

EDO.js: A comprehensive JavaScript library for interaction with musical set theory in any tuning

Michael Seltenreich¹

¹ Department of Music, New York University, New York, New York 10003

DOI: [10.21105/joss.03784](https://doi.org/10.21105/joss.03784)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Brian McFee](#) ↗

Reviewers:

- [@napulen](#)
- [@albincoreya](#)

Submitted: 23 September 2021

Published: 16 November 2021

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

EDO.js is a JavaScript library for generating, analyzing, and visualizing pitch-collections within any tuning system of Equal Divisions of an Octave (EDO). The current focus is on psychological experiments pertaining to musical scale perception and pitch-centric music research more broadly. EDO.js implements major algorithms presented in the music theory and music cognition literature (e.g., [Balzano, 1982](#); [Carey, 2007](#); [Carey & Clampitt, 1989](#); [Lerdahl, 2001](#); [Rahn, 1991](#); [Vassilakis, 2001](#)), alongside a wide array of additional tools for analyzing and manipulating midi-data and melodies in the framework of musical set theory ([Forte, 1973](#)) and diatonic set theory. EDO.js sets itself apart from existing music analysis libraries (such as, music21 ([Cuthbert & Ariza, 2010](#))), in that while most libraries are centered around engaging with the canon of Western music through the lens of applied (and adopted) music-theoretical concepts (e.g., figured bass, roman numerals, etc.), EDO.js is approaching music from a more fundamental level and is aimed at engaging with music's primitives from a cognitive and psycho-acoustical point of view. As such, EDO.js provides analytical tools that can be applied also to non-Western music, making the library more exploratory in nature. Most critically, EDO.js is, to the best of our knowledge, the most comprehensive library to support non-standard tuning to date.

Two classes are at the core of the package. The EDO class, and a daughter class, the Scale class. The EDO class and the Scale class conceptually diverge only on a single aspect: pitches in the Scale class are regarded as pitch-classes (exist only within an octave) and they have an enharmonic meaning. However, in the EDO class pitches are regarded as specific, and are not confined to a single octave. That is to say, while the Scale class treats C3 and C4 simply as "C," the EDO class would regard them as separate entities. As such, the Scale class is better suited to engage with scales, their properties, and their structures, while the EDO class is an implementation of useful functions for engaging with more abstract musical structures.

The EDO Class

The class contains eight sets of functions:

- Functions used for converting between equivalent representations of the input in various formats (cents, ratios, pitches, intervals, frequencies, etc.)
- Functions used for quantifying various parameters of a given input (e.g., the number of common tones between two inputs).
- Boolean assertions on input data (e.g., is a set of pitches a transposition of another?)
- Functions used for visualization (e.g., contours, fractal trees, necklaces, fractal necklaces, nested necklaces, etc.).
- Functions used for importing and processing of midi files.
- Functions used for importing and processing MusicXml files.

- Functions used for exporting data in various formats (e.g., images).
- Other functions for analysis, and manipulation (e.g., extracting motives, extracting contour, generating harmonic progressions, extracting pitch distribution, generating pseudo-random melodies, and more).

The Scale Class

The class contains six sets of functions:

- Functions for converting between equivalent representations of the scale (e.g., for converting pitches to step sizes).
- Functions for quantifying various parameters of a given scale (cardinality, trichords, dissonance, coherence, and others).
- Boolean assertions about the scale (e.g., is it invertible? Is it a mode-of-limited-transposition?)
- Functions used for exporting (e.g., Scala files).
- Functions used for scale structure visualization (e.g., scalar fractal trees).
- Analysis, manipulation, and generation functions (e.g., calculating the coherence quotient, extracting diatonic motives, returning the Rothernberg Propriety, calculating Vassilakis Roughness, retrieving available n-chords, etc.).

In addition to the sets of functions described above, the Scale class also contains five chainable methods commonly used in set-theory.

- `Scale.invert()` returns the inversion of the original set.
- `Scale.mode(n)` returns the *n*th mode of the original set.
- `Scale.normal()` returns the set in normal order.
- `Scale.prime()` returns the set in its prime form.
- `Scale.complement()` returns the complement of the set in the current EDO.

For instance, `Scale.mode(3).invert().complement()` will return an instance of a Scale that is equivalent to the complement scale of the inversion of the 3rd mode of the original scale.

Statement of need

Edo.js is written in JavaScript for deployment on a Node.js server or in a web-browser. The full documentation for the package is available [here](#), and the package is available on [GitHub](#) or in Node Package Manager ([NPM](#)).

This library is aimed at music theorists, musicologists, and cognitive scientists working on pitch-centric musical research, for the creation of stimuli for experiments, and for the analysis of musical structures within and outside of standard tuning systems.

References

- Balzano, J. G. (1982). *The Pitch Set as a Level of Description for Studying Musical Pitch Perception*. Springer, Boston, MA. https://doi.org/10.1007/978-1-4684-8917-0_17
- Carey, N. (2007). Coherence and sameness in well-formed and pairwise well-formed scales. *Journal of Mathematics and Music*, 1(2), 79–98. <https://doi.org/10.1080/17459730701376743>

- Carey, N., & Clampitt, D. (1989). Aspects of Well-Formed Scales. *Music Theory Spectrum*, 11(2), 187–206. <https://doi.org/10.2307/745935>
- Cuthbert, M. S., & Ariza, C. (2010). Music21: A toolkit for computer-aided musicology and symbolic music data. In J. S. Downie & R. C. Veltkamp (Eds.), *ISMIR* (pp. 637–642). International Society for Music Information Retrieval. <https://doi.org/10.5281/zenodo.1416114>
- Forte, A. (1973). *The structure of atonal music* (Vol. 304). Yale University Press. ISBN: [9780300021202](https://doi.org/10.2307/745935)
- Lerdahl, F. (2001). *Tonal pitch space*. Oxford University Press, USA. <https://doi.org/10.1093/acprof:oso/9780195178296.001.0001>
- Rahn, J. (1991). Coordination of interval sizes in seven-tone collections. *Journal of Music Theory*, 35(1/2), 33–60. <https://doi.org/10.2307/843809>
- Vassilakis, P. (2001). Auditory roughness estimation of complex spectra—roughness degrees and dissonance ratings of harmonic intervals revisited. *The Journal of the Acoustical Society of America*, 110(5), 2755–2755. <https://doi.org/10.1121/1.4777600>