Sound Innovation Engine 1.0

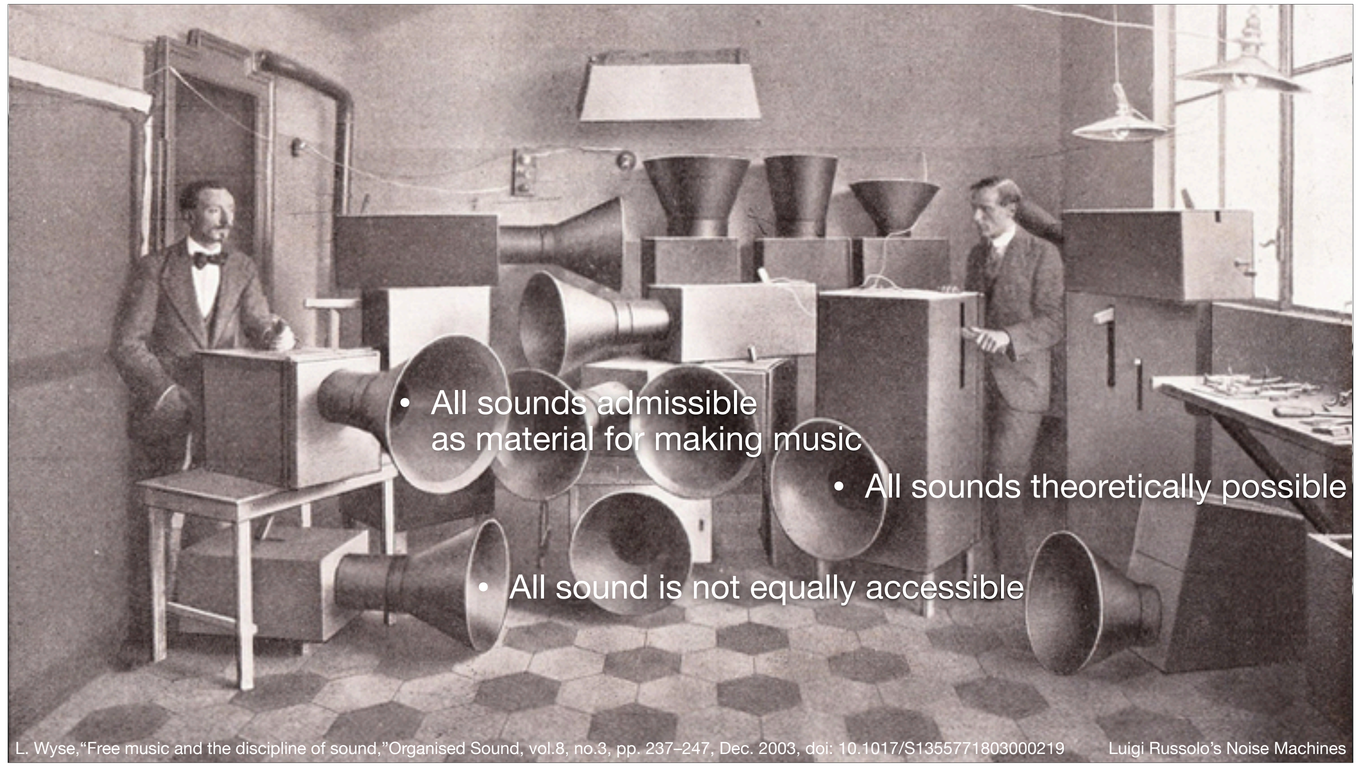
Towards Sound Innovation Engines Using Pattern-Producing Networks and Audio Graphs

Björn Þór Jónsson, Çağrı Erdem, Stefano Fasciani and Kyrre Glette, January 2024

UiO : RITMO Centre for Interdisciplinary Studies in Rhythm, Time and Motion
University of Oslo

… about steps towards enabling access to *all* sounds with evolutionary algorithms…

… ways to find sounds you might not have heard before, and might also like, with evolutionary algorithms …



…because all sounds have become admissible, in the recent history of music,
- at least since Russolo worked on the Art of Noise
- substitute for the limited variety of timbres provided by traditional instruments
 with the infinite variety of timbres in noises, “reproduced with appropriate mechanisms”

and all sounds are in principle accessible, with modern technology, especially digital sound synthesis

but all sounds are not yet equally accessible,
where access to sounds can for example depend on technical expertise and dedication

recognising sounds you’ve never heard

and maybe you don't know what you're looking for, so you can't prompt for specific results

finding sounds recognised as pleasing but unfamiliar

- maybe you would rather like to be surprised, by serendipitous discoveries,
- and go with the flow of those and see where they lead your creative process

Quality Diversity to discover stepping stones with goal switching on a path to greatness

we're interested in enabling further access to all sounds

with automatic exploration through the space of sounds,

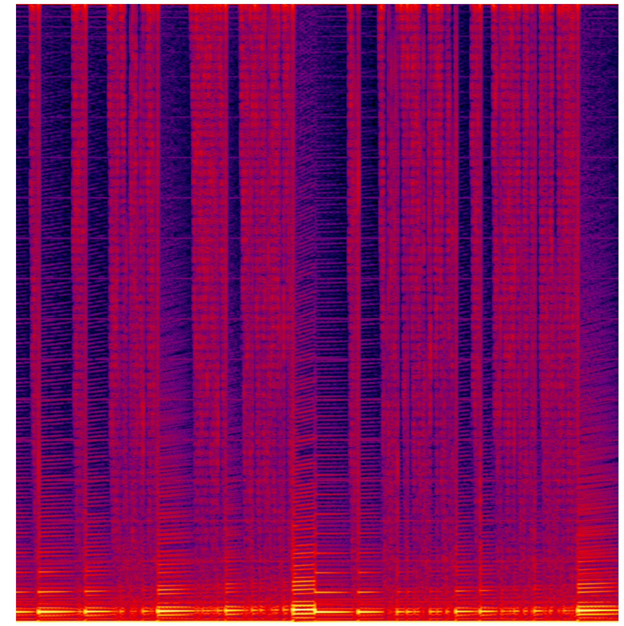
and hopefully expanding the horizon towards all sounds

our approach to this is based on Quality Diversity algorithms
which keep track of many different classes of solutions
and check the performance of offspring from one class
in other classes,
which may lead to the discovery of stepping stones through many classes
on path to interesting discoveries.

Innovation Engines

Automate QD exploration with a model capable of:

- distinguishing novelty
- evaluating quality



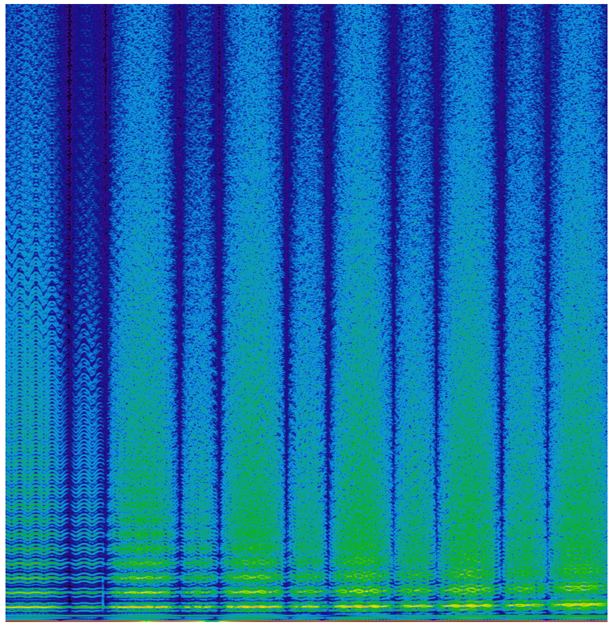
To automate the exploration,
Innovation Engines combine such QD algorithms with a model that is capable of evaluating
whether new solutions are
Interestingly new.

…
To drive automated exploration with such diversity- promoting algorithms, the Innovation Engine algorithm abstracts the process of human curiosity, replacing human judgement with a discriminative model that identifies interesting ideas [28,26]. Innovation Engines integrate two key components: Evolutionary Algorithms (EAs), such as those from the family of QD, capable of generating and gathering various novel outputs; and a model capable of distinguishing that novelty and evaluating its quality, such as Deep Neural Network (DNNs), creating niches and competition within them, thus providing selection pressure to quide QD search.

Innovation Engines

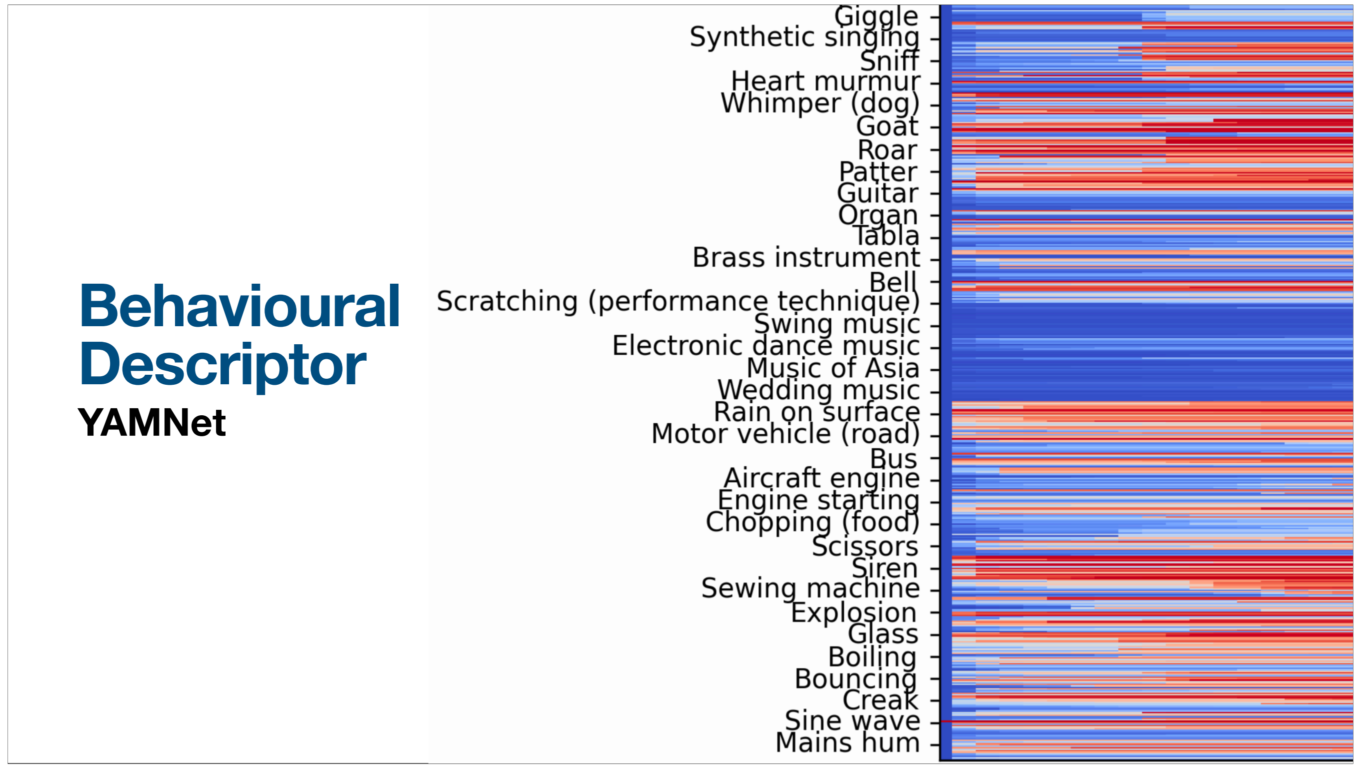
Ultimate goal:

- Unsupervised classification
- Produce new types of things



A long term goal of innovation engines is to learn to
- classify the things they have seen so far
- and seek to produce new types of things

Unsupervised, without labeled data…



But to start our Innovation Engine explorations with sound, we start with a pre-trained model:

- YAMNet, a DNN classifier, to define the measurement space:
    - by using the classification to define diversity
    - and confidence levels for each class as quality

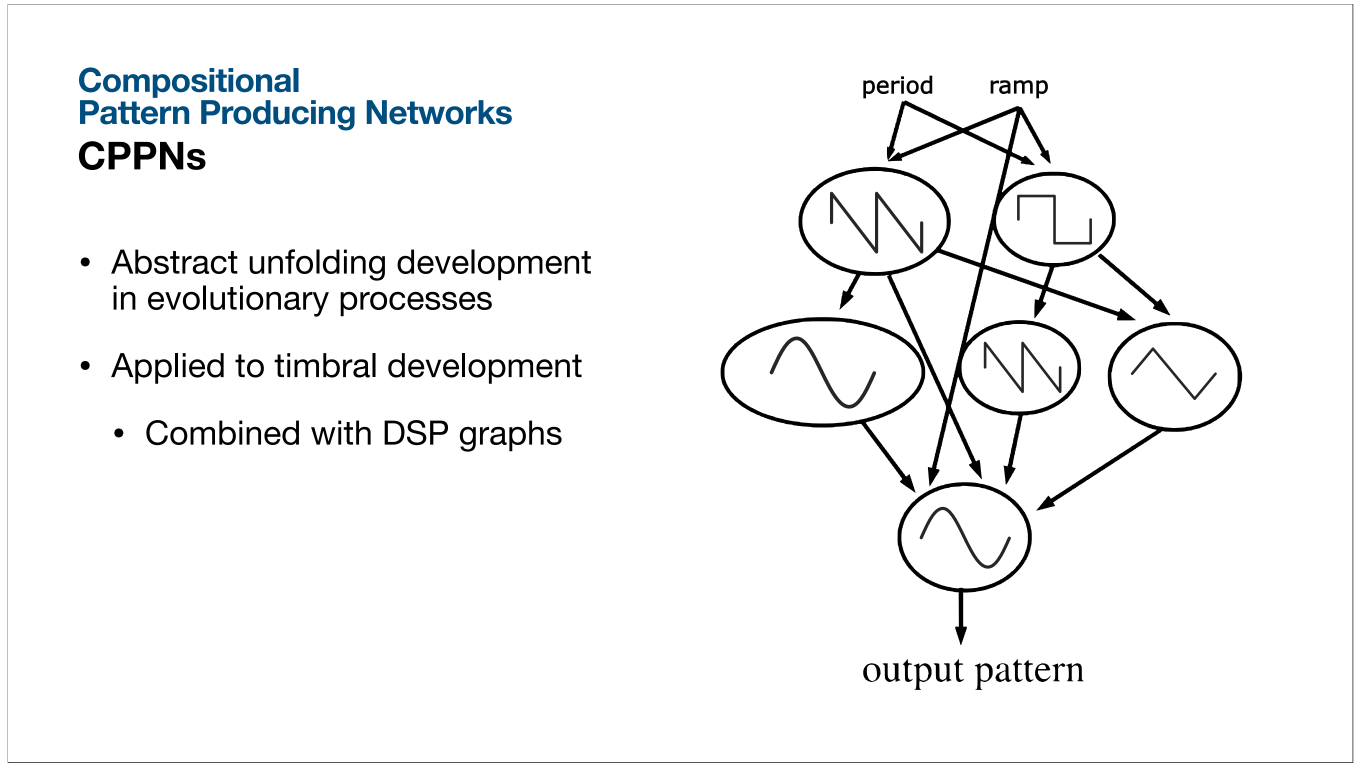
…

Behavioural Descriptor

To guide the QD search, we chose the Yet Another Mobile Network (YAMNet) DNN classifier to define our search space. The con- fidence scores output by the classifier for each class are used as selection signals for the QD algorithm, as discussed in section 3.1. While this pre-trained network

Towards Sound Innovation Engines 5

may limit our exploration, it was adopted in an effort to replicate a setup from previous evaluations of the Innovation Engine algorithm in the visual domain. That classifier is trained on AudioSet [9], which can be considered as a sonic sibling of the DNN classifiers trained on the ImageNet dataset [5]. YAMNet outputs 521 scores from a logistic (softmax) layer, corresponding to AudioSet classes. The classifier's output is intended "as a stand-alone audio event classifier that provides a reasonable baseline across a wide variety of audio events."4. Our approach to sound generation can be somewhat likened to a unique type of sound synthesiser, which is not crafted with the intention of mimicking natural sounds or creating textures that easily fit into well-known categories. Many modern gen- erative models excel at such tasks [1], building on their prior training, but we considered the varied signal provided by this model as a good starting point for driving the EA towards diversity. We also considered it interesting to mirror the overall setup from experiments [28,26,18] that inspire our sonic investigations.

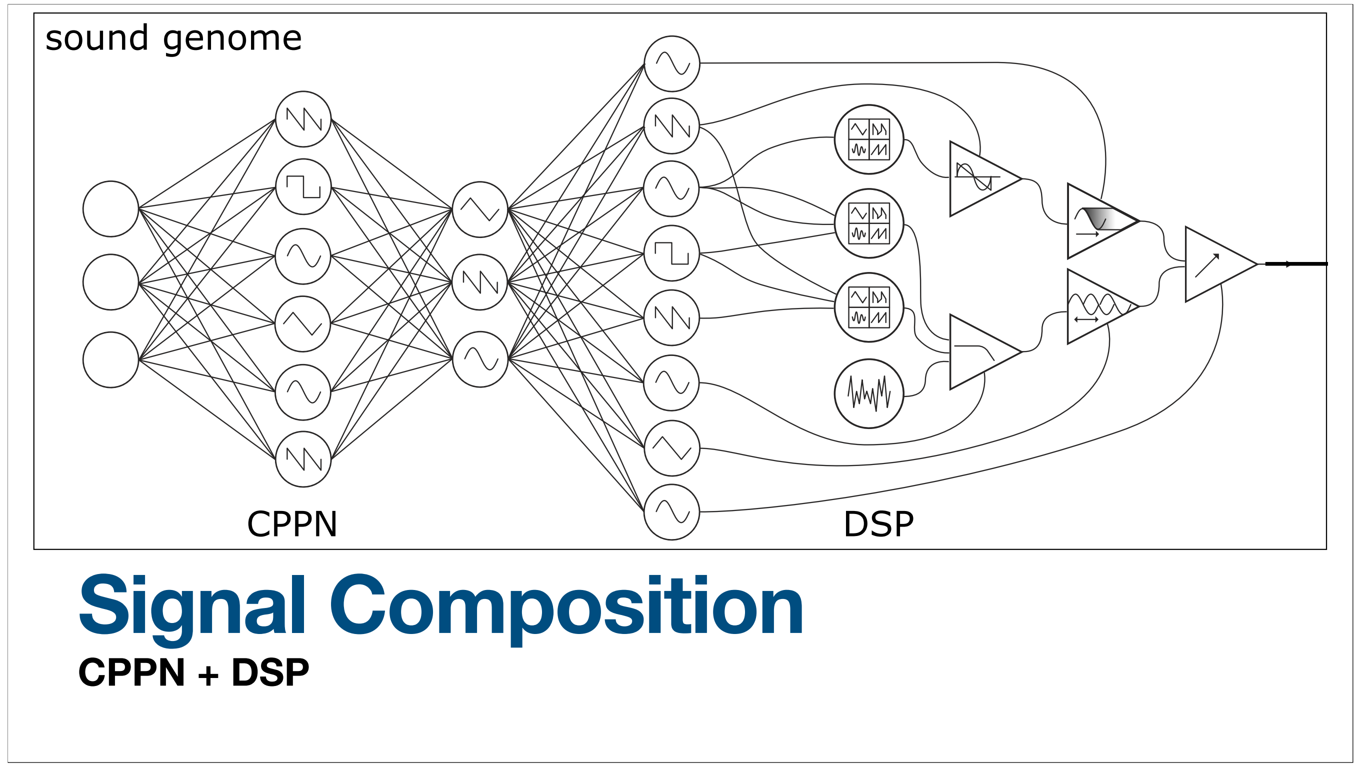


Compositional Pattern Producing Networks
are a part of our approach to synthesising sounds:

CPPN networks abstract unfolding development during evolutionary processes
- by composing different functions at each node

This can be compared with the process of timbral development,
where musical expression depends on changes and nuances over time.

…
Compositional Pattern Producing Networks (CPPNs) [35] are a foundation of the explorations leading to the Novelty Search and Innovation Engine algorithms. The networks abstract unfolding development in evolutionary processes, which build a phenotype over time. This is done by using any variety of canonical functions at each node, based on the idea that the order in which the networks compose functions can provide that abstraction. This can be compared with the
Towards Sound Innovation Engines 3
process of timbral development, where musical expression depends on changes and nuances over time.



For synthesising sounds with CPPNs we use the classic synthesiser waveforms as potential activation functions:
- sine, square, triangle and sawtooth

A corresponding DSP graph can contain a variety of nodes, such as filters, noise, distortion, reverb, and specialised wavetable and additive synthesis nodes.

Each DSP node can receive audio and control signals from the CPPN, at any frequency
- each unique frequency requires a separate CPPN activation, for reach sample

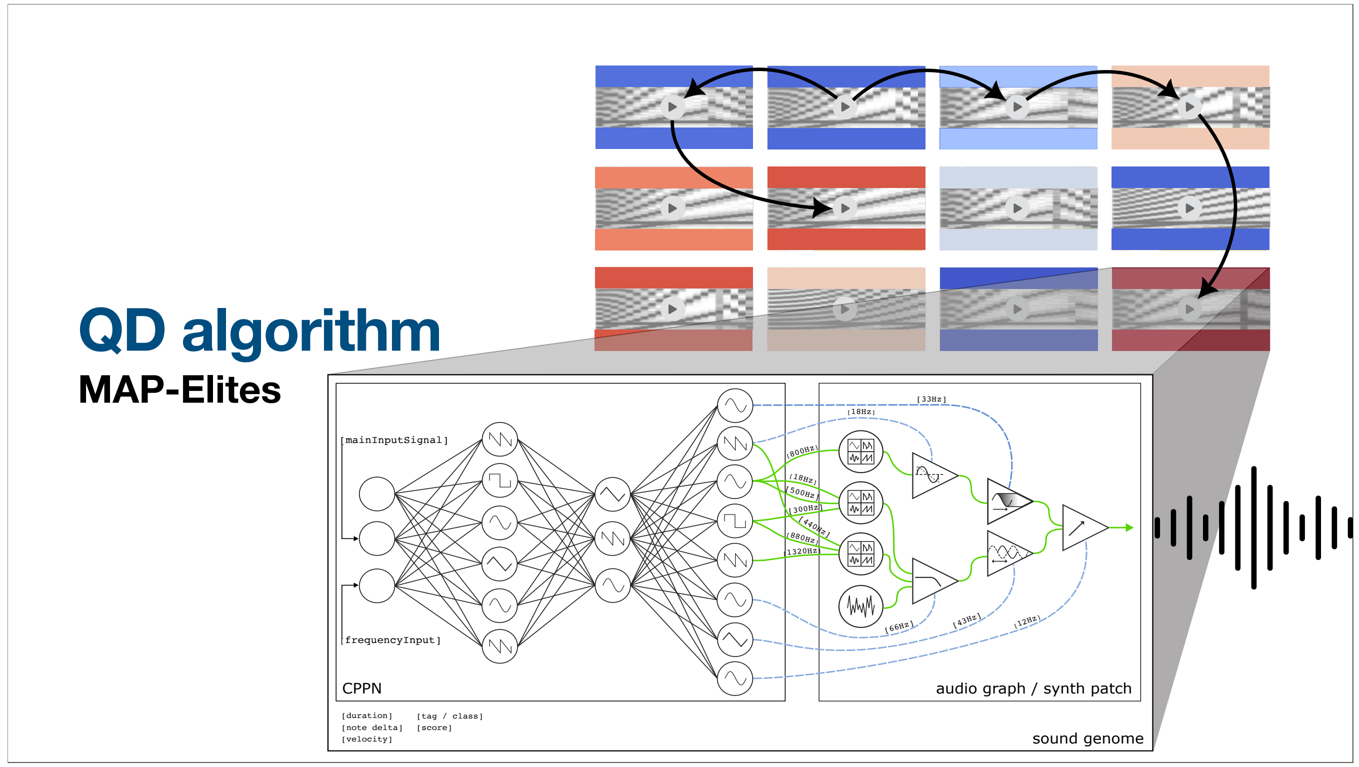
Linear ramp and periodic signals are input to the CPPN.

…

Periodic Signal Composition

One factor potentially influencing the search space is our choice of CPPN activation functions and node types in the DSP graph. CPPNs have commonly been used to compose Gaussian, sigmoid, and periodic functions, such as in [35,34]. In our case, the pattern-producing network can only compose periodic functions, commonly used as oscillators in a variety of sound synthesis techniques: sine, square, triangle, and sawtooth. The node types in the DSP graph are the same as in [33], in addition to custom nodes, which were added to the repertoire in an effort to widen the search space. Those additional nodes are a wavetable and a specialised additive synthesis node, where multiple audio signals are sourced from the CPPN to fill a table in the former and represent partials or harmonics in the latter. The wavetable is traversed according to a control signal, also sourced from the CPPN, in a manner similar to vector synthesis. The partials in the additive synthesis node can be slightly inharmonic, according to a mutable parameter to each.

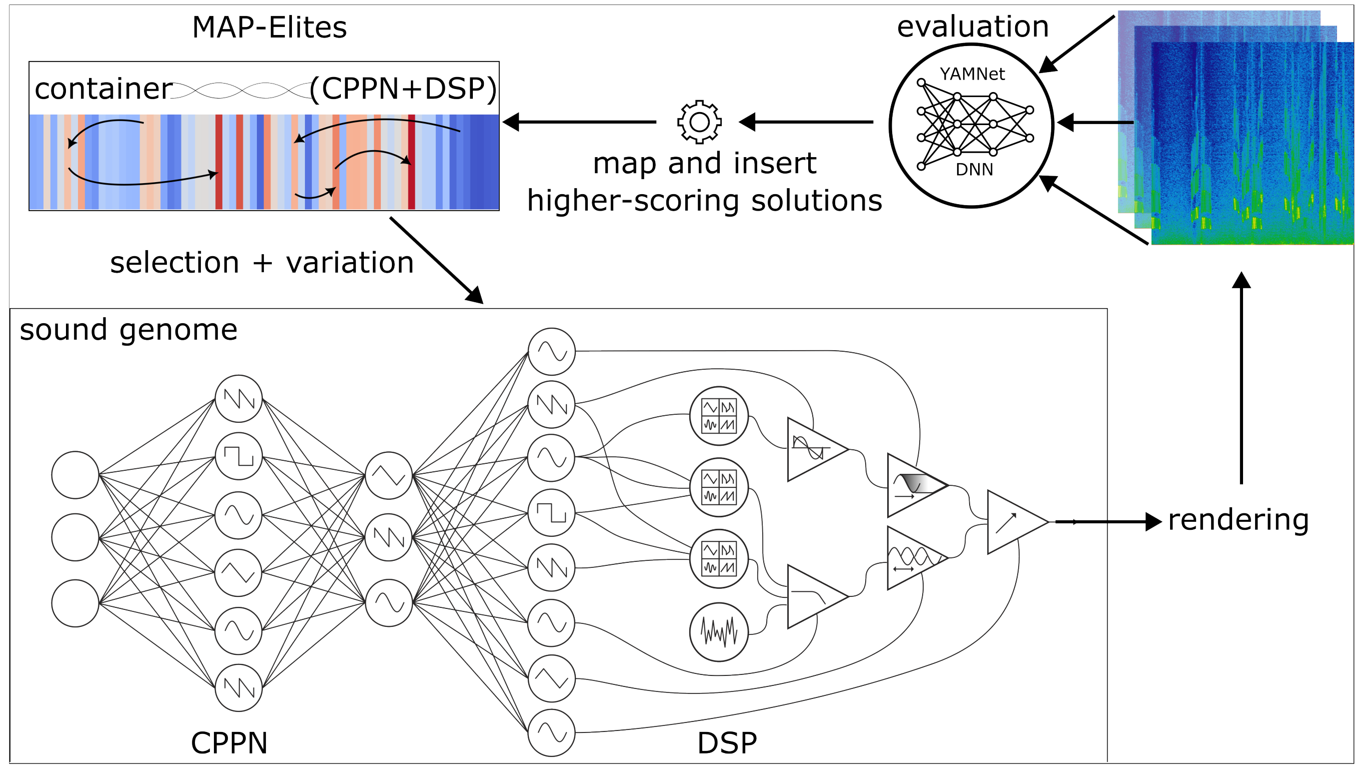
The duration of sounds rendered from each genome is defined by a linear ramp of values from –1 to 1 supplied to one CPPN input, while the pitch is controlled by the rate of a periodic (sine) signal at another input. Velocity is intended to simulate stimuli of different intensities when interacting with physical instruments, which is achieved by scaling the sine wave input by a velocity factor. The inputs are sampled at the same rate as the sampling rate of the audio graph.



We use MAP-Elites as our QD algorithm:
It usually divides the solution space, or container, into a grid, or cells:
In this experiment the classes from YAMNet define the cells.
So each cell in the container which MAP-Elites works on contains a sound genome,
which is rendered into a sound for evaluation,
and if it performs better on another class,
then it becomes the elite in that class.

…

Automating the discovery of new sounds is the goal of this study. We achieve this by applying the Innovation Engine algorithm to the sound synthesis approach developed in previous research on interactive novelty discovery. By using the proposed technique for sound synthesis, the system does not need to be trained beforehand as the evolutionary method starts from networks with no hidden nodes and progressively evolves primitive individuals by adding nodes and connections with the NeuroEvolution of Augmenting Topologies (NEAT) algorithm [36]. In our initial experiments, we use a signal from a pre-trained discriminative model to guide QD search, without human feedback in the evolutionary loop. Investigating this setup is intended to pave the way for further explorations of unbounded discovery of interesting sounds.



So to feed the Innovation Engine with Sound Objects, we use CPPN networks to emit patterns as waveforms
- they can be used as audio signals, either raw or fed through a Digital Signal Processing (DSP) graph
- when combined with a DSP graph, the CPPN patterns can also be used as control signals for various nodes in the DSP graph

The classification model - YAMNet - evaluates the Sound Objects, for diversity and quality,
and based on that evaluation, the QD algorithm - MAP-Elites -
potentially declares the new Sound Object as an elite in a new class,
if it's the highest performer there.

…

The use of patterns produced by CPPNs as sources of sound- and control signals for sound synthesis has been explored in a novelty seeking Interactive Evolutionary Computation (IEC) [37] configuration, which was inspired by previous work on the generation of visual artefacts [16]. The representation of temporal unfolding provided by CPPNs has been combined with the evolution of Digital Signal Processing (DSP) graphs during several iterations of investigation, detailed in [14]. This resulted in a distinct approach to sound synthesis, where any combination of the two graphs, depicted in figure 1, can be rendered at any duration, revealing the sub-patterns encoded by CPPNs over varying periods of time.

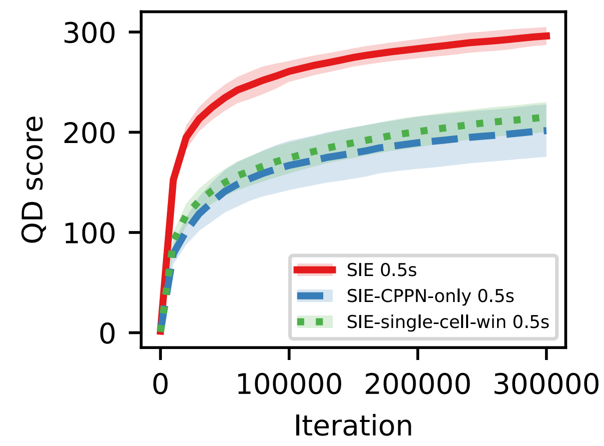
…

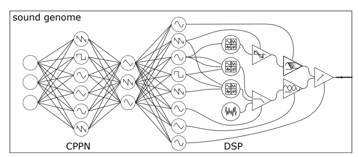
To start evaluating the applicability of the Innovation Engine algorithm in the domain of sounds, we combine a sound synthesis technique with a supervised discriminative model. The foundation of our sound-generating system relies on using the patterned outputs from CPPNs as the raw materials for sound and control signals. These signals can be utilised in their original form or further shaped through a DSP graph. Such a design choice enables the evolutionary state to begin from a blank slate, established with random initialization of the CPPN and DSP graph counterparts. This avoids dataset constraints that might limit the potential for discovery of novel sounds. The genome evolved by the evolutionary (QD) processes is composed of the CPPN and DSP networks and the evolvable connections between them. Details of this genome configuration are discussed and diagrammed in [14]. Figure 1 illustrates the data flow of our experimental setup and shows how the genome fits within the data pipeline.

Results

Signal Processing Graph

- QD-score:
  - CPPN + DSP
    - vs
  - CPPN only





So what did we find?

One quantitative measure is the QD-score, which summarises the performance from all classes in the container / map:

Here we see the the red line of our baseline experiment reaching a QD-score of around 300, out of a maximum of 500:
- 500 class and maximum classifier confidence of 1 for each

But the question came up: why we were coupling CPPNs with DSP graphs,
why not just let the CPPNs alone emit the pattern of a sound waveform?

So we tried that:
instead of wiring multiple CPPN outputs to multiple DSP graph nodes,
we tried using just one CPPN output for the sound,
with no DSP:
We can see that the performance of that configuration, in terms of a QD-score, is less
- also at the cost of higher CPPN network complexity and resulting longer rendering times, as we'll see later

…
Signal Processing Graph
To investigate the impact of merging CPPNs with DSP graphs, we set up evolutionary runs in two distinct configurations: one in which an evolved CPPN functioned solely

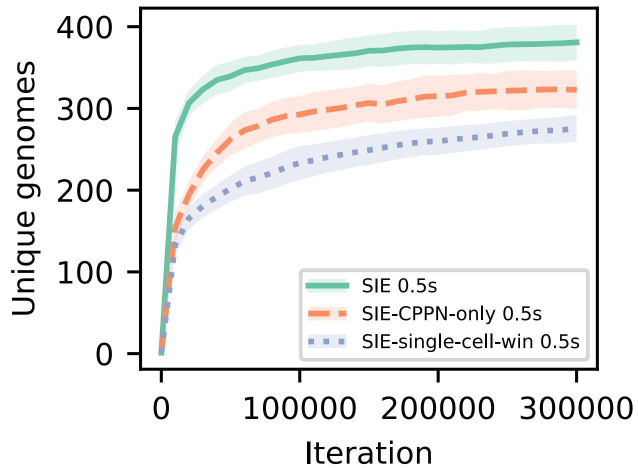
as the audio signal source, providing a single output, and another where the CPPN was paired with an evolving DSP graph, allowing it to offer a multitude of audio and control signals, from up to 18 outputs.

In our experiments, we quantify the QD algorithm performance by calculat- ing the QD-score [32,31]. This score is determined by summarising the confidence levels of the elites across the various classes delineated by YAMNet. When com- paring the results from these runs, we observe in Figure 2 that the phenotypes (i.e., sound objects) produced from the genomes where CPPNs and DSP graphs were co-evolved achieved the highest overall QD-score. Through informal listen- ing sessions conducted by the authors, it was observed that the sounds rendered from runs where the evolution of DSP graphs was allowed alongside CPPNs ex- hibited a higher degree of subjective aesthetic appeal. This phenomenon could potentially be attributed to the prevalence of classical synthesizer sounds, to which our ears have grown accustomed. In this context, the DSP graph can be seen as functioning akin to a modular synthesizer patch, rendering us less inclined to perceive the raw output generated by CPPNs as inherently pleas- ing. The rendered sounds can be auditioned in an online explorer (sec. 3.7) or accompanying dataset [15].

Results

Elite Populations

- More diversity when evaluating sound objects from CPPN + DSP genomes



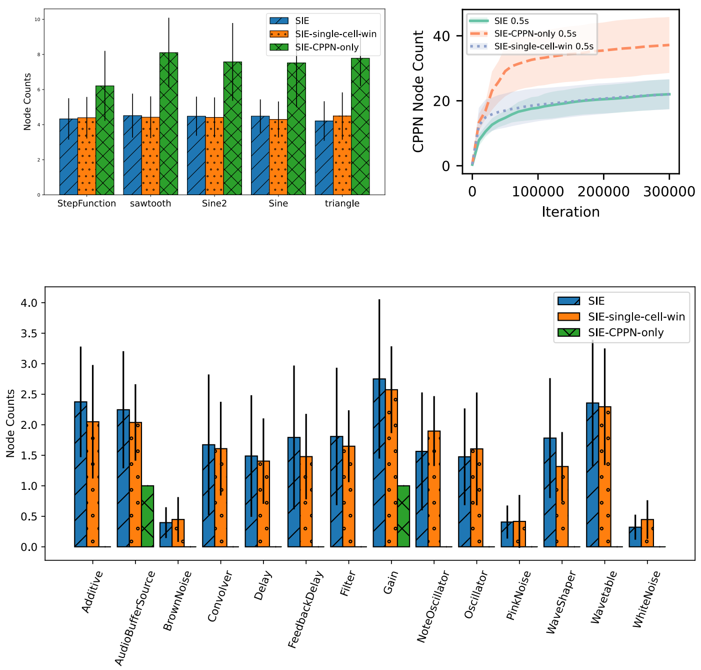
We observed more diversity when coupling CPPNs with DSP graphs:
The set of unique elites at the end of CPPN-only runs is smaller than when co-evolving the DSP graphs

…
Elite Populations
Figure 3 (left) shows that the range of iterations where the current elites are found at the end of each run is sharply delimited around iterations 150K to 250K of the CPPN-only runs, while the CPPN+DSP runs continue to discover new elites more gradually throughout the latter half of the runs.
The set of unique elites at the end of CPPN-only runs is smaller than when co-evolving the DSP graphs, as plotted in figure 3 (right). Instead of distinguish- ing between individuals by their ID, where the differences could be only slight changes in e.g. connection weights, this plot is based on distinction between unique combinations of CPPN and DSP node and connection counts.

Results

Genome Complexity

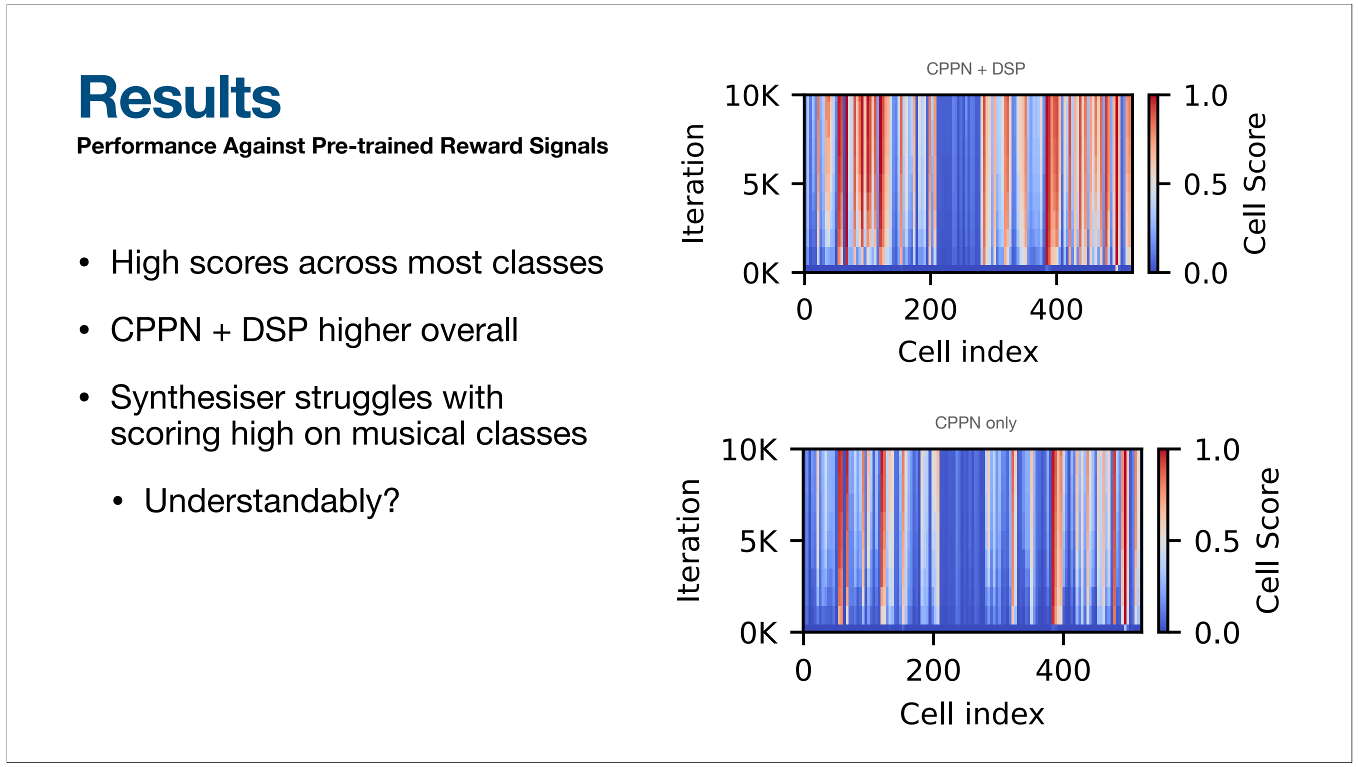
- Uniform set of CPPN activation functions
- Custom wavetable and additive synthesis DSP nodes prominent
- CPPN only genomes with higher CPPN node count
    - Compensating for the lack of DSP?



- The distribution of CPPN activation function types is quite uniform in all variants of our runs
- Apart from the stock Web Audio API nodes, custom DSP nodes, for wavetable and additive synthesis are prominent
    - Might be worthwhile to implement other classic synthesis techniques for the DSP part
- CPPN-only runs resulted in more complex function compositions, likely to compensate for the lack of a co-evolving DSP graph

…

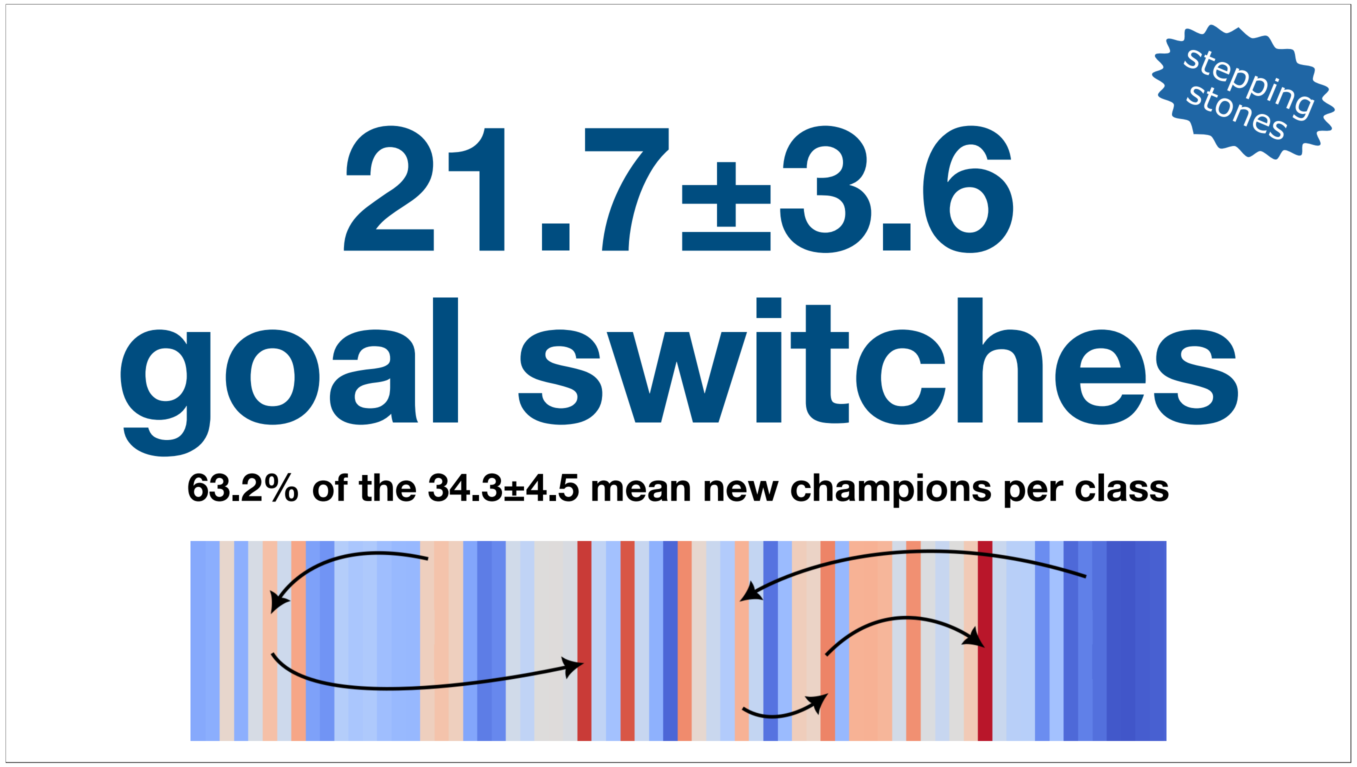
The composition of audio graph nodes and CPPN activation functions can be seen in figure 4, where the prominence of the custom audiograph nodes (wavetable and additive synthesis, fig. 4, bottom) suggest that implementing other known techniques from the history of sound synthesis may be worthwhile. The distribution of CPPN activation function types is quite uniform in all vari- ants of our runs (fig 4, top). It's also interesting to observe in the left plot of fig- ure 6 that the CPPN-only runs resulted in more complex function compositions, likely to compensate for the lack of a co-evolving DSP graph. This increased CPPN complexity resulted in longer rendering times and thus increased durations of the evolution runs, as that part of the genome is more computationally expensive, with potentially many network activations required for each sample, as discussed in [14].



- High scores across most classes
- CPPN + DSP higher overall
- Synthesiser struggles with scoring high on musical classes
    - Understandably?

…

The YAMNet classifier chosen in this iteration of our investigations assigned high scores to the sounds generated by our system across most classes, as can be seen in figure 5. There we can see again how the co-evolution of CPPNs with DSP graphs achieves higher scores overall. The figure also reveals how the synthesiser struggles in the range of classes between 214 and 276, which classify musical genres, rather than distinct sounds or instruments, such as “Pop music”, “Rhythm and blues”, “Flamenco”, etc. This is reasonable as the system is expected to generate sounds useful in the process of creating e.g. music, rather than entire musical compositions. Nonetheless it can be interesting to observe what the system came up with for those low-confidence classes, such as “Theme music”: a filter can be set in the online explorer (sec. 3.7) to audition classes containing the phrase “music” while scrubbing through the runs with a slider.



How did evolution leverage the diversity promoted by our classifier?

We measured the stepping stones across the classes by counting goal switches:

"the number of times during a run that a new class champion was the offspring of a champion of another class"

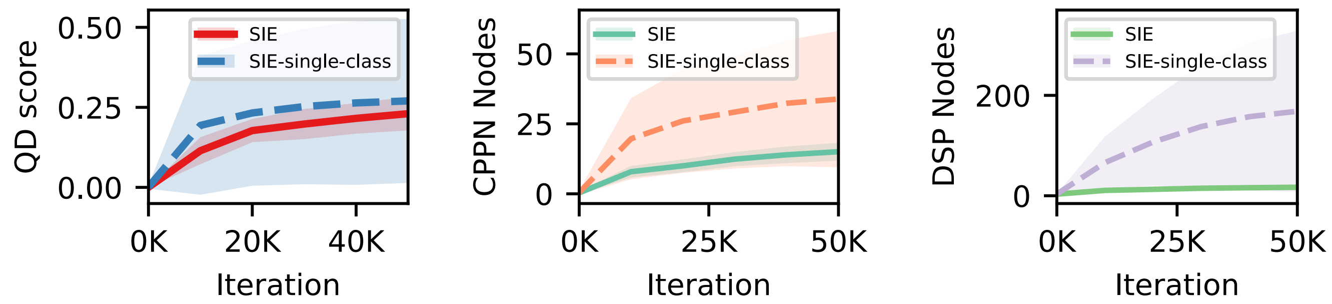
we measured a mean of 21.7±3.6 goal switches

63.2% of the 34.3±4.5 mean new champions per class

This can be compared to the 17.9% goal switches in previous Innovation Engine experiments with image generation.

…
Evolutionary Stepping-Stones
To assess how evolution leveraged the diversity promoted by our classifier, we conducted two measurements that explored the stepping stones across various classes. One has been called goal switching and defined as "the number of times during a run that a new class champion was the offspring of a champion of another class" in [28,26]. From our runs we measured a mean of 21.7±3.6 goal switches, 63.2% of the 34.3±4.5 mean new champions per class. This can be compared to the 17.9% goal switches reported in [26]. Another way of measuring how the evolutionary paths flow though the stepping stones laid out by the classifier is to trace through the phylogenetic tree leading to each elite and then count how often its parent comes from a class different from the one it occupies. Counting from the current elites of each class at the end of the evolution runs, we found a mean of 44.9±14.7 such occurrences. In lieu of a visual phylogenetic presentation, the generation slider of the evolution

runs explorer (section 3.7) can dynamically reveal how elites for each class come from different, often unrelated classes during the course of evolution.



Abandoning Diversity

Similar performance at the cost of significantly higher complexity

We were also curios to see how a single objective search would compare in the domain of sound,
- where all effort spent on one goal

So we selected 10 classes as single objectives of separate runs and
compared the performance
and genome complexity with the performance from the QD runs on those same classes.

The single objective runs scored similarly to the QD runs, though with a high variance,
But at the cost of much higher genome complexity.

The unexpected result of higher performance from the single-class runs may be attributed to the narrow set of chosen classes

…
Growth of genome complexity seems to have stayed within reasonable limits, even when CPPNs were left alone to the task of performing against the clas- sifier (fig. 6). An exception to this is when we experimented with abandoning diversity and adopting single objectives. Though the benefit of diversity has been demonstrated [28,26], we investigated how a similar experiment fares in the sound domain. To that end, we selected 10 classes as single objectives of separate runs and compared the performance and genome complexity with the performance from the QD runs on those same classes.

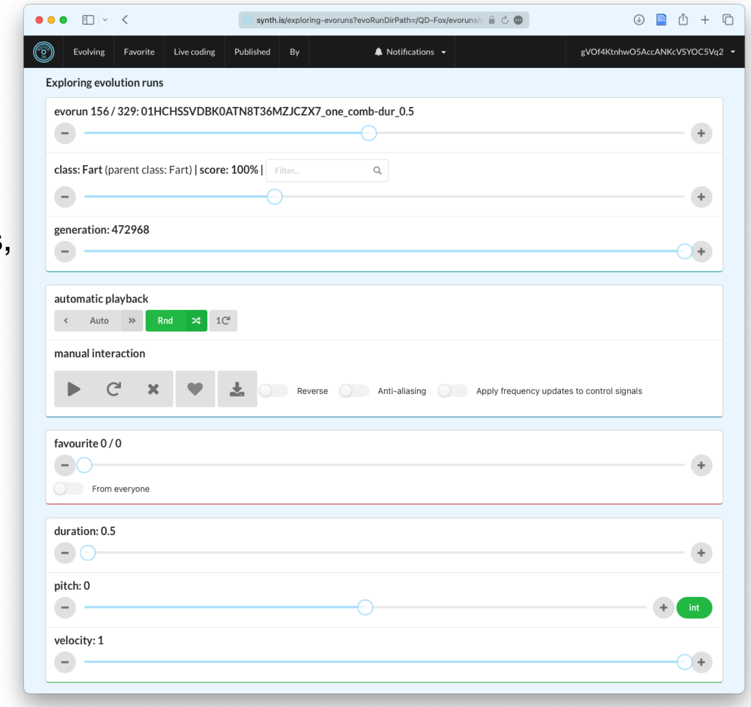
Interestingly, although the performance in single objective runs is higher on average than in multi-class runs, as shown in the first plot in figure 7, the difference is accompanied by a higher level of deviation and much higher genome complexity. The second and third plots in figure 7 indicate that the CPPN and DSP node counts in genomes from single objective runs are significantly higher than those of genomes from the same set of classes in QD runs. The computational effort required for the complex genomes evolved during the single class runs limited our iteration count to 50 thousand, 1/6th of the iterations performed for the baseline QD runs. The unexpected result of higher performance from the single-class runs may be attributed to the narrow set of chosen classes; this experiment could benefit from further investigation.

Single-class runs were performed on the classes Aircraft, Banjo, Beatboxing, Boom, Choir, Dubstep, Fusillade, Mandolin, Synthetic singing and Whistling.

Sound Objects and their Application

Evoruns Explorer

- Scrub through evolutionary runs, classes and generations

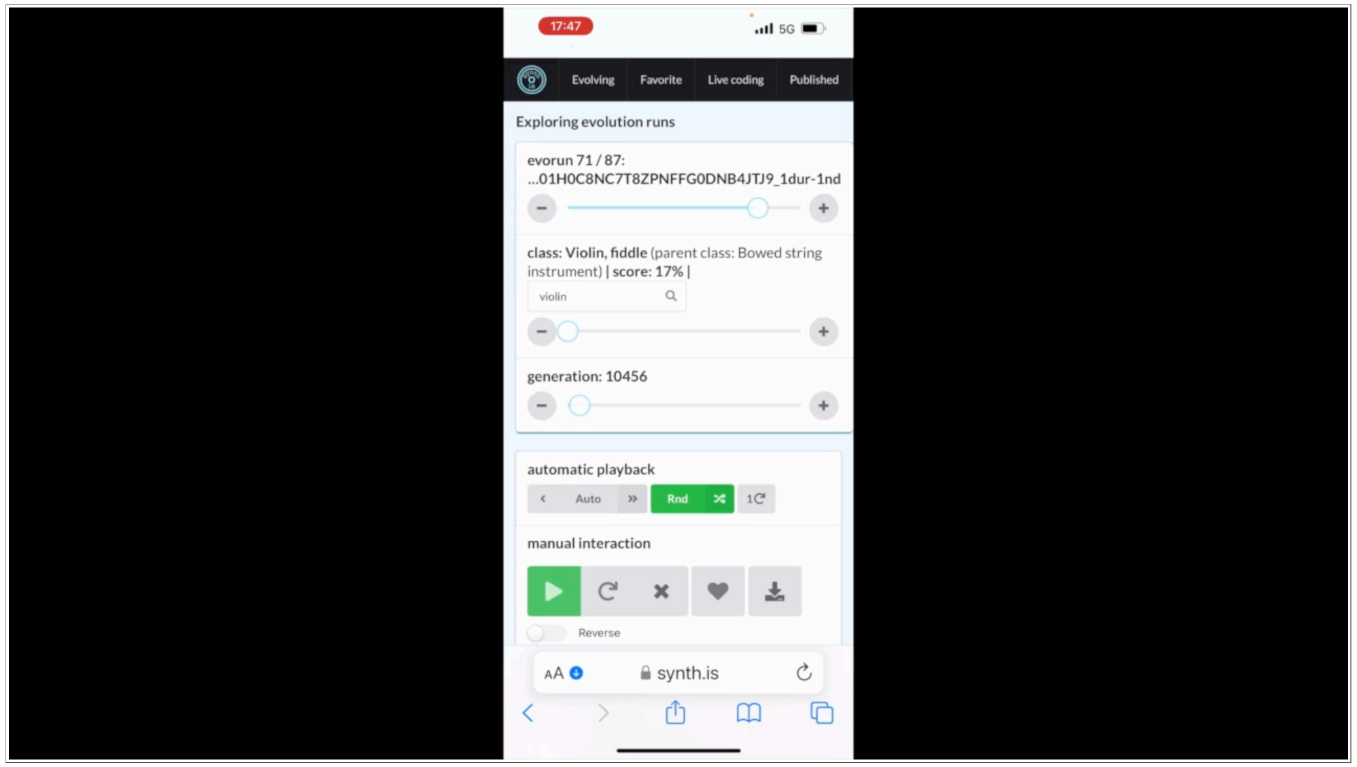


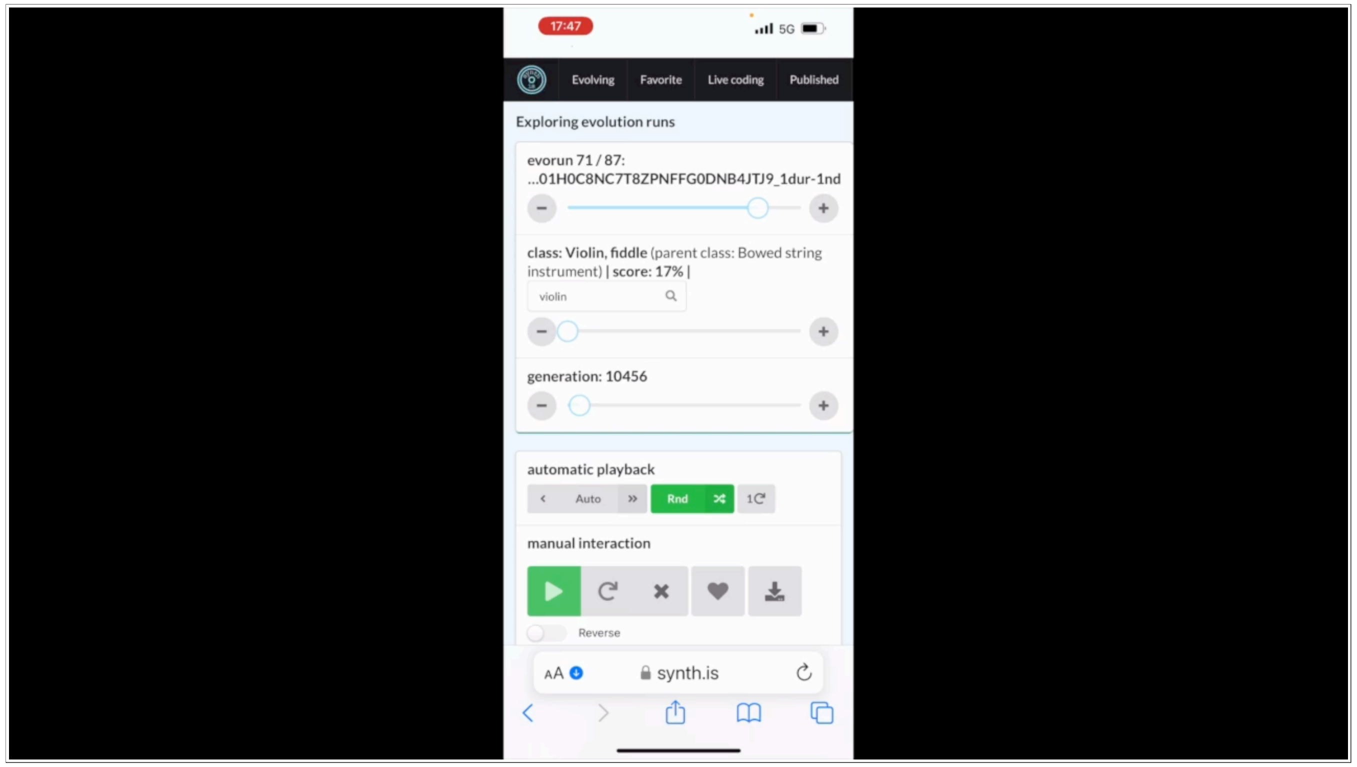
In addition to numbers and plots, it's also interesting to hear the sounds discovered:
Instead of showing a visual phylogenetic tree of the evolutionary paths,
We offer an online explorer which offers sonic access to the results from all our experiments:

There you can scrub through evoruns and their classes,
and for each class you can scrub through the generations throughout the run,
which can in many cases reveal the goal switching behaviour that we've measured.

…

We have facilitated open access to the generated artifacts through different means. Those include an evolution runs explorer7, depicted in figure 8a. Fi- nal elites from all runs have also been rendered to (128563) WAV files, which have been included in the accompanying dataset [15]. The sound objects in the pre-rendered files reflect the render-settings used to evaluate the corresponding genome that became an elite. The online explorer7 provides greater flexibility as it dynamically renders sounds with the default settings, but the interface also enables users to modify these settings. This modification can potentially reveal other intriguing sonic behaviors from the same genome.

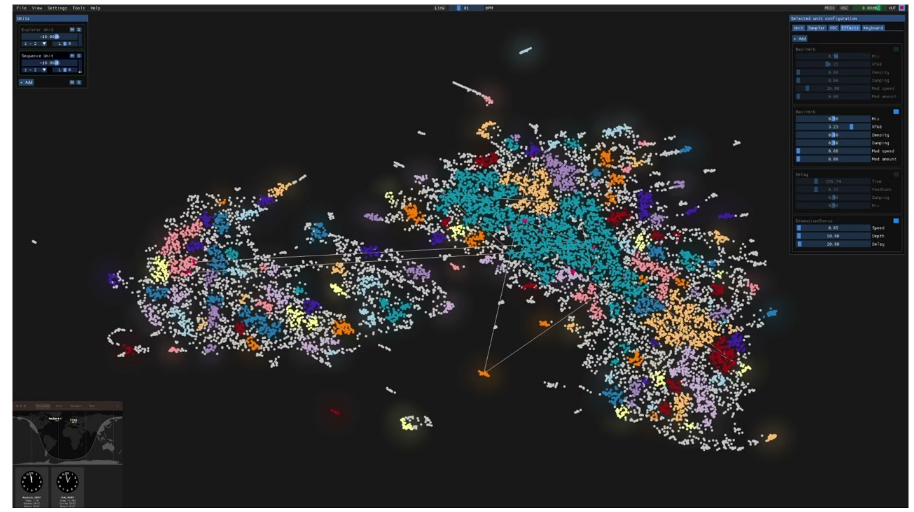

17:47
5G
Evolving
Favorite
Live coding
Published
Exploring evolution runs
evorun 71 / 87:
...01H0C8NC7T8ZPNFFG0DNB4JTJ9_1dur-1nd
class: Violin, fiddle (parent class: Bowed string instrument) | score: 17% |
violin
generation: 10456
automatic playback
Auto
Rnd
1
manual interaction
Reverse
AA
synth.is


17:47
5G
Evolving
Favorite
Live coding
Published
Exploring evolution runs
evorun 71 / 87:
...01H0C8NC7T8ZPNFFG0DNB4JTJ9_1dur-1nd
class: Violin, fiddle (parent class: Bowed string instrument) | score: 17% |
violin
generation: 10456
automatic playback
Auto
Rnd
manual interaction
Reverse
AA
synth.is

Sound Objects and their Application

Evolutionary Sequences

- Live-stream playlist:



It’s also interesting to observe if the discovered sounds are of any use in music composition or sound organisation.

To test that applicability, the sounds have been loaded into the experimental sampler AudioStellar,
Which is then configured to drive evolutionary sequences through the sound objects.

Several live streams with such sequencing have been broadcast and recordings from those are available online.

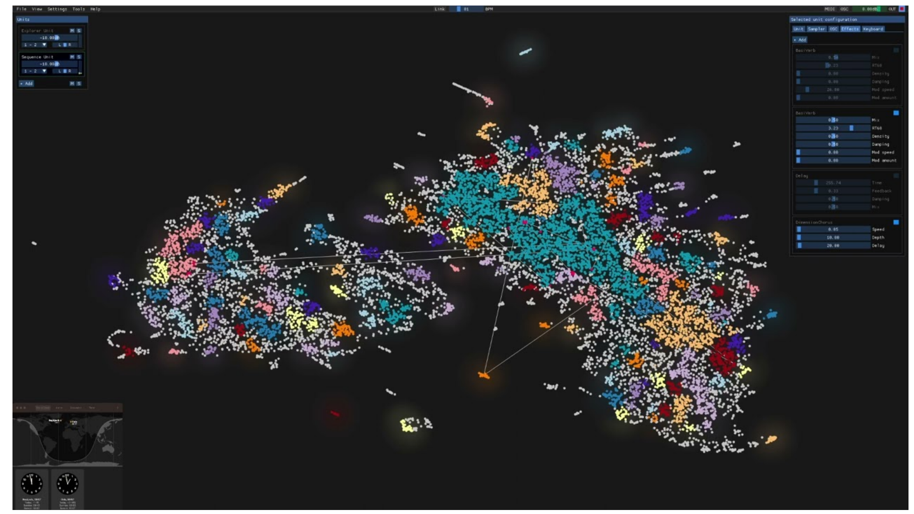
Many hours of recordings - let’s listen to one of those hours…

…
As part of our investigation into the applicability of the discovered artefacts for creating other art, we loaded subsets of them into the experimental sampler AudioStellar [8] and used that software to drive evolutionary sequences through the phenotypes. A playlist of live-stream recordings showcasing evolutionary sequences using sounds discovered by QD runs is accessible online8. These com- positions are largely automated, with human input limited to initial settings like evolutionary sequencing rates and fundamental sound effects. Nonetheless, they demonstrate the potential of the discovered sound objects to inspire creative endeavors.

Sound Objects and their Application

Evolutionary Sequences

- Live-stream playlist:

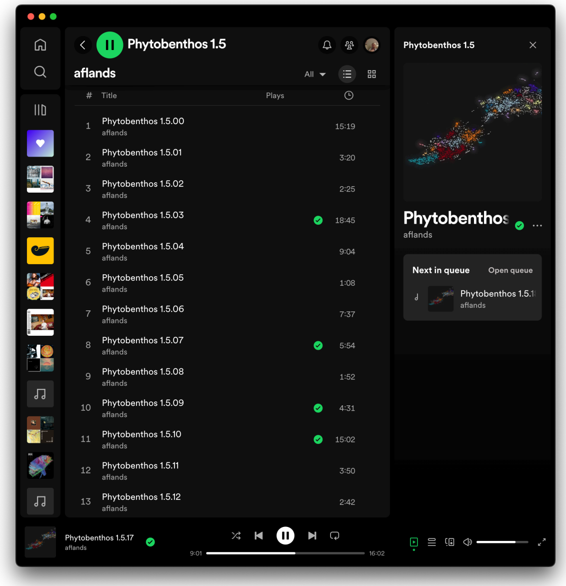


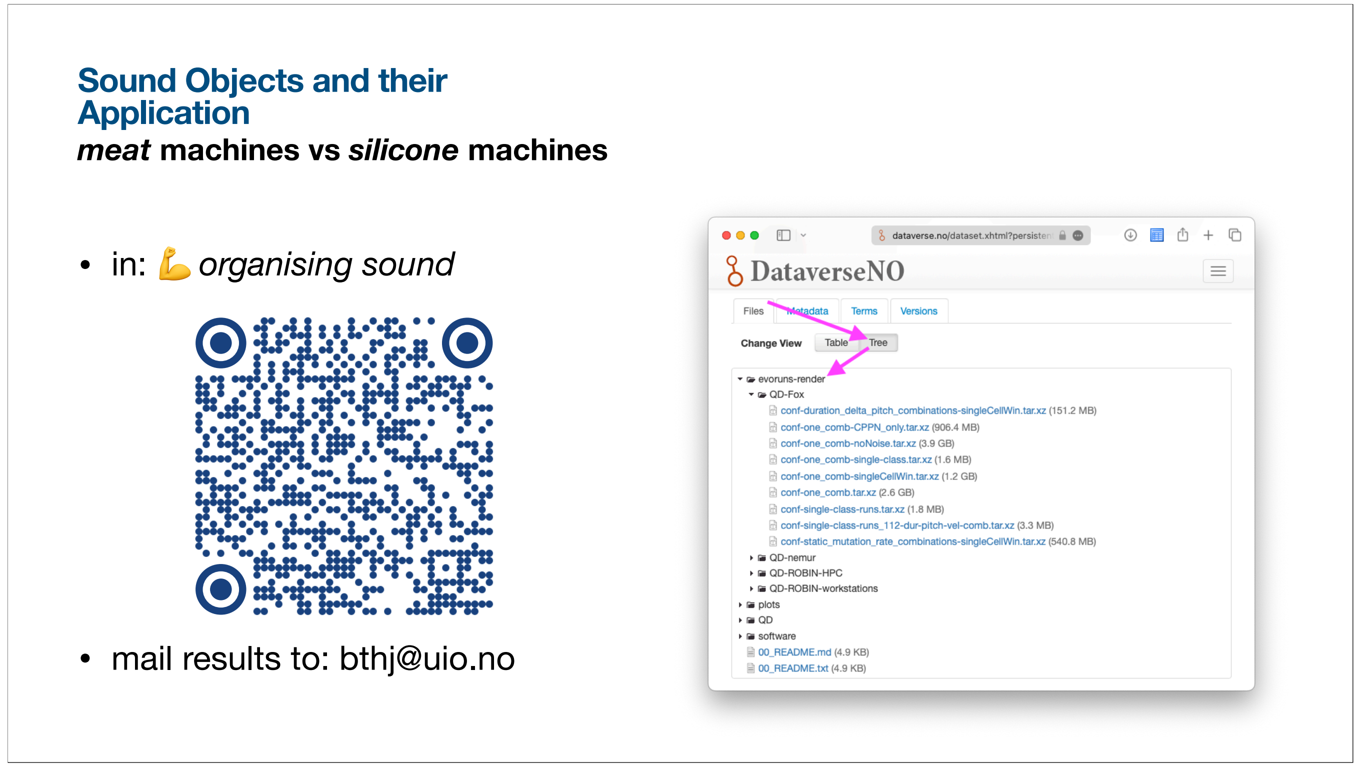
Sound Objects and their Application

Evolutionary Sequences

- Some of this stuff is actually on main-streaming services (WIP):

- IndieWeb publication planned





So we’ve tried to have silicone based machine compose stuff with those sounds.

But what can meat based machines - we - do with those sounds?

Here’s an opportunity to give that a try…

It is thought-provoking to consider if a human, given the same dataset, could craft more aesthetically pleasing arrangements with these sonic artefacts. We encourage the reader to obtain a copy of the files and engage in such experimentation [15].

…

In the broadest sense, the concept of instruments has evolved from being a mere means to an end to a starting point for a journey into the unknown [21, p. 49]. The evolutionary system explored here is not intended as an instrument for serving requests from preconceived ideas but rather as a tool for discovering interesting sound objects that can steer the creative journey. The sound artefacts generated by our system, as discussed in this paper, are intended to facilitate or inspire the creation of further sonic art. This is different from the visual artefacts produced by many generative systems, which are often seen as standalone pieces without further utility. Instead of a top-down approach—where the end goals and characteristics of the desired sound are pre-defined—our method encourages a bottom-up process of exploration. This reflects the evolutionary path of human development, where cognitive skills have been shaped by the very tools that humans have uncovered. This echoes the saying, “the tool writes the toolmaker as much as the toolmaker writes the tool” ([4] as cited in [21, p. 5]). An instrument that promotes such exploratory discovery can enable us to continue on our path of evolution by developing human abilities through technology.

con clu si on

Evoruns Explorer

Silicon Machine Composer

For Meat Machine Composers

- Applying diversity-promoting algorithms with classifier reward signals is a viable approach for sound discovery
- The sound synthesis approach employed achieved high confidence from a DNN classifier
- Diverse and innovative sound objects were generated, suggesting further explorations in this system

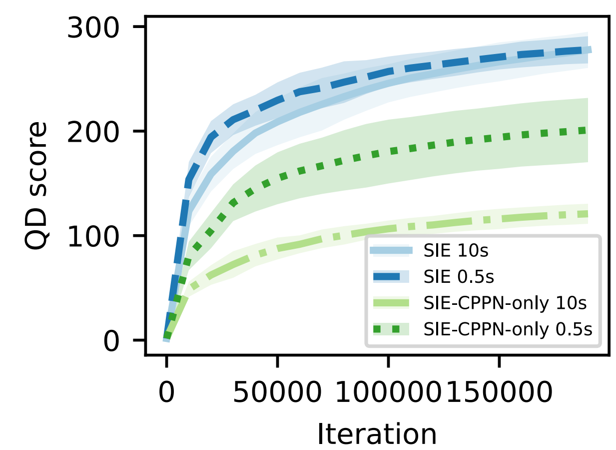
- demonstrated that it is a viable approach to apply diversity-promoting algorithms with classifier reward signals for sound discovery
- our current sound synthesis approach can achieve high confidence from a DNN classifier
- the diverse set of sounds generated suggests further explorations in this system

<...>

Results

Temporal Pattern Revelation and Classifier Characteristics

- Longer time for sounds to develop doesn't benefit CPPN-only sounds
  - Lack of DSP more apparent?
- CPPN + DSP duration agnostic



TODO: skip

Our baseline runs rendered sounds with the duration of half a second,
but we were curios to see the effect of allowing more time for the patterns from the CPPNs to reveal against the classifier:
YAMNet is trained on 10 second sounds, so we configured a set of runs rendering sounds of that duration.

we hypothesised that allowing the classifier to sample in more detail
the patterns developed by the CPPNs,
when processing more frames over a longer duration, would result in higher confidence:

For the CPPN-only runs, the opposite turned out to be the case,
where rendering 10s sounds achieved a lower QD score than corresponding runs rendering 0.5s sounds.
Perhaps the lack of DSP becomes more significant in the evaluation of longer duration sounds.
Duration has little effect when DSP graphs evolve alongside the CPPNs,

…
Although half a second sounds were the most prevalent renditions of successful individuals in our manual parameters search (section 2), comparing sets of runs with two large variations in the duration of the evaluated phenotypes was in- teresting. We chose to compare runs evaluating half a second renditions of the evolved genomes with a set of runs evaluating ten-second renditions. One mo- tivation for the choice of the longer duration, is that "YAMNet is trained on 1,574,587 10-second YouTube soundtrack excerpts from within ... AudioSet"6. While CPPN-only runs achieved less overall confidence when rendering 0.5s sounds for evaluation by the classifier, as

can be seen on the left of figure 2, we hypothesised that allowing the classifier to sample in more detail the patterns developed by the CPPNs, when processing more frames over a longer duration, would result in higher confidence. The opposite turned out to be the case, where CPPN-only runs, rendering 10s sounds for evaluation achieved a lower QD score than corresponding runs rendering 0.5s sounds. Perhaps the lack of DSP becomes more significant in the evaluation of longer duration sounds. Duration has little effect when DSP graphs evolve alongside the CPPNs, as the right plot in figure 2 shows.

57.4%±3.4%

- map coverage, when incremental - otherwise immediately full coverage

Because we have a classifier capable of measuring placement and performance across the whole measurement space at once,
we can immediately reach full coverage.

To examine the effect of gradually covering the map of cells by allowing each candidate to potentially win only one cell,
the one where it receives the highest confidence from the classifier,
we performed an identical set of runs except with that restriction in place.
That configuration reached a coverage of 57.4%±3.4%,
with their QD-score following a trajectory similar to that of full coverage CPPN-only runs, as we saw on the previous plot.

…
Behaviour Space Coverage
The default behaviour of our MAP-Elites im- plementation allows each evaluated individual to win all cells where it performs better or where there is a vacancy, so it reaches full coverage from the first seed. To examine the effect of gradually covering the map of cells by allowing each candidate to potentially win only one cell, the one where it receives the highest confidence from the classifier, we performed an identical set of runs except with that restriction in place. Runs where at most one cell at a time is won reached a coverage of 57.4%±3.4%, with their QD-score following a trajectory similar to that of full coverage CPPN-only runs, as depicted in figure 2, left.