

kromosynth

A Sound Innovation Engine with Evolutionary Algorithms

bthj - 2023-03

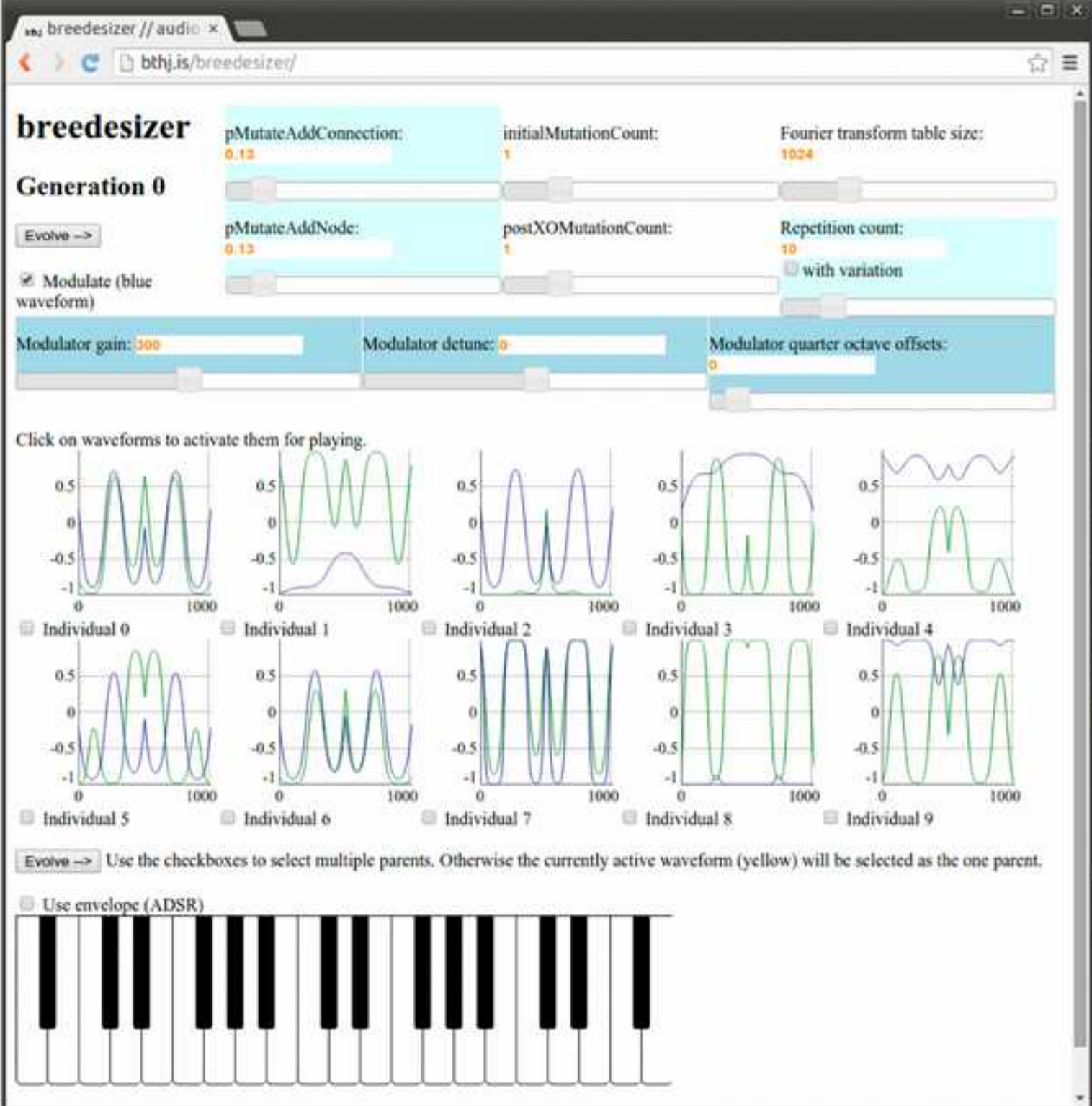




IEC

for generative art

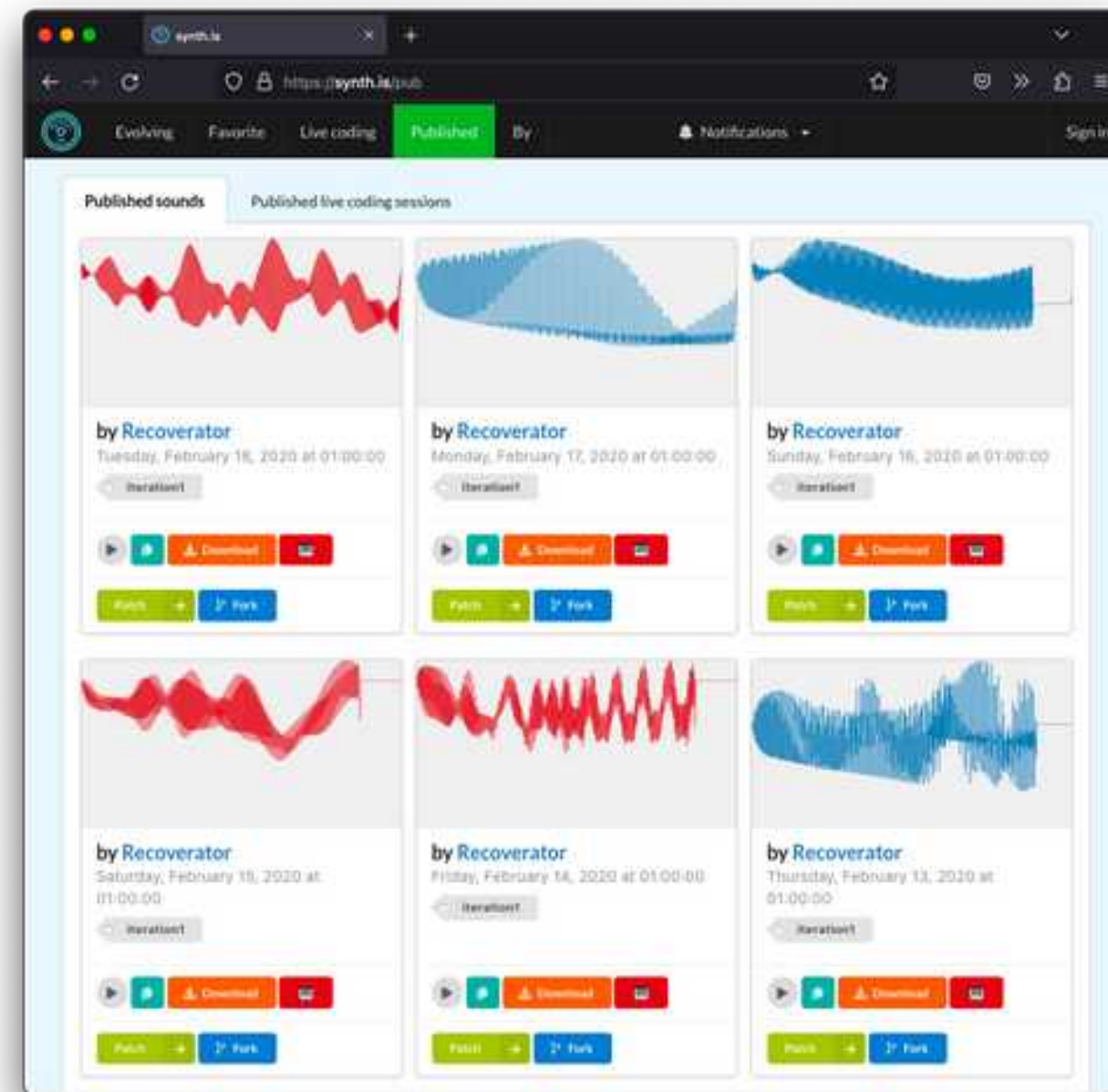
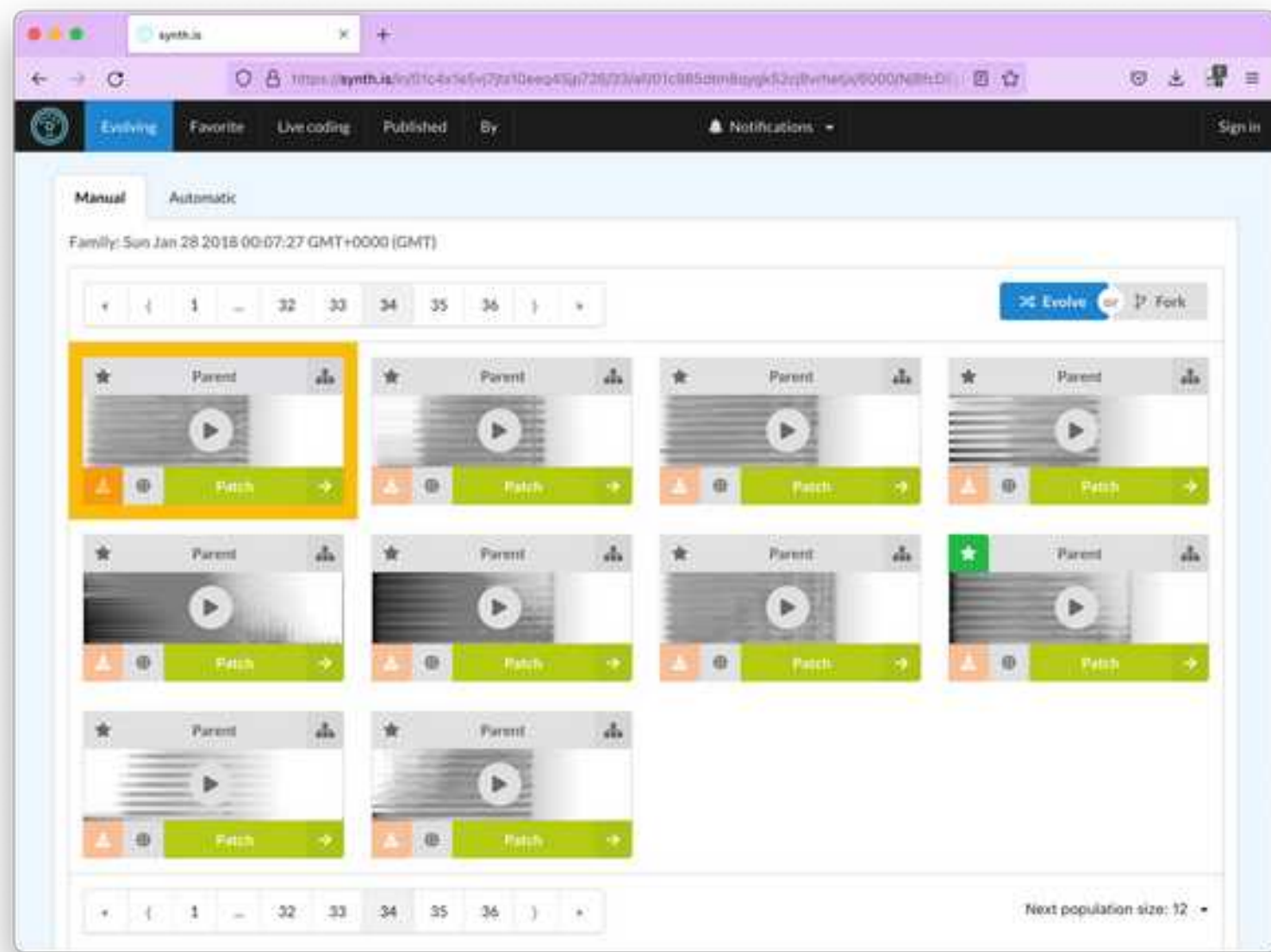
<p>shiny Red apple By acow</p>  <p>Evolve DNA ★★★★★</p>	<p>night Bridge By ddambro</p>  <p>Evolve DNA ★★★★★</p>	<p>alien duck By burndirt</p>  <p>Evolve DNA ★★★★★</p>	<p>butterfly in sky By happybethel</p>  <p>Evolve DNA ★★★★★</p>
<p>Landscape By shmoopy</p>  <p>Evolve DNA ★★★★★</p>	<p>behind a chicken By ken</p>  <p>Evolve DNA ★★★★★</p>	<p>basket By ken</p>  <p>Evolve DNA ★★★★★</p>	<p>Chick bird By burndirt</p>  <p>Evolve DNA ★★★★★</p>



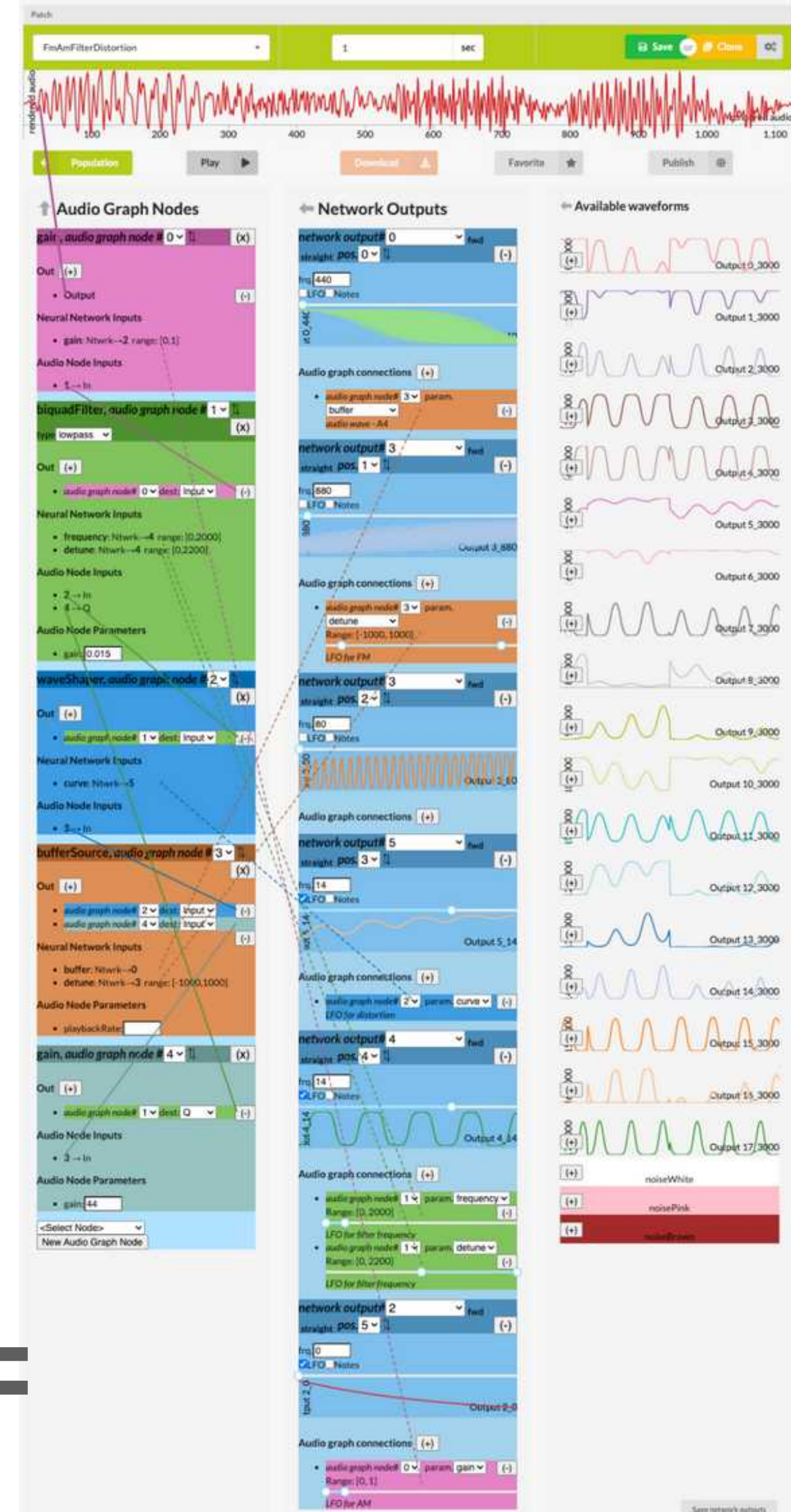
The screenshot shows the 'breedesizer' web application interface. At the top, there are several sliders and input fields for parameters: 'pMutateAddConnection' (0.13), 'initialMutationCount' (1), 'Fourier transform table size' (1024), 'pMutateAddNode' (0.13), 'postXOMutationCount' (1), and 'Repetition count' (10). Below these are checkboxes for 'Modulate (blue waveform)' and 'with variation'. There are also sliders for 'Modulator gain' (500), 'Modulator detune' (0), and 'Modulator quarter octave offsets' (0). The main area contains ten waveform plots labeled 'Individual 0' through 'Individual 9'. A yellow highlight is on the first waveform (Individual 0). Below the waveforms is a piano keyboard interface with a 'Use envelope (ADSR)' checkbox. The browser address bar shows 'bthj.is/breedesizer/'.

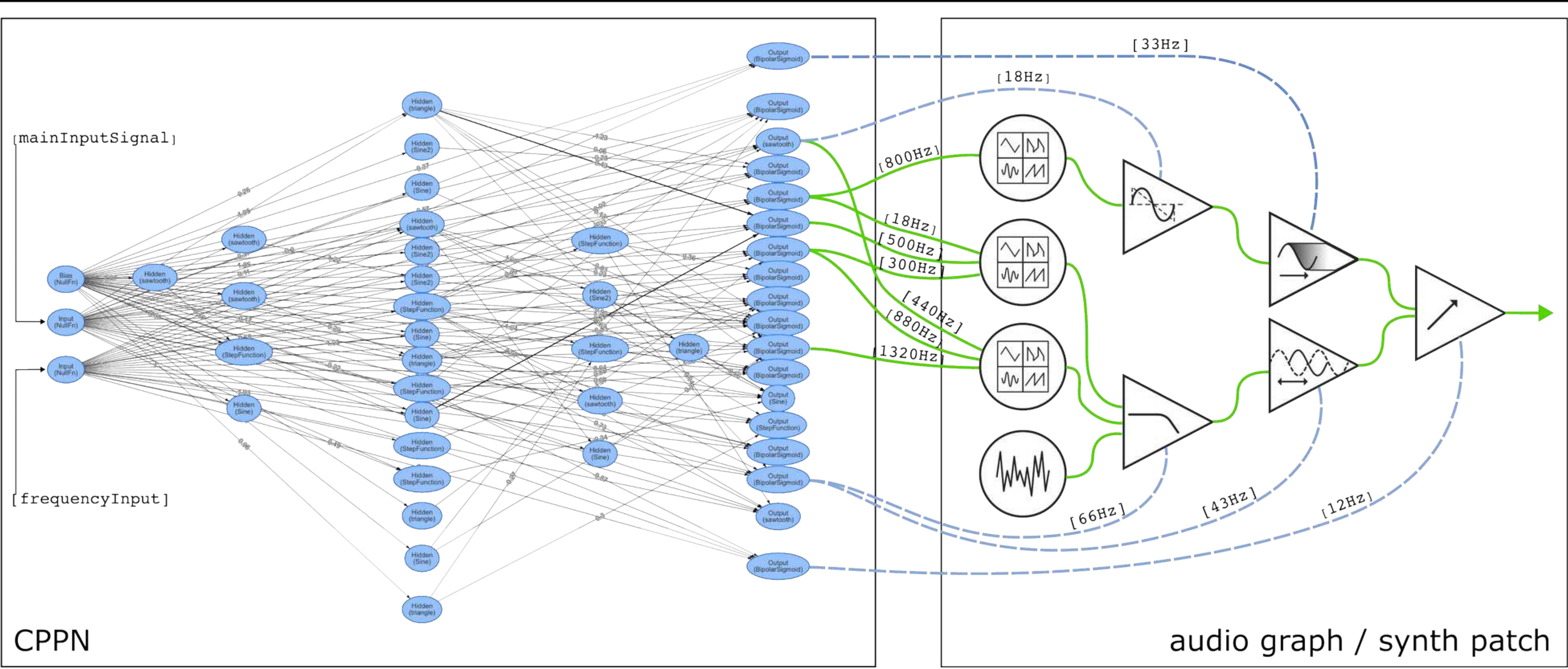
IEC

and timbral development



=





CPPN

audio graph / synth patch

[duration] [tag / class]
 [note delta] [score]
 [velocity]

sound genome

Innovation Engines

Full Citation: Nguyen A, Yosinski J, Clune J (2015) *Innovation Engines: Automated Creativity and Improved Stochastic Optimization via Deep Learning*. Proceedings of the Genetic and Evolutionary Computation Conference.

Innovation Engines: Automated Creativity and Improved Stochastic Optimization via Deep Learning

Anh Nguyen
University of Wyoming
anguyen8@uwyo.edu

Jason Yosinski
Cornell University
yosinski@cs.cornell.edu

Jeff Clune
University of Wyoming
jeffclune@uwyo.edu

Keywords

Deep Neural Networks; Deep Learning; MAP-Elites

ABSTRACT

The Achilles Heel of stochastic optimization algorithms is getting trapped on local optima. Novelty Search avoids this problem by encouraging a search in all interesting directions. That occurs by replacing a performance objective with a reward for novel behaviors, as defined by a human-crafted, and often simple, behavioral distance function. While Novelty Search is a major conceptual breakthrough and outperforms traditional stochastic optimization on certain problems, it is not clear how to apply it to challenging, high-dimensional problems where specifying a useful behavioral distance function is difficult. For example, in the space of images, how do you encourage novelty to produce hawks and heroes instead of endless pixel static? Here we propose a new algorithm, the Innovation Engine, that builds on Novelty Search by replacing the human-crafted behavioral distance with a Deep Neural Network (DNN) that can recognize interesting differences between phenotypes. The key insight is that DNNs can recognize similarities and differences between phenotypes at an abstract level, wherein novelty means interesting novelty. For example, a novelty pressure in image space does not explore in the low-level pixel space, but instead creates a pressure to create new types of images (e.g. churches, mosques, obelisks, etc.). Here we describe the long-term vision for the Innovation Engine algorithm, which involves many technical challenges that remain to be solved. We then implement a simplified version of the algorithm that enables us to explore some of the algorithm's key motivations. Our initial results, in the domain of images, suggest that Innovation Engines could ultimately automate the production of endless streams of interesting solutions in any domain: e.g. producing intelligent software, robot controllers, optimized physical components, and art.

Figure 1: Images produced by an Innovation Engine that look like example target classes. In each pair, an evolved image (left) is shown with a real image (right) from the training set used to train the deep neural network that evaluates evolving images.

synth.is

synth.is/mining-sounds/01FH5VVHW29WQTW4T3N22DF894

Evolving Favorite Live coding Published By

Notifications Björn Þór Jónsson

Generation 99999 Algorithm mapElites

1 2 3 ... 99993 99994 99995 99996 99997 99998 99999

- Speech
 - [100%] [100%] [100%] [99%] [99%] [99%] [99%] [99%] [99%] [99%]
 - [99%] [99%] [99%] [99%] [99%] [98%] [98%] [98%] [98%] [98%]
 - [97%] [96%] [94%] [87%] [82%] [77%] [54%] [22%] [8%] [6%]
 - [5%] [5%] [4%] [4%] [4%] [3%] [3%] [1%] [0%]
- Child speech, kid speaking
 - [26%] [26%] [26%] [26%] [25%] [25%] [21%] [11%] [6%]
 - [3%] [3%] [2%] [2%] [2%] [1%] [1%] [1%] [1%]
 - [0%] [0%] [0%] [0%]
- Conversation
 - [16%] [16%] [16%] [14%] [14%] [12%] [12%] [12%] [12%]
 - [12%] [12%] [9%] [9%] [9%] [9%] [5%] [5%] [5%]
 - [4%] [3%] [3%] [3%] [3%] [2%] [2%] [0%] [0%]
- Narration, monologue
 - [10%] [9%] [8%] [8%] [8%] [8%] [6%] [4%] [4%]
 - [4%] [3%] [3%] [3%] [2%] [2%] [2%] [2%] [2%]
 - [1%] [1%] [1%] [0%] [0%]
- Babbling
 - [58%] [41%] [39%] [37%] [36%] [15%] [15%] [15%] [15%] [9%]
 - [8%] [7%] [7%] [5%] [1%] [1%] [1%] [0%] [0%] [0%]
 - [0%] [0%] [0%] [0%] [0%] [0%] [0%] [0%] [0%]
- Speech synthesizer
 - [28%] [28%] [28%] [23%] [23%] [21%] [16%] [15%] [15%] [15%]
 - [15%] [15%] [12%] [11%] [11%] [6%] [6%] [5%] [4%]
 - [4%] [4%] [4%] [4%] [3%] [3%] [2%] [0%]
- Shout
 - [49%] [48%] [46%] [45%] [43%] [43%] [41%] [39%] [34%]
 - [33%] [33%] [28%] [28%] [28%] [28%] [28%] [27%] [27%]
 - [23%] [23%] [22%] [22%] [21%] [21%] [19%] [16%] [12%]
 - [11%] [9%] [8%] [8%] [6%] [5%] [5%] [4%] [4%]
 - [3%] [3%] [1%] [0%] [0%] [0%] [0%]
- Below
 - [55%] [53%] [47%] [41%] [30%] [21%] [21%] [19%] [15%] [11%]
 - [10%] [9%] [9%] [6%] [3%] [2%] [2%] [1%] [0%] [0%]
 - [0%] [0%]
- Whoop
 - [60%] [58%] [54%] [53%] [51%] [50%] [48%] [29%] [28%]
 - [28%] [26%] [25%] [25%] [19%] [18%] [18%] [17%] [11%]
 - [11%] [9%] [9%] [7%] [6%] [4%] [4%] [4%] [4%]
 - [1%] [0%] [0%] [0%]
- Yell
 - [53%] [51%] [50%] [46%] [43%] [41%] [34%] [34%] [34%]



escape from the web browser

lib

The screenshot shows the npm website interface for the package 'kromosynth'. At the top, there's a search bar with the text 'Search packages' and a 'Search' button. To the right are 'Sign Up' and 'Sign In' buttons. The main header for the package 'kromosynth' includes the version '1.0.2', the status 'Public', and the publication date 'Published 2 months ago'. Below this, there are navigation links for 'Readme', 'Code' (with a 'Beta' badge), '9 Dependencies', '1 Dependents', and '3 Versions'. The main content area features the package name 'Kromosynth' and a description: 'sonic design with evolutionary algorithms'. A detailed description follows: 'The engine behind synth.is and kromosynth-cli for audio waveform breeding with neuro-evolution of pattern producing networks and quality diversity search.' Below the description is a 'Keywords' section with the following terms: 'audio', 'synthesis', 'neuro-evolution', 'quality-diversity', 'map-elites', 'cppn', and 'neat'. On the right side, there's an 'Install' section with a code block containing the command '> npm i kromosynth'. Below that are links for the 'Repository' (github.com/synth-is/kromosy...) and 'Homepage' (github.com/synth-is/kromosy...). A 'Weekly Downloads' section shows a bar chart with a value of '4'. At the bottom right, there's a table with columns for 'Version' and 'License', showing '1.0.2' and 'AGPL-3.0-or-...' respectively.

npm Search packages Search Sign Up Sign In

kromosynth
1.0.2 • Public • Published 2 months ago

Readme Code Beta 9 Dependencies 1 Dependents 3 Versions

Kromosynth

- sonic design with evolutionary algorithms

The engine behind [synth.is](#) and [kromosynth-cli](#) for audio waveform breeding with neuro-evolution of pattern producing networks and quality diversity search.

Keywords

audio synthesis neuro-evolution quality-diversity map-elites
cppn neat

Install

```
> npm i kromosynth
```

Repository
github.com/synth-is/kromosy...

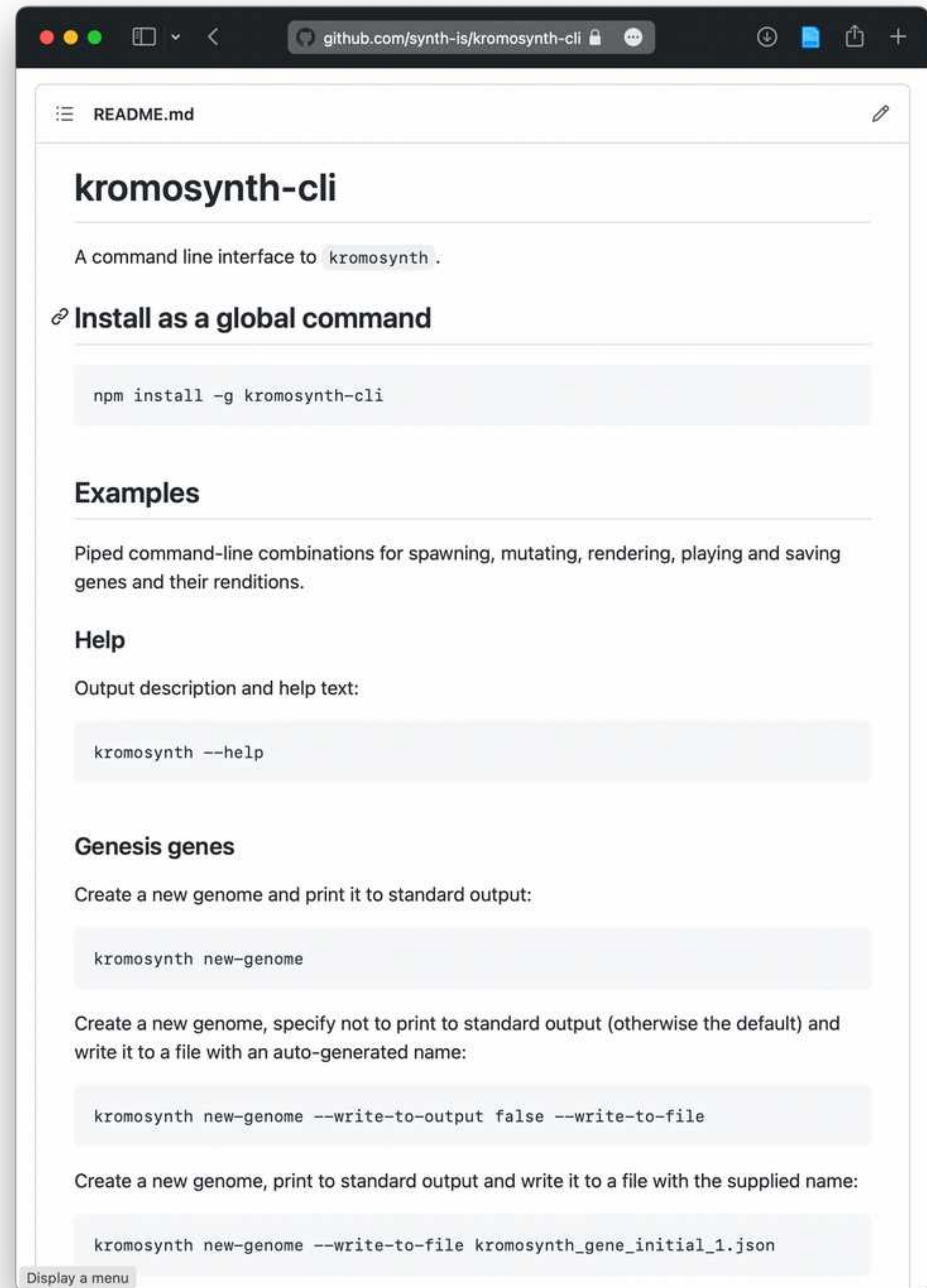
Homepage
github.com/synth-is/kromosy...

Weekly Downloads

4

Version	License
1.0.2	AGPL-3.0-or-...

CLI



The image shows a browser window displaying the README for the 'kromosynth-cli' project on GitHub. The browser's address bar shows the URL 'github.com/synth-is/kromosynth-cli'. The README content includes a title, a brief description, an installation instruction, and several usage examples.

README.md

kromosynth-cli

A command line interface to `kromosynth`.

Install as a global command

```
npm install -g kromosynth-cli
```

Examples

Piped command-line combinations for spawning, mutating, rendering, playing and saving genes and their renditions.

Help

Output description and help text:

```
kromosynth --help
```

Genesis genes

Create a new genome and print it to standard output:

```
kromosynth new-genome
```

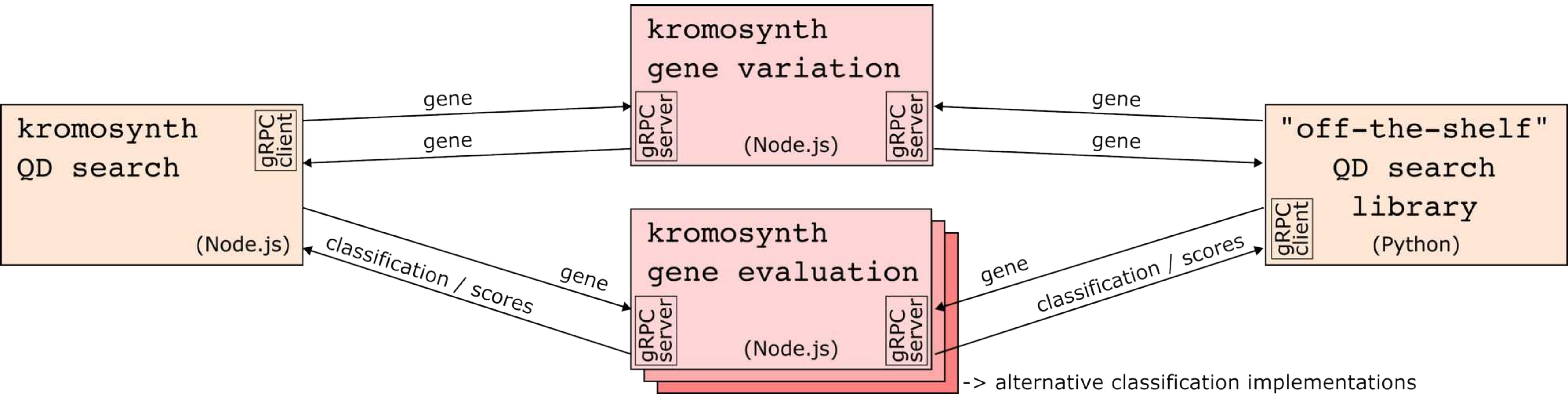
Create a new genome, specify not to print to standard output (otherwise the default) and write it to a file with an auto-generated name:

```
kromosynth new-genome --write-to-output false --write-to-file
```

Create a new genome, print to standard output and write it to a file with the supplied name:

```
kromosynth new-genome --write-to-file kromosynth_gene_initial_1.json
```

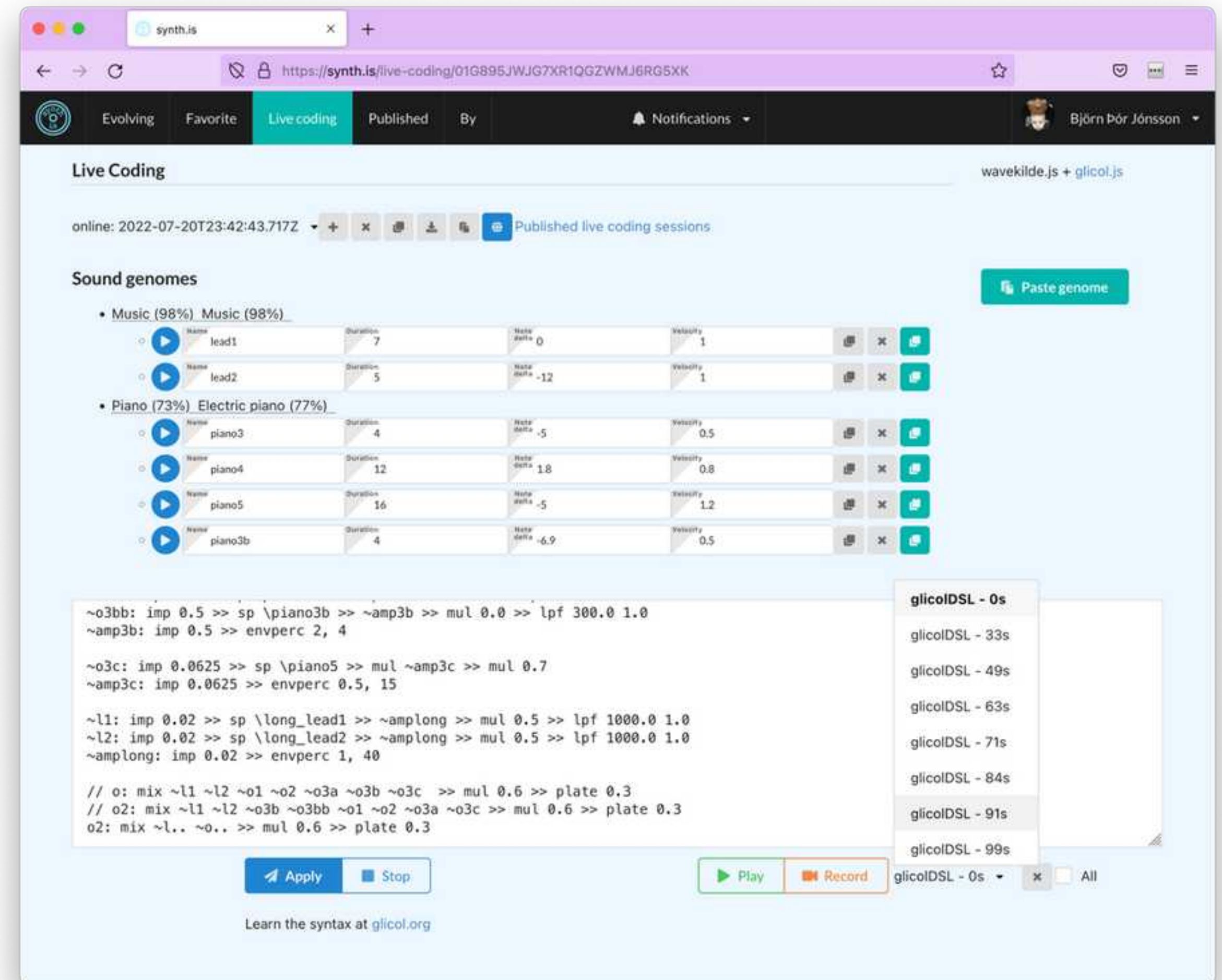
Display a menu



Artefacts and Interfaces

Instrument interaction

- SFZ sample based virtual instruments
 - <https://sfzformat.com/> - *sforzando*
 - Plug-in:
<https://www.plogue.com/products/sforzando.html>
- VST / AU(v3) plugins
 - -PoC with the JUCE framework, works on GarageBand@iPhone and Ableton Live @ desktop
- Live Coding
 - Glicol integration



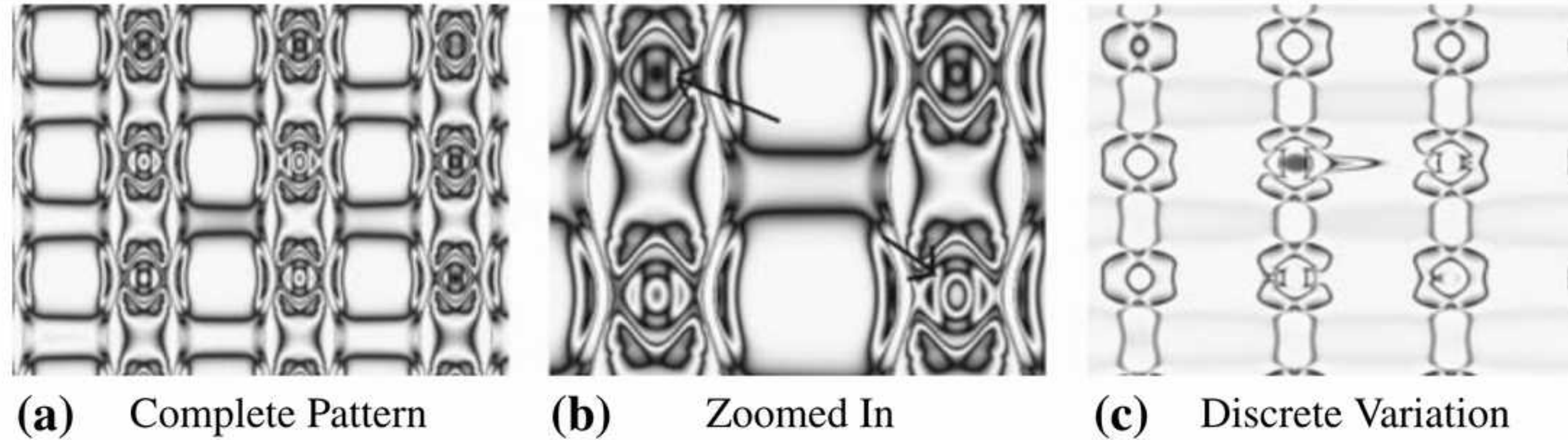


Fig. 12 *Repetition with variation.* The pattern in (b) is a closeup of the upper-left portion of (a). Arrows in (b) point to significant difference between motifs that are otherwise similar. By mixing inputs from separate coordinate frames, one repeating and one not, a CPPN can easily generate a large variety of patterns with this property. This phenomenon is similar to the interaction of two chemical gradients in a developing embryo, one periodic and one not. In (c), the object at the center of the pattern exhibits a strong dissimilarity from its repeated relatives, showing that the effect of interacting coordinate frames can be highly pronounced. In these patterns, the CPPN makes a strong analogy with nature and displays some of its essential capabilities

demotime