The Concatenator: A Bayesian Approach To Concatenative Musaicing

Christopher J. Tralie
Department of Mathematics And Computer Science
Ursinus College
ctralie@alumni.princeton.edu

Ben Cantil ("Encanti")
DataMind Audio
bencantil@gmail.com

Motivation

- Ben found Chris online by his open source implementation of Let It Bee[4]
- Ben[6] and others[5, 7] were making awesome music with Driedgers' technique[1], but it was too slow. We wanted a similar effect on **large corpora** in **real time**
- Music producers have so many sample packs on their computer that they haven't even listened to all of the sounds. We want a system that can efficiently pick out sounds for them to match with a target, preserving rhythm and pitch

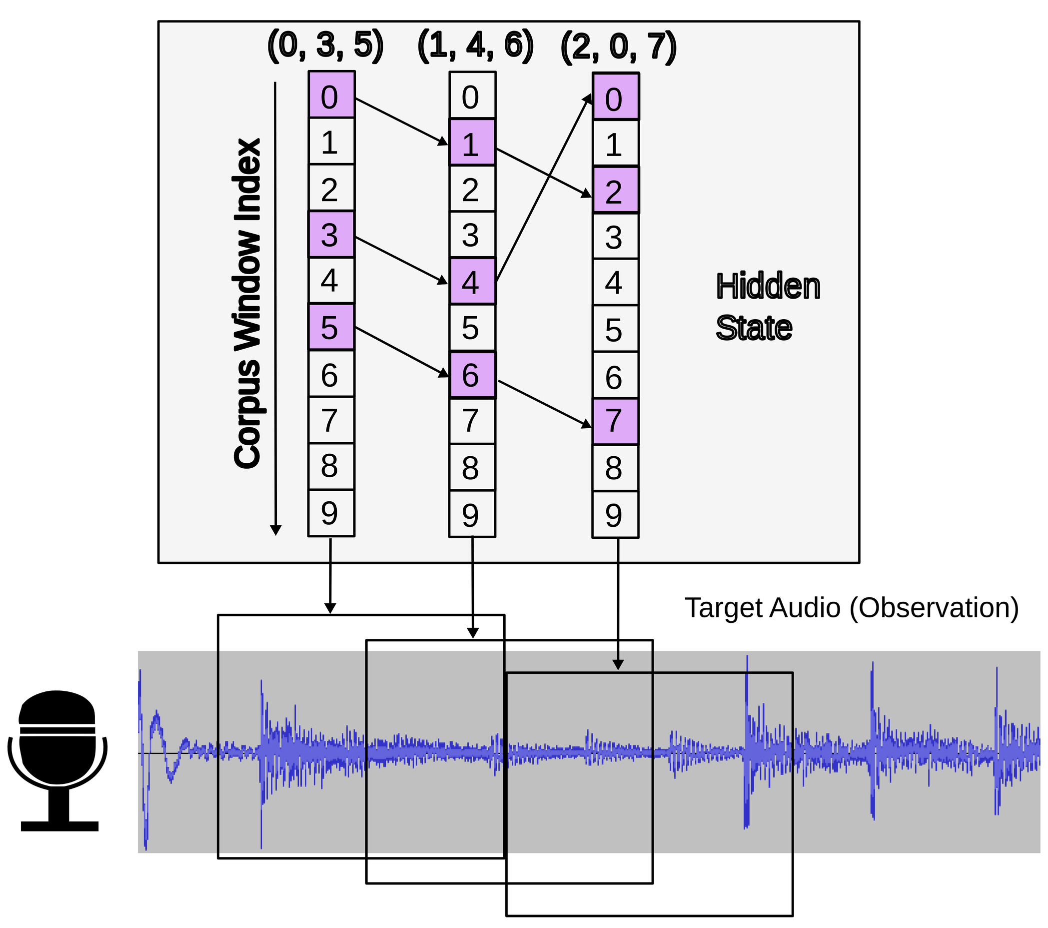
State Space

- A state st at time **t** represents a **sparse** selection out of **N** possible corpus windows as a **p**-length vector of indices

$$\vec{s_t}[k] \in \{0, 1, ..., N-1\}, k = 0, 1, ..., p-1$$

- Given associated weights **h**, and a spectogram **W** for the corpus, the spectral approximation of the target at frequency bin **m** is

$$\vec{\Lambda_t}[m] = \sum_{k=0}^{p-1} \vec{h_t}[k] W_{m, \vec{s_t}[k]}$$

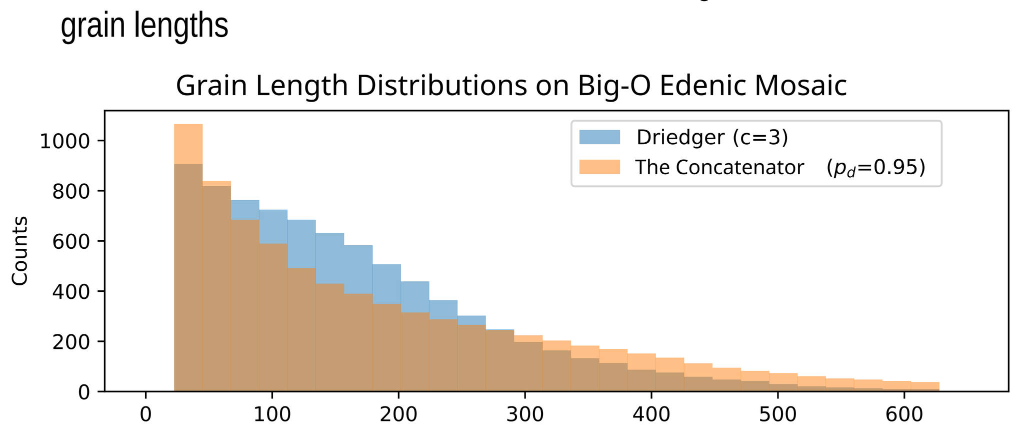


Transition Model

- **Markov assumption**: This state only depends on the last state
- Move to the next window in the corpus in time order with probability $\mathbf{p_d}$
- Different activations transition independently; **factorial hidden markov model**[13]

$$p_T(\vec{s_t} = \vec{b} | \vec{s_{t-1}} = \vec{a}) = \prod_{k=0}^{p-1} \begin{cases} p_d & \vec{b}[k] = \vec{a}[k] + 1 \\ \frac{1-p_d}{N-1} & \text{otherwise} \end{cases}$$

- In the absence of observation corrections, leads to a geometric distribution of grain lengths



Observation Model

- Cost for the i[th] particle based on the target spectrogram $\mathbf{v_t}$ is:

$$D_i(\vec{v_t} || \vec{\Lambda_i}) = \left(\sum \vec{v_t} \odot \log\left(\frac{\vec{v_t}}{\vec{\Lambda_i}}\right) - \vec{v_t} + \vec{\Lambda_i} \right) + \frac{||\boldsymbol{\alpha} \odot \vec{\boldsymbol{h_i}}||_2^2}{\mathbf{2}}$$

- Important to apply L2 regularization for near silence with a factor α
- To find the activations $\mathbf{h_i}$ minimizing $\mathbf{D_i}$ for each particle, we perform **L**=10 iterations of the regularized KL update equation for NMF[10], using only the activations $\mathbf{s_i}$ for this particle

$$\vec{h_i}^{\ell}[k] \leftarrow \vec{h_i}^{\ell-1}[k] \left(\frac{\sum_m (W_{m,\vec{s_i}[k]})(\vec{v_t}[m]) / (\vec{\Lambda_i}^{\ell-1}[m])}{(\sum_m W_{m,\vec{s_i}[k]}) + \boldsymbol{\alpha}[\mathbf{k}] \vec{\boldsymbol{h_i}}^{\boldsymbol{\ell-1}}[\mathbf{k}]} \right)$$

- This is very fast for each particle, and it is embarrassingly parallelizable

- Finally, we convert each divergence to an observation **probability** over all particles using softmax with "temperature" parameter τ, where higher τ means we promote a better fit to the target

$$p_O[i] = \frac{e^{-\tau D_i}}{\sum_j e^{-\tau D_j}}$$

Mixing

- Audio is mixed together using the same weights learned for spectral fits, using a Hann window (similar to [2] and [8])
- We can choose to zero out repeated activations for some amount of time when the particles vote on the final activations. Driedger is also mindful of this

Probability of Choosing "Good Enough" Particles

$$1 - \left(p_d + (1-p_d) \frac{(N-1-wk)}{N-1} \right)^{(2\delta+1)pP}$$

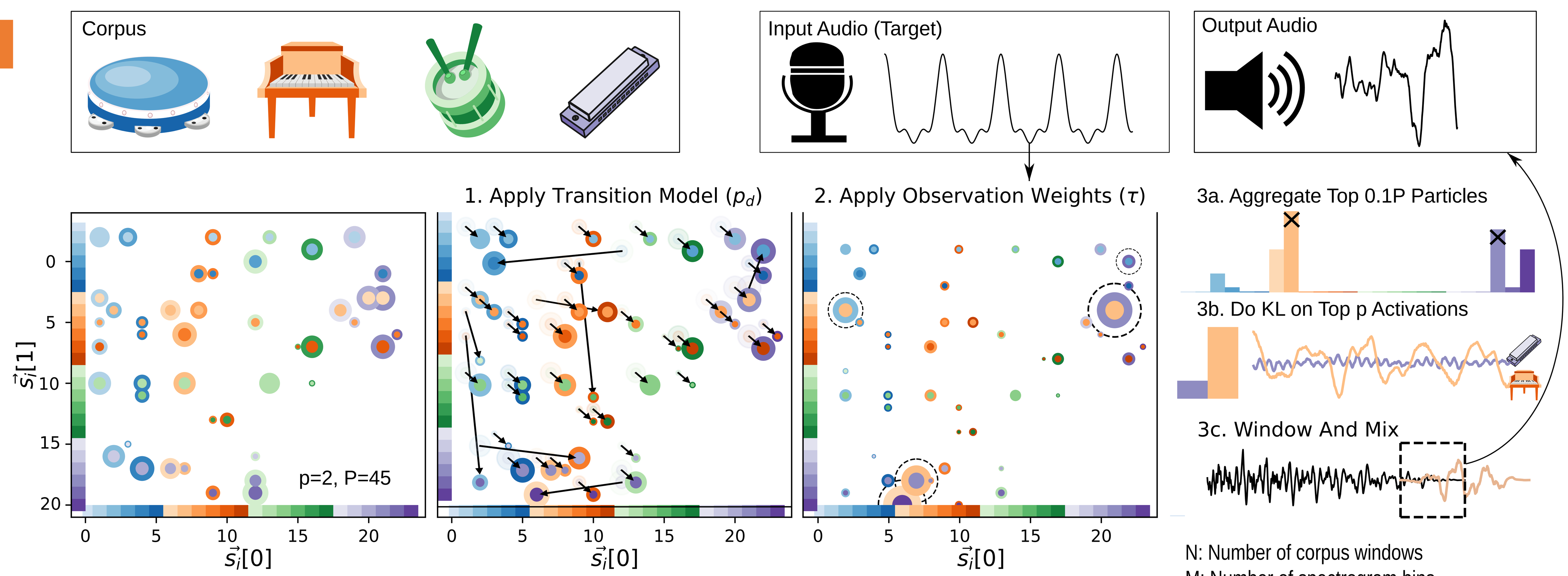
Allowed time offset from ideal window

Top **w** similar windows to ideal window

Particle Filter Pipeline for Concatenative Musaicing

Beginning: Randomly throw a bunch of "particles" (darts) into the corpus. Then repeat the following steps as time progresses

1. With high probability $\mathbf{p_d}$, slide particles forward in time in the corpus. With small probability **(1-$\mathbf{p_d}$)**, randomly jump to new locations in the corpus to adapt to new sounds coming in.
2. Give higher weight to particles that fit the target better, according to KL divergence of the best possible fit *using only the activations from that particle*
3. Resample particles periodically according to weight to replicate ones with a good fit ("survival of the fittest")
4. At every hop length time increment, have the top 10% particles vote to determine the final corpus activations, and solve one final NMF problem to mix in audio



Driedger[1] NMF Time Complexity: O(LMNT): *Linear* dependence on corpus size

The Concatenator Time Complexity O(LMPpT); **Does not scale with corpus size**! (We also use smaller L)

N: Number of corpus windows
M: Number of spectrogram bins
P: Number of particles
p: Activations per window
L: NMF Iterations (we can use fewer than Driedger)
T: Number of windows in target

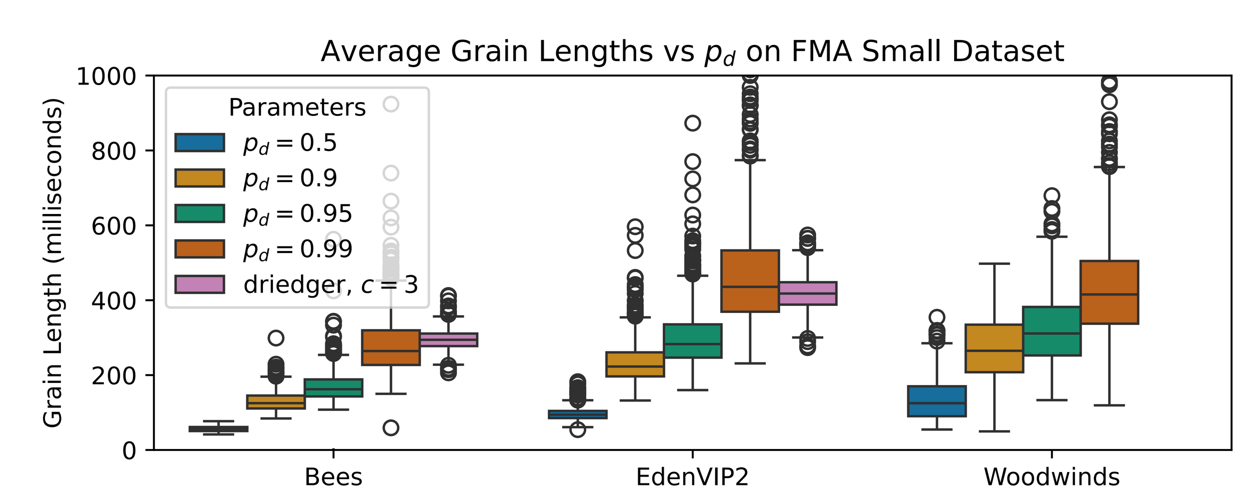
Quantitative Evaluation: Free Music Archive (FMA) Dataset

- Randomly subsampled 1000 30 second clips from the Free Music Archive (FMA)-small dataset, each of which we used as a target for 3 different corpora:

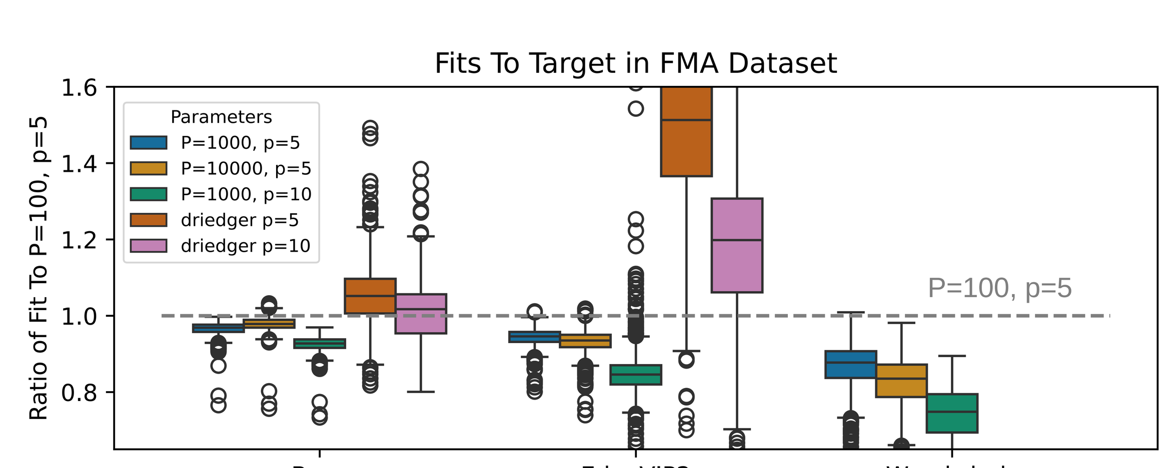
Small: Driedger Bees[1] (~1 minute)

Medium: Eden VIP2[6] (~10 minutes)

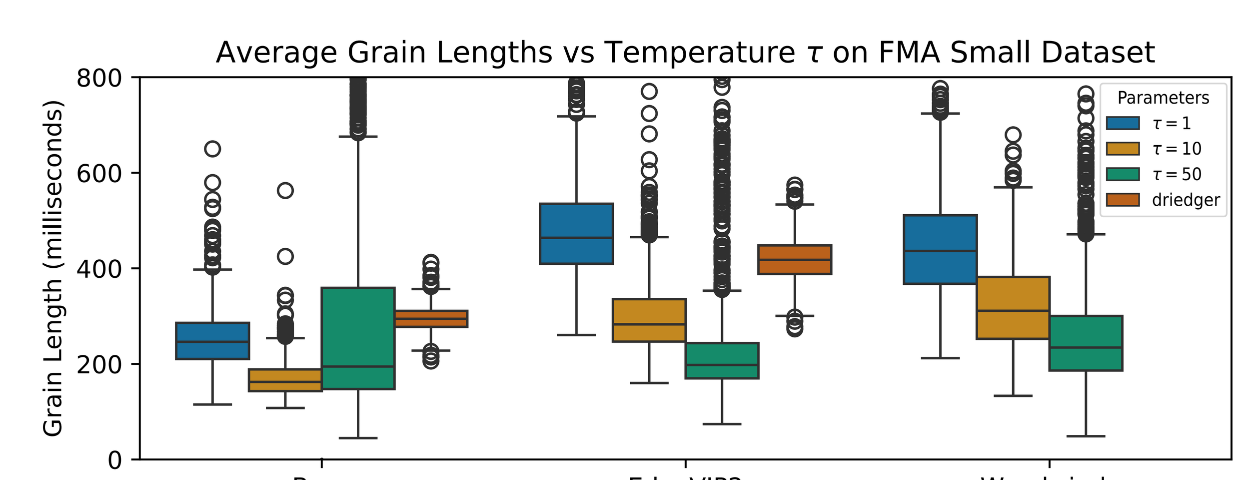
Large: Ulowa Woodwinds[12] (~1.6 hours)



- Increasing p_d **increases** the average grain length since windows are less likely to jump at each timestep.

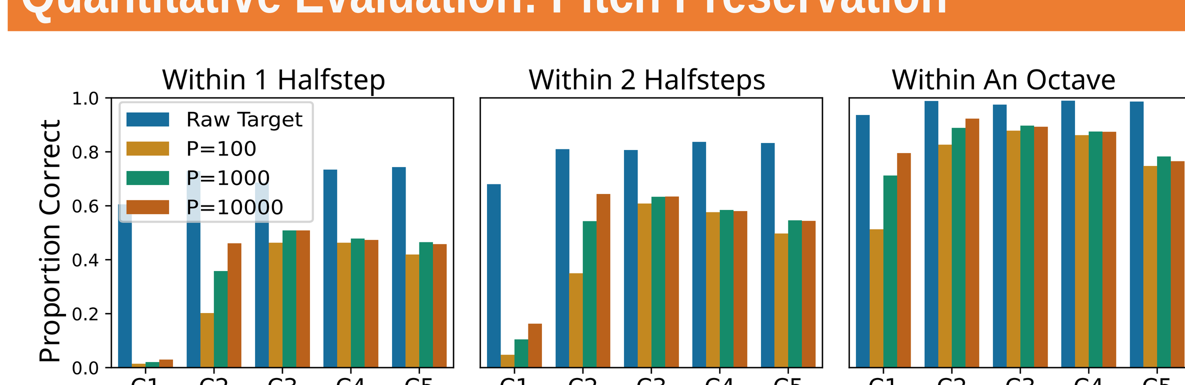


- Increasing polyphony leads to a better fit (ratios <1), and increasing particles leads to a better fit, especially for larger corpora like the Woodwinds



Increasing temperature τ **decreases** the average grain length since this prioritizes the observation probability at each timestep.

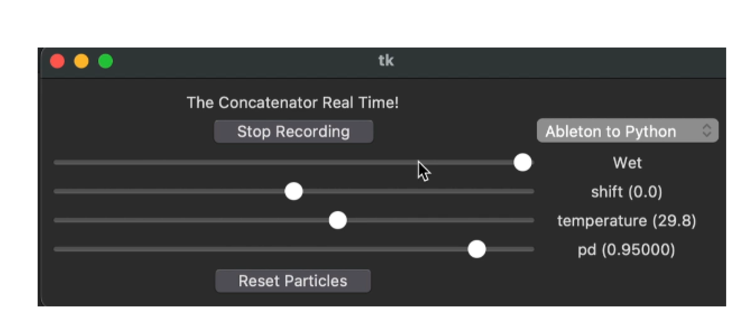
Quantitative Evaluation: Pitch Preservation



- Tested out on stems in Musdb-18hq dataset [14]
- Good pitch preservation in all but the lower octaves (could be improved with CQT)

Qualitative Results / Open Source Plugin

https://www.ctralie.com/TheConcatenator



Built on portaudio[15] for real time mode

- Huge corpora (e.g. *The Wall* album, hours long sample packs)
- Corpora recreating themselves
- Various target melodies, counterpoint, full mixes, basses, drums, vocals, noise

DataMind Audio Plugin Coming Soon...

- Special thanks to **Robin Leathart** for tireless software development!
- Special thanks to **Jacob Crider** for UI layout and skin design
- Special thanks to **Luca Ayscough** for help optimizing the initial prototype
- Special thanks to **Yashique Chalil** and **Martin Parker** for value feedback during beta testing

References

[1] J. Driedger, T. Prätzlich, and M. Müller, "Let it bee: towards nmf-inspired audio mosaicing." in Proceedings of 16th International Society for Music Information Retrieval (ISMIR), 2015, pp. 350–356.
[2] D. Schwarz, "A system for data-driven concatenative sound synthesis," in 3rd International Conference on Digital Audio Effects (DAFx), 2000, pp. 97–102.
[3] B. L. Sturm, "Matconcat: An application for exploring concatenative sound synthesis using matlab." in 7th International Conference on Digital Audio Effects(DAFx), 2004.
[4] C. Tralie, "Let it bee," https://github.com/ctralie/ LetItBee, 2018.
[5] R. Clouth, "Zero point," zero-point, 2020. https://robclouth.com/
[6] B. Cantil, "Edenic mosaics," 2021.
[7] V. Drakes, "Hate devours its host," https://amekcollective.bandcamp.com/album/hate-devours-its-host, 2023.
[8] M. Buch, E. Quinton, and B. L. Sturm, "Nichtnegativematrixfaktorisierungnutzendesklangsynthesensystem (nimfks): Extensions of nmf-based concatenative sound synthesis," in Proceedings of the 20th International Conference on Digital Audio Effects, 2017,p. 7.
[9] H. Foroughmand Aarabi and G. Peeters, "Multi-source musaicing using non-negative matrix factor 2-d deconvolution," in 18th International Society for Music Information Retrieval (ISMIR) Late-Breaking Demo Session, 2017.
[10] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," IEEE Transactions on Audio, Speech, And Language Processing, vol. 15, no. 3, 2007.
[11] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," ACM Transactions on Graphics (Proc. SIGGRAPH), vol. 28, no. 3, Aug. 2009.
[12] "The university of iowa musical instrument samples," https://theremin.music.uiowa.edu, last Accessed:2024-04-05.
[13] Z. Ghahramani and M. Jordan, "Factorial hidden markov models," Advances in neural information processing systems, vol. 8, 1995.
[14] J. Salamon, R. M. Bittner, J. Bonada, J. J. Bosch, E. Gómez Gutiérrez, and J. P. Bello, "An analysis/synthesis framework for automatic f0 annotation of multitrack datasets," in Hu X, Cunningham SJ, Turnbull D, Duan Z. ISMIR 2017 Proceedings of the 18th International Society for Music Information Retrieval Conference; 2017 Oct 23-27
[15] R. Bencina and P. Burk, "Portaudio-an open source cross platform audio api," in ICMC, 2001.