PRAAT AUDIOTOOLS: ANALYSIS OBJECTS AS COMPOSITIONAL CONTROLLERS FOR INTERPRETABLE SOUND TRANSFORMATION

Shai Cohen *

Department of Music
Bar-Ilan University
Ramat Gan, Israel
shai.cohen@biu.ac.il

ABSTRACT

This demonstration presents **Praat AudioTools**, an open-source hybrid toolkit that repurposes Praat's phonetic-analysis environment for electroacoustic composition, sound design, and offline analysis–resynthesis workflows. Rather than treating analysis data as temporary measurements hidden inside an audio processor, Praat AudioTools exposes pitch contours, formant structures, temporal segmentations, spectral descriptors, phrase boundaries, stochastic trajectories, and host-application exchange files as editable compositional objects. These objects can be inspected, modified, chained, reused, and rendered into new sound transformations. The demonstration focuses on seven offline workflows: Neural Ambient Drone Designer, Praat for Max and Max for Live, Phase-Space Composer, Reich Generator, MCMC Musical Variation, Messagesquisse Opening, and Vector/Full-Chain composition workflows. None of the examples are presented as real-time effects. Instead, they show an "edit-in-the-middle" model in which sound is analyzed, intermediate representations are made visible, compositional decisions are applied to those representations, and the result is rendered as audio. The aim is to demonstrate a transparent alternative to both conventional black-box audio effects and end-to-end generative audio systems: a compositional environment where analysis objects become controllers, traces, scores, and reproducible technical artifacts.

1. INTRODUCTION

Analysis–resynthesis has been central to digital audio effects since the development of vocoders, LPC methods, spectral modeling, phase-vocoder processing, granular techniques, and feature-driven transformation [1, 2]. In many implementations, however, analysis is only an internal stage: the system extracts features, applies processing, and produces an output sound, while the intermediate representations disappear from the user's compositional workspace.

This demonstration proposes a different model. In **Praat AudioTools**, analysis objects are not merely diagnostic results; they are compositional materials. A pitch contour may become a melodic controller, a segmentation may become a formal grid, a spectral or timbral descriptor may become a navigation space, and a stochastic or machine-generated path may become a reusable control trajectory. The important shift is from "effect as signal transformation" to "effect as editable representation."

Praat AudioTools extends the established software Praat [3], widely used in speech science, into an offline environment for creative sound transformation. The toolkit contains a large collection of scripts and hybrid workflows for analysis, synthesis, adaptive processing, spectral transformation, spatialization, algorithmic composition, and host-application exchange. The demonstration presents a focused set of case studies that illustrate the same technical idea at different scales: analysis data, feature spaces, stochastic processes, and host bridges become visible controllers for sound transformation.

The contribution is therefore both practical and conceptual. Practically, the demonstration introduces a working open-source toolkit for offline sound design and composition [4, 5]. Conceptually, it presents a workflow in which interpretability is achieved not by simplifying algorithms, but by preserving the intermediate objects that connect analysis, decision-making, and synthesis [6].

2. SYSTEM OVERVIEW

The demonstrated part of Praat AudioTools is best understood not as a collection of isolated effects, but as a set of **offline composition engines**. Each engine begins with one source sound and converts it into an intermediate representation that can be inspected, edited, stored, or passed to another script. The representation may be a grain table, phrase list, event-boundary CSV, feature-space projection, attractor path, variation state, Max/Live exchange file, or multi-section rendering plan. The sound transformation is therefore not hidden inside a single processing block; it is distributed across analysis, representation, decision-making, and rendering.

The seven examples in this demonstration share a common architecture:

- **The source is analyzed as material.** The scripts first divide a recording into usable units: overlapping grains in Neural Ambient Drone Designer, silence-based events in Phase-Space Composer, repeated loop fragments in Reich Generator, phrases in MCMC Musical Variation, source layers in Messagesquisse Opening, or host-exported audio in the Max and Max for Live workflow.
- **The units are described or organized.** Each unit receives a compositional identity. Neural Ambient Drone Designer uses descriptors such as centroid, bandwidth, harmonicity, and pitch stability. Phase-Space Composer builds a feature space from centroid, flatness, entropy, flux, and RMS energy. MCMC Musical Variation represents phrases through pitch,

* This work was carried out at the Department of Music, Bar-Ilan University.

time, and dynamic parameters. Messagesquisse Opening organizes the source through the SACHER pitch set and Morse-derived timing.

- **A process chooses a path through the material.** The system applies a compositional logic to the representation: clustering and recombination in Neural Ambient Drone Designer; attractor-driven navigation in Phase-Space Composer; gradual phasing in Reich Generator; annealed Metropolis-style variation in MCMC Musical Variation [7, 8]; serial/harmonic inscription in Messagesquisse Opening [9]; and scripted chaining in the Vector/Full-Chain example.
- **The result is rendered, not streamed.** None of the demonstrated examples are real-time audio effects. They are offline or semi-offline rendering workflows. This allows the intermediate objects to remain available for inspection, comparison, revision, and reproduction.

This design makes the compositional decision visible. In a conventional effect, the user adjusts parameters and hears the output. In these scripts, the user can also examine the intermediate object that produced the output: the selected grains, the event plan, the attractor trajectory, the accepted variation states, the phasing relations, or the section chain. The central claim of the demonstration is that these intermediate structures are not implementation details—they are the actual compositional controllers.

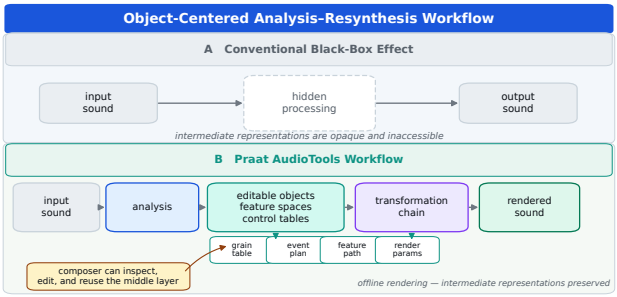

Figure 1: Object-centered analysis–resynthesis workflow contrasting conventional black-box processing (top row) with the Praat AudioTools model (bottom row) in which intermediate representations are exposed as editable compositional objects.

3. DEMONSTRATION STRUCTURE

The demonstration is designed as a table-based, offline presentation. A short loop of approximately five minutes introduces the general workflow and several prepared outputs. Longer interactions allow visitors to inspect intermediate objects, compare different parameter settings, and hear alternative renderings.

Because the examples are not real-time effects, the live component consists of inspection, explanation, parameter adjustment, and comparison between source material, intermediate representations, and rendered audio. When processing time permits, short excerpts can be re-rendered during the session; otherwise, pre-rendered examples are used to keep the demonstration reliable and repeatable.

3.1. Neural Ambient Drone Designer

The **Neural Ambient Drone Designer** shows how one source recording can become an atmospheric drone through analysis-driven selection and recombination. The script divides the source into overlapping grains, extracts four descriptors—spectral centroid, spectral bandwidth, harmonics-to-noise ratio, and pitch—and normalizes them as a feature space. A k-means clustering stage then groups grains into related textural regions. Presets such as Dark Ambient, Bright Shimmer, Dense Texture, Sparse Minimal, and Evolving Pad change the balance between grain size, density, clustering, stereo width, and octave-shimmer behavior.

The audience hears the original source, sees how regions have been analyzed or clustered, and then hears the rendered drone. The machine-assisted stage does not output an unexplained audio result; it creates an interpretable compositional structure—selected grains, cluster identities, shimmer probabilities, stereo distribution, and recombination decisions—that can be inspected or changed before rendering.

3.2. Praat for Max and Max for Live

Praat for Max and Max for Live demonstrates the toolkit's hybrid workflow between Praat, Max/MSP, and Ableton Live. Audio can be sent from Max or Ableton into Praat, processed offline by Praat AudioTools scripts, and returned to the host environment as rendered audio or control data.

This example clarifies that Praat AudioTools is not only a closed collection of scripts inside Praat. It can function as an offline analysis–resynthesis back end for existing creative environments. The workflow is deliberately not presented as a real-time plugin. Its value lies in opening a slower, inspectable processing stage inside otherwise real-time-oriented environments: Max and Live become front-end compositional spaces, while Praat AudioTools provides analysis, transformation, and rendering.

3.3. Phase-Space Composer

Phase-Space Composer generates composition by moving through an acoustic phase space rather than by following a conventional score. The Praat script first detects sounding events using intensity-based silence segmentation and writes a temporary WAV file and event-boundary CSV. A Python engine then extracts per-event features and maps a deterministic dynamical trajectory onto the event space.

The state space can use two to five dimensions: centroid and flatness; centroid, flatness, and flux; centroid, flatness, entropy, and flux; or centroid, flatness, entropy, flux, and RMS energy. The trajectory may be generated by a Hopf limit cycle, Lorenz attractor, Rössler attractor, or logistic map with time-delay embedding. Mapping from trajectory points to events uses weighted Euclidean distance, optional velocity/direction alignment, feedback coupling, tabu anti-repetition, and temperature-controlled stochastic selection. The engine writes a plan file and statistics file, which Praat uses to reconstruct the audio montage and draw a visualization of the event space and path.

3.4. Reich Generator

The **Reich Generator** demonstrates phasing, gradual process, and controlled drift as an audio-effect workflow [10]. A short fragment

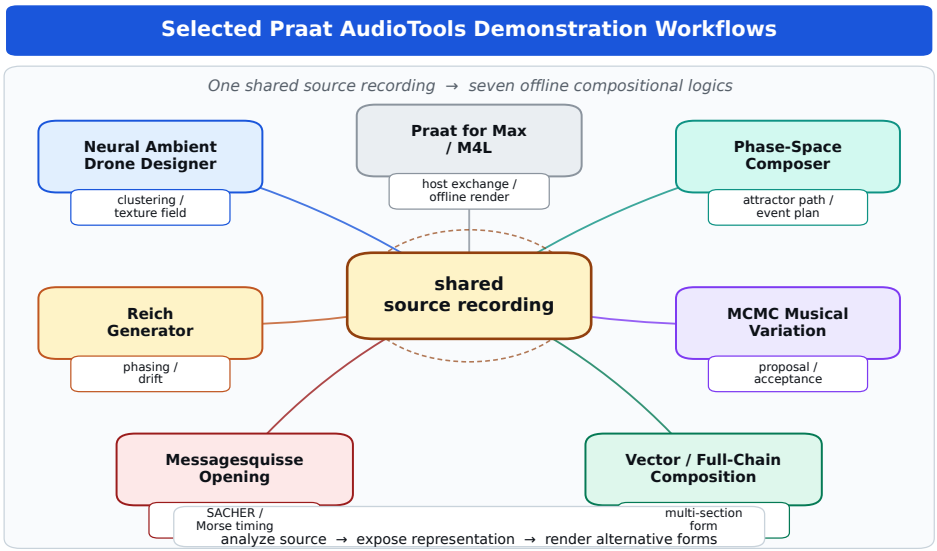

Figure 2: Overview of the seven demonstration workflows, each reinterpreting the same source material through a distinct compositional logic.

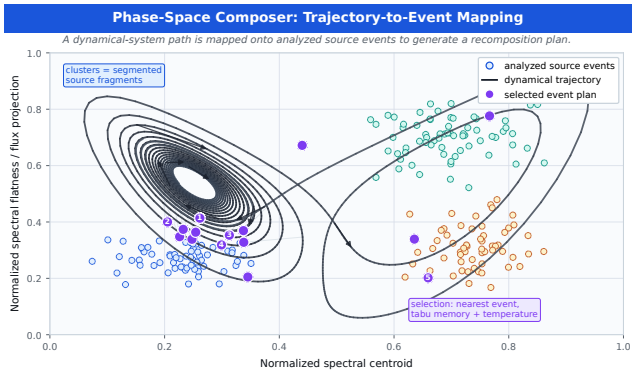

Figure 3: Phase-Space Composer: sound events mapped as points in feature space (centroid × flatness/flux), with a dynamical-attractor trajectory selecting events for recomposition.

is extracted from one source sound—long enough to carry a recognizable character, but compact enough to repeat. The script creates multiple voices from this fragment. One voice remains stable, while another is slightly altered and allowed to drift over time. Because its playback is minutely offset, the layers slowly separate, collide, and realign, producing changing rhythmic and harmonic patterns from a very small amount of material. An optional third voice can function as an anchor.

The intermediate parameters—loop boundaries, phase offsets, voice count, harmonic transformations, and drift curves—are visible and reproducible. A small change in offset or speed relation produces large-scale rhythmic and spectral consequences.

3.5. MCMC Musical Variation

MCMC Musical Variation shows how chance can generate musical variation while still being guided by taste. The script begins by dividing a source recording into phrases using silence detection. Each phrase becomes part of a musical state defined by pitch contour, time mapping, and dynamic shaping.

The system proposes small changes step by step: a pitch nudge, dynamic swell, micro-rubato shift, phrase transposition, temporal swap, global transposition, tempo warp, or larger dynamic arch. After each proposal, an energy model evaluates musical criteria such as scale conformity, smooth voice leading, range, rhythmic stability, dynamic coherence, phrase integrity, and contour variety. The method is best understood as annealed exploration with a Metropolis-Hastings-style acceptance rule [7, 8]. At selected thinning intervals, accepted states are rendered phrase by phrase through tape-speed pitch shifting, overlap-add time stretching, dynamic scaling, and crossfaded reassembly.

3.6. Messagesquisse Opening

Messagesquisse Opening demonstrates historically informed compositional transformation within the same analysis–resynthesis environment. A single source tone, such as a cello sound, is unfolded into a six-layer harmonic field inspired by Boulez's *Messagesquisse* [9]. The script uses the SACHER pitch set—E♭, A, C, B, E, and D—as a structural model for six emerging layers.

Each layer is pitch-shifted by sample-rate reinterpretation, so the original tone is recast into a harmonic field rather than simply copied. The timing of the layers is also composed: entries are shaped by the Morse-code rhythm of the name SACHER, so temporal spacing becomes part of the inscription. Later voices may enter from later moments in the source recording, creating the impression that the sound is advancing through its own memory. The layers are then spread across stereo space, blended with the dry original, and shaped into a dense but transparent opening texture.

3.7. Vector / Full-Chain Composition Workflows

The **Vector/Full-Chain** example shows that a composition in Praat AudioTools does not have to come from one script. A single source can become the raw material for several processes combined into a larger form. In one prepared example, the chain unfolds as a three-part structure: first an atmospheric neural drone, then a percussive groove, and finally a crystalline reverberant outro. Each section is generated from the same source, but each reimagines that source through a different compositional logic.

The chain then aligns the parts in time, crossfades them into one another, mixes them in stereo, trims excess silence, applies a final fade, and normalizes the result. The composition is therefore not just a sequence of effects but a designed form—intro, body, and release—built from multiple transformations of the same sound. Scripts can be run in sequence, combined in parallel, or prepared as performance chains.

4. TECHNICAL CHARACTERISTICS

The demonstration emphasizes the following technical characteristics:

- **Offline operation**: all examples are processed offline or semi-offline; none are presented as real-time effects.
- **Inspectable intermediate representations**: feature vectors, event lists, control tiers, paths, parameter tables, and host-exchange files are visible and reusable.
- **Reproducible rendering**: transformations can be repeated from stored parameters rather than manually reconstructed.
- **Hybrid architecture**: Praat scripts can interact with Python helpers, Max/MSP, Max for Live, and external audio environments.
- **Open-source distribution**: the toolkit is available through both a public Praat AudioTools webpage and a GitHub repository [4, 5].
- **Minimal demo requirements**: the table demonstration requires one laptop, headphones or small speakers, and one power outlet. Internet access is not required for the prepared examples.

5. RELEVANCE TO DAFX

Praat AudioTools is relevant to the DAFx community in several ways.

First, it expands the notion of a digital audio effect from a signal processor to an editable workflow. The effect is not only the rendered sound; it is the chain of analysis objects, control paths, transformations, and resynthesis decisions that produce the sound.

Second, it contributes to current discussions of interpretability in audio machine learning and algorithmic sound design [6]. The demonstration does not rely on end-to-end black-box generation. Even when clustering, stochastic search, latent navigation, or adaptive procedures are involved, the system foregrounds the intermediate representations that make the transformation explainable and modifiable.

Third, it shows the creative reuse of a mature scientific tool. Praat [3] was designed primarily for phonetic analysis, but its object model, scripting language, pitch and formant analysis, tiers, grids, and manipulation objects can be repurposed as a compositional audio environment.

Finally, the demonstration addresses a practical need for composers and sound designers who work between offline analysis tools and real-time creative environments. By including both internal Praat workflows and bridges to Max/MSP and Max for Live, the demo shows how analysis-driven transformation can enter existing production and performance ecosystems without needing to behave like a conventional live plugin.

6. CONCLUSIONS

This demonstration presents Praat AudioTools as a hybrid open-source toolkit and as a model for interpretable sound transformation. Across the selected examples—Neural Ambient Drone Designer, Praat for Max and Max for Live, Phase-Space Composer, Reich Generator, MCMC Musical Variation, Messagesquisse Opening, and Vector/Full-Chain workflows—the same principle recurs: analysis results, generated parameter structures, and scripted transformation chains become compositional controllers.

The demo argues for the value of offline editorial time in the design of digital audio effects. By exposing intermediate representations and allowing them to be inspected, edited, and reused, Praat AudioTools offers an alternative to both real-time patching environments and opaque generative systems. It invites DAFx participants to consider representation, workflow, and interpretability as central technical concerns in contemporary audio-effect design.

7. REFERENCES

[1] U. Zölzer, Ed., *DAFx: Digital Audio Effects*, 2nd ed. Chichester, UK: John Wiley & Sons, 2011.

[2] X. Serra and J. O. Smith, "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, 1990.

[3] P. Boersma and D. Weenink, "Praat: doing phonetics by computer [Computer program]," 2024, [Online]. Available: https://www.fon.hum.uva.nl/praat/.

[4] S. Cohen, "Praat AudioTools: An offline analysis–resynthesis toolkit for experimental composition," 2024–2025, [Online]. Available: https://mashav.com/sha/Praat%20AudioTools/.

[5] ——, "Praat AudioTools GitHub repository," 2024, [Online]. Available: https://github.com/ShaiCohen-ops/Praat-plugin_AudioTools.

[6] Z. C. Lipton, "The mythos of model interpretability," *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[7] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[8] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[9] P. Boulez, "Messagesquisse for solo cello and six cellos," 1976, score published by Universal Edition, Vienna.

[10] S. Reich, *Writings on Music, 1965–2000*. Oxford: Oxford University Press, 2002.