Notochord Homunculus: A Playground for Low-Latency Deep Generative MIDI

Victor Shepardson
Intelligent Instruments Lab
University of Iceland
`victorshepardson@hi.is`

Thor Magnusson
Intelligent Instruments Lab
University of Iceland
`thormagnusson@hi.is`

ABSTRACT

Notochord is a large-scale statistical generative model for MIDI prioritizing low latency inference. Homunculus is a graphical and MIDI interface to Notochord implementing a variety of real-time co-creative use cases. This demonstration will present the features of Homunculus to visitors, who can play with it via a MIDI key and pad controller.

1. INTRODUCTION

Large-scale statistical machine learning has become an important paradigm for generative music and algorithmic composition. Many models have been developed for MIDI sequences [1], since MIDI is widely compatible and MIDI files are widely available as training data, spanning the gap between notation and performance capture.

Notochord [2] is a MIDI model which emphasizes immediacy of interaction, processing MIDI Note On and Note Off events with extremely low latency. In this demonstration, we introduce Homunculus, a new real-time engine and UI for extemporaneous play with Notochord models.

2. PRIOR WORK

Notochord is a recurrent neural network model for MIDI supporting low-latency inference, which admits various types of low-level conditioning and constraints. Compared to lightweight MIDI systems based on e.g. hidden Markov models, Notochord is more general. It has many corners to explore, tending toward the uncanny aesthetic of large-scale generative AI. Compared to other machine learning models of similar scope, Notochord prioritizes very low latency (circa 10 milliseconds) event processing, meaning it can be used with perceptual instantaneity in a performance setting. Though extremely responsive, it is not especially coherent, lending it a character which is alternately charming, annoying, humorous, or absurd.

Notochord represents MIDI Note On and Note Off events broken into four modalities: note number (pitch), velocity, elapsed time since the previous event, and an operative MIDI program number. It handles polyphonic and multi-channel MIDI; when fit to a dataset with Program Change events indicating General MIDI (GM) instruments, such as the Lakh MIDI dataset [3], it can model music with multiple parts. Such a model can be used for co-improvisation, with a performer supplying events for certain GM instruments while those on other GM instruments are sampled from the model.

Notochord is available as an open source Python package [1], including model weights pre-trained on the Lakh dataset. The Python API can be queried over Open Sound Control for use in computer music environments like SuperCollider or Pure Data. However, building sensible musical constraints out of the low-level affordances of the API can be challenging, especially while preserving low latency. So, we developed a real-time engine with a MIDI and graphical interface which makes it easy to experience co-improvising with a Notochord model and can be configured for various uses cases without writing code.

3. NOTOCHORD HOMUNCULUS

Homunculus is a MIDI-processing application with a text-mode graphical user interface (TUI), implemented in Python as part of the `notochord` package (version 0.7.1 at time of writing). It provides a real-time engine for combining live MIDI input with sampling from a Notochord model at low latency suitable for tactile performance with a MIDI controller as well as live interaction with other computer music software.

When Homunculus is run from the command line, a text-mode user interface (figure 1) is drawn in the terminal. Diagnostic messages, recent MIDI events, and the forecasted next event are displayed. Clickable buttons control each voice mode, activation, and instrument, switch presets, and execute global actions. Detailed per-voice settings and MIDI/OSC control mappings can be specified on the command line or assigned to presets in a config file.

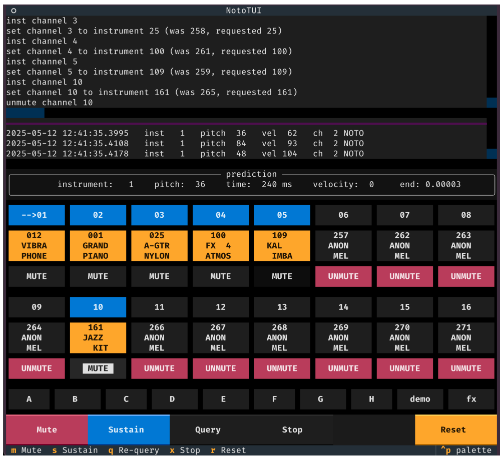
Each MIDI Note On or Note Off event processed by Homunculus is fed to a Notochord model, which is queried for a forecasted next event in the updated context.

3.1 Voice Modes

Homunculus is built around 16 voices, one for each MIDI channel. A voice can be in one of three modes: `input`, `auto`, or `follow`, and has an associated General MIDI instrument (i.e. MIDI program number).

An `input` voice is controlled externally, but affects the Notochord model. Homunculus accepts MIDI on `input`

[1] `https://pypi.org/project/notochord`



```
inst channel 3
set channel 3 to instrument 25 (was 258, requested 25)
inst channel 4
set channel 4 to instrument 100 (was 261, requested 100)
inst channel 5
set channel 5 to instrument 109 (was 259, requested 109)
inst channel 10
set channel 10 to instrument 161 (was 265, requested 161)
unmute channel 10

2025-05-12 12:41:35.3995   inst   1   pitch  36   vel  62   ch  2 NOTO
2025-05-12 12:41:35.4108   inst   1   pitch  84   vel  93   ch  2 NOTO
2025-05-12 12:41:35.4178   inst   1   pitch  48   vel 104   ch  2 NOTO

                                  prediction
        instrument:   1    pitch:  36    time:  240 ms    velocity:  0    end: 0.00003
```

Figure 1. The Homunculus TUI. Top to bottom: message console, event log, forecasted event, voice bay, preset rack, global controls.

channels and feeds it into the model. If Notochord forecasts an `input` voice to play, Homunculus immediately samples a new event constrained to occur at a later time. That is, it waits for expected input to happen, and prepares an action in case that it doesn't.

An `auto` voice plays autonomously. Homunculus ignores MIDI input on `auto` channels, but does execute forecasted events, feed them to the model, and transmit them on the MIDI output port.

A `follow` voice acts like a harmonizer for another voice. Homunculus ignores MIDI input for `follow` voices and excludes them from forecasts. Instead, it copies the timing of another voice. For every note played by the followed voice, a coextensive note is played by the follower, with just its MIDI note number sampled from the model.

The mode and instrument of each voice can be changed interactively; 'muting' a voice doesn't merely silence MIDI output, but intervenes in the generative process. A 'punch-in' feature can automatically change voices to `input` mode on MIDI input, and back to `auto` after a given duration.

3.2 Constraints & Prompting

Pitch range can be constrained for each voice, and optionally extracted from a SoundFont file so all notes are playable; Each `auto` voice can have hard constraints on polyphony and note duration. E.g., a piano might be constrained to play only notes shorter than one second while an organ plays single notes which are five seconds long.

Three global steering parameters adjust the overall rate of MIDI events, polyphony, and pitch register across `auto` voices. These have a 'soft' effect, truncating the model distributions by probability mass rather than value. Since Notochord processes one MIDI event at a time, latency can accumulate when event density is high; Homunculus automatically manipulates timing constraints when this happens, limiting worst-case latency.

Each preset can have an associated MIDI file which is used as a prompt for the Notochord model. The contents of that file are fed through the model, so that Notochord will act as if continuing past the end of the file in a similar style. Homunculus voice configurations can also reflect the active channels and programs in the MIDI file.

3.3 Applications

Homunculus is envisioned as a means to study the phenomenology of generative machine learning by bringing large models into the realm of tactility and improvisation.

It has been used in live musical performances: in *Notochord Arcs and Scrambled Signals* [4] it allowed a Notochord model to communicate with another generative MIDI system, with a performer changing voice modes and instruments live. In *Associative Memories* (2023), `follow` voices were used to expand a simple improvisation by a performer into chords played on a church organ. In *Haunting* (2025) punch-in and prompting were used with four voices to animate the organ manuals and foot pedals in a 'haunted fugue'. Homunculus has also been used by artists experimenting with semi-autonomous instruments including self-playing guitars [2,3] and woodblocks. [4]

4. DEMONSTRATION

In this demonstration, [5] Homunculus is running on a laptop with a small MIDI key, knob and pad controller. Visitors can interact with the model by playing a chromatic instrument on the keys and/or a drumkit on the pads, and setting constraints via knobs, and changing the ensemble of computer-controlled instruments with the mouse.

MIDI is sent to a SoundFont synthesizer with the full range of General MIDI sounds. Audio is distributed to headphones via a splitter, so that the presenter and multiple visitors can hear and take part in the same performance.

Acknowledgments

The Intelligent Instruments Lab is supported by the European Research Council as part of the INTENT project, under the EU's Horizon 2020 programme (Grant 101001848), and an NVIDIA hardware grant.

5. REFERENCES

[1] Y. Ma *et al.*, "Foundation Models for Music: A Survey," Aug. 2024, arXiv:2408.14340 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2408.14340

[2] V. Shepardson, J. Armitage, and T. Magnusson, "Notochord: a Flexible Probabilistic Model for Embodied MIDI Performance," in *Proceedings of the 1st Conference on AI Music Creativity*, 2022. [Online]. Available: https://zenodo.org/record/7088404

[3] C. Raffel, "Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching," Ph.D. dissertation, Columbia University, 2016.

[4] V. Shepardson and N. Privato, "Notochord Arcs & Scrambled Signals," in *Proceedings of the 2nd Conference on AI Music Creativity*, 2023, https://aimc2023.pubpub.org/pub/tii5jy7j.

[2] `https://federicovisi.com/the-sophtar`

[3] `https://craigscottslobotomy.com/against-the-machine`

[4] `https://iil.is/openlab/82`

[5] `https://youtu.be/u1ntK2Qg8vo`