Beyond the Beats: Can Downbeat Trackers Predict Hypermetre?

Jose Esquivel and Jordan B. Smith[0000−0002−0316−1235]

Queen Mary University of London, 327 Mile End Road, Bethnal Green, London E1 4NS, UK
{j.esquiveldejesus, jordan.smith}@qmul.ac.uk

Abstract. Hypermetre is the metric structure between the timescales of bars and sections in a piece of music. Modelling hypermetre could make existing downbeat tracking and structure analysis algorithms more robust. However, this topic is under explored compared to downbeat tracking and structure analysis, perhaps due to the lack of annotated data for training. In this work, we evaluate the success of using existing downbeat tracking algorithms to perform hypermetre tracking, proposing five methods to do so. We define four levels of hypermetre complexity, and find that a straightforward baseline is best for the simplest cases, but that adaptations of downbeat trackers can perform better for more complex cases.

Keywords: Hypermetre · Downbeat tracking · Music Information Retrieval.

1 Introduction

Hypermetre is the metrical structure that lies between the shorter timescale of downbeats and bars and the longer timescale of structural sections (e.g., chorus, verse and bridge) [16]. Figure 1 illustrates how hypermeasures are built out of bars, and how sections are in turn built on hypermeasures.

Meter tracking is harder when the time signature is inconsistent. Similarly, hypermetre tracking is harder when the hypermetre varies. Changes in hypermetre appear to be much more common than changes in metre [20]. This appears to be the case for the McGill Billboard Dataset (MBD) [3], a set of chord and section annotations for a set of pop music from the 1950s to the 2000s that appears to also indicate hypermetric groupings of bars (see Section 4.1). Out of 1092 annotations, fewer than 1% contain a change in time signature (e.g., from 4/4 to 3/4); by contrast, a majority of songs (64%) contain an apparent change in hypermetre; e.g., from a pace of 4 bars per hypermeasure to 5 bars per hypermeasure.

Proc. of the 17th Int. Symposium on Computer Music Multidisciplinary Research, London, United Kingdom, 2025

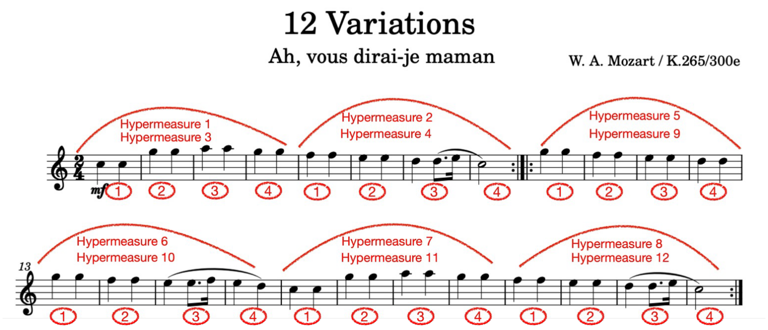

Fig. 1. Illustration of how hypermeasures relate to other units of time in a piece.

Using structural information as a constrain template has been found to be effective in guiding systems that generate music [17]. Despite this, and despite hypermetre being a well-known topic among music theorists [6], most prior research on segmentation has focused on downbeat tracking and structure analysis, making hypermetre an under explored topic in comparison.

In this work we explore using existing downbeat tracking algorithms to perform hypermetre tracking. We propose five methods that use the output of existing beat and downbeat trackers to predict hypermetre. We build simple systems, including: one that makes strong assumptions about the hyper-time signature; one that uses a downbeat tracker with extreme parameter settings; two that apply standard downbeat tracking to modified versions of the audio; and one that takes an structure analysis approach.

Finally, we identify the MBD as a valuable source of hypermetre annotations. We describe its properties and discuss the appropriateness of using this data to evaluate hypermetre prediction systems.

2 Related Work

2.1 Beat and Downbeat Tracking

Hypermetre tracking is similar to Beat and downbeat tracking, but at a different timescale, as shown in Figure 1. In this figure, the beat timescale would correspond to each crotchet, the downbeat timescale would correspond to the first beat of each bar, and the hypermetre timescale would correspond to the first crotchet in the bars with the circled annotation of 1.

Research of systems that predict metre focused firstly on the beat and downbeat timescales. Those early examples of metre tracking took a feature engineering approach like [7, 9] in which induction and dynamic programming algorithms complement the feature engineering approach to predict beats and downbeats.

Feature engineering based methods were later surpassed by Machine Learning (ML) methods in performance, and seemed to have shifted the focus to the downbeat timescale.

A typical modern approach to beat and downbeat tracking (shared by [10, 12, 5, 21, 4, 2]) is to train a neural network to estimate beat and downbeat times, and post-processes the estimates with a Dynamic Bayesian Network (DBN) to resolve metrical ambiguity, stabilise the tempo, and fix the time signature.

A second approach that uses ML is to use self-supervised model [13, 14] to predict metre at different timescales as a hierarchical tree. In these works, the different timescales are modelled as hierarchical levels, assuming binary structure regularity. It is also acknowledged that the performance drops when this binary regularity is not present, which is often the case at higher metrical hierarchies (any metre level beyond the bar is regarded as hypermetre). They use their own predicted beat and downbeat information as a basis to predict the metric hierarchy instead of relying on existing downbeat trackers or annotations.

In this work, we use one of the models [2] included in the widely available Python library *Madmom* [1], which uses the first approach of training a ML model. We propose, use and evaluate systems that use [2] that are incapable of modelling time signature changes, which is important for the hypermetre timescale. However, our aim in doing so is to assess how much this limits performance in a different context: the prediction of hyperdownbeats.

2.2 Music Structure Analysis and Segmentation

The aim of music structure analysis (MSA) is to split a piece of music into meaningful sections (e.g., intro, chorus and bridge). In Figure 1, we can see that a section is typically formed of several hypermeasures. Thus, another reasonable way to predict hypermetre could be to approach it "top-down", starting with a structural analysis and predicting hyperdownbeats based on that.

Segmentation analysis usually works with hierarchies that refer to 'sections', 'phrases', 'motives' and 'notes' [18]. In contrast, the metrical hierarchies are 'beat', 'downbeat', and 'hyperdownbeat' [13]. Thus, MSA methods focus on identifying repeated melodies and sequences and sudden changes [19], whereas hypermetre may relate instead to a sense of pulse. This approach is reasonable since at the hypermetre timescale, there is a sense of pulse that is strongly related to the segmentation hierarchies, in special the phrase hierarchy, which may overlap perfectly with hypermetre for certain musical genres like pop or rock.

The fact that MSA systems do not model hypermetre explicitly may be due to a lack of training data. For example, the All-In-One system [15] takes an integrated approach that performs beat, downbeat tracking and segmentation analysis in the same system, recognising that these are all related tasks. In the next section, one of the proposed systems attempts to insert hypermeasure predictions into the output of All-In-One.

3 Methods

None of the tools we have used in this work (Madmom and All-In-One) were designed to handle time signature changes; therefore, we don't anticipate that any of them will be able to predict hyperdownbeats all that accurately, especially when there are hypermetre changes. Training a model to predict hypermetre seems like a reasonable approach, since hypermetre is similar to downbeat tracking but at a different timescale, but because of the lack of hypermetre annotated training data, we firstly propose and evaluate these baselines.

Baseline This method assumes that the hypermetre is in four: i.e., that bars occur in groups of four. This assumption makes a reasonable baseline since, of the songs in MBD, 93.5% have a 4/4 time signature and 77% of the hyperbars are four bars long.

This algorithm estimates the downbeats using the Madmom library. Once we have the downbeats, we assume that every fourth downbeat is a hyperdownbeat, starting with the first. This method relies on the accuracy of prediction of the downbeat: if it is predicted incorrectly (e.g. the first downbeat predicted as beat 3 in a metre of 4), then predictions of hypermetre would propagate this error to the remaining predictions.

Slow-Tempo Hyperdownbeats move at a slower timescale than downbeats. Thus, by setting the downbeat tracker included in the Madmom library to target the hypermetre timescale, we can use it to perform hypermetre tracking.

The default tempo range used by Madmom for inference is 55–215 beats per minute (according to its documentation), or 13–53 downbeats per minute in a 4/4 metre. However, we can set the tempo range parameter freely. If we direct Madmom to use a tempo range of 13–53 bpm, then it may predict "beats" that are in fact downbeats, and "downbeats" that are in fact hyperdownbeats, in the range of 3–13 per minute.

With this algorithm, we expect poor predictions of downbeats at the hypermetre scale because the downbeat tracker [2] included in Madmom was trained with music that includes the range 60–224 bpm (this is the tempo range for the ballroom dataset [11], which was used along other datasets to train [2]) and we are using it for 3–10 bpm.

Accelerated The approach of the **Accelerated** method is to speed up the music, so that we can use Madmom with its default target tempo range (i.e., the default range of 55 - 215 bpm). Then, as with Slow-Tempo, we interpret the predicted downbeats as hyperdownbeats.

We use the "effects.time_stretch" acceleration function provided by librosa to accelerate the songs. Then, we divide the predictions of the accelerated version by the speed-up factor to get hyperdownbeat predictions on the non accelerated song. Looking at different acceleration factors (2, 3 and 4) on a subset of 26 songs (we chose the first 26 songs from the MBD, we used this subset later to validate the annotations), we observed that a 3x speed-up worked best. A factor of 2x predicts the same points in time as downbeats and for a factor of 4x, consecutive fast onsets started to be considered a single one, shifting the location of the prediction outside the tolerance interval of 0.5s. Accelerating by 3x seemed to reach a happy medium.

Decimated An alternative way to speed up the audio without warping the timbre is to 'decimate' the song by deleting all but the first beat of every bar. If we predict the beats and downbeats of such a 'decimated' track, this could indicate the downbeats and hyperdownbeats of the original audio.

That is the approach of the **Decimated** method: given an input track A, we first use Madmom to predict the beats and downbeats of A. Then we select the first 'beat-span' of each bar (i.e., from the onset of the first beat to the onset of the second) and concatenate these to produce a new 'decimated' audio track A_D. Finally, we use Madmom again to estimate the beats and downbeats of A_D. The predicted beats of A_D should correspond to the downbeats of A, and the downbeats of A_D should correspond to the hyperdownbeats of A. For this algorithm, we expect phase error propagation in cases when downbeats are predicted incorrectly.

Interpolated The Interpolated method relies on a segmentation approach and can be considered a *top-down* approach.

This algorithm takes the segmentation predictions at the section level performed by the All-In-One system [15] and interpolates between them to predict additional hyperdownbeats at each section's midpoint (i.e. take the average of start and end times of the section as an interpolated hyperdownbeat).

This assumption is based on the fact that a symmetrical organization of musical ideas is common according to [20]. It also matches the assumption of [13] that sections have binary form. Thus, the **Interpolated** method should produce poor results when this assumption isn't met.

4 Experiments

4.1 Data: The McGill Billboard Dataset

The McGill Billboard Dataset [3] annotations contain time signature and key, beat-level chord labels, and structural information. According to the dataset creators, "Each transcription is broken with line breaks into phrases, which are defined loosely as any point where a group might choose to start playing during a rehearsal" [3]. Although this guideline for annotators seems vague, we inspected a subset of the first 26 songs (which contains at least one example for the decades from 1960–1990) and found that every line break downbeat was more important than the downbeats of the rest of the line, even when the line breaks were not

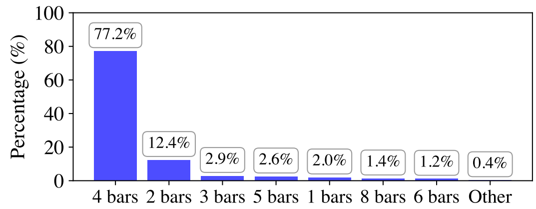

Fig. 2. Distribution of hyperbar lengths in bars for the MBD. The remaining 0.4% of the data consists of hyperbar lengths measuring 7, 9, 10, 11, 12, 13, 14, 16, and 17 bars. These annotations require review and modification because they sometimes indicate the repetition of a different hyperbar length. For instance, an annotation might show that a hyperbar of 4 bars repeats four times, resulting in a 16-bar line. This correctly reflects the final length of a 4 hyperbar section, but it is needed to insert additional linebreaks to make the annotations more consistent with hypermetre.

equally important. Then we used the whole dataset (1092 songs) to perform the analysis and the experiments.

In Figure 2 we can see the distribution of the bar count per line in the MBD. Musical phrases that have a duration of four measures are common (77%).

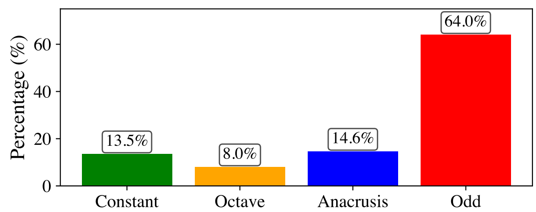

Fig. 3. Distribution of different types of hypermetre changes in the MBD songs.

An analysis of bar-based line lengths in the MBD, particularly where these lengths change, shows different perceived effects. For example, a song might change from a 4-bar line to an 8-bar line before returning to 4. A change that doubles the line length in this way is often imperceptible. However, a change from a 4-bar line length to a 2-bar line length is perceived differently, more similar to a temporary change in time signature. This would be the case also for line length changes by a factor different of two (e.g. from 4 to 5 and then back to 4)

We define four categories of songs based on how the hypermetre changes within them:

Constant: hypermetre is constant in the whole song.

Anacrusis: the changes in hypermetre occur only at the first or the last hyperbar, thereby acting as a kind of 'pickup', but constant otherwise.

Octave change: the duration of two consecutive hyperbars differ in length, doubling the bar (e.g., changing from 4 to 8, or from 3 to 6).

Odd change: the duration of two consecutive hyperbars differ in length by a factor other than two (e.g., a change from 4 to 5), or the length is halved (e.g., a change from 4 to 2)

Figure 3 shows that only 28% of the songs in the MBD have a regular (Anacrusis and Constant) hypermetre (constant except for at the beginning and end), whereas 72% of songs are irregular in some way.

4.2 Evaluation Metrics

To evaluate the systems, we use the same evaluation metrics as in beat and downbeat prediction: precision, recall and F-measure.

We choose a tolerance of 0.5 s for a prediction to be considered as correct. This tolerance is a balance between the tighter tolerance of 0.07 s used for downbeat tracking [8, 2] and the largest 3 s tolerance used in Music Structure Analysis [19, 18].

4.3 Results

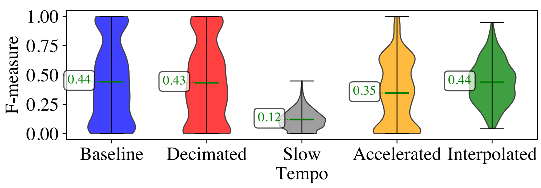

Fig. 4. Violin plot of the distribution of F-measure (with mean) achieved by the five systems for all 1,092 songs in the MBD.

Results in Figure 4 show that the methods with the best average F-measure are Interpolated and Baseline ($\bar{F} = 0.44$), followed by Decimated ($\bar{F} = 0.43$) and Accelerated ($\bar{F} = 0.35$), with Slow-tempo in a distant last place ($\bar{F} = 0.12$). The distribution in F-measure (we take each song's F-measure for the violin plot per method in Figure 4 and take the average as a summary metric of the method) has a roughly uniform shape for Baseline and Decimated: some songs are modelled perfectly and others very poorly. In contrast, the Interpolated method's results have a bell curve shape: the system usually gives a mediocre prediction.

We expected each method to perform worse when there were changes in hypermeasure length. To assess the impact of these changes, we plot the F-measure again in Figure 5 according to whether the song's hypermetre is Constant or whether it contains Anacrusis, Octave changes, or Odd changes.

For Baseline, when applied only to songs with constant hypermetre, the average F-measure is $\bar{F} = 0.77$, much higher than the overall average $\bar{F} = 0.44$. The performance of the Decimated method also decreases when the complexity of the hypermetre increases (from $\bar{F} = 0.55$ for Constant to $\bar{F} = 0.37$ for Odd).

When we consider songs with irregular hypermetre (i.e., Odd cases), which comprise 64% of the dataset, the best performing method is Interpolated ($\bar{F} = 0.42$), by a slight margin over the Baseline (35%) and Decimated (37%).

The F-measure for Slow Tempo is $\bar{F} = 0.12$ as shown in Figure 4, a poor result that is not affected much by the presence of hypermetre changes.

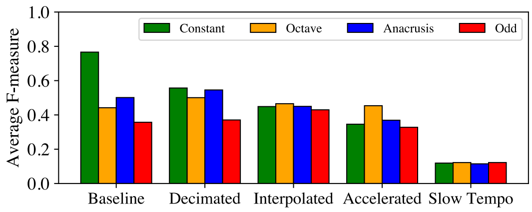

Fig. 5. Average F-measure performance per method and per classification of hypermetre change.

5 Conclusion

Can downbeat trackers predict hypermetre? We found that they can, sometimes, but not reliably: the distribution in F-measure ranged from 0.0 to 1.0 for most systems, but the average F-measure was always below 0.5.

The success of these methods appears to be limited by the complexity of hypermetre changes, from Figure 5, we can observe that each method performed worse on songs with odd hypermetre changes compared to those with constant hypermetre.

A hyperdownbeat tracking algorithm that is unable to handle hyper-time signature changes or anacrusis should expect to predict perfectly at most 15% of the songs in the MBD.

Based on the trends evident in Figure 5, we can see that the inability to handle changes in metre reduces the accuracy of the predicted hyperdownbeats. We are relying on beat and downbeat trackers to perform hypermetre tracking, which in turn relies on the assumption of a constant time signature. The presence of non-regular hyper-time signatures indicates that the assumption of constant metre would not yield good results at the hypermetre timescale, highlighting the need of research algorithms that handle metre changes at this timescale.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Böck, S., Korzeniowski, F., Schlüter, J., Krebs, F., Widmer, G.: Madmom: A new python audio and music signal processing library. In: Proceedings of the 2016 ACM Multimedia Conference. pp. 1174–1178. Association for Computing Machinery, Inc (10 2016)
2. Böck, S., Krebs, F., Widmer, G.: Joint beat and downbeat tracking with recurrent neural networks. In: Proceedings of the International Society for Music Information Retrieval Conference (2016)
3. Burgoyne, J.A., Wild, J., Fujinaga, I.: An expert ground truth set for audio chord recognition and music analysis. In: Proceedings of the 12th International Society for Music Information Retrieval Conference. pp. 633–638. Miami, FL (2012)
4. Cheng, T., Goto, M.: Transformer-based beat tracking with low-resolution encoder and high-resolution decoder. In: Proceedings of the International Society for Music Information Retrieval Conference. pp. 466–473. Milan, Italy (2023)
5. Chiu, C.Y., Ching, J., Hsiao, W.Y., Chen, Y.H., Su, A.W.Y., Yang, Y.H.: Source separation-based data augmentation for improved joint beat and downbeat tracking. In: Proceedings of the European Signal Processing Conference. pp. 391–395. Dublin, Ireland (2021)
6. Cohn, R.: Complex hemiolas, ski-hill graphs and metric spaces. Music Analysis **20**(3), 295–326 (2001)
7. Dixon, S.: Automatic extraction of tempo and beat from expressive performances. Journal of New Music Research **30**(1), 39–58 (2001)
8. E. P. Davies, M., Böck, S., Fuentes, M.: Tempo, Beat and Downbeat Estimation. https://tempobeatdownbeat.github.io/tutorial/intro.html (11 2021)
9. Ellis, D.P.: Beat tracking by dynamic programming. Journal of New Music Research **36**(1), 51–60 (3 2007)
10. Fuentes, M., McFee, B., Crayencour, H.C., Essid, S., Bello, J.P.: A Music Structure Informed Downbeat Tracking System Using Skip-chain Conditional Random Fields and Deep Learning. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 481–485. IEEE (5 2019)
11. Gouyon, F., Dixon, S., Pampalk, E., Widmer, G.: Evaluating rhythmic descriptors for musical genre classification. In: Proceedings of the AES 25th International Conference. vol. 196, p. 204 (2004)
12. Hung, Y.N., Wang, J.C., Song, X., Lu, W.T., Won, M.: Modeling Beats And Downbeats With A Time-Frequency Transformer. In: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 401–405. IEEE (2022)
13. Jiang, J., Chin, D., Zhang, Y., Xia, G.: Learning hierarchical metrical structure beyond measures. In: Proceedings of the International Society for Music Information Retrieval Conference. pp. 201–209. Bengaluru, India (2022)
14. Jiang, J., Xia, G.: Self-supervised hierarchical metrical structure modeling. In: ICASSP (2023)
15. Kim, T., Nam, J.: All-in-One Metrical and Functional Structure Analysis with Neighborhood Attentions on Demixed Audio. In: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics. vol. 2023-October. Institute of Electrical and Electronics Engineers Inc. (2023)

16. Krebs, H.: Hypermeter and Hypermetric Irregularity in the Songs of Josephine Lang. Engaging Music: Essays in Music Analysis (2005)
17. Lattner, S., Grachten, M., Widmer, G.: Imposing Higher-Level Structure in Polyphonic Music Generation using Convolutional Restricted Boltzmann Machines and Constraints. Journal of Creative Music Systems (2018)
18. McFee, B., Nieto, O., Bello, J.P.: Hierarchical evaluation of segment boundary detection. In: Proceedings of the International Society for Music Information Retrieval Conference, ISMIR. pp. 406–412 (2015)
19. Nieto, O., Mysore, G.J., Wang, C.I., Smith, J.B.L., Schluter, J., Grill, T., McFee, B.: Audio-Based Music Structure Analysis: Current Trends, Open Challenges, and Applications (2020)
20. Temperley, D.: Hypermetrical Transitions. Music Theory Spectrum **30**(2), 305–325 (10 2008)
21. Zhao, J., Xia, G., Wang, Y.: Beat Transformer: Demixed beat and downbeat tracking with dilated self-attention. In: Proceedings of the International Conference on Music Information Retrieval. pp. 169–177. Bengaluru, India (2022)