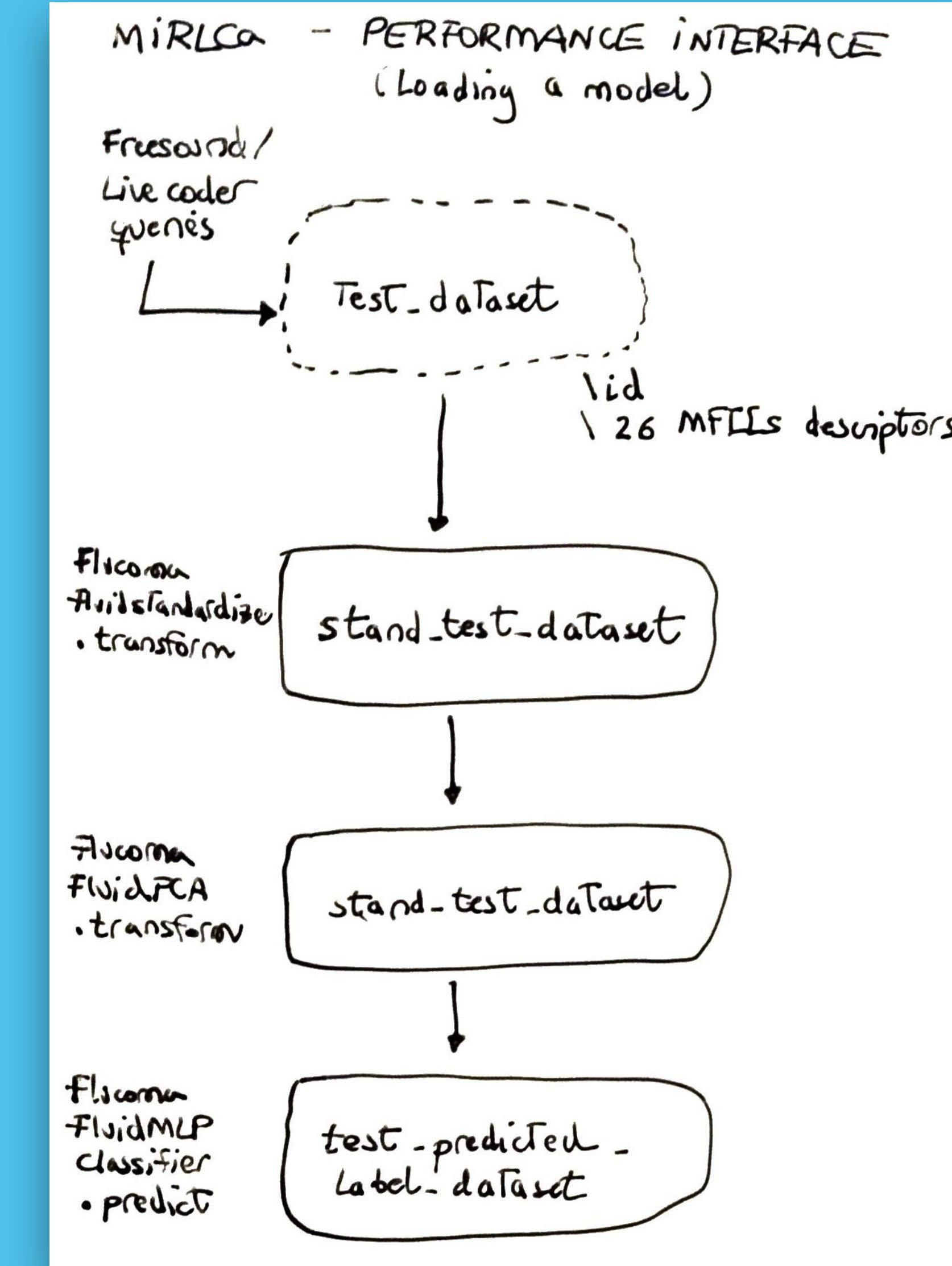
The right pane shows the "Post window" containing the output of the code execution, which includes lists of analysis data and command-line interactions with curl.

```
26  
List[ -1053.298471421769, 111.25091792495392, 87.62835313957727, 58.4014552  
Set[ distance_to_target, name, id, analysis ]  
16540  
temp_list:  
26  
List[ -992.583890981369, 166.74956102050794, 89.05534298445274, 38.25804378  
Set[ analysis, name, id, distance_to_target ]  
217743  
temp_list:  
26  
List[ -958.832182538593, 173.02894182389306, 61.095056387217795, 29.8234074  
Set[ analysis, name, id, distance_to_target ]  
138438  
temp_list:  
26  
List[ -967.9915490981258, 142.332397196993, 51.10733519833398, 30.378236337  
Set[ analysis, name, id, distance_to_target ]  
376413  
temp_list:  
26  
List[ -1064.3803748936773, 91.69737108567735, 62.15880204197317, 34.5658256  
Set[ analysis, name, id, distance_to_target ]  
15131  
temp_list:  
26  
List[ -1008.7018486209183, 165.88189484058393, 125.5415763563707, 83.747395  
Set[ analysis, name, id, distance_to_target ]  
3272  
temp_list:  
26  
List[ -1059.4759497904608, 101.71304136442335, 78.22846097916032, 52.867722  
test_dataset_content  
Dictionary[ (data -> Dictionary[ (217743 -> [ -958.83218253859, 173.0289418  
Dictionary[ (data -> Dictionary[ (3272 -> [ -1059.4759497905, 101.713041364  
Dictionary[ (data -> Dictionary[ (3272 -> [ -1.4355532402558, -0.0397590976  
Done  
Dictionary[ (data -> Dictionary[ (3272 -> [ -0.35204856781929, -4.994735759  
Test complete  
3272  
curl -H 'Authorization: Token 5a837b803eb5a6da25dd3b42346fd6550080b919' 'ht  
MIRLCa: Do you like this sound?  
{"id":3272,"url":"https://freesound.org/people/Jovica/sounds/3272/","name":  
curl -H 'Authorization: Token 5a837b803eb5a6da25dd3b42346fd6550080b919' 'ht  
[0]: id: 3333 name: Dronetail 63.wav by: Jovica dur: 13.0  
[1]: id: 3272 name: Dronetail 105.wav by: Jovica dur: 13.0
```

How to organise the files:



Behind the scenes...



Tutorial MiRLCA Performance: Loading a ML Model

Training Mode

The screenshot shows the SuperCollider IDE interface with four tabs open: MIRLCa-intro-training.scd, MIRLCa-intro-performance.scd, MIRLCa.sc, and Post window.

The MIRLCa-intro-training.scd tab contains the following code:

```
8 p = "/Users/anna/Desktop/MIRLCa/"
9
10 // MIRLCa instantiation
11
12 a = MIRLCa.new(path: p)
13
14 // Start the training
15
16 a.starttraining
17
18 // If you like the sound, execute this command
19
20 a.ok
21
22 // If you don't like the sound, execute this command
23
24 a.ko
25
26 // You can either pause the process or stop training. Pause should be executed when a
   new sound has been downloaded, otherwise you might need to execute this command
   twice.
27
28 a.pause
29
30 // A new sound will appear, sometimes you need to wait. Keep saying if you like the
   sound or not.
31
32 // Once you are done, you can stop the training. Stop training should be executed
   when a new sound has been downloaded. Otherwise you might need to execute "Pause" to
   stop playing the latest sound.
33
34 a.stoptraining
35
36 // If you don't like the result, you can continue training
```

The Post window tab displays the following output:

```
*****
You have 9 sounds in your dataset
The sound IDs are: Set[ 362349, 464127, 152562, 280091, 220698, 304655, 178
*****
Fading out the previous sound...
Number of sounds fading out: 1
curl -H 'Authorization: Token 5a837b803eb5a6da25dd3b42346fd6550080b919' 'ht
-> a MIRLCa
{"detail":"Not found."}
Sound analysis does not exist
Either SoundID or sound analysis does not exist
I'm getting another sound...
curl -H 'Authorization: Token 5a837b803eb5a6da25dd3b42346fd6550080b919' 'ht
{"detail":"Not found."}
Sound analysis does not exist
Either SoundID or sound analysis does not exist
I'm getting another sound...
curl -H 'Authorization: Token 5a837b803eb5a6da25dd3b42346fd6550080b919' 'ht
{"detail":"Not found."}
Sound analysis does not exist
Either SoundID or sound analysis does not exist
I'm getting another sound...
curl -H 'Authorization: Token 5a837b803eb5a6da25dd3b42346fd6550080b919' 'ht
{"detail":"Not found."}
Sound analysis does not exist
Either SoundID or sound analysis does not exist
I'm getting another sound...
curl -H 'Authorization: Token 5a837b803eb5a6da25dd3b42346fd6550080b919' 'ht
{"detail":"Not found."}
Sound analysis does not exist
Either SoundID or sound analysis does not exist
I'm getting another sound...
curl -H 'Authorization: Token 5a837b803eb5a6da25dd3b42346fd6550080b919' 'ht
{"id":164176,"url":"https://freesound.org/people/bmoreno/sounds/164176/","n
curl -H 'Authorization: Token 5a837b803eb5a6da25dd3b42346fd6550080b919' 'ht
curl -H 'Authorization: Token 5a837b803eb5a6da25dd3b42346fd6550080b919' 'ht
{"id":164176,"url":"https://freesound.org/people/bmoreno/sounds/164176/","n
curl -H 'Authorization: Token 5a837b803eb5a6da25dd3b42346fd6550080b919' 'ht
[0]: id: 164176 name: 0371 People_ambience.wav by: bmoreno dur: 181.075
{"lowlevel":{"mfcc":{"min":[-1138.4200593123592,1.396759147763041e-05,-45.4
List[ 164176, good, -720.9884737485957, 136.8912376372041, -4.0182580596830
*****
You have 10 sounds in your dataset
The sound IDs are: Set[ 178232, 304655, 464127, 400170, 308073, 164176, 280
*****
Fading out the previous sound...
Number of sounds fading out: 1
curl -H 'Authorization: Token 5a837b803eb5a6da25dd3b42346fd6550080b919' 'ht
-> a MIRLCa
{"id":343299,"url":"https://freesound.org/people/Kalou/sounds/343299/","nam
curl -H 'Authorization: Token 5a837b803eb5a6da25dd3b42346fd6550080b919' 'ht
curl -H 'Authorization: Token 5a837b803eb5a6da25dd3b42346fd6550080b919' 'ht
{"id":343299,"url":"https://freesound.org/people/Kalou/sounds/343299/","nam
curl -H 'Authorization: Token 5a837b803eb5a6da25dd3b42346fd6550080b919' 'ht
[0]: id: 343299 name: Goodge St - TCR ambience.wav by: Kalou dur: 257.29
{"lowlevel":{"mfcc":{"min":[-1138.4200593123592,1.396759147763041e-05,-65.9
```

Post window Auto Scroll

```
Fading out the previous sound...
Number of sounds fading out: 1
*****
There are: 15 training_candidates
There are: 7 testing_candidates
*****
-> a MIRLCa
Dictionary[ (data -> Dictionary[ (231466 -> [ -744.40758462342,
Dictionary[ (data -> Dictionary[ (337924 -> [ bad ]),
(95845 ->
(444556 -> [ bad ]),
(48974 -> [ good ]),
(147323 -> [ bad ]),
(204404 -> [ bad ]),
(456631 -> [ good ]),
(34938 -> [ good ])
Dictionary[ (data -> Dictionary[ (175929 -> [ -809.1963472979,
1
Dictionary[ (data -> Dictionary[ (175929 -> [ good ]),
(464129 -
(273835 -> [ bad ]),
(393040 -> [ good ]) ]),
(cols -> 1) ]
Standardizer training Done
Standardizer training Done
PCA training Done
PCA test Done
Trained 0.48051404953003
Test complete
7
*****
Accuracy (0%-100%): 0.71428571428571
Continue training or Save to JSON files?
*****
```

Errata

Accuracy is actually given as a proportion and not percentage:

- This means that with 22 sounds I already achieved 71% of accuracy.
- You will find the code updated to percentage in the next version to avoid misleading results. Sorry for the typo!
- See: Line 563 of MIRLCA.sc.



File Edit Options Buffers Tools SCLang Help

```
// different similar sounds

s.boot;

p = "/home/hvillase/Escritorio/MIRLCa/";
a = MIRLCa.new(path: p);

a.random
a.similar(1)
a.delay
a.play
a.autochopped(36, 4)
a.mute
a.fadeout(4)

b = MIRLCa.new(path: p)
b.random
b.similar

-:***- different-similar-sounds.scd All L18 (SCLang)
now playing...[0]: id: 276150 name: Coins_Single_10.wav by: Little
Synth('synth_mono_fs' : 1014)
now playing...[1]: id: 8364 name: HTickC.aiff by: hanstimm dur:
Synth('synth_mono_fs' : 1015)
now playing...[2]: id: 100040 name: saz_hyena.wav by: soundbytez
Synth('synth_mono_fs' : 1016)
Play backwards <<
localhost (127.0.0.1) 1.8|1.8 % u: 95 s: 29 g: 6 d: 296
```

File Edit View Selection Find Packages Help

QBRNTHSS_MIRLC.scd tidal.scd sclang://localhost:57120

```
1
2 p = "/home/rcasamajo/MIRLC2/"; // Define a patch for out data
3
4 // Hi! This is a little test of how to integrate the use of MIRLC
5 // with Tidal Cycles (the language I usually use to live coding)
6
7 a = MIRLCa.new(path: p);
8 a.tag("voice");
9 a.similar;
10 a.similarauto(0,3,10);
11 a.solo();
12 a.mute(6);
13 a.stop;
14 a.info;
15 a.play;
16 a.playauto // args: |times = 4, tempo = 30|
17
```

scratch.tidal

```
1 -- tempo
2 setcps (140/60)
3
4 hush
5
6 d2
7 -- $ sometimes (# coarse 15)
8 $ pan "0.5*12"
9 # s (slow 5 $ (segment 1 $ choose ["Fairchild808:3","glitchtoys:3","vermonaDM:3"])
10 # gain (range 0 0.85 $ slow 5 $ tri)
11 # pan (slow 2 $ (segment 1 $ rand))
12 # ...+ 1
```

~/Documents/scratch.tidal 4:5

LF UTF-8 TidalCycles GitHub Git (0) 2 updates

Collaboration with l'ull cec and Phonos in *virtual* Barcelona

Follow-up workshop: 4 special teasers in preparation (to be announced soon!)

Follow-up concert with TOPLAP Barcelona (TBD)

A ML System, What For?

- **Music style**
- **Music preference** (Gerard Roma)
- **Different instruments** (Hernani Villaseñor)
- **Parts in the composition** (Iván Paz)
- **Snapshots of a musical biography/life journey** (Jonathan Moss)
- Other?

“Situated Musical Actions”

- Lucy Suchman (1987)’s introduced the term “**situated action**” to refer any action as being linked to the context where it happens (from a study on users using an expert photocopier system designed to help them).
- A “**situated musical action**” refers to any musical action related to a specific context (where we expect the VA to help us in that action within that context).

Hands-on MIRLCa training (breakout rooms)

Break-out Group Worksheet

- 1. One of the group can share the screen at a time. Try the tutorial.**
 - 1. Do you understand what is the purpose of training a ML learning model?**
 - 2. How many sounds do you need to train to obtain a decent accuracy result?**
- 2. How do you envision the use of this tool in your practice?**

<https://pad.riseup.net/p/online-workshop-lhs-keep>

Breakout rooms: Reflections on machine learning tasks and live coding

Break-out Group Worksheet

- 1. What problem would you like to solve from a machine learning perspective?**
- 2. How would you go beyond the approach of following live coder actions (also known as the call-response strategy)?**
- 3. How would you create legible and negotiable actions (not only to the live coder but also to the audience)?**

<https://pad.riseup.net/p/online-workshop-lhs-keep>

Follow-up

Follow-up

- **Feedback Survey:** Your participation is essential to justify these activities and to help us improved for the next workshop. Please participate!
- **Showcasing your work:** Adaptation of the tools from the workshop to your practice, you are all welcome to participate:
 - **Next week:** Help-desk sessions if interested
 - **MTI^2 final concert**

References

- Great blog on ML: <https://machinelearningmastery.com>
- Alpaydin, Ethem. *Machine Learning: The New AI*. Cambridge, MA: MIT Press, 2016.
- Suchman, L. (1987). Plans and Situated Actions: The Problem of Human-Machine Communication. Cambridge, UK: Cambridge University Press.

Thank you!!!