Organised Sound (2026), 1–9
doi:10.1017/S1355771826101204

CAMBRIDGE UNIVERSITY PRESS

Article

Praat Audiotools: An offline analysis-resynthesis toolkit for experimental composition

Shai Cohen
Department of Music, Bar-Ilan University, Ramat Gan, Israel

Abstract

Praat AudioTools is an open-source library of over 300 scripts that transforms the phonetic analysis software Praat (Boersma and Weenink) into an offline, object-centric laboratory for electroacoustic composition. While Praat is the standard tool for linguistic analysis, its potential for musical creation has remained largely untapped due to its interface design. We propose a workflow in which phonetic analysis objects – specifically PitchTiers (frequency curves), FormantGrids (resonance tracks) and TextGrids (temporal segmentation) – function as editable musical scores. Unlike real-time performance environments (e.g., Max/MSP, SuperCollider), which prioritise low-latency interaction, this toolkit emphasises 'compositional deep time', embedding analysis within an iterative edit–render–listen loop. Small modifications to analysis data produce structural consequences in timbre, gesture and form, enabling a research-creation practice rooted in the acousmatic tradition and spectromorphological thinking. By treating phonetic measurement as compositional material, AudioTools bridges phonetics and poetics. We contextualise this framework within the lineage of speech synthesis in electroacoustic music and demonstrate, through case studies, how it enables compositional strategies grounded in analysis as composition. The toolkit integrates neural network processing while maintaining interpretability, positioning it critically against black-box neural synthesis and arguing for transparent, parametric control in research-creation practice.

Keywords: acousmatic practice; analysis-driven synthesis; electroacoustic composition; formant synthesis; neural audio processing; Praat; reproducible workflows; research-creation; spectromorphology

(Received 6 October 2025; revised 11 February 2026; accepted 6 March 2026)

1. Introduction

The history of electroacoustic composition is defined by a persistent tension between the immediacy of performance and the 'deep time' of studio construction (Wishart 1996). Since the earliest days of musique concrète, composers have sought tools that allow for the precise manipulation of sonic material. Today, this landscape is dominated by two paradigms: the Digital Audio Workstation (DAW), which arranges audio on a timeline, and real-time coding environments like Max/MSP or SuperCollider, which focus on signal flow and generative processes.

However, a third paradigm exists, often overlooked in musical discourse but ubiquitous in linguistics: the Analysis-Resynthesis model. Here, sound is not merely treated as a waveform (as in a DAW) nor as a stream of numbers (as in digital signal processing (DSP) code), but as a set of discrete, semantic objects derived from the sound itself.

Praat AudioTools intervenes in this discourse by repurposing Praat, the standard tool for phonetic analysis in linguistics developed by Paul Boersma and David Weenink (2025), as a dedicated offline composition environment. It is crucial to clarify that AudioTools is not a standalone audio engine. Rather, it is a comprehensive suite of over 300 scripts written in the Praat scripting language that wraps the software's robust analysis algorithms in a composer-friendly interface. By treating analysis results as malleable scores rather than static descriptions, the system aligns with the historic Music-N paradigm (Mathews 1969), which distinguishes between the score (data) and the orchestra (synthesis). However, AudioTools introduces a crucial inversion: the 'score' is not derived from symbolic notation, but from the rigorous analysis of existing sound materials.

The toolkit has evolved significantly since its initial development, expanding from 250 to over 300 scripts organised into ten functional categories: AI and Adaptive Processing, Analysis and Feature Extraction, Distortion, Dynamics and Envelope, Filter and Color, Generative and Synthesis, Modulation, Pitch Processing, Reverberation and Spatial and Surround. This taxonomy reflects both the phonetic heritage of Praat and the specific affordances required for electroacoustic composition, from neural network-assisted timbre sequencing to physics-based room acoustics simulation, from cellular automata texture generation to higher-order ambisonic spatial processing.

1.1. The phonetic lineage in electroacoustic music

To understand the relevance of a linguistic tool in a musical context, we must contextualise the long, intertwined history of phonetics and electroacoustic music. The desire to decompose and

Corresponding author: Shai Cohen; Email: shai.cohen@biu.ac.il
Cite this article: Cohen S. (2026). Praat Audiotools: An offline analysis-resynthesis toolkit for experimental composition. *Organised Sound*, 1–9, https://doi.org/10.1017/S1355771826101204

© The Author(s), 2026. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

https://doi.org/10.1017/S1355771826101204 Published online by Cambridge University Press

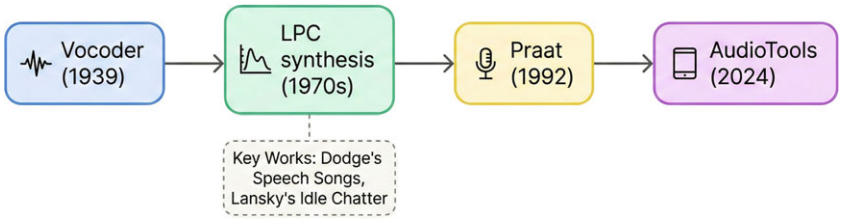


Figure 1. Historical lineage of phonetic tools in electroacoustic music.

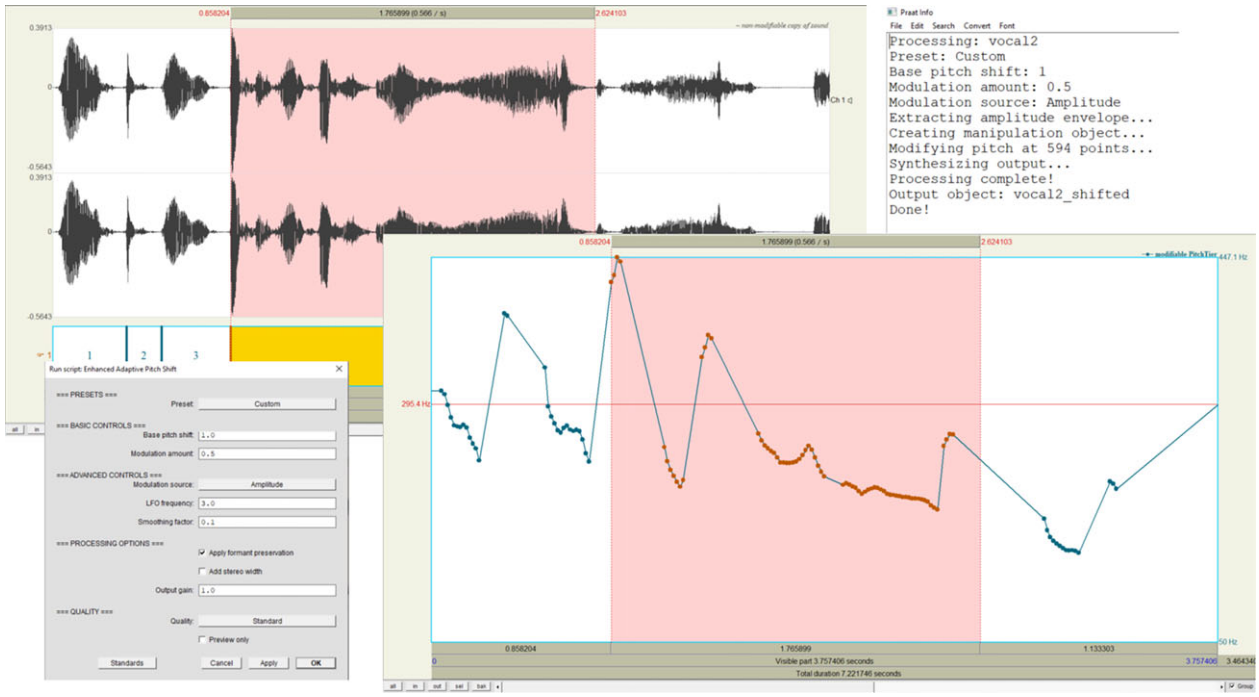


Figure 2. Praat GUI screenshot showing PitchTier editing.

resynthesise the voice has been a driving force in electronic music technology since the 1930s.

The lineage begins with Homer Dudley's Vocoder (Dudley, 1939), originally designed for telecommunications compression but rapidly adopted for its ability to decouple the spectral envelope (speech) from the excitation source (pitch). This 'source-filter' model, fundamental to phonetics, became the bedrock of subtractive synthesis.

In the computer music era, this relationship deepened. Composers like Charles Dodge (in Speech Songs, 1972) and Paul Lansky (in Idle Chatter, 1985) utilised linear predictive coding (LPC), a technique borrowed directly from speech telecommunications. They demonstrated that by manipulating the data frames of speech analysis, stretching the vowels, quantising the pitch, one could unlock a 'musicality' hidden within the spoken word.

Praat AudioTools situates itself as a modern successor to this lineage. However, whereas Dodge and Lansky worked with mainframe computers and non-real-time code, Praat offers a graphical user interface (GUI). It democratises the 'LPC studio' of the 1970s, allowing contemporary composers to visualise formants and pitch curves not as abstract numbers, but as editable vectors. It represents a return to the 'slow', rigorous engagement with the internal anatomy of sound that defined the early computer music era, but enhanced by modern visualisation and, significantly, augmented by contemporary machine learning techniques that maintain the transparency of parametric control (see Figure 1).

1.2. Why Praat? A linguistic tool in a musical context

Beyond its history, we must look at the specific affordances of Praat that are absent in standard musical software. Its longevity, continuous development by Boersma and Weenink since the 1990s, offers a stability that contrasts sharply with the rapid obsolescence of Virtual Studio Technology (VST) plugins and commercial DAWs.

But the primary contribution is the graphical interface for spectral editing. In a DAW, automation lanes are secondary overlays. In Praat, the 'Editor' window treats the analysis object (e.g., a PitchTier) as the primary document. The interface allows for vector-based drawing of frequency curves over a spectrogram backdrop with a precision designed for scientific measurement. When hijacked for composition, this allows users to draw microtonal glissandi or complex formant trajectories with an accuracy of fractions of a hertz/millisecond. This visual feedback loop, seeing the 'shape' of the sound's anatomy, invites a different kind of listening and interaction, one that is analytical yet deeply creative (see Figure 2).

https://doi.org/10.1017/S1355771826101204 Published online by Cambridge University Press

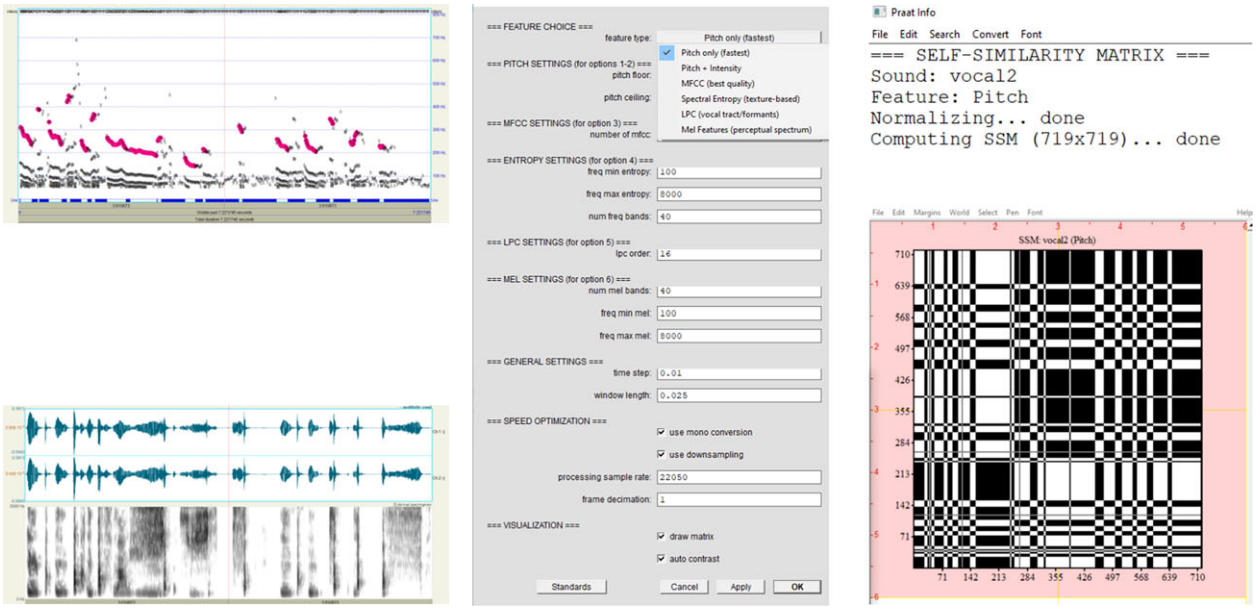


Figure 3. Core data objects visualisation.

1.3. The anatomy of analysis objects

To understand the compositional potential of this system, one must first grasp the core data structures of the Praat environment. In the context of this toolkit, we repurpose these linguistic tools as musical objects. Unlike generic data arrays in Python or Csound, these objects are semantically rich and natively editable.

PitchTier: In phonetics, this represents the fundamental frequency (F0) of speech over time. In composition, the PitchTier becomes a fully editable melodic contour. Structurally, it is a series of time-frequency coordinate pairs (t, f). Unlike MIDI, which is discrete (note-on/note-off) and quantised to semitones, a PitchTier is a continuous vector. It can be drawn, smoothed, exaggerated, or mathematically transformed independent of the audio's original duration or timbre. This allows the composer to extract the 'intonation' of a cello and apply it to a synthesiser, or to draw a 'melody' that consists entirely of microtonal glissandi. Because it is a vector, it can be mathematically manipulated without the artefacts associated with time-stretching audio.

FormantGrid: This is perhaps the most powerful object in the toolkit. It represents the resonant frequencies (formants) and bandwidths of a sound over time, effectively, the 'shape' of the vocal tract or instrument body. The FormantGrid acts as a 'timbral score'. By editing this grid, the composer can manually sculpt the spectral envelope, morphing the resonance of a violin into that of a vowel, or deleting specific resonances to 'hollow out' a sound. It consists of multiple tiers (Formant 1, Formant 2, etc.), each defining a centre frequency and a bandwidth. This separation allows for 'Impossible Acoustics', such as a vocal tract where F1 remains static (like a drone) while F2 and F3 articulate rapid, bird-like gestures.

TextGrid: A multi-tiered annotation file used in linguistics to mark phonemes, words, or syllables. In AudioTools, the TextGrid functions as a rhythmic sequencer and event trigger. It allows signal processing operations to be quantised to specific sonic events. For example, a script can be instructed to 'apply granular synthesis only to segments labeled "fricative", and reverb only to segments labeled "vowel"'. This moves automation away from the timeline (minutes: seconds) and onto the content (phonemes:events).

Table: While often invisible in other software, the Table object is central here. It is a spreadsheet-like structure used for algorithmic manipulation. Scripts can convert a PitchTier into a Table, apply complex statistical operations (e.g., 'randomize values within a Gaussian distribution'), and convert it back to a PitchTier. This serves as the bridge between manual editing and algorithmic generation.

These four object types form the core editable data structures of the system (see Figure 3).

2. Methodology: Object-centric composition

The core methodology of AudioTools relies on decoupling sound into independent parameter streams (objects), manipulating those objects and resynthesising. This differs fundamentally from standard signal processing.

2.1. The edit-in-the-middle workflow

In a typical Music Information Retrieval (MIR) workflow using Python libraries such as librosa, the process follows a linear sequence: analysis leads to coding transformations, which then render the output. The intermediate data are often hidden in variables. If the analysis is imperfect, for example, a pitch tracker jumps an octave, the user must tweak the code parameters and re-run the entire pipeline.

Praat AudioTools offers a transparent alternative. First, the user runs a script to extract a PitchTier or FormantGrid from the source audio. Second, the object appears in the object list and can be opened for visualisation and editing. The user sees the error (such as the octave jump) and corrects it directly with the mouse, or decides to creatively alter the curve. Finally, the modified object is fed back into a synthesis script for resynthesis. This workflow embeds human judgement and creative intervention at the analysis stage itself, rather than treating analysis as a fixed preprocessing step.

This 'edit-in-the-middle' approach fundamentally distinguishes AudioTools from automated pipelines. The composer maintains agency over the analytical representation, ensuring that extracted parameters align with perceptual intuition and compositional intent

https://doi.org/10.1017/S1355771826101204 Published online by Cambridge University Press

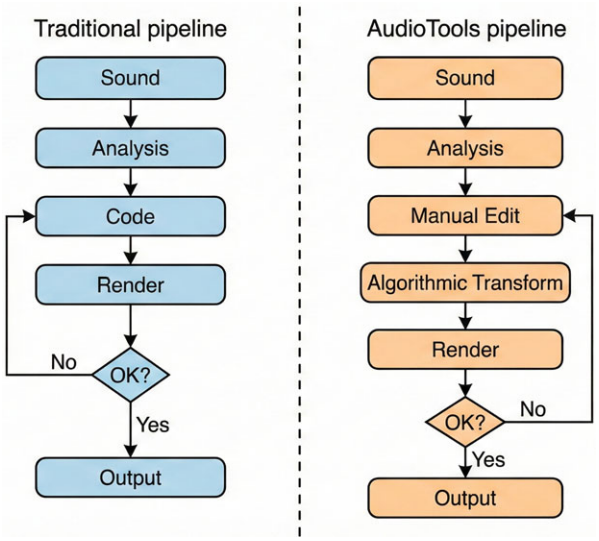


Figure 4. Workflow comparison diagram.

before transformation. Errors become opportunities; artefacts can be preserved or enhanced rather than algorithmically corrected.

Consider a concrete example: extracting the pitch contour from a violinist's vibrato. An automated tracker might produce a jagged, octave-jumping curve due to the rapid frequency modulation. In a Python workflow, the composer would write code to smooth or filter these errors, a time-consuming trial-and-error process. In AudioTools, the PitchTier appears visually superimposed on the spectrogram. The composer sees exactly where the tracker failed, manually removes spurious points with mouse clicks, and smooths the curve by adjusting control points, all while listening to playback. The corrected tier can then be scaled (widening the vibrato), inverted (creating an inverse melodic contour), or mapped to a synthesis parameter (driving granular density). This immediate visual-auditory feedback loop, where the object remains editable throughout the creative process, exemplifies the edit-in-the-middle paradigm (see Figure 4).

2.2. Comparative platform analysis

To situate AudioTools within the contemporary landscape of computer music tools, we compare it across several dimensions with established platforms. Table 1 demonstrates how AudioTools occupies a distinctive niche: offline operation with native visual editing and exceptional reproducibility.

This comparison reveals AudioTools' distinctive position: it combines the visual editability of GUI tools with the reproducibility of code-based systems, while maintaining the offline 'deep time' essential for studio electroacoustic practice. The 30-year longevity of Praat addresses archival concerns that plague contemporary music software.

3. Toolkit architecture and core processes

AudioTools comprises over 300 scripts organised into a ten-category taxonomy that reflects both the phonetic heritage of Praat and the specific requirements of electroacoustic composition. Rather than presenting an exhaustive catalogue, we highlight representative processes that demonstrate the object-centric paradigm and spectromorphological engagement. This organisation emerged organically from compositional practice, scripts clustered by the type of sonic transformation they enable rather than by underlying algorithm. Complete technical documentation is available in supplementary materials.

3.1. Taxonomic organisation

The toolkit's organisation prioritises compositional intention over technical implementation. The ten categories, AI and Adaptive Processing, Analysis and Feature Extraction, Distortion, Dynamics and Envelope, Filter and Colour, Generative and Synthesis, Modulation, Pitch Processing, Reverberation and Spatial and Surround, each address distinct aspects of spectromorphological practice. For instance, 'Filter & Color' groups processes that sculpt spectral profiles (formant manipulation, frequency shifting, resonant filtering), while 'Spatial & Surround' contains multi-channel spatialisation tools (Distance-based amplitude panning (DBAP), higher-order ambisonics, prosody-driven diffusion). This compositional taxonomy allows users to navigate by sonic goal rather than algorithmic category.

3.2. AI and adaptive processing

This category integrates neural network and statistical learning approaches while maintaining transparency. The Hidden Markov Model (HMM) Timbre Sequencing analyses sound corpora to learn timbral states, outputting state assignments as editable TextGrids. The PCA Tone Shaper extracts dominant spectral components, allowing manipulation of interpretable dimensions derived from the sound itself. The Neural Granular Texture Morpher generates parameter trajectories as editable Tables rather than direct audio output. The composer views and modifies learned interpolation paths. This exemplifies interpretable neural processing: machine learning assists rather than replaces human agency.

3.3. Analysis and feature extraction

This category includes tools for detailed timbral analysis. The Spectral Flatness and Roughness script computes metrics that guide sample selection or drive synthesis parameters. The Self-Similarity Matrix Calculator reveals structural repetition, identifying loop points or recurring motifs. The Tempo Curve Estimator tracks timing fluctuations, enabling composers to extract tempo gestures from performances. The Speech to MusicXML Rhythm Converter translates syllabic timing into notated rhythm. These tools transform analysis into compositional material.

3.4. Spectral and timbral processing

The Filter and Colour category operates on spectral envelopes and resonance structures. The Creative Formant Manipulations suite allows swapping formant order or applying mathematical operations to trajectories, creating impossible vocal geometries. The MFCC-based Sound Controller uses timbral fingerprints to modulate synthesis parameters. The Whisper Morph removes voicing while preserving formant structure. These transformations operate on perceptually meaningful dimensions rather than abstract spectral bins.

The MFCC-based Sound Controller extracts Mel-Frequency Cepstral Coefficients (timbral fingerprints) from a source sound and uses them to modulate parameters in real-time synthesis. For instance, the brightness of a voice (reflected in higher MFCCs) might control filter cut-off in granular synthesis. The Whisper Morph gradually removes voicing from speech by reducing

https://doi.org/10.1017/S1355771826101204 Published online by Cambridge University Press

Table 1. Comparison of analysis-resynthesis platforms

Feature	Praat AudioTools	Python/librosa	Max/MSP	SuperCollider	CSound
Paradigm	Object-centric GUI	Code-based	Visual patcher	Code-based	Code-based
Analysis Editing	Native visual editor	External tools required	Via patchers	Via code	Text file editing
Reproducibility	Scripts + data objects	Code + data files	Patches (version-dependent)	Code files	Score files
Real-time	Offline only	Offline	Real-time (primary)	Real-time (primary)	Offline
Learning Curve	Moderate (GUI)	Steep (programming)	Moderate (visual)	Steep (syntax)	Steep (syntax)
Visualisation	Built-in spectrograms	matplotlib required	Limited	External tools	External tools
Longevity	30+ years (1992–present)	~10 years (Python 2→3)	25+ years	20+ years	40+ years

harmonic content while preserving formant structure, creating a continuum from voiced speech to breathy whisper. These transformations operate on perceptually meaningful dimensions, formant structure, voicing and breathiness, rather than abstract spectral bins.

3.5. Spatial and temporal processing

Spatial processing leverages prosodic contours for multichannel diffusion. The 8-Channel Speech-Driven Spatialisation maps pitch and intensity to speaker positions. DBAP allows trajectory definition as editable coordinate Tables. The higher-order ambisonic tools support detailed spatial imaging, with decoders for common arrays. The Partial Panner distributes spectral partials across speakers by frequency, a spectromorphological strategy impossible with amplitude panning.

The higher-order ambisonic encoder and decoder support up to third-order encoding, providing detailed spatial imaging for fixed-medium work. The toolkit includes decoders for common speaker arrays (stereo, quad, octagon, cube) and tools for importing custom decoder matrices. The Partial Panner distributes individual spectral partials across the speaker array based on their frequency, creating a spatialised spectrum where low partials emerge from one location and high partials from another, a spectromorphological strategy impossible with amplitude-based panning.

3.6. Generative and synthesis systems

This category enables algorithmic composition through cellular automata, Markov chains and chaos functions. The Cellular Automata generator maps cell states to grain parameters. The Lorenz Deep Analog produces continuous control signals from chaotic attractors. Image sonification tools convert visual data via multiple strategies: Spectral Image Sonification treats images as spectrograms, Percussive Image Sonification uses edge detection for rhythmic grains and Photo Brightness-Controlled Pitch creates melodic contours from brightness values. These embody transparent cross-domain translation.

4. Neural and AI integration: Interpretability in machine-assisted composition

The integration of neural networks and statistical learning within AudioTools represents a deliberate intervention in contemporary discourse around AI music tools. Rather than adopting the black-box paradigm of systems like RAVE or DDSP, where latent spaces remain opaque and transformations inscrutable, AudioTools implements what we term interpretable neural processing: machine learning assists compositional decision-making while preserving transparent, parametric control. This approach maintains the toolkit's core commitment to epistemic agency: the composer understands not just what changed, but how and why.

4.1. Interpretable neural processing

The distinction between black-box and interpretable neural processing is methodological, not merely technical. In black-box systems, a neural network receives audio input and produces audio output, with intermediate representations hidden in high-dimensional latent spaces. The composer inputs sound A and receives sound B, but cannot inspect or edit the transformation pathway. This opacity, while enabling remarkable results, fundamentally limits compositional agency: one cannot systematically explore the transformation space, cannot reverse-engineer why a particular output emerged and cannot document the process for scholarly discourse.

AudioTools addresses this through architectural choices that prioritise transparency. Neural networks are used for analysis, feature extraction and parameter generation, not end-to-end audio synthesis. The Parametric Autoencoder Resynthesis script exemplifies this approach. It trains an autoencoder on MFCC features extracted from a sound corpus, learning a compressed representation. However, rather than using the decoder to directly generate audio, the system outputs the encoded parameters as a Table object. The composer can then inspect these parameters, understand their range and distribution, manually edit them and use them to drive synthesis algorithms. The network compresses timbral complexity into manageable dimensions, but the composer retains control over the resynthesis process.

This paradigm extends to real-time control. The Neural Delay Control script uses a feed-forward network to analyse incoming audio features (spectral centroid, flux, roughness) and adjust delay parameters (feedback, delay time, mix) accordingly. Crucially, the network's output is not an audio signal but a parameter trajectory, a DurationTier or IntensityTier that the composer can view, edit and archive. The system learns adaptive mappings (e.g., 'increase feedback when spectral flux is high') but exposes these mappings as editable curves. This allows post hoc refinement: if the network over-emphasises feedback on certain transients, the composer can manually attenuate those peaks while preserving the learned behaviour elsewhere.

4.2. Statistical learning approaches

Beyond neural networks, AudioTools incorporates statistical learning methods from machine learning and computational linguistics, adapted for musical contexts. The HMM Timbre

https://doi.org/10.1017/S1355771826101204 Published online by Cambridge University Press

Sequencing provides a case study in transparent algorithmic composition.

The HMM learns timbral states from a sound corpus, for example, analysing a spoken phrase to identify distinct phonetic regions (vowels, fricatives, stops). Each audio segment is assigned to a hidden state, and the system learns transition probabilities between states. In traditional HMM usage, these states remain abstract. AudioTools makes them concrete: the state assignments are output as a TextGrid with labelled intervals, and the transition matrix is saved as a Table. The composer can inspect which segments the model grouped together, listen to them and modify the groupings. The transition probabilities are editable numbers; increasing p(state2|state1) makes transitions from state 1 to state 2 more likely in generated sequences.

When generating new sequences, the HMM probabilistically transitions between states, selecting corresponding audio segments. But the composer can intervene at any step: force a particular state at a particular time, bias transitions towards or away from certain states, or manually reorder the generated sequence. The HMM provides intelligent suggestions based on learned patterns, but the composer retains veto power. This exemplifies research-creation: the process of building, training and interrogating the model becomes part of the compositional methodology, with each step documented and replicable.

The PCA-based tools (PCA_Timbre_Selector, PCA_Tone_Shaper) similarly expose the learned structure. Principal Component Analysis reduces a high-dimensional feature space (e.g., 13 MFCCs across 100 time frames) to its principal axes. AudioTools displays these axes graphically: PC1 might represent overall brightness, PC2 might capture attack sharpness and PC3 might encode roughness. The composer can then navigate this learned timbral space by adjusting PC weights, effectively sculpting timbre along interpretable dimensions derived from the sound itself. Moreover, the PCA projection is reversible: one can move to a desired point in PC space and resynthesise audio. This differs fundamentally from Variational Autoencoder (VAE) latent spaces, where interpolation often produces incoherent sounds due to the non-linear decoder; PCA operates in feature space, where interpolation is perceptually meaningful.

4.3. Neural-assisted composition: A workflow example

To demonstrate interpretable neural processing in practice, consider a complete workflow using the Neural Granular Texture Morpher. The goal is to create a smooth evolution from a rough, percussive texture (broken glass) to a smooth, continuous texture (ocean waves).

First, the composer creates granular syntheses of both source sounds: glass with short grains (20 ms) at high density (100 grains/sec), ocean with long grains (200 ms) at low density (10 grains/sec). These establish the endpoints. Next, the Neural Granular Texture Morpher script is invoked. It analyses both textures, extracting features (grain statistics, spectral flux, roughness measures) and trains a small feed-forward network to learn a mapping between feature spaces. The network discovers that morphing between the textures requires coordinated changes: grain size must increase while density decreases and spectral emphasis must shift from high frequencies to mid-low.

The crucial step: rather than generating the morph audio directly, the system outputs parameter trajectories as Tables, one for grain size (interpolating 20 ms to 200 ms), one for grain density (100 to 10 grains/sec), and one for spectral centre frequency (8 kHz to 500 Hz). The composer views these as curves in Praat. Perhaps the grain size transition is too abrupt; the composer manually adjusts the curve to be more gradual. Perhaps the density drops too quickly; it can be reshaped. The network provided an intelligent starting point, but the composer sculpts the exact morphological trajectory.

Finally, these edited parameter curves drive a granular synthesis script, generating the audio. Because every step is documented (the source sounds, the feature extraction parameters, the network architecture, the edited Tables, the synthesis parameters), the entire composition is reproducible. Another composer could load the same sources, retrain the network and modify the parameter curves differently, exploring alternative morph paths. This transparency, the ability to inspect, understand, modify and document each stage, exemplifies how AudioTools integrates contemporary machine learning while maintaining the interpretability essential for research-creation.

This approach contrasts sharply with end-to-end neural synthesis. In a system like RAVE, one would encode both textures to latent vectors, interpolate in latent space and decode back to audio. The result might sound compelling, but the interpolation pathway remains opaque: one cannot inspect what changes are happening, cannot selectively modify certain aspects while preserving others, and cannot easily communicate the method beyond 'I used a pre-trained RAVE model'. AudioTools sacrifices some of the 'magic' of neural synthesis, the surprising, emergent results, in favour of epistemic clarity. For composers and researchers who value understanding their methods as deeply as their results, this trade-off is worthwhile.

5. Case studies

We present three extended case studies demonstrating how the object-centric workflow shapes compositional outcomes across different analytical approaches: spectral analysis-driven granular synthesis, cross-domain translation from visual data to percussive rhythm and formant analysis as harmonic material. Each includes detailed parameter documentation and demonstrates the iterative edit-render-listen cycle central to AudioTools methodology.

5.1. Case study 1: Adaptive grain cloud synthesis via spectral content analysis

Concept: To create granular textures where grain duration adapts dynamically to the spectral content of the source material, short grains for high-frequency transients, long grains for low-frequency sustained tones, thereby producing spectral-morphological coherence unavailable through fixed-parameter granulation.

Workflow: The composition began with a 12-second field recording of wind through metal pipes, containing both breathy high-frequency turbulence and resonant low-frequency tones. Using the Adaptive Grain Cloud Synthesis script, the system first converted the audio to mono and calculated baseline grain parameters (50 ms duration, 75% overlap, density factor 2.0). The critical innovation: before extracting each grain, the script analysed a temporary extraction to compute its spectral centroid. If the centroid exceeded 2000 Hz (indicating high-frequency content), the grain duration was reduced to 60% of baseline (30 ms) and pitch scatter was doubled. Conversely, for centroids below 2000 Hz (low-frequency content), grain duration increased to 140% of baseline (70 ms) and pitch scatter was halved. This adaptive strategy ensured that rapid, bright transients produced pointillistic textures while sustained, dark tones produced smooth, overlapping clouds.

https://doi.org/10.1017/S1355771826101204 Published online by Cambridge University Press

The editing phase involved analysing the intermediate output and adjusting the spectral centroid threshold. Initial tests at 1500 Hz caused too many grains to receive long durations, creating muddy overlap. Raising the threshold to 2000 Hz produced clearer differentiation. The composer also enabled pitch shifting (using resampling for accurate transposition) and random grain reversal, adding timbral variety. The script's dynamics preservation mode maintained the original recording's amplitude envelope throughout the granulation process, preventing the common problem of uniform grain loudness that destroys gestural shape.

Result: The resulting texture exhibits clear spectro-morphological evolution: high-frequency consonants produced rapid grain clusters (30 ms), while low-frequency vowels produced smooth drones (70 ms), directly mirroring the source's spectral morphology. This demonstrates parametric coupling where spectral analysis drives synthesis parameters transparently – the composer controls the mapping function (2000 Hz threshold, 0.6/1.4 scaling factors) rather than relying on opaque neural network weights.

5.2. Case study 2: Image-driven percussive

Concept: To translate visual architectural geometry into rhythmic, percussive audio textures where image brightness controls temporal density, colour balance controls stereo positioning and visual structure generates timbral evolution, demonstrating cross-domain mapping where spatial visual information becomes temporal sonic gestures.

Workflow: A digital photograph of a modernist building facade (800 × 600 pixels, colour) was loaded into Praat as a Photo object. The Percussive Image Sonification script extracted RGB channels as three Matrix objects, then computed per-column statistics: brightness (average of normalised R, G, B values) and pan position (derived from red-blue balance, where red-dominant columns pan left and blue-dominant pan right). Green channel data contributed to brightness but not panning, maintaining timbral neutrality.

The sonification algorithm scanned the image column-by-column (left to right), generating percussive clicks whose parameters mapped directly to visual properties. Brightness controlled three simultaneous dimensions: inter-click interval (bright columns produced rapid clicks, dark columns produced sparse timing), click volume (brighter = louder) and spectral content (brightness mapped to base frequency 800–2000 Hz). Each click consisted of three frequency components (fundamental, 2.5x overtone, 4x overtone) with 60:30:10 amplitude ratios, creating a spectrally rich impact. Click duration remained constant at 50 ms, but the raised-cosine envelope ensured smooth attacks and decays. Stereo panning followed the computed pan values, creating spatial motion as the 'scan' progressed through the image.

Result: The percussive texture directly reflects the building's visual geometry.

5.3. Case study 3: Formant-to-chord mapping for harmonic sonification

Concept: To extract formant frequencies from vocal recordings and map them to musical chords with microtonal precision, treating the spectral resonance structure of speech as harmonic material that can be notated, performed by acoustic instruments and subjected to conventional music-theoretical analysis, bridging phonetic analysis and compositional harmony.

Workflow: A recording of a female voice reciting a four-line poem was selected as source material. The voice exhibited clear formant structure with distinct vowel articulations and consonant transitions. Using the Formant to MusicXML Chord Converter, the system first created a Formant object via Burg's LPC analysis (time step 10 ms, 5 formants tracked up to 5500 Hz, 25 ms analysis window). The script then segmented the recording into 8 equal portions (2 seconds each), analysed formants at each segment's temporal midpoint and converted the first four formant frequencies (F1-F4) to MIDI note numbers.

The critical innovation: rather than rounding to the nearest semitone (which would destroy the phonetic specificity of the formants), the system calculated microtonal alterations with 1/8-tone precision. For each formant frequency, the script computed a floating-point MIDI number, separated it into an integer (base note) and a fractional part (microtonal deviation) and then quantised the fraction to the nearest eighth tone. This allowed the chord notation to reflect the actual formant frequencies to within ±6.25 cents, preserving phonetic detail while remaining writable in conventional notation with microtonal accidentals.

Result: The resulting MusicXML file, imported into MuseScore, displayed eight measures each containing a three- or four-note chord corresponding to one analysis segment. Visual inspection revealed clear correspondences between vowel phonetics and chord structure. For instance, the vowel /a/ (first formant ∼800 Hz, second ∼1200 Hz) produced chords resembling major thirds (roughly 5:4 frequency ratio), while /i/ (F1 ∼300 Hz, F2 ∼2300 Hz) produced wider intervals approaching sevenths or ninths. Consonantal transitions (reduced formant clarity) resulted in chords with fewer pitches or increased microtonal deviations.

6. Discussion: Critical positioning

The development of Praat AudioTools is not merely a software engineering project but a critical intervention in electroacoustic music. It raises key questions about agency, opacity and the nature of compositional instruments in contemporary practice.

6.1. Against the black box: Interpretability versus neural synthesis

We currently stand at a juncture in music technology. As discussed in Section 4.1, neural synthesis tools often operate as 'black boxes' with opaque latent spaces.

Praat AudioTools champions interpretability. When the system alters a sound, it does so via explicit, editable parameters: Formant Frequency, Bandwidth, Duration and Amplitude. Every parameter is exposed and manipulable. This transparency allows for a different kind of virtuosity: not the virtuosity of the prompt engineer, but the virtuosity of the analyst-composer who understands the inner anatomy of their sounds. We argue that for research-creation, where method is as important as result, this transparency is non-negotiable.

This is not to dismiss neural synthesis; it has legitimate creative applications, but to argue for methodological pluralism. When the goal is understanding how and why a transformation works, parametric transparency remains essential. AudioTools offers an alternative to algorithmic opacity: a space where compositional decisions remain traceable, inspectable and communicable.

6.2. Authorship and the found score

By deriving the score from analysis, AudioTools challenges traditional notions of authorship. The composer does not invent the melody from scratch; they 'find' it within the recording of a bird, a

https://doi.org/10.1017/S1355771826101204 Published online by Cambridge University Press

machine or a voice. The creative act shifts from generation to curation and transformation. This resonates with the 'surrealist' approach to musique concrète: the object is found, then manipulated to reveal its hidden qualities (Schaeffer 1966). The PitchTier is a 'found object' that the composer polishes and recontextualises.

This has implications for compositional discourse. When presenting work created with AudioTools, composers must acknowledge the analytical origins of their materials, the source recordings, the extraction parameters and the nature of editorial interventions. This documentation becomes part of the work's scholarly contribution, aligning creative practice with research-creation methodologies (Magnusson 2019).

6.3. Archival longevity and reproducibility

The choice of Praat addresses a crisis in digital art preservation. Commercial plugins (VSTs) frequently become obsolete as operating systems update. Proprietary DAW formats fragment into incompatible versions. Praat, maintained by the University of Amsterdam, has prioritised backward compatibility for over 30 years. Scripts written in 1995 still execute today. By building AudioTools on this bedrock, we ensure that compositions created with this toolkit remain not just documented, but executable, the highest standard of reproducibility.

This stability is essential for establishing electroacoustic music as a serious, cumulative research discipline. When compositional methods can be exactly replicated decades later, analysis objects, scripts and renders all function, the field moves beyond ephemeral experimentation toward systematic knowledge building. AudioTools proposes a model where the archive of process (scripts, objects, presets, logs) stands alongside finished works as a scholarly contribution.

6.4. Pedagogical implications: Developing parametric listening

In pedagogical contexts, AudioTools serves a dual function: as a creative toolkit and as an analytical instrument for developing what we term 'parametric listening', the ability to perceive acoustic structure as editable dimensions rather than holistic textures. This transformation of perception, cultivating awareness through measurement and manipulation, represents the toolkit's deepest contribution to electroacoustic pedagogy.

Traditional ear training focuses on pitch, rhythm and harmony, the parameters of Western tonal music. Spectromorphological listening, advocated by Smalley and others, attends to sound shapes and gestural energies but remains primarily qualitative. AudioTools bridges these approaches: students learn to hear formant trajectories, spectral centroids, roughness indices, quantifiable dimensions that can be extracted, visualised and modified. A student analysing a vowel sound learns to hear not just 'an /a/ sound' but 'a sound with F1 at 800 Hz, F2 at 1200 Hz, with 50 Hz bandwidth', parameters that can then be exaggerated, inverted or transferred to other sounds.

The pedagogical workflow, analyse, visualise, edit, resynthesise, listen, develops analytical skills through experimentation. Students record their voice, extract pitch contours, then ask: 'What if we invert this? Double its range? Flatten it?' This hypothesis-testing cycle builds understanding without requiring programming knowledge. Advanced projects demonstrate research-creation pedagogy where methodology is documented as rigorously as aesthetic results, positioning composition as inquiry.

Student feedback has consistently emphasised two aspects: first, that the visual representation has lowered the barrier to understanding ('I can finally see what formants are'), and second, that the iterative edit–render–listen loop enables learning through experimentation without requiring programming knowledge. One student remarked: 'In Max/MSP I was always guessing which numbers to change. Here I can see the curve and shape it like I want'. This directness, treating analysis objects as graphical scores, makes complex DSP accessible to musicians without engineering backgrounds.

Several students have published their AudioTools-based compositions in student journals, including technical appendices detailing their workflows, something rarely seen in undergraduate composition portfolios.

6.5. Research-creation implications: Process as contribution

AudioTools aligns with contemporary research-creation discourse by positioning compositional process, not just finished works, as a scholarly contribution. The toolkit enables a methodology where documentation is intrinsic rather than supplementary: scripts, presets, analysis objects, parameter logs and intervention records constitute an archive of process that can be shared, critiqued and built upon by other researchers.

This addresses artistic research's challenge: communicating tacit knowledge. AudioTools scripts and analysis objects serve as executable documentation; other researchers can replicate methods and understand iterative decision-making. This enables cumulative knowledge building: when methods are reproducible with stable, open-source software, the field moves beyond ephemeral experimentation. The research-creation cycle becomes: develop compositional hypothesis, design analytical workflow, implement and refine, evaluate and document. Furthermore, shared analysis objects enable distributed creativity, multiple researchers contributing to collaborative projects, each stage documented and attributable.

This reproducibility enables cumulative knowledge building. When a composer publishes a paper describing an analysis-driven composition workflow, they can include the actual scripts and analysis objects as supplementary materials. Other composers can then adapt these methods, modifying the analysis algorithms, trying different editorial strategies and applying the workflow to their own materials. This differs fundamentally from black-box systems, where replication requires access to proprietary models or cloud APIs. AudioTools compositions are reproducible with free, open-source software that has remained stable for three decades.

The research-creation cycle becomes: develop a compositional hypothesis (e.g., 'Can prosodic contours from speech control spatial trajectories in multichannel diffusion?'), design an analytical workflow (extract pitch and intensity from speech, map to DBAP coordinates), implement and refine (edit the extracted curves, adjust mapping functions, generate spatial audio), evaluate (listen, analyse spectrograms, adjust) and document (publish the scripts, analysis objects and critical reflection). The methodology is as rigorous as scientific research, but retains artistic agency; the composer's editorial decisions, aesthetic judgements and creative intuitions remain central.

Furthermore, AudioTools supports what we might call distributed creativity: multiple researchers can collaborate by sharing analysis objects. One researcher extracts formant data from a field recording, another edits the formants to create impossible vocal geometries and a third applies these to instrumental synthesis.

https://doi.org/10.1017/S1355771826101204 Published online by Cambridge University Press

Each stage is documented, traceable and attributable. This collaborative model, common in scientific research, remains rare in electroacoustic composition, where individual authorship is jealously guarded. AudioTools enables collective research-creation where methods accumulate and evolve.

7. Conclusion

This article makes three linked contributions to electroacoustic discourse:

Methodological: We formalise an object-centric framework where phonetic descriptors, PitchTiers, FormantGrids and TextGrids become editable controllers within sound-based composition practice. By treating analysis objects as compositional scores, we bridge speech science and musical poetics. Small changes to these objects produce structural consequences in timbre, gesture and spatial dramaturgy, enabling research-creation where method and musical meaning interweave. The toolkit's expansion to over 300 scripts demonstrates the generativity of this paradigm: once analysis objects are understood as compositional material, a vast landscape of transformational strategies becomes accessible.

Empirical: Extended case studies with detailed parameter documentation demonstrate how the edit–render–listen cycle shapes compositional outcomes. Discoveries emerge through iterative refinement, formant compensation during pitch transformations, cross-domain translation of visual structure to sonic texture, and algorithmic generation tempered by human curation. These strategies surface only through prolonged engagement with materials characteristic of studio electroacoustic practice. The three case studies, cross-synthesis, image sonification and generative texture design, span the toolkit's methodological range while maintaining a consistent object-centric workflow.

Critical: We position AudioTools against contemporary alternatives, neural synthesis toolchains, real-time performance environments and automated MIR pipelines, arguing that interpretability, editorial control and reproducibility remain vital aesthetic and scholarly values. In an era of algorithmic opacity, we champion transparency: the ability to inspect, understand and communicate one's methods as integral to compositional discourse and research-creation practice. The toolkit's neural integration demonstrates that machine learning and interpretability are not opposed but complementary when properly architected.

Praat AudioTools validates the studio composition model, proving that deep-time engagement with analysis data yields sonic results unattainable through real-time improvisation alone. It proposes a methodology where the micro-structure of sound, formants, jitter, spectral trajectories and temporal grains, becomes the primary domain of the composer's attention. The toolkit transforms the phonetic into the aesthetic, providing a structured yet open environment for the next generation of spectromorphological research.

Between real-time improvisation and offline reflection, between algorithmic automation and human agency, between neural opacity and parametric transparency, AudioTools occupies a distinctive position: reproducible, data-informed composition that preserves interpretability and editorial control within the electroacoustic studio tradition. It offers composers a laboratory where phonetics and poetics converge, where measurement becomes material, and where the archive of process stands alongside finished works as a scholarly contribution to research-creation discourse.

The pedagogical dimension strengthens this contribution. By cultivating 'parametric listening' in students, the ability to perceive acoustic dimensions as editable parameters, AudioTools bridges the gap between analytical understanding and creative practice. This perceptual transformation, where one learns to hear formants, roughness and spectral centroids as manipulable qualities, represents a form of embodied knowledge that enriches both composition and analysis.

However, the core commitment remains unchanged: deep engagement with sound, understanding acoustic structure through rigorous analysis, mapping perceptual qualities to parametric control, designing systematic transformations while maintaining artistic agency, continues to be vital to sound-based composition. This is composition as research: methodical, documented, replicable, invested in understanding sound as deeply as transforming it. In a technological landscape increasingly dominated by opaque AI systems, AudioTools asserts that transparency, reproducibility and epistemic agency remain essential values for compositional practice that seeks not just to make sounds, but to understand them.

Acknowledgements. The author thanks Bar-Ilan University's Department of Music for institutional support, the Praat development team (Paul Boersma, David Weenink) for the foundational platform and responsive engagement with the research community, and colleagues in the electroacoustic composition community for feedback on early versions. Special thanks to beta-testing students whose discoveries informed pedagogical materials and whose creative applications of the toolkit challenged initial assumptions. Thanks to anonymous reviewers whose critiques shaped this article's focus towards compositional practice, critical positioning within electroacoustic discourse and integration of neural processing while maintaining interpretability.

References

Books and Chapters

Dudley, H. (1939). Remaking speech. *The Journal of the Acoustical Society of America*, **11**(2), 169–177. https://doi.org/10.1121/1.1916020

Magnusson, T. 2019. *Sonic Writing: Technologies of Material, Symbolic, and Signal Inscriptions*. New York: Bloomsbury Academic.

Mathews, M. V. 1969. *The Technology of Computer Music*. Cambridge, MA: MIT Press.

Schaeffer, P. 1966. *Traité des objets musicaux*. Paris: Éditions du Seuil.

Wishart, T. 1996. *On Sonic Art* (new & rev. ed., ed. S. Emmerson). London: Routledge.

Journal Articles

Smalley, D. 1997. Spectromorphology: Explaining Sound-Shapes. *Organised Sound* **2**(2): 107–126. https://doi.org/10.1017/S1355771897009059

Boersma, P. and Weenink, D. 2025. Praat: Doing Phonetics by Computer [v6.x]. http://www.praat.org/

Caillon, A. and Esling, P. 2021. RAVE: A Variational Autoencoder for Fast and High-Quality Neural Audio Synthesis. arXiv:2111.05011.

Cohen, S. 2025. Praat AudioTools [Software]. GitHub: https://github.com/ShaiCohen-ops/Praat-plugin_AudioTools

Engel, J., *et al.* 2020. DDSP: Differentiable Digital Signal Processing. Proceedings of ICLR 2020. arXiv:2001.04643.

Dodge, C. 1972. Speech Songs [audio recording]. CRI Records.

Lansky, P. 1985. Idle Chatter. On Idle Chatter and Other Conversations. Bridge Records.

https://doi.org/10.1017/S1355771826101204 Published online by Cambridge University Press