2021

Automatic Hierarchy Expansion for Improved Structure and Chord Evaluation

Katherine M. Kinnaird
Smith College, kkinnaird@smith.edu

Brian McFee
New York University

Recommended Citation

Kinnaird, K.M. and McFee, B., 2021. Automatic Hierarchy Expansion for Improved Structure and Chord Evaluation. Transactions of the International Society for Music Information Retrieval, 4(1), pp.81–92. DOI: http://doi.org/10.5334/tismir.71

TISMIR

Kinnaird, K. M., & McFee, B. (2021). Automatic Hierarchy Expansion for Improved Structure and Chord Evaluation. *Transactions of the International Society for Music Information Retrieval,* 4(1), pp. 81–92. DOI: https://doi.org/10.5334/tismir.71

RESEARCH

Automatic Hierarchy Expansion for Improved Structure and Chord Evaluation

Katherine M. Kinnaird* and Brian McFee†

Partitioning a recording into non-overlapping time intervals comes in many forms. There is the structural segmentation task which labels structures either syntactically as *A* and *B*, or structurally as *verse* or *chorus*. The chord annotation task is similar, labeling segments by their chords. While many of these annotations are flat, this article extends the method by McFee and Kinnaird (2019) for automatically enhancing structural annotations by inferring (and expanding) hierarchical information from the segment labels. One of our extensions adds new rules that allow for structural labels with a wider vocabulary than the syntactical ones in the SALAMI dataset. Using this first extension, we compare annotations from the Beatles-TUT and Isophonics datasets to investigate similarities between these annotations. Our second extension creates a multi-level annotation for chords that addresses a number of current challenges in chord evaluation. Using a large collection of chord annotations (manually and automatically generated), we investigate how and where the multi-level hierarchies can enhance (or detract from) comparing chord annotations.

Keywords: Structure segmentation; chord recognition; hierarchy; evaluation

1. Introduction

In the music information retrieval (MIR) literature, the issue of partitioning recordings into labeled segments is well studied. Whether those labels are chord annotations or structural annotations (such as *verse* and *chorus*), musical structure analysis broadly concerns methods for automatically inferring relationships between moments in time within a piece (Dannenberg and Goto, 2008; Paulus et al., 2010). Much of the computational work in this area models musical structure as having exactly one partition of the recording. The resulting segments of these flat annotations are neither merged nor subdivided to form larger or smaller structures.

This restrictive view on partitioning a piece can lead to a number of challenges in both structural segmentation and chord annotations. In both tasks, there can be differences due to levels of expertise. For example in the structural segmentation task, expert annotators may encode latent hierarchical information by using variation markers in their segment labels, *e.g.*, *A*, ..., *A′* or *verse, ..., verse_ (instrumental)* (Smith et al., 2011; Paulus and Klapuri, 2006). Similar issues of ambiguity and subjectivity exist in the chord annotation task that can be further complicated due to differing chord vocabularies (Pauwels et al., 2019). These variation markers or increasingly complicated chord grammars may be clearly informative, but these nuances are often overlooked by standard comparison methods. McFee and Kinnaird (2019) created an automatic hierarchy expansion that sought to leverage the inherent yet latent hierarchical structure in these annotations. This automatic hierarchy expansion is built on the recent trend of developing datasets (Smith et al., 2011; Nieto and Bello, 2016), computational methods (Ullrich et al., 2014), representations (McGuirl et al., 2018), and evaluation criteria (McFee et al., 2017) for hierarchically structured music segmentations. This article continues in this tradition, by extending the automatic hierarchy expansion by McFee and Kinnaird (2019) to further examples in segmentation.

1.1 Our contributions

In this work, we apply the idea of automatic hierarchy expansion from McFee and Kinnaird (2019) to segmentation problems where the labels convey not only syntactic, but "semantic" information as well. Specifically, we develop methods for inferring latent hierarchical structure from segmentation annotations (such as *verse*, *chorus*, *bridge*, etc.), and from chord annotations, exposing hierarchical relations among related chords (e.g., `C:maj` and `C:7`). For the structure example, we compare two collections of human-generated annotations of the Beatles dataset: Beatles-TUT (Paulus, 2010) and the Isophonics structural segmentations (Harte, 2010). For this example, we follow the procedure from McFee and Kinnaird (2019) for creating the automatic hierarchy expansions, with minor

* Smith College, 44 College Lane, Northampton MA, US

† New York University, 60 5th Ave., New York, NY, US

Corresponding author: Katherine M. Kinnaird (kkinnaird@smith.edu)

edits due to differences in the segmentation vocabulary when compared to the vocabulary in SALAMI (Smith et al., 2011).

For the chord annotations, we develop an automatic hierarchy construction scheme based on iterative simplification of chords in Section 3.2. This progressive simplifying of the chord annotations combined with the original annotation creates a hierarchical annotation, which can then be compared to other hierarchies using existing techniques. This application of hierarchy expansion results in a new approach for comparing chord annotations which accounts for similarity between the internal structure of the annotations. To demonstrate the method, we compare the resulting structural similarity derived from automatically induced chord hierarchies to standard chord recognition evaluation metrics over a large corpus of previous published annotations.

2. Related work

This paper concerns two issues: 1) adding flexibility to the automatic hierarchy expansion for use on structural labels in the structural segmentation task, and 2) extending the automatic hierarchy expansion for the chord annotation task. In both cases, we are concerned with hierarchical evaluation methods. In the latter case, we seek to address a number of challenges in the chord annotation task.

2.1 Hierarchical evaluation

Recent work by McFee and Kinnaird (2019) created an automatic hierarchy expansion on flat (i.e., single-level) structure annotations for simple form structure labels, such as *A*, *B*, *A′*, etc. These hierarchies allowed for a more nuanced comparison between annotations by exploiting the latent structure inherent in variation markers such as a section labeled with *A′* compared to an *A* section versus comparing a section with the label *A′* to another labeled as *B*. The contribution of a multi-level evaluation for segmentation is an important one, but McFee and Kinnaird (2019) did not address structural labels such as *verse* or *chorus*. The first contribution of this article extends the automatic hierarchy expansion to structural labels.

For the sake of consistency, we adopt the notation and conventions of McFee and Kinnaird (2019). That is for a signal of duration T, we define a (flat) *segmentation* as a function $S : [0, T] \rightarrow V$ where V denotes a set of segment labels, e.g., $V = \{A, B, \ldots\}$. A *multi-level segmentation* (or *hierarchy*) is defined as a sequence of segmentations $H = (S_0, S_1, \ldots)$, where S_0 maps to a single label, and subsequent segmentations S_i are ordered from *coarse* to *fine*. We assume that each segmentation S_i maps to a distinct vocabulary.

Like McFee and Kinnaird (2019), the approach taken in this work is based on the *L*-measure method for multi-level segmentation comparison (McFee et al., 2017). Specifically, given two hierarchies H^R (the reference) and H^E (the estimate), we compare them by using the *L*-Measure, which is the harmonic mean of *L*-precision and *L*-recall. Both the precision and recall scores are defined by comparing two collections of time instant triplet sets $A(H^E)$ and $A(H^R)$. The time instance triplets encode t, u, v such that the depth that instances t and u share is deeper than the depth that t and v share. Specifically, if two time instants t and u are both contained in the same segment, while another time instant v lies outside that segment, then (t, u, v) would constitute an element of the set derived from the segmentation. Similarity between segmentations—and by extension, multi-level segmentations—is derived by counting the proportions of triplets shared between the derived sets. Specifically, the precision score is defined as:

$$\text{L-Precision}\left(H^R, H^E\right) := \frac{\left|A\left(H^R\right) \cap A\left(H^E\right)\right|}{\left|A\left(H^E\right)\right|}. \qquad (1)$$

Recall is defined analogously by reversing the roles of reference and estimate. Typically, the terms *reference* and *estimate* distinguish between annotations derived from the method's use in comparing algorithm outputs to manual annotations. Using the *L*-measure instead of either *L*-precision or *L*-recall removes the need to confer privileged status to one of two different annotations.[1]

2.2 Chord evaluation

Automatic chord estimation is a long-standing problem in music information retrieval. The recent survey by Pauwels et al. (2019) provides a comprehensive overview of the topic, and highlights several outstanding challenges for chord recognition research. The first three challenges—1. finding an appropriate feature representation; 2. describing what a chord looks like in the feature space; and 3. the mismatch between [audio] processing rate and chord rate—relate primarily to signal processing, and are beyond the scope of the present work. The remaining four, however, all relate in one way or another to structural aspects of chords: 4. achieving long-term consistency in chord sequences; 5. exploiting relationships with related musical concepts; 6. handling ambiguity and subjectivity; and 7. chord vocabulary and associated balance problems. Our investigation of hierarchical structure analysis for chord evaluation aims to address (to varying extents) each of these challenges. By directly comparing the repetition structure of reference and estimated chord annotations, we provide a way to compare internal consistency of annotations over the entire track. The hierarchy construction we propose incorporates both the (implied) key of the piece encoded in the chord labels and bass (inversions) in a unified scheme, thereby exploiting similarity between hierarchically related chord labels. By comparing simultaneously across multiple simplifications of the chord annotation, we provide a metric which is robust to particular kinds of ambiguity and subjectivity, such as tuning disagreements. Finally, exploiting the grammar of chord labels to construct a hierarchical representation does not directly address class imbalance problems, but as noted by McFee and Bello (2017), it does simplify the problem of selecting a chord vocabulary, as any chord label which validates under the formal grammar of Harte et al. (2005) can be directly incorporated in the evaluation.

The standard chord evaluation metrics provided by `mir_eval` (Raffel et al., 2014)—itself based on the MIREX 2013 chord metrics (Bay et al., 2010) —provide ways to compare two chord annotations at varying degrees of specificity. Following the work of Pauwels and Peeters (2013), `mir_eval` provides, among others, the following evaluations:

roots	Requires enharmonic agreement only at the root of the chord, ignoring quality or bass. Example: *C*♯:min ≡ *D*♭:maj ≢ *C*:min.
thirds	Requires agreement only at the root and the third scale degree, ignoring other pitch classes. Example: *C*:maj ≡ *C*:aug ≢ *C*:min.
triads	As above, but including the fifth, and ignoring additional pitch classes. Example: *C*:maj ≡ *C*:7 ≢ *C*:aug.
sevenths	Compares the root, third, fifth, and seventh, ignoring above-octave extensions.[2] Example: *C*:9 ≡ *C*:7 ≢ *C*:maj7.
tetrads	Compares all 12 pitch classes (including the root).

Although these evaluations follow a clear hierarchical pattern of increasing specificity, they are calculated, normalized, and reported separately. While this is beneficial from the perspective of inspecting behavior and failure modes of an individual estimator, the lack of a unified metric presents a challenge for succinctly comparing different estimators, or getting a holistic measure of similarity between two annotations.

Finally, the chord segmentation metrics of Mauch (2010) and Harte (2010) are similar in spirit to the structural annotation approach we take here, but differ in a few critical ways. The chord segmentation metrics operate by computing the *directional Hamming distance* to determine the amount of over- or under-segmentation of the estimated chord annotation relative to the reference (and vice versa). However, this is performed purely based on the time interval data, and does not depend on the chord segment labels. The approach we take here has a similar inspiration, but inducing a hierarchical structure by simplifying chord labels provides a more detailed view of the segmentation problem. In effect, the proposed hierarchical approach can measure the extent to which it is possible to automatically simplify one chord annotation to match another, *e.g.*, by discarding inversions or simplifying upper extensions.

2.3 Learning from hierarchical labels

Many MIR tasks involve taxonomies or otherwise hierarchically structured labels. Most of the published research on these topics simplifies classification problems to flat 1-of-*K* formulations, which facilitates modeling by standard machine learning algorithms. However, there is a relatively small collection of works which build hierarchical structure directly into the learning problem (e.g., by modifying the training objective, model architecture, or both), such as for tagging or genre classification (Burred and Lerch, 2004); instrument recognition (Essid et al., 2005); and chord estimation (McFee and Bello, 2017; Carsault et al., 2018). Specifically in the case of chord estimation, it has been demonstrated that representing chord labels in a way that exposes structure—e.g., root and pitch classes (McFee and Bello, 2017), or hieararchical relations between qualities (Carsault et al., 2018)—can improve accuracy over unstructured token representations used by general-purpose classification methods. While these approaches exploit structure between labels during training, this is distinct from our focus in this work on *evaluation*.

3. Methods

This work proposes two extensions to the automatic hierarchy expansion by McFee and Kinnaird (2019). The first extends the application of the automatic hierarchy expansion for the structural segmentation task to include structural labels such as *verse* and *coda*, instead of just syntactical ones (such as *A*, *A′*, and *B*) which the original automatic hierarchy expansion restricted itself to.

The second extension of the automatic hierarchy expansion concerns chord annotations, expanding flat chord annotations into a hierarchy that contains progressively coarser chord information. The resulting expansion seeks to address four of the challenges identified by Pauwels et al. (2019) of chord recognition, as noted in the previous section.

3.1 Automatic Hierarchy Expansion for Structure Annotations

Building on the automatic hierarchical expansion for any 'flat' annotations proposed by McFee and Kinnaird (2019), our method still expands a flat annotation into a hierarchy with three levels but includes updates to accommodate the vocabularies in the Beatles-TUT (Paulus, 2010) and Isophonics (Harte, 2010) datasets. As defined by McFee and Kinnaird (2019), the first level is a *contraction* of variation markers, such as removing the '*A*' from a label '*VerseA*.' The second level is the original flat annotation. The third level is a *refinement* of the labels by making each instance of a label from the contraction level unique by adding counters to the label (either as numbers for syntactical labels or using repeated ′ for structural ones. Our extension of the automatic hierarchy expansion will demonstrate that one can apply these methods to structural labels.

For concrete examples, consider the two examples in **Figure 1**. Each shows a flat annotation on the left side, which is then repeated on the right side of the panel as the middle level of the resulting automatic hierarchy expansion. The contraction level, shown in green, removed the variation markers of the *A* repetition in the left panel, and in the right panel, it removed the variation markers of the *verse* label. In both cases, the result is that the contraction part of the hierarchy has two kinds of repetitions instead of three; *A* and *B* for the left example, and *verse* and *chorus* for the right example.

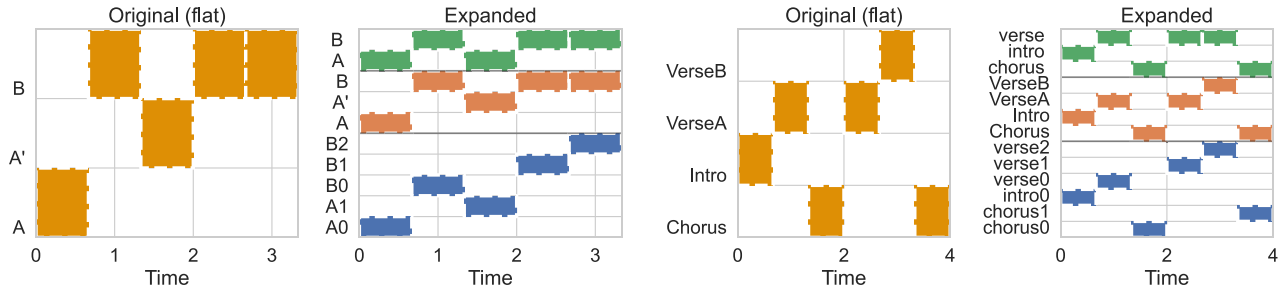

Figure 1: Two examples of automatic hierarchy expansion. In both examples, the contraction level (green, top) removes variation markers, while the refinement level (blue, bottom) adds counters to each instance of a segment label. The center level (orange) preserves the original annotation. The left example is a flat segmentation with segments (*A*, *B*, *A*′, *B*, *B*) expanded into a three-level hierarchy. The right example has a flat segmentation with structural labels (Intro, VerseA, Chorus, VerseA, VerseB, Chorus).

The refined level of the hierarchy for structure annotations, shown in blue, has at most one block per line. This refinement level is created directly from the contraction level of the hierarchy. For each instance of a syntactical label in the contraction level, we append a counter (starting with 0) to form a new label. For the structural labels (such as *verse* and *chorus*), we append a counter using the prime symbol (′) to form each new label. If instead we had conducted this refinement starting at the middle level, we would have ended up with the annotation labels {A0,A′0,B0,B1,B2} instead of {A0,A1,B0,B1,B2}. Similarly, in our second example with the structural labels, we have {chorus, chorus′, intro, verse, verse′, verse″} instead of {chorus, chorus′, intro, verseA, verseA′, verseB}. Creating the refined level from either the contraction level or from the original flat annotations produces equivalent results, but the former is easier to interpret. What is more, given the broader range of variations in structure labels for the Beatles-TUT and Isophonics datasets, building from the contraction level provides more constrained labels in the refined level than from the flat level.

We note that although many label contraction rules can be automatically defined (such as stripping variation markers), the general problem is non-trivial due to the unrestricted vocabularies used by annotators for structural segmentation. In this work, we employed a combination of automatic rules defined by regular expressions with manual corrections specific to each collection.[3]

3.2 Automatic Hierarchy Expansion for Chords

In this section, we apply the idea of automatic hierarchical expansion (McFee and Kinnaird, 2019) to 'flat' chord annotations. Like the case for structure, each level of our chord hierarchy is a contraction of the following one. This means that the original chord labels are at the deepest level of the hierarchy and that the levels above the original chord labels correspond to successive contractions (or simplifications) of the labels derived by discarding details. We will describe this succession using the concrete example shown in **Figure 2**. The lowest part of the image shows the original chord annotation, which is iteratively contracted to form the higher levels of the hierarchy. The first simplification normalizes enharmonic equivalences across keys to use only sharps (so *D*♭ becomes *C*♯). The

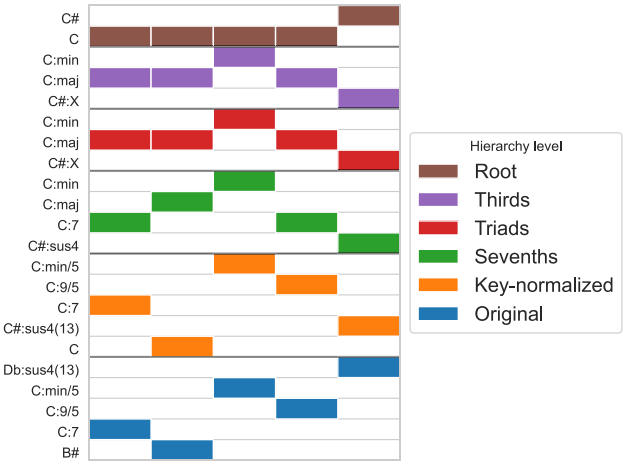

Figure 2: An example of automatic hierarchy expansion applied to chord annotations. The original (full detail) labels form the bottom layer of the hierarchy, and each successive layer represents simplification: key normalization, extension elimination, then simplification to triads, thirds, and finally the roots at the top of the hierarchy.

second simplification discards inversions, suppressed notes, and above-octave extensions. The third, fourth, and fifth simplifications discard the seventh, fifth, and third scale degrees (respectively). The result is a 6-layer hierarchy, where the top layer consists of only chord roots, and the bottom layer contains the original labels in full detail.

To support out-of-gamut chords, we deviate slightly from the notation of Harte et al. (2005), which encodes all out-of-gamut chords as the symbol `X`. This would discard root information, resulting in a premature merging of segments in the middle of the hierarchy. For example, at the *triads* level, `C:sus4` and `G:sus4` are both out of vocabulary, and would map to `X`, losing their root information. Instead, we retain root information for out-of-gamut chords, mapping instead to, *e.g.*, `C:X` or `G:X`. This allows us to preserve root information and retain a well-formed hierarchy.

Additionally, we propose a pruned version of the above hierarchy. Effectively in the pruned version, we start with the first level (the roots level) but only add the subsequent levels if they are distinct from the previous one. This means that the pruned version of these hierarchies may

have less than six levels. If we had pruned the hierarchy shown in **Figure 2**, then the triad level would have not been included as it is not distinct from the thirds level. However, this pruning still guarantees that we will have the original annotations. We note that this specific scheme for hierarchical expansion is one among many possibilities, and is intended to align with existing chord evaluation metrics. Alternative schemes are possible, such as not merging triads at the thirds level (to keep diminished distinct from minor), but this is not pursued in the present work. We also note that the number of levels in a hierarchy is not important to the L-measure, as it is concerned with proportions of time instant triplets as defined by the first two sharing a common level that is deeper than the third time instant. Neither the depth of the difference nor what the levels correspond to (such as tetrads) is taken into account in the L-measure.

These automatic chord hierarchies differ from the automatic hierarchy expansion for structure annotations in several ways. First, there are at most three levels in the automatic hierarchy expansion for structure annotations, while there can be six levels in the automatic chord hierarchies. Second, the automatic chord hierarchies do not have to contend with variation markers like those present in structure annotations. While variation markers add nuance to structure annotations in a similar way to increasing complex chord grammars, variation markers are more subjective than chord annotations. The challenges in comparing structure annotations introduced by this subjectivity is partially addressed by the automatic hierarchy expansion with the contraction and refinement levels (McFee and Kinnaird, 2019). In contrast, flat chord annotations, while less subjective, are more reliant on expertise to hear increasingly complex chords and to be consistent in their annotations. To address this curse of expertise, the automatic chord hierarchies iteratively coarsen the original chord annotations from the bottom up, but do not introduce any refinements.

4. Experiments

We apply our two extensions of the automatic hierarchy expansion to two examples. The first experiment concerns the structural segmentation task with semantic labels instead of just syntactical ones. The second experiment uses our chord extension of the automatic hierarchy expansion.[4]

4.1 Comparing Beatles-TUT and Isophonics

Our first experiment investigates the differences between two collections of structural annotations: Beatles-TUT (Paulus, 2010) and Isophonics (Harte, 2010). There are 174 tracks that are in both datasets and can be matched.[5] Both sets of annotations were derived from human annotators, which raises the question of how much agreement there is between the two. In each collection, annotators applied slightly different conventions and vocabulary to describing sections. Changes in vocabulary are not intrinsically problematic, but there are also differences in how variation markers are applied, and how specifically sections are annotated. Our goal in applying automatic hierarchy expansion here is to obtain a more robust assessment of how closely these two collections of annotations agree.

Figure 3 compares the L-measure scores derived from comparing the two collections of flat annotations (horizontal axis) to the scores derived after applying automatic hierarchy expansion (vertical axis). The flat annotations achieve a relatively high mean agreement score of 0.85 ± 0.14. We expect high agreement here because both collections were created by human annotators operating on similar principles and drawing on common prior work (Pollack, 2000).

After expanding the annotations into hierarchies, the L-measure exhibits a modest increase to 0.89 ± 0.10. While the average change is relatively small, there are a few cases where the change is substantial: the minimum agreement increases from 0.13 (flat) to 0.53 (expanded), and in general, the distribution of scores is more tightly concentrated as illustrated by the marginal histograms in **Figure 3**. This indicates that hierarchy expansion indeed exposes common latent structure between these two collections of annotations.

Figures 4 and **5** illustrate two extreme cases where automatic hierarchy expansion dramatically changes the L-measure between the annotations in the Beatles-TUT and Isophonics dataset. In the first case, "Dig It" (**Figure 4**) improves from 0.131 to 0.726, because the contraction of the Isophonics annotation (i.e. removing the additional nuances around *refrain*) agrees most closely with the TUT annotation (that only has one kind of *refrain*), and the refinement of the TUT annotation (i.e. adding nuances to the repeated refrains) matches more closely with the Isophonics annotation. As such, these two annotations are effectively very similar, with the majority of their differences due to the use of variation markers and small deviations in boundary placement.

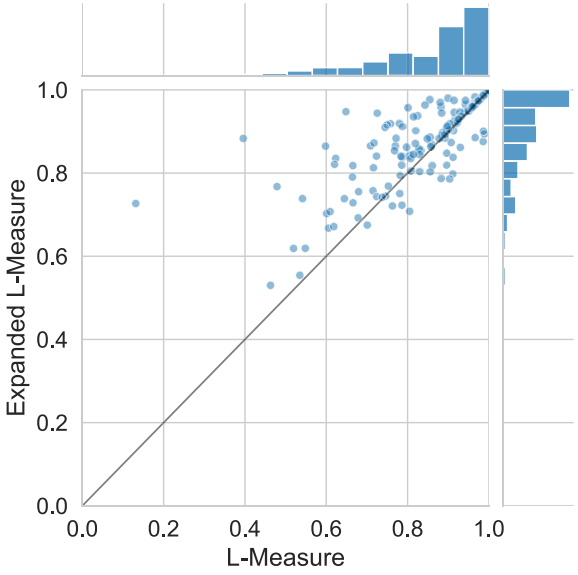

Figure 3: A comparison of structure agreement between the TUT and Isophonics examples using the L-measure before automatic hierarchy expansion (horizontal axis) and after expansion (vertical axis).

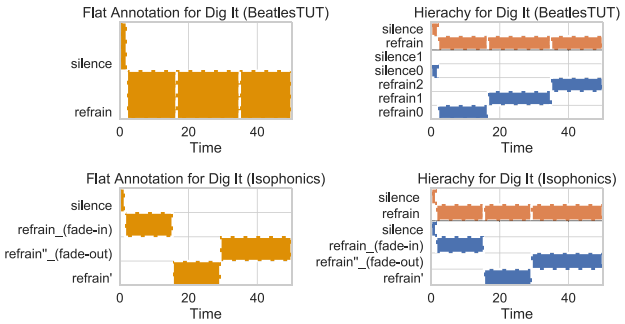

Figure 4: Extreme example "Dig It" where hierarchy expansion increases the L-measure from flat annotations by +0.595. Top left: Flat annotation in Beatles-TUT; Top right: Automatic hierarchy expansion for Beatles-TUT annotation; Bottom left: Flat annotation in Isophonics; Bottom right: Automatic hierarchy expansion for Isophonics annotation.

The second case, "Michelle" (**Figure 5**) decreased from 0.903 to 0.786 after expansion. This is explained by the annotation from Isophonics that uses two labels that contain conflicting structural labels: *outro_verse_(instrumental)* and *outro_bridge*. The first simplifies to "verse" in the contraction level (shown in green level on the top right of **Figure 5**) and the second simplifies to "bridge." This results in the contraction level (shown as the green level on the bottom right of **Figure 5**) having six instances of a verse, four instances of the bridge, and no segment labeled as the outro. In the Beatles-TUT annotation the time-steps within these labels are simply marked as *outro*. This results in the contraction level (shown as the green level on the top right of **Figure 5**) having five instances of a verse, three instances of the bridge, and one segment representing the outro (labeled as 'out' in the contraction level).

Summarizing the results of this investigation, the original TUT and Isophonics annotations do broadly agree with each other, though there are some notable cases where annotations superficially disagree. Automatic hierarchy expansion is able to infer latent semantic structure encoded in the segment labels, and exposing this structure reveals more agreement than was initially detectable. We note that the expansion method used here is relatively naive, consisting of a handful of manually constructed string substitution rules. The rules we have implemented do in some cases lead to a decrease in

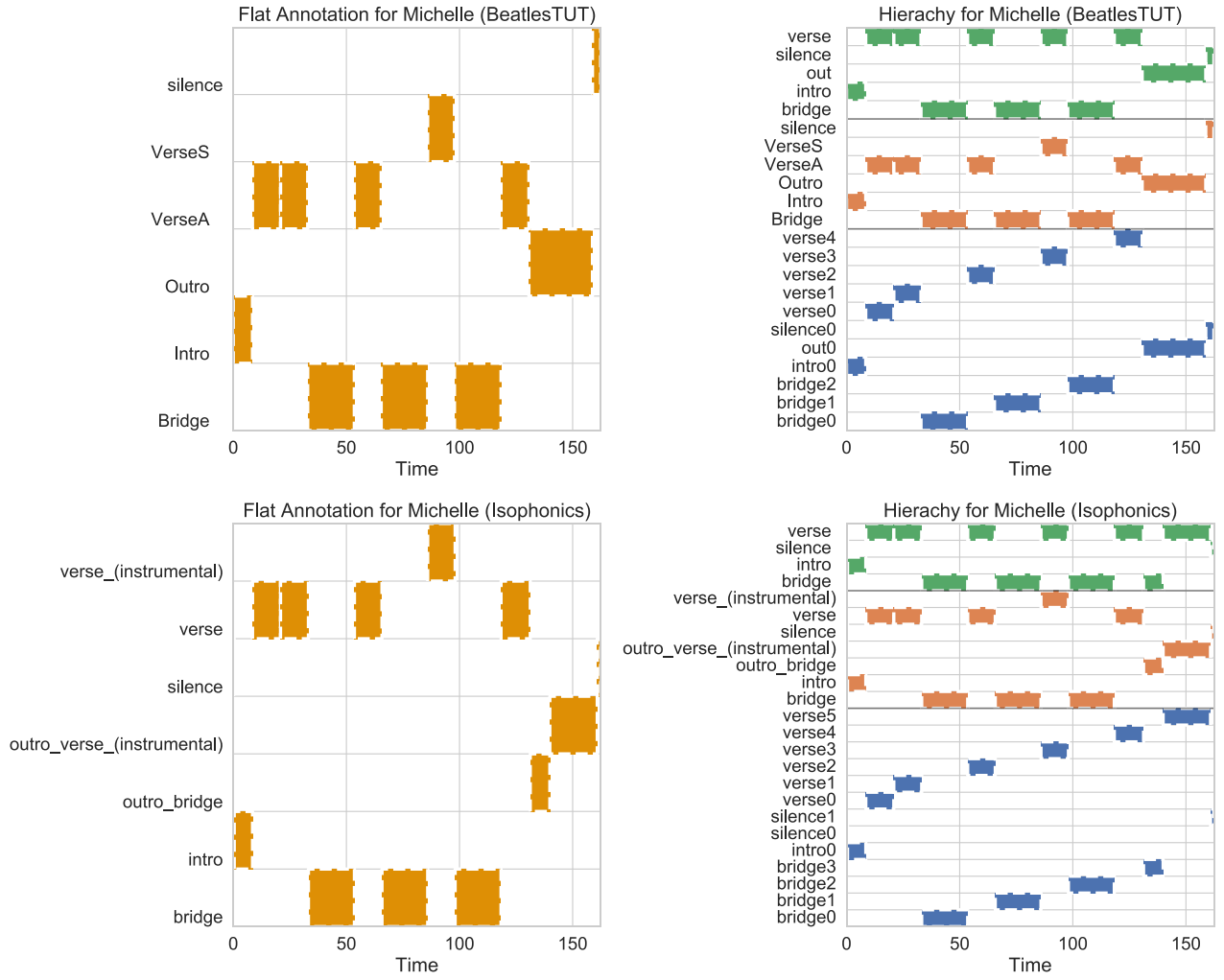

Figure 5: Extreme example "Michelle" where hierarchy expansion decreases by −0.118 the L-measure from flat annotations. Top left: Flat annotation in Beatles-TUT; Top right: Automatic hierarchy expansion for Beatles-TUT annotation; Bottom left: Flat annotation in Isophonics; Bottom right: Automatic hierarchy expansion for Isophonics annotation.

agreement (e.g., **Figure 5**), but overall, the method appears to be relatively robust.

4.2 Expansion on Chord Annotations

We evaluated the impact of the automatic chord expansion using the collection of 1217 annotated recordings from Humphrey and Bello (2015) (the *reference*) and the predictions given by the model of McFee and Bello (2017) (the *estimate*). For each track, we selected one reference annotation and compared it to the estimate annotation using first the standard chord metrics provided by `mir_eval` (version 0.5) (Raffel et al., 2014), and then with the L-measure applied to the automatically expanded chord hierarchy annotations.

Figure 6 illustrates the comparisons between each of the standard chord evaluation metrics — *roots*, *thirds*, *triads*, *tetrads* — and the L-precision, L-recall, and L-measure applied to the automatically generated hierarchies derived from both reference and estimated chord annotations. For illustrative purposes, a robust linear regression (Huber-T weighted (Huber, 1981) to reduce the influence of outliers) is performed for each pair of metrics using the Python `statsmodels` package (Seabold and Perktold, 2010). The 95% confidence intervals on the regression are provided by bootstrap-sampling ($n = 100$ trials). **Figure 7** reports the (Spearman) correlation between each pair of chord evaluation metrics.[6]

The first thing to observe in **Figure 6** is that the precision and recall scores exhibit similar trends. Low precision is generally interpreted as "over-predicting", which in this context would mean that the estimate contains more (deeper) structure than the reference, and conversely for low recall. However, in this context, both annotations are automatically derived by the same process, and should therefore be expected to have comparable depth when the underlying chord annotations use similarly complex vocabulary. In this dataset, the L-precision and

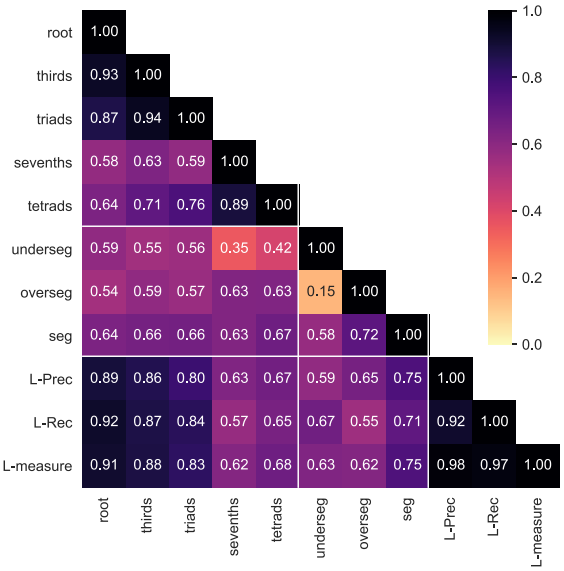

Figure 7: The Spearman correlation between each pair of chord metrics on the 1217 dataset. In addition to the basic metrics (roots, thirds, triads, sevenths, tetrads), we include the directional Hamming distance metrics (underseg, overseg, and seg).

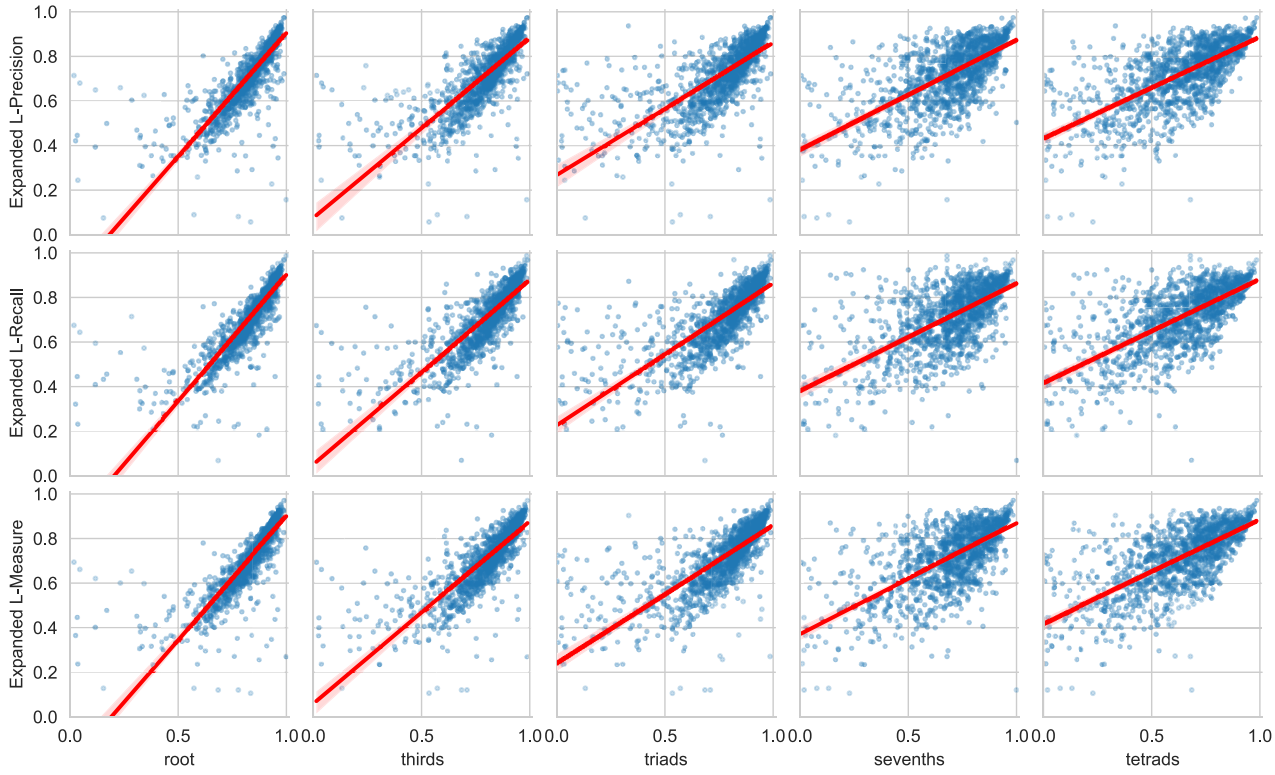

Figure 6: Basic chord metrics (roots, thirds, triads, sevenths, tetrads) are compared to hierarchy expansion metrics on the collection of 1217 songs. The solid line (and shaded region) indicates the linear regression between each combination of metrics (and 95% confidence interval).

L-recall scores exhibit a Spearman correlation of +0.92 (**Figure 7**), indicating that the depth of the hierarchy is generally consistent between the references and estimates. To simplify the discussion for the remainder of this experiment, we will therefore focus on the single "L-measure" score derived from the harmonic mean of precision and recall.

Overall, **Figure 6** demonstrates that the automatic hierarchy expansion metric (L-measure) generally agrees with the standard chord-based metrics, though the correlation decreases as the chord metrics become more specific (going from roots to tetrads). This should be expected, since the chord structure hierarchy is derived from the same principles which underlie the standard chord metrics. From the correlation summary in **Figure 7**, we can also observe that the L-measure scores correlate strongly with both the basic chord metrics and the directional Hamming distance metrics (*underseg*, *overseg*, and *seg*). Note that the directional Hamming distance metrics have comparatively weak correlation with the basic metrics, which should be expected because these metrics completely ignore the semantic content of the labels.

While the hierarchy-based comparison correlates with each of the prior metrics, this is only on average. There are some notable disagreements, and investigating these reveals interesting behavior in the chord metrics as well as the quality of the data.

Figure 8 demonstrates two extreme examples of the comparison between the *roots* and L-measure metrics. In the first case— "Lovely Rita" by The Beatles—the estimator produces a similar chord progression to the reference, but sharp by one semitone due to a disagreement in tuning.[7] While extreme, this example is typical of tracks which produce low root estimation scores, which often arise from tuning discrepancies. Because the annotations disagree at the roots, all standard chord metrics produce scores near 0. However, the L-measure is robust to this disagreement, as it relies on internal structural consistency between the annotations, rather than absolute pitch agreement. Conversely, the second example in **Figure 8**—"Jungle Boogie" by Kool & The Gang—shows the reverse situation, where the two annotations have dramatically different structure (and correspondingly low L-measure), but produce a relatively high roots score and middling-to-low scores on the remaining metrics.

Figure 9 illustrates three cases from the opposite end of the hierarchy, comparing L-measure score to the *tetrads* metric. In the first case, "The Way You Do The Things

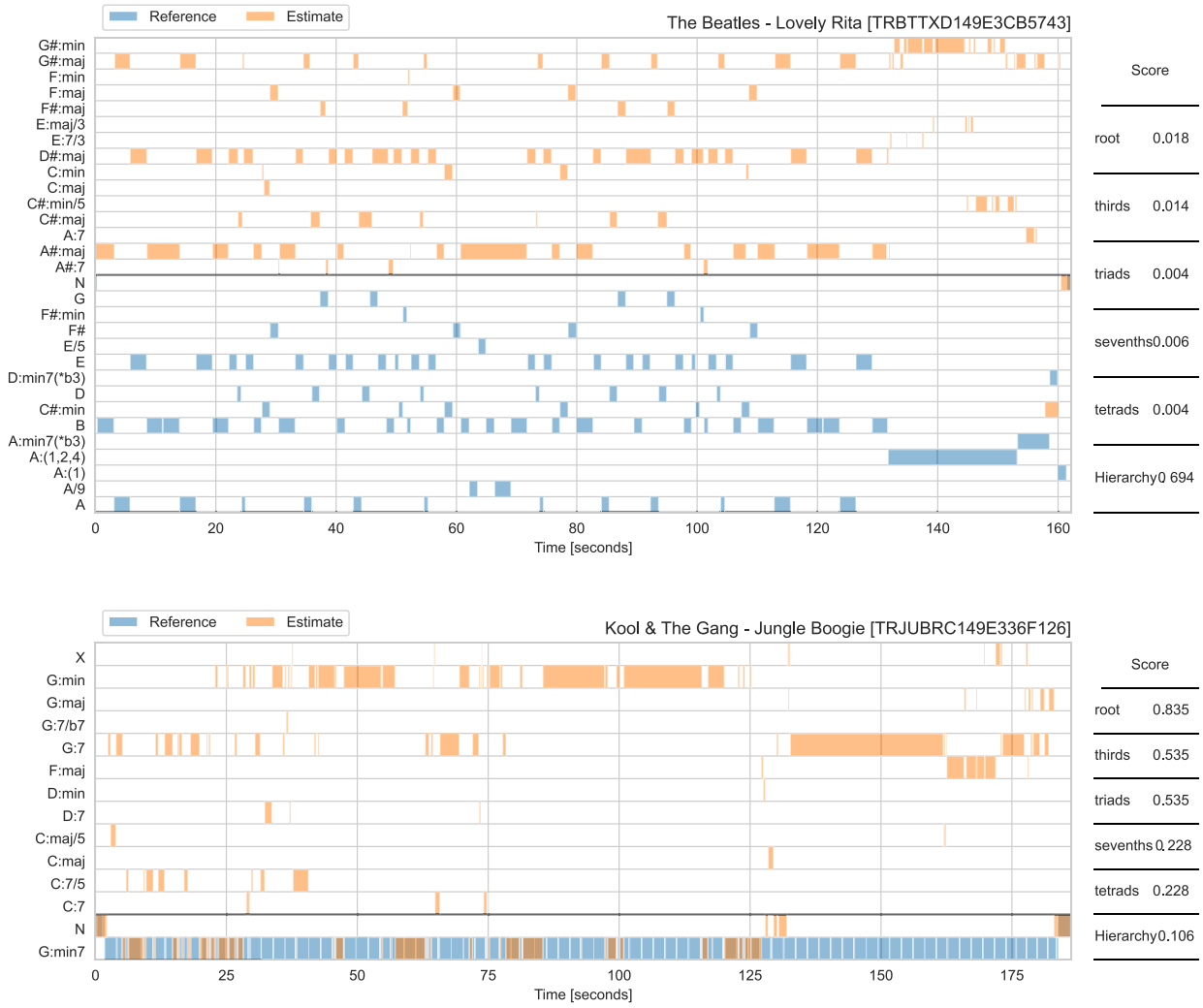

Figure 8: Examples of extreme disagreements between *roots* and hierarchy measures. Top: a high expanded L-measure, but a low roots score due to disagreement in tuning. Bottom: a high roots score, but a low hierarchy score due to large structural discrepancies.

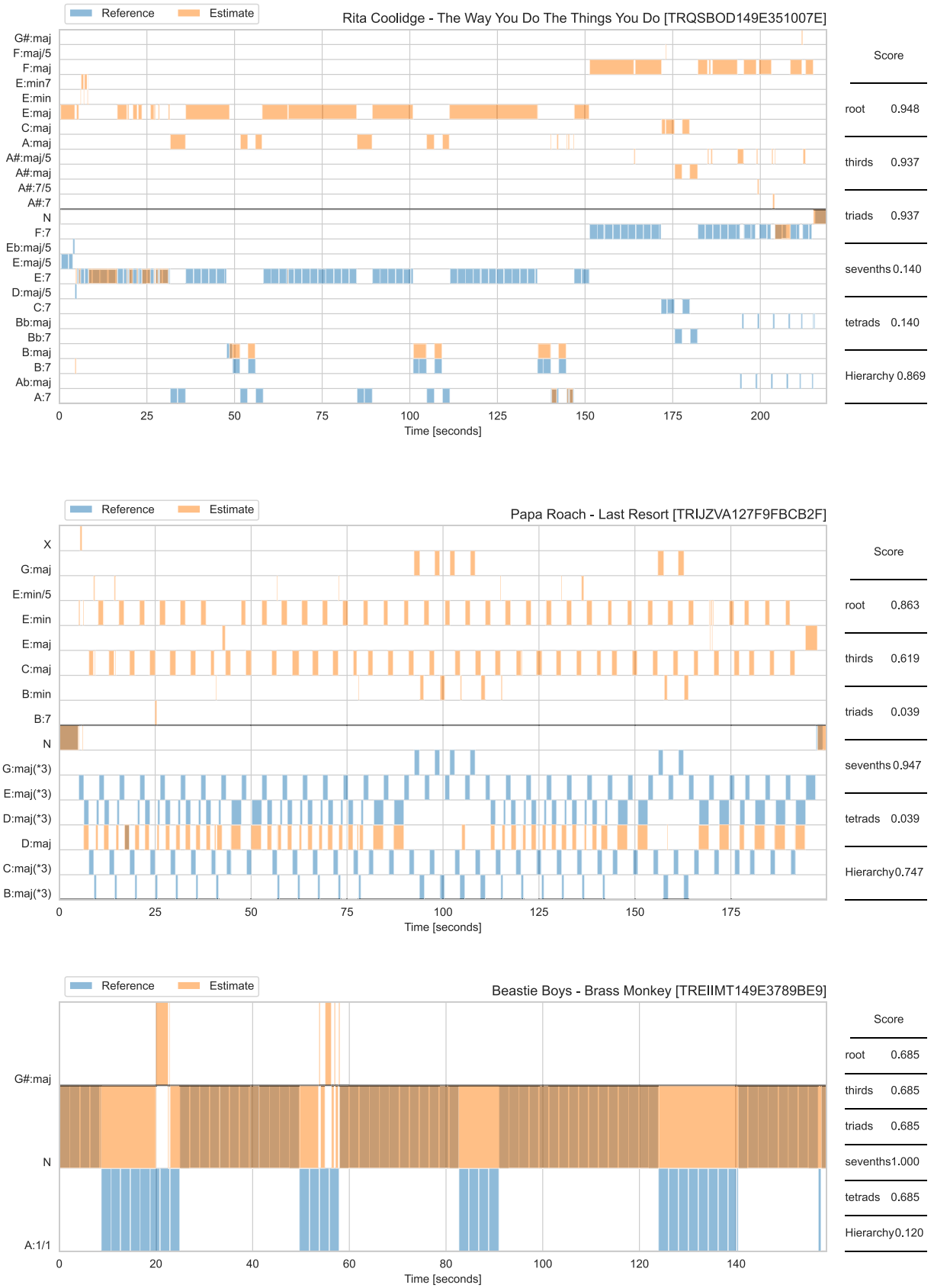

Figure 9: Examples of extreme disagreements between *tetrads* and hierarchy measures. Top and middle: high L-measure but a low tetrads score. Bottom: high tetrads score, but low L-measure.

You Do" by Rita Coolidge, the estimate is consistently predicting triads instead of full seventh chords, but the overall structure is largely preserved, resulting in a relatively high hierarchy score. This does not imply that the estimate is necessarily *correct*, just that it is structurally similar to the reference annotation.

The second example, "Last Resort" by Papa Roach, exhibits a similar score discrepancy, but this time arising

from "power chord" annotations where the third scale degree is excluded in the reference annotations but not in the estimate. This issue is pervasive in chord datasets derived from popular rock music, and power chord annotations do not fit comfortably into the standard chord evaluation metrics, which generally assume the presence of the third. In this example, the standard scores are (arguably) artificially low, but the structure of the annotations is largely in agreement.[8]

Conversely, "Brass Monkey" by the Beastie Boys, produces a low hierarchy score, but relatively high scores on all standard metrics (including tetrads). This is explained by the estimate guessing "no chord" almost everywhere, with few brief diversions to `G♯:maj`. Inspection of the reference annotation reveals that it is a relatively extreme case, alternating between "N" (no-chord) and "A:1/1" (meaning only a root note A with no harmony). It could be argued that this track should be excluded from the chord evaluation overall as it has no harmonic content. That said, the two annotations do exhibit substantially different structure, and encode different harmonic content, so a low score seems reasonable in this case.

The second example in **Figure 9** exposes an interesting aspect of mir_eval's chord evaluation methods: scores are normalized according to their occurrence within the song. For example, if neither the reference nor annotation contain seventh chords, the *sevenths* score is assigned a value of 1 (0/0). This problem is not isolated to total absence: the example in question also results in relatively inflated scores for all standard metrics, compared to the harmonic content actually encoded in the annotations. While this score normalization may be correct from the perspective of quantifying false discovery errors, it raises other problems, particularly when scores are aggregated over an entire collection to report a statistical summary of algorithm performance. Specifically, this normalization can artificially inflate aggregated scores in non-trivial ways. This may in part explain the low correlation between the sevenths metric and others, reported in **Figure 7**. The hierarchical approach is not immune to this problem either: if (and only if) a chord annotation has trivial structure (e.g., is entirely one label), then the L-measure will also encounter a similar problem arising from interpretation of 0/0 in Equation (1). However, these cases are exceedingly rare in chord annotation corpora.

To summarize the results of this study, it does appear that automatic hierarchy expansion can be used to holistically compare chord annotations, though some care should be taken in interpreting the resulting score. A low score (e.g., the second examples in **Figures 8** and **9**), does generally indicate significant structural disagreements between the annotations that cannot be automatically reconciled. A high score indicates structural agreement, and can be robust to errors arising from tuning error, assuming that the annotations are otherwise structurally similar. However, a high score does not necessarily mean that the annotations fundamentally agree, as it is still possible to have high structural similarity while missing critical details (**Figure 9**, top example). This observation suggests the following rubric for effective use of automatic hierarchy expansion in chord evaluation: 1) if the hierarchy agreement is low, the annotations have fundamental differences; 2) if the hierarchy agreement is high, the annotations are structurally similar, but the standard metrics should be checked in fine-to-coarse order (tetrads to roots) to determine absolute agreement.

5. Conclusion

Several tasks compare various structural partitions of recordings, either to evaluate the "correctness" of various human or algorithmic annotations or simply to compare them for consistency. Examples include the chord recognition and structural segmentation tasks. Recent work by McFee and Kinnaird (2019) proposes expanding flat annotations (that assume the existence of a single "correct" segmentation of a recording) into hierarchical ones for use in evaluating syntactical structure labels. McFee and Kinnaird (2019) then assert that such multi-level evaluations could be a robust alternative to evaluations on flat annotations. In this work, we extend the automatic hierarchy expansion method by McFee and Kinnaird (2019) in two ways.

Our first extension of the automatic hierarchy expansion method proposed (McFee and Kinnaird, 2019) concerns structural segment labels. We apply this extension to comparing segmentation annotations of Beatles recordings in the Isophonics and Beatles-TUT datasets. In this extension, we created rules to exploit latent hierarchical structure inherent in labels such as *verse*, *verseA*, and *verse_instrumental* before creating the automatic hierarchy expansion from McFee and Kinnaird (2019).

Our second investigation using hierarchical structure analysis addresses several existing challenges for chord evaluation. We are able to compare the internal consistency of annotations over an entire track, while also leveraging hierarchical relationships between chord labels. Additionally, by comparing across several levels of chord simplifications, we address issues arising from ambiguous and subjective annotations, such as differences in the spelling of root pitch classes, varying amounts of detail in chord labels, and ambiguities arising from tuning discrepancies. We also address the challenge of selecting a chord vocabulary noting that any chord label that is within the formal grammar of Harte et al. (2005) can be directly incorporated in this hierarchical evaluation, and we have provided an extension for chords that are labeled with *X* denoting being "out of grammar." This second investigation showed that the automatic hierarchy expansion can be used to holistically compare two chord annotations, but that it should not be used as a singular measure without careful interpretation. This is especially true if the *L*-measure (which acts similarly to a weighted average across all simplifications) between two annotations is high, as this could mean that the annotations have strong agreement in the coarsest chord simplifications but have more nuanced disagreement in the finer layers.

In any segmentation task comparing two sets of annotations can be challenging in part due to differing

richness of annotation vocabularies. For example, the annotations within the SALAMI dataset has a very restrictive syntactical vocabulary labeling sections with letters (Smith et al., 2011; McFee and Kinnaird, 2019), while the structure labels in Isophonics are more semantic in nature including structural labels such as *verse*, *bridge*, and *outro*. In this work, we demonstrate through two extensions applied to two different examples that comparison between datasets with differing vocabularies is possible. This means that one can be less restrictive with the 'allowable' annotations and then create hierarchies that exploit the semantic structure that is latent in whatever resulting annotations are created during labeling.

Notes

[1] The L-measure can also be applied to compare flat segmentations. See McFee et al. (2017) for details and McFee and Kinnaird (2019) for a summary.

[2] Interestingly, the "sevenths" evaluation exactly does not count all categories of seventh chords, and excludes `dim7` and `hdim7`.

[3] A Jupyter notebook with these rules can be found at: https://github.com/kmkinnaird/tismir2020-hierarchy.

[4] The associated code for this paper is available in the above listed GitHub repository.

[5] The matching process is contained in a Jupyter notebook in the GitHub repository listed above.

[6] We report the Spearman correlation here, rather than Pearson, because there is no reason to generally expect a linear relationship between bounded, normalized metrics. Rather, we are more interested in the rank-ordering induced by these metrics.

[7] In fact, this song was originally recorded one half-step sharper than it appears on the record (Lewishon, 1988). This explains the source of the tuning discrepancy between the reference and estimate.

[8] As an aside, the use of `E:maj(*3)` in the reference annotation for this track is questionable, as the key of the song is `E:min`.

Acknowledgements

The first author is the Clare Boothe Luce Assistant Professor of Computer Science and Statistical and Data Science at Smith College and as such, is supported by Henry Luce Foundation's Clare Boothe Luce Program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Luce Foundation.

Competing Interests

The authors have no competing interests to declare.

References

Bay, M., Bello, J. P., Burgoyne, J. A., Chew, E., Ehmann, A. F., Khadkevich, M., Mauch, M., McVicar, M., Pauwels, J., and **Rocher, T.** (2010). The Utrecht agreement on chord evaluation. https://www.music-ir.org/mirex/wiki/The_Utrecht_Agreement_on_Chord_Evaluation. Accessed: 2020-08-13.

Burred, J. J., and **Lerch, A.** (2004). Hierarchical automatic audio signal classification. *Journal of the Audio Engineering Society*, 52(7/8): 724–739.

Carsault, T., Nika, J., and **Esling, P.** (2018). Using musical relationships between chord labels in automatic chord extraction tasks. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*.

Dannenberg, R. B., and **Goto, M.** (2008). Music structure analysis from acoustic signals. In *Handbook of Signal Processing in Acoustics*, pages 305–331. Springer. DOI: https://doi.org/10.1007/978-0-387-30441-0_21

Essid, S., Richard, G., and **David, B.** (2005). Instrument recognition in polyphonic music based on automatic taxonomies. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1): 68–80. DOI: https://doi.org/10.1109/TSA.2005.860351

Harte, C. (2010). *Towards automatic extraction of harmony information from music signals*. PhD thesis, Queen Mary University of London.

Harte, C., Sandler, M. B., Abdallah, S. A., and **Gómez, E.** (2005). Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 66–71.

Huber, P. J. (1981). *Robust Statistics*. John Wiley & Sons, New York. DOI: https://doi.org/10.1002/0471725250

Humphrey, E., and **Bello, J.** (2015). Four timely insights on automatic chord estimation. In Müller, M. and Wiering, F., editors, *Proceedings of the 16th International Society for Music Information Retrieval Conference*, pages 673–679.

Lewishon, M. (1988). *The Complete Beatles Recording Sessions: The Official Story of the Abbey Road Years 1962—1970*. Bounty Books, London.

Mauch, M. (2010). *Automatic chord transcription from audio using computational models of musical context*. PhD thesis, Queen Mary University of London.

McFee, B., and **Bello, J.** (2017). Structured training for large-vocabulary chord recognition. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*.

McFee, B., and **Kinnaird, K. M.** (2019). Improving structure evaluation through automatic hierarchy expansion. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pages 152–158.

McFee, B., Nieto, O., Farbood, M. M., and **Bello, J. P.** (2017). Evaluating hierarchical structure in music annotations. *Frontiers in Psychology*, 8: 1337. DOI: https://doi.org/10.3389/fpsyg.2017.01337

McGuirl, M. R., Kinnaird, K. M., Savard, C., and **Bugbee, E. H.** (2018). SE and SNL diagrams: Flexible data structures for MIR. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, pages 341–347.

Nieto, O., and **Bello, J. P.** (2016). Systematic exploration of computational music structure research. In *Proceedings of the 17th International Society for Music Information Retrieval Conference*, pages 547–553.

Paulus, J. (2010). Improving Markov model based music piece structure labelling with acoustic information. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pages 303–308.

Paulus, J., and **Klapuri, A.** (2006). Music structure analysis by finding repeated parts. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, pages 59–68. ACM. DOI: https://doi.org/10.1145/1178723.1178733

Paulus, J., Müller, M., and **Klapuri, A.** (2010). State of the art report: Audio-based music structure analysis. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pages 625–636.

Pauwels, J., O'Hanlon, K., Gómez, E., and **Sandler, M. B.** (2019). 20 years of automatic chord recognition from audio. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pages 54–63.

Pauwels, J., and **Peeters, G.** (2013). Evaluating automatically estimated chord sequences. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 749–753. IEEE. DOI: https://doi.org/10.1109/ICASSP.2013.6637748

Pollack, A. W. (2000). 'notes on ...' series. https://www.recmusicbeatles.com/public/files/awp/awp.html, accessed 2020-08-31.

Raffel, C., McFee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., and **Ellis, D. P. W.** (2014). Mir_eval: A transparent implementation of common MIR metrics. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, pages 367–372.

Seabold, S., and **Perktold, J.** (2010). statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*. DOI: https://doi.org/10.25080/Majora-92bf1922-011

Smith, J. B. L., Burgoyne, J. A., Fujinaga, I., De Roure, D., and **Downie, J. S.** (2011). Design and creation of a large-scale database of structural annotations. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 555–560.

Ullrich, K., Schlüter, J., and **Grill, T.** (2014). Boundary detection in music structure analysis using convolutional neural networks. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, pages 417–422.

How to cite this article: Kinnaird, K. M., & McFee, B. (2021). Automatic Hierarchy Expansion for Improved Structure and Chord Evaluation. *Transactions of the International Society for Music Information Retrieval,* 4(1), pp. 81–92. DOI: https://doi.org/10.5334/tismir.71

Submitted: 01 September 2020 **Accepted:** 28 April 2021 **Published:** 29 June 2021

]u[*Transactions of the International Society for Music Information Retrieval* is a peer-reviewed open access journal published by Ubiquity Press.

OPEN ACCESS