# HIERARCHICAL EVALUATION OF SEGMENT BOUNDARY DETECTION

**Brian McFee[1,2], Oriol Nieto[2], and Juan P. Bello[2]**

[1]Center for Data Science, New York University
[2]Music and Audio Research Laboratory, New York University
[1,2]{brian.mcfee, oriol, jpbello}@nyu.edu

## ABSTRACT

Structure in music is traditionally analyzed hierarchically: large-scale sections can be sub-divided and refined down to the short melodic ideas at the motivic level. However, typical algorithmic approaches to structural annotation produce flat temporal partitions of a track, which are commonly evaluated against a similarly flat, human-produced annotation. Evaluating structure analysis as represented by flat annotations effectively discards all notions of structural depth in the evaluation. Although collections of hierarchical structure annotations have been recently published, no techniques yet exist to measure an algorithm's accuracy against these rich structural annotations. In this work, we propose a method to evaluate structural boundary detection with hierarchical annotations. The proposed method transforms boundary detection into a ranking problem, and facilitates the comparison of both flat and hierarchical annotations. We demonstrate the behavior of the proposed method with various synthetic and real examples drawn from the SALAMI dataset.

## 1. INTRODUCTION

The analysis of structure in music is a principal area of interest to musicologists. Its goal is to identify and characterize the form of a musical piece by investigating the organization of its components, such as sections, phrases, melodies, or recurring motives. Traditional analyses usually provide multiple levels of annotation (*e.g.*, Schenkerian analysis), which suggest that music is structured hierarchically [3], and can be modeled and analyzed using tree representations [2].

In the music information research literature, *music segmentation* (also known as *music structure analysis*) is a task that aims to automatically identify the structure of a musical recording [6]. The segmentation task has historically been geared toward algorithms which produce a flat partition of the recording into disjoint segments. This formalization contrasts with our intuition that music exhibits hierarchical structure [7, 8]. Even though a large dataset of hierarchically-structured human annotations is now publicly available [8], current evaluation methodologies are defined only for *flat* segmentations. As a result, the dimension of *depth* has been practically ignored in the evaluation of music segmentation algorithms.

In contrast to segmentation, the *pattern discovery* task formulation allows output segments to overlap, and the annotation is not required to cover the entire piece. These two tasks share multiple attributes [5], and steps toward a general formulation musical structure analysis could be made by accounting for depth in segmentation. Numerous metrics to evaluate pattern discovery have been proposed [1]. However, they are designed to capture repeated patterns, and would be inappropriate for evaluating non-repeating, hierarchical structure.

### 1.1 Our contributions

We present the *Tree Measures* ($T$-measures): an evaluation framework designed to measure the accuracy of boundary detection in hierarchical segmentations. The $T$-measures infer frame-wise similarity from a hierarchical annotation, and then compare the induced rank-orderings to assess agreement between reference and estimated annotations. The $T$-measures integrate information from all layers of a hierarchy, trivially specialize to handle flat annotations, and require no explicit correspondence between the depth of the estimated and reference hierarchies. Thus, the $T$-measures encourage the development of new algorithms to produce richer representations of structure. Although not all music can necessarily be modeled using trees [11], we argue that tree-based evaluation represents a first step toward moving beyond flat structure analyses. We demonstrate the properties of $T$-measures with multiple synthetic, human, and algorithmic examples.

## 2. SEGMENT BOUNDARY EVALUATION

Segmentation algorithms are typically evaluated for two distinct goals. The first goal, *boundary detection*, evaluates the algorithm's ability to detect the times of transitions between segments. The second goal, *structural grouping*, evaluates the labeling applied to the estimated segmentation, and thus quantifies the ability of an algorithm to detect repeated forms, such as verses or refrains. In this paper, we focus exclusively on the boundary detection task.

Boundary estimates are typically evaluated by precision and recall [10]. Estimated and reference boundaries are

matched within a specified tolerance window — typically either 0.5 or 3 seconds — and the hit rate $n_h$ (number of matches) is used to define precision and recall scores:

$$P := \frac{n_h}{n_e}, \qquad R := \frac{n_h}{n_r}, \tag{1}$$

where $n_e$ and $n_r$ denote the number of boundaries in the estimated and reference annotations, respectively. $P$ and $R$ are typically combined into a single $F$-measure by computing their harmonic mean.

Boundary detection has also been evaluated by *deviation* [10]. This is done by measuring the median time (absolute) differential between each reference boundary and the nearest estimated boundary (*R2E*), and vice versa (*E2R*). Boundary deviation is useful for quantifying the temporal accuracy of a detection event. However, it can be sensitive to the number of estimated boundaries.

### 2.1 The limitations of flat evaluation

The precision-recall paradigm has been critical to quantifying improvements in segmentation algorithms, but it has numerous limitations with hierarchical annotations. The most obvious limitation is that both the reference and estimated annotations must have flat structure. This is sometimes resolved by collecting multiple flat reference annotations for each track, each corresponding to different levels of analysis [8].

When only the estimation is flat, it is still not obvious how to compute accuracy against multiple layers. Aggregating reference boundaries across layers prior to evaluation would imply that all boundaries are equally informative. However, high-level boundaries often convey more information about the overall structure of the piece, but their contribution to the total score may be diluted by the abundance of low-level boundaries, which necessarily outnumber high-level boundaries in hierarchical annotations.

Flat evaluation followed by aggregation across layers can be similarly problematic, since it discards the relational structure between layers in the reference annotation. This can complicate interpretation of the scores by conflating inaccurate boundary detection with mismatch between the target levels of the estimate and reference annotations [9].

Finally, the above strategies provide no means to directly compare two hierarchical annotations. While one may imagine simple comparison strategies when both hierarchies have a small number of layers with an obvious layer-wise correspondence — *e.g.*, SALAMI's *large*- and *small*-scale annotations — it is unclear how to proceed in more general settings.

## 3. THE TREE MEASURES

In this section, we derive the *tree measures* for evaluating multi-level segment boundary detection. The evaluation is based on a reduction to ranking evaluation, which we describe in detail below.

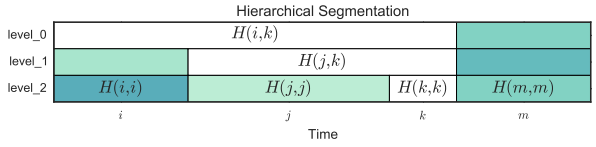



**Figure 1**: An example of a three-level hierarchical segmentation. Frames $i, j$, and $k$ are indicated along the $x$-axis, and their containing segments are indicated within the figure, *e.g.*, $H(j, k)$.

### 3.1 Preliminaries

Let $X$ denote a set of sample frames generated from the track at some fixed resolution $f_r$ (*e.g.*, 10Hz).[1] Let $S$ denote a flat, temporally contiguous partition of $X$, and let $S(i)$ identify the segment containing the $i$th frame in $X$. We will use the subscripts $S_R$ and $S_E$ to denote *reference* and *estimated* annotations, respectively.

A *hierarchical segmentation* $H$ is defined as a tree of flat segmentations $(S^0, S^1, \ldots, S^d)$ where each layer is a *refinement* of the preceding layer.[2] Let $H(i, j)$ identify the smallest (most refined) segment containing frames $i$ and $j$. We will denote precedence (containment) of segments by $\prec$: *e.g.*, $H(j, k) \prec H(i, k)$. Note that flat segmentations are a special case of hierarchical segmentations, where there are only two levels of segmentation, and the first layer contains no boundaries.

As illustrated in Figure 1, hierarchical segmentations can be represented as tree structures. Here, $H(i, i), H(j, j)$ and $H(k, k)$ denote the most specific segments containing frame $i, j$ and $k$, respectively. From the figure, we observe that $H(j, k)$ identifies the least common ancestor of frames $j$ and $k$. We can generally infer membership and precedence relations from the hierarchy, *e.g.*,

$$j \in H(j, j) \prec H(j, k) \prec H(i, j) = H(i, k). \tag{2}$$

### 3.2 Flat segmentation and bipartite ranking

Segmentation evaluation can be reduced to a ranking evaluation problem as follows. Let $q$ denote an arbitrary frame, and let $i$ and $j$ denote any two frames such that $S_R(q) = S_R(i)$ and $S_R(q) \neq S_R(j)$. In this case, $i$ may be considered *relevant* for $q$, and $j$ is considered *irrelevant*. This leads to the following per-frame recall metric:

$$f(q; S_E, S_R) := \sum_{\substack{i \in S_R(q) \setminus \{q\}, \\ j \notin S_R(q)}} \frac{[\![ S_E(q) = S_E(i) \neq S_E(j) ]\!]}{Z_q} \tag{3}$$

$$Z_q := (|S_R(q)| - 1) \cdot (n - |S_R(q)| + 1),$$

where $[\![\cdot]\!]$ is the indicator function, $n = |X|$ denotes the total number of frames, and $Z_q$ counts the number of terms in the summation. The score for frame $q$ is the fraction of

[1] Non-uniform samplings (*e.g.*, beat- or onset-aligned samples) are also easily accommodated.

[2] A partition $S^{i+1}$ is a refinement of partition $S^i$ if each member of $S^{i+1}$ is contained within exactly one member of $S^i$.

pairs $(i, j)$ for which $S_E$ agrees with $S_R$ with respect to $q$. Averaging over all $q$ yields a mean recall score:

$$\rho(S_E, S_R) := \frac{1}{n} \sum_q f(q; S_E, S_R). \tag{4}$$

### 3.3 Hierarchies and partial ranking

Equation (3) is defined in terms of segment membership equality, but it has a straightforward generalization to hierarchical segmentations. If we restrict attention to a query sample $q$, then $H(q, \cdot)$ induces a partial ranking over the remaining samples. Frames contained in $H(q, q)$ are considered maximally relevant, followed by those in $H(q, q)$'s immediate ancestor, and so on.

Rather than compare frames $q$, $i$, and $j$ where $S(q) = S(i) \neq S(j)$, we can instead compare where $H(q, i) \prec H(q, j)$: *i.e.*, the pair $(q, i)$ merge deeper in the hierarchy than do $(q, j)$. This leads to the following generalization of Equation (3):

$$g(q; H_E, H_R) := \sum_{\substack{(i,j),\\ i \neq q,\\ H_R(q,i) \prec H_R(q,j)}} \frac{[\![H_E(q, i) \prec H_E(q, j)]\!]}{Z_q}, \tag{5}$$

where $Z_q$ is suitably modified to count the number of terms in the summation. This definition is equivalent to Equation (3) for flat hierarchies, but it applies more generally to hierarchies of arbitrary (and unequal) depth.

Just as in Equation (3), $g$ can be viewed as a classification accuracy of correctly predicting pairs $(i, j)$ as positive ($q$ and $i$ merge first) or negative ($q$ and $j$ merge first). Ties ($H(q, i) = H(q, j)$) are precluded by the strict precedence operator in the summation. Equation (5) can be alternately be viewed as a generalized area under the curve (AUC) over the partial ranking induced by the hierarchical segmentation, where depth within the estimated hierarchy $H_E$ plays the role of the detection threshold.

Averaging over $q$ yields the *tree-recall* $T$-measure:

$$\mathcal{T}_R(H_E, H_R) := \frac{1}{n} \sum_q g(q; H_E, H_R). \tag{6}$$

The *tree-precision* metric $\mathcal{T}_P(H_E)$ is defined analogously by swapping the roles of $H_E$ and $H_R$:

$$\mathcal{T}_P(H_E, H_R) := \mathcal{T}_R(H_R, H_E). \tag{7}$$

Intuitively, $\mathcal{T}_R$ measures how many triplets generated by the reference $H_R$ can be found in the estimate $H_E$, while $\mathcal{T}_P$ computes the converse. The $T$-measures retain interpretation as recall and precision scores, albeit at the level of frame triplets rather than boundaries. Finally, an analogous $F$-measure $\mathcal{T}_F$ can be defined in the usual way by computing the harmonic mean of $\mathcal{T}_P$ and $\mathcal{T}_R$.

### 3.4 Windowing in Time

The $T$-measures defined above capture the basic notion of hierarchically nested, frame-level relevance, but they pose three technical limitations. First, the score for each query will generally depend on the track duration $n$, which makes comparisons between tracks of differing length problematic. Second, for large values of $n$ (long tracks), Equation (5) can be dominated by trivial comparisons where $j$ lies far from $q$ in time, *i.e.*, $|q - i| \ll |q - j|$. Longer tracks will produce inflated scores compared to shorter tracks, simply by having more "easy" comparisons. Finally, the calculation of Equation (6) can be expensive, taking $\mathcal{O}(n^3)$ time using a direct implementation.

To resolve these issues, we introduce a time window of $w$ seconds to both simplify the calculation of the metric and normalize its range. This is achieved by restricting the triples $(q, i, j)$ in the summation such that $i$ and $j$ both lie within a window of $w$ seconds centered at $q$. Adding this windowing property to equations (5, 6) yields the windowed $T$-measures:

$$g(q; H_E, H_R, w) := \sum_{\substack{i,j \in \{x : |q-x| \leq w/2\}\\ i \neq q,\\ H_R(q,i) \prec H_R(q,j)}} \frac{[\![H_E(q, i) \prec H_E(q, j)]\!]}{Z_q(w)}, \tag{8}$$

$$\mathcal{T}_R(H_E, H_R; w) := \frac{1}{n} \sum_q g(q; H_E, H_R, w), \tag{9}$$

and $Z_q(w)$ is again modified to count the terms in the summation. This reduces computational complexity from $O(n^3)$ to $\mathcal{O}(nw^2)$. Each query frame $q$ now operates over a bounded number of comparisons, so the windowed $T$-measures are calibrated across tracks of different lengths. This property is useful when compiling score statistics over a test collection.

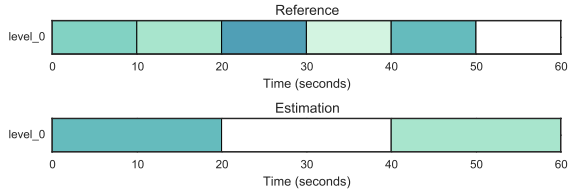
### 3.5 Transitive reduction

Just as Equation (5) can be dominated by long-range interactions in the absence of windowing, deep hierarchies can also pose a problem. To see this, consider the sequence $H_R(q, i) \prec H_R(q, j) \prec H_R(q, k)$. Since the summation in Equation (5) ranges over all precedence comparisons, and $i \in H_R(q, j)$, the triple $(q, i, k)$ is double-counted. Since segments grow in size at higher levels in the hierarchy, over-counting can dominate the evaluation.

To counteract this effect, the summation can be restricted to include only direct precedence relations. This is accomplished by comparing samples only from successive levels in the hierarchy, *i.e.*, replacing the partial ranking generated by $q$ with its transitive reduction. This both eliminates redundant comparisons and increases $g$'s effective range. We refer to the resulting metrics as *reduced* $T$-measures.

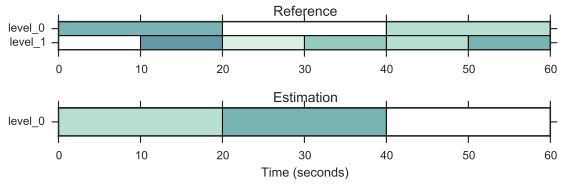
## 4. SYNTHETIC EXAMPLES

In this section we discuss the behavior of the $T$-measures by showing various synthetic examples, and comparing them against other existing methods when possible. For each example in this section, we illustrate the behavior of our proposed metric under different window times $w$. This section is subdivided by the types of annotations under consideration.



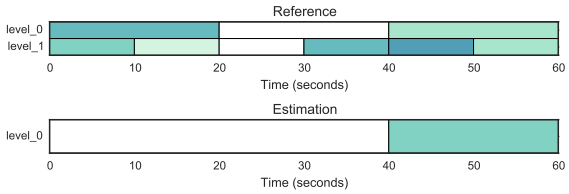

| | $T$-measures | | Hit Rate | |
|---|---|---|---|---|
| $w$ | $\mathcal{T}_R$ | $\mathcal{T}_P$ | $R$ | $P$ |
| 0.5 | 0.40 | 1.00 | 0.40 | 1.00 |
| 3 | 0.40 | 1.00 | 0.40 | 1.00 |
| 15 | 0.39 | 0.53 | | |
| 30 | 0.69 | 0.50 | | |
| $\infty$ | 0.80 | 0.50 | | |

**Figure 2***:* Flat vs. flat boundaries (top), $T$-measures and boundary detection (hit-rate) scores (bottom).




| | Reduced | | Full | |
|---|---|---|---|---|
| $w$ | $\mathcal{T}_R$ | $\mathcal{T}_P$ | $\mathcal{T}_R$ | $\mathcal{T}_P$ |
| 0.5 | 0.00 | 1.00 | 0.40 | 1.00 |
| 3 | 0.00 | 1.00 | 0.40 | 1.00 |
| 15 | 0.37 | 1.00 | 0.51 | 1.00 |
| 30 | 0.70 | 1.00 | 0.82 | 1.00 |
| $\infty$ | 0.80 | 1.00 | 0.89 | 1.00 |

**Figure 3***:* Hierarchical reference vs. flat (large-scale) estimation (top) and $T$-measures (bottom). *Reduced* uses the transitive reduction method of section 3.5, while *Full* uses comparisons across all layers.




| | Reduced | | Full | |
|---|---|---|---|---|
| $w$ | $\mathcal{T}_R$ | $\mathcal{T}_P$ | $\mathcal{T}_R$ | $\mathcal{T}_P$ |
| 0.5 | 0.0 | 1.00 | 0.20 | 1.00 |
| 3 | 0.0 | 1.00 | 0.20 | 1.00 |
| 15 | 0.19 | 0.94 | 0.26 | 0.94 |
| 30 | 0.37 | 0.71 | 0.44 | 0.71 |
| $\infty$ | 0.53 | 0.67 | 0.59 | 0.67 |

**Figure 4***:* Hierarchical reference vs. flat under-segmentation (top) and $T$-measures (bottom).

### 4.1 Flat vs. flat annotations

We first compare two flat boundary annotations to demonstrate how the $T$-measures behave compared to standard boundary detection. When both annotations are flat, the reduced $T$-measures behave identically to the full measures, so we omit them from this section. The synthesized flat boundaries are displayed on the top of Figure 2, and they aim to capture a situation where an algorithm correctly detects a subset of the reference boundaries.

The hit rate scores obtain a recall of 0.40 and a precision of 1.0, since all estimated boundaries are also in the reference, but only two out of five boundaries were retrieved. [3] When $w$ does not exceed the minimum segment duration, the $T$-measures coincide exactly with the boundary detection metrics. For larger $w$, $\mathcal{T}_P$ decreases, while $\mathcal{T}_R$ increases as $w$ approaches the track duration. The dependency on $w$ is further explored in Section 5.1.

To understand the relationship between $\mathcal{T}_P$ and $w$, consider the example $(q, i, j) = (5, 15, 25)$. The estimation considers $i$ to be relevant for $q$ (since they belong to the segment $[0, 20]$), and $j$ to be irrelevant for $q$. Meanwhile, the reference considers both $i$ and $j$ to be equally irrelevant for $q$, so this triple contributes 0 to the precision metric. Note that this comparison is counted only when $w$ is large enough to span multiple segments.

In general, sensitivity to long-range interactions increases with $w$. This illustrates how the window size depends on the duration and scale of structure that the practitioner wishes to capture.

### 4.2 Flat vs. hierarchical annotations

Here we present four examples of flat estimations against a fixed hierarchical reference, but note that the reverse comparisons can be inferred by swapping $T_P$ and $T_R$.

#### *4.2.1 Large-scale and under-segmentation*

Figure 3 illustrates a flat estimation corresponding to the highest layer of a hierarchical reference. We report $T$-measures with and without the transitive reduction strategy described in Section 3.5. The $T$-measures behave as expected: the tree-precision score $\mathcal{T}_P$ is always 100%, since the reference contains the estimation. We also observe the general trend that *full* scores exceed *reduced* scores.

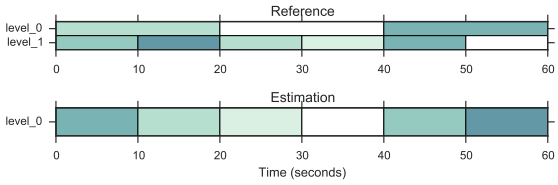
For small time windows ($w \leq 3$), the full tree-recall score is 40%, just as in the previous example. The *reduced* recall scores in this case are 0 because no frame $q$ in the estimation has two frames $i, j$ both within $w \leq 3$ seconds that merge within one layer of each-other in the reference.

Figure 4 illustrates an example of under-segmentation: the estimation misses a high-level structural change at 20s. Again, small $w$ yields $T$-measures which coincide with standard boundary detection metrics. Larger $w$ increases the tree-recall (and decreases precision) since only long-range interactions are well represented in the estimation.

#### *4.2.2 Small-scale and over-segmentation*

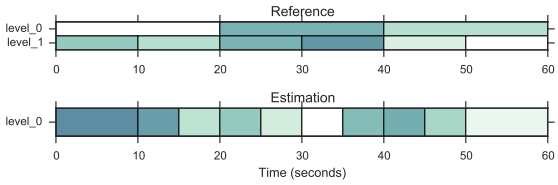
Figure 5 illustrates an example comparable to Figure 3, except that the estimation now corresponds to the bottom layer of the reference annotation. Again, since the reference contains the estimation, precision is maximal for all $w$. However, the reference provides strictly more informa-

[3] The first and last boundaries (0 and 60s) mark the beginning and end of the track, and since they are constant across all estimates, we suppress them during the evaluation to avoid score inflation.




| | Reduced | | Full | |
|---|---|---|---|---|
| $w$ | $\mathcal{T}_R$ | $\mathcal{T}_P$ | $\mathcal{T}_R$ | $\mathcal{T}_P$ |
| 0.5 | 1.00 | 1.00 | 1.00 | 1.00 |
| 3 | 1.00 | 1.00 | 1.00 | 1.00 |
| 15 | 0.63 | 1.00 | 0.76 | 1.00 |
| 30 | 0.30 | 1.00 | 0.59 | 1.00 |
| $\infty$ | 0.20 | 1.00 | 0.55 | 1.00 |

**Figure 5***:* Hierarchical reference vs. flat, small-scale estimation (top) and $T$-measures (bottom).




| | Reduced | | Full | |
|---|---|---|---|---|
| $w$ | $\mathcal{T}_R$ | $\mathcal{T}_P$ | $\mathcal{T}_R$ | $\mathcal{T}_P$ |
| 0.5 | 1.00 | 0.56 | 1.0 | 0.56 |
| 3 | 0.98 | 0.56 | 0.98 | 0.56 |
| 15 | 0.46 | 0.86 | 0.53 | 0.86 |
| 30 | 0.22 | 0.92 | 0.40 | 0.92 |
| $\infty$ | 0.13 | 0.94 | 0.37 | 0.94 |

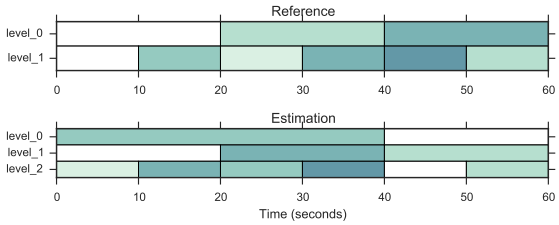
**Figure 6***:* Hierarchical reference vs. flat over-segmentation (top) and $T$-measures (bottom).

tion: namely, it encodes structure over the low-level segments. The $T$-measures quantify the missing information in the estimation When $w$ exceeds the smallest segment duration (10s), $\mathcal{T}_R$ decreases. This information would be obscured by independent, layer-wise boundary evaluation.

Similarly, Figure 6 illustrates an *over-segmentation* where the estimation predicts more boundaries than the deepest layer of the reference. Again, the $\mathcal{T}_R$ decays when the window captures multiple short segments. Unlike the under-segmented example in Figure 4, long-range interactions derived from $H_E$ are mostly satisfied by $H_R$, so $\mathcal{T}_P$ increases rather than decreases.

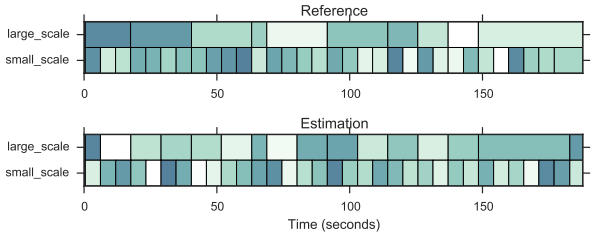
### 4.3 Hierarchical vs. hierarchical

Figure 7 compares two different hierarchical segmentations. The estimation contains an additional high-level layer, but is otherwise identical to the reference. At small $w$, both $T$-measures agree perfectly, since the window is not large enough to resolve differences. As $w$ increases, $\mathcal{T}_P$ decreases as expected, since the estimation has found an additional structural element not captured in the reference. The $\mathcal{T}_R$ scores remain at 100% for all $w$.




| | Reduced | | Full | |
|---|---|---|---|---|
| $w$ | $\mathcal{T}_R$ | $\mathcal{T}_P$ | $\mathcal{T}_R$ | $\mathcal{T}_P$ |
| 0.5 | 1.00 | 1.00 | 1.00 | 1.00 |
| 3 | 1.00 | 1.00 | 1.00 | 1.00 |
| 15 | 1.00 | 0.98 | 1.00 | 0.99 |
| 30 | 1.00 | 0.79 | 1.00 | 0.89 |
| $\infty$ | 1.00 | 0.62 | 1.00 | 0.79 |

**Figure 7***:* 2-layer vs. 3-layer hierarchical boundaries (top) and $T$-measures scores (bottom).




| | Reduced | | Full | |
|---|---|---|---|---|
| $w$ | $\mathcal{T}_R$ | $\mathcal{T}_P$ | $\mathcal{T}_R$ | $\mathcal{T}_P$ |
| 0.5 | 0.76 | 0.77 | 0.81 | 0.79 |
| 3 | 0.95 | 0.95 | 0.96 | 0.93 |
| 15 | 0.75 | 0.75 | 0.80 | 0.84 |
| 30 | 0.62 | 0.83 | 0.71 | 0.89 |
| $\infty$ | 0.57 | 0.96 | 0.68 | 0.98 |

**Figure 8***:* Hierarchical annotations for SALAMI track #636 from the two different human annotators. Top: annotations; bottom: $T$-measures scores.

## 5. LARGE-SCALE EVALUATION

In this section, we apply the $T$-measures to quantify inter-annotator agreement in the SALAMI corpus, and evaluate the hierarchical predictions of the agglomerative clustering method (OLDA) of McFee and Ellis [4].

### 5.1 Human annotator agreement

Figure 8 illustrates hierarchical annotations obtained from two human annotators on one track in the SALAMI dataset. While the two annotators tend to agree at the small scale, they differ at the large scale. This is reflected in the $T$-measures: at large $w$, the recall skews low because the reference's large-scale annotations are coarser than those of the estimation.

To further investigate inter-annotator agreement, we computed $T$-measure scores between hierarchical reference annotations for the 410 tracks in the SALAMI dataset where two annotations are available and both mark the start and end times of the song equally at both levels. To simplify exposition, we summarize agreement by $\mathcal{T}_F$. Figure 9

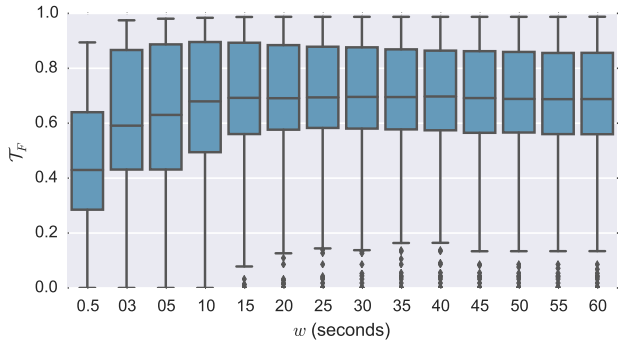



**Figure 9**: $\mathcal{T}_F$ scores between human annotators for SALAMI tracks over a range of window sizes $w$.

illustrates the distribution of per-track $\mathcal{T}_F$ scores as a function of $w$. We observe that the score distribution is relatively stable for $w \geq 15$.[4] The example in Figure 8 is generally representative of inter-annotator agreement, achieving $\mathcal{T}_F = 0.75$ at $w = 15$. The out-lying low scores tend to be examples where one annotator ignored structure annotated by the other: *e.g.*, in track #68, one annotator only marked *silence* boundaries.

This analysis quantitatively substantiates prior observations that humans do not perfectly agree upon structural annotations [9], and suggests an accuracy ceiling near 70% for hierarchical annotation. Similarly, it suggests that $w = 15$ provides a reasonable default value for the SALAMI dataset. This setting is large enough to capture multiple small-scale segments: in the tracks considered for this evaluation, the median small-scale segment duration was 6.66s, with a 95th percentile of 15.69s.

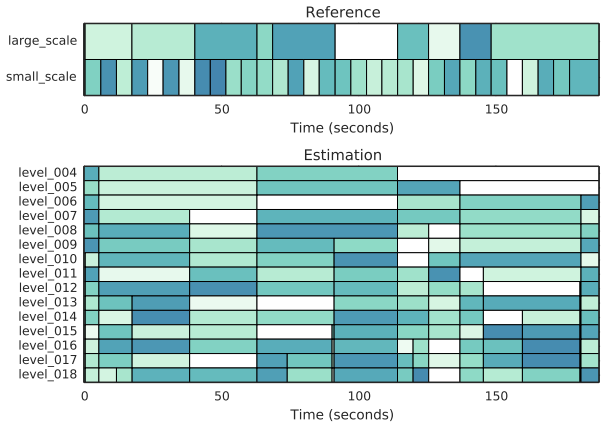
### 5.2 Annotator vs. algorithm

Finally, we evaluated the quality of hierarchical segmentations produced by OLDA [4].[5] Figure 10 illustrates one example output of OLDA and the resulting $T$-measures. The reference provides two levels of segmentation (large and small), while the estimation produces several layers with generally large segments. For sufficiently large $w$, the estimation achieves high recall and low precision. This behavior is typical of the OLDA method, which constructs hierarchies in a bottom-up fashion by agglomerative clustering, adding only a single boundary at each layer. Due to the depth of the estimated boundaries, the *full* scores are inflated compared to the *reduced* scores.

Figure 11 displays the $\mathcal{T}_F$ score distribution for OLDA, measured against annotator 1 on 726 tracks from SALAMI. These results reveal a gap of around 30% between inter-annotator agreement (Figure 9) and the performance of OLDA. This suggests that there is substantial room for improvement in hierarchical boundary estimation algorithms.

[4] The analogous plots for $\mathcal{T}_P$ and $\mathcal{T}_R$ are omitted for brevity, but illustrate the same trend.

[5] To the authors' knowledge, this is the only published method for hierarchical boundary detection.




| | Reduced | | Full | |
|---|---|---|---|---|
| $w$ | $\mathcal{T}_R$ | $\mathcal{T}_P$ | $\mathcal{T}_R$ | $\mathcal{T}_P$ |
| 0.5 | 0.14 | 1.00 | 0.28 | 0.55 |
| 3 | 0.20 | 1.00 | 0.34 | 0.72 |
| 15 | 0.62 | 0.56 | 0.66 | 0.70 |
| 30 | 0.76 | 0.53 | 0.80 | 0.58 |
| $\infty$ | 0.90 | 0.16 | 0.93 | 0.42 |

**Figure 10**: Hierarchical reference annotation vs. OLDA on SALAMI track #636. (top) and $T$-measures (bottom).

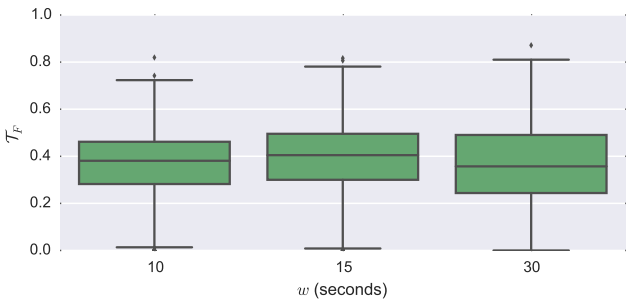



**Figure 11**: $\mathcal{T}_F$ scores between OLDA and human reference annotations on the SALAMI dataset.

## 6. DISCUSSION AND CONCLUSIONS

The implementation of $T$-measures depends upon two critical parameters: the time window $w$, and whether to use the *reduced* or *full* metrics. While the setting of $w$ ultimately depends upon the practitioner's preference and characteristics of the dataset, the results on SALAMI suggest that $w = 15$ provides a reasonable balance between capturing high-level structure and resilience to long-range interactions. As illustrated in section 4.2.1, when $w$ is large enough to capture multiple short segments, the transitive reduction approach can also be used to enhance the range of the metrics while eliminating redundant comparisons.

In this paper, we focused only on the problem of evaluating estimated boundaries. In future work, we plan to extend general ideas behind $T$-measures to other structural annotation problems, such as segment label agreement.

## 7. ACKNOWLEDGEMENTS

BM acknowledges support from the Moore-Sloan Data Science Environment at NYU.

## 8. REFERENCES


[1] Tom Collins. Discovery of Repeated Themes & Sections, 2013.

[2] Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, 1983.

[3] Fred Lerdahl and Ray Jackendoff. An Overview of Hierarchical Structure in Music. *Music Perception: An Interdisciplinary Journal*, 1(2):229–252, 1983.

[4] Brian McFee and Daniel P. W. Ellis. Learning to Segment Songs with Ordinal Linear Discriminant Analysis. In *Proc. of the 39th IEEE International Conference on Acoustics Speech and Signal Processing*, Florence, Italy, 2014.

[5] Oriol Nieto and Morwaread M. Farbood. Identifying Polyphonic Patterns From Audio Recordings Using Music Segmentation Techniques. In *Proc. of the 15th International Society for Music Information Retrieval Conference*, pages 411–416, Taipei, Taiwan, 2014.

[6] Jouni Paulus, Meinard Müller, and Anssi Klapuri. Audio-Based Music Structure Analysis. In *Proc of the 11th International Society of Music Information Retrieval*, pages 625–636, Utrecht, Netherlands, 2010.

[7] Geoffroy Peeters and Emmanuel Deruty. Is Music Structure Annotation Multi-Dimensional? A Proposal for Robust Local Music Annotation . In *Proc. of the 3rd International Worskhop on Learning Semantics of Audio Signals*, pages 75–90, Graz, Austria, 2009.

[8] Jordan B. Smith, J. Ashley Burgoyne, Ichiro Fujinaga, David De Roure, and J. Stephen Downie. Design and Creation of a Large-Scale Database of Structural Annotations. In *Proc. of the 12th International Society of Music Information Retrieval*, pages 555–560, Miami, FL, USA, 2011.

[9] Jordan B. L. Smith and Elaine Chew. A Meta-Analysis of the MIREX Structure Segmentation Task. In *Proc. of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil, 2013.

[10] Douglas Turnbull, Gert RG Lanckriet, Elias Pampalk, and Masataka Goto. A supervised approach for detecting boundaries in music using difference features and boosting. In *ISMIR*, pages 51–54, 2007.

[11] GeraintA. Wiggins and Jamie Forth. Idyot: A computational theory of creativity as everyday reasoning from learned information. In Tarek R. Besold, Marco Schorlemmer, and Alan Smaill, editors, *Computational Creativity Research: Towards Creative Machines*, volume 7 of *Atlantis Thinking Machines*, pages 127–148. Atlantis Press, 2015.