AUDIO-BASED COVER SONG RETRIEVAL USING APPROXIMATE CHORD SEQUENCES: TESTING SHIFTS, GAPS, SWAPS AND BEATS

Juan Pablo Bello
Music Technology, New York University
jpbello@nyu.edu

ABSTRACT

This paper presents a variation on the theme of using string alignment for MIR in the context of cover song identification in audio collections. Here, the strings are derived from audio by means of HMM-based chord estimation. The characteristics of the cover-song ID problem and the nature of common chord estimation errors are carefully considered. As a result strategies are proposed and systematically evaluated for key shifting, the cost of gap insertions and character swaps in string alignment, and the use of a beat-synchronous feature set. Results support the view that string alignment, as a mechanism for audio-based retrieval, cannot be oblivious to the problems of robustly estimating musically-meaningful data from audio.

1 INTRODUCTION

The term *musical similarity* can be used to imply a relationship between songs that goes beyond texture, genre or artist, and that is more akin to purely musicological comparisons between songs, e.g. in terms of their melody, harmony and/or rhythm. In this context, cover song identification in popular music can be seen as a good, albeit limited, test of the ability to model musical similarity. The task of identifying cover songs poses many difficulties for audio-based music retrieval since renditions are often quite different from the original in one or many attributes including instrumentation, key or genre to name a few.

In this paper we propose an approach to cover song identification based on the use of string alignment for the scoring of approximate chord sequences. These sequences are extracted from audio using chroma features and hidden Markov Models [2]. They are *approximate* because chord estimation from audio is never 100% accurate. Song sequences in a collection are ranked according to the score of their alignment with a query sequence. The use of approximate string matching is favored as the sequential ordering of events in the signal is taken into account. We argue that in order to maximize retrieval results, one has to consider not only the key or tempo differences between cover song sequences, but also the ways in which these sequences approximate (or not) the songs they represent.

1.1 Previous Work and Motivation

There is a long history of using approximate string matching in Music Information Retrieval. A notable example in the symbolic domain is the use of string alignment for the characterization of melodic similarity in both monophonic and polyphonic databases [9, 13]. This is unsurprising if we consider that melodies are well posed to be characterized as sequences of symbols representing, for example, pitches or intervals.

This reasoning is also behind early attempts to incorporate audio into MIR systems in the context of *Query by Humming* (QBH). This problem is largely defined as one of matching between a monophonic audio query and a symbolic and often polyphonic database. The transformation of signals into strings can be achieved using well-known signal processing algorithms, thus sequence alignment is featured prominently in QBH research ([1] is a recent example). However, audio-based analysis, even in the monophonic case, adds an extra layer of complexity that is bound to negatively impact the performance of these systems [12]. This is all the more acute for the case when polyphonic audio signals, the format in which most music is available, are used both as queries and as documents in the database. In [10] the cosine distance is used between the most repeated melodic fragments of songs, represented as key-invariant and beat-synchronous spectral lines, to measure pairwise similarity. This approach uses cover-song identification as a test of melodic similarity in audio collections. While showing great promise, it suffers from the great difficulties of robustly estimating melody from complex signals.

Alternatively, music similarity can be characterized by harmonic, rather than melodic, content using so-called chroma features, or pitch class profiles. In [11], a successful system is presented for the identification of excerpts (10-30s) in orchestral music. The method relies on short-time statistics, quantization and resampling of chroma features in order to find similar excerpts despite tempo variations. In [6] a cover song ID system is proposed that cross-correlates beat-synchronous chroma features to characterize pairwise similarity. Key invariance is achieved by performing all 12 shifted versions of the cross-correlation. This approach performed best on the 2006 MIREX cover song identification task.

An interesting variation on the theme of using chroma fea-

tures for characterizing music similarity is proposed in [3]. In this work, chroma features are collapsed into string sequences using vector quantization (VQ) and best retrieval is achieved by calculating the string-edit distance between these sequences. Success is demonstrated for finding repeated patterns within a song. This paper also provides a strong argumentation in favor of using string-based methods that take into account the ordering of events in the signal, an issue which is consistently ignored in models for texture-based similarity. However, the lack of interpretability of the VQ-produced strings encourages the use of metrics that consider all character swaps to be the same, a strategy that we purposefully avoid in this paper. Alternatively, [8] uses strings representing chord sequences. This approach to cover song ID, on which our work is based, relies on the calculation of chord sequences by means of HMM-based analysis (supervised in their case, unsupervised in ours) and the computation of pairwise similarity on key-transposed sequences using DTW. While results are promising, this work fails to justify why the chosen scoring methodology is a good fit to the nature of the problem and to the data it uses. This, in turn, branches out into other fundamental questions: How does this approach cope with the inexactitude of the sequence estimation? What is the impact of attempting to introduce key invariance? Is there a purpose for introducing beat-based rhythmic invariance (as proposed by [6] and [10]) into the process? These questions motivate our work, and our attempts to answer them constitute its main contributions.

1.2 Organization of this paper

Section 2 briefly explains how chord sequences are estimated and analyzes their likely pattern of confusion. Section 3 discusses the very basics of sequence alignment, introduces our strategy for scoring character substitutions and explains our approach to key-invariant alignment. Section 4 presents the results and discussion of four experiments on cover song identification aimed at measuring the impact that certain parameter configurations have on retrieval. Section 5 presents conclusions and future work.

2 ESTIMATING CHORD SEQUENCES

In [2] a methodology was introduced for robustly generating sequences of major and minor triads from audio signals. The approach, to be briefly summarized in the following, is used as the front end to our cover song identification system. First, 36-dimensional chroma vectors, or pitch-class profiles, are calculated from the audio signal by collapsing constant-Q spectral data into one octave. These vectors are tuned and, optionally, averaged within beats, before being quantized into 12-bin vectors representing the spectral energy distribution across notes of the chromatic scale. These features are used as observations on a 24-state hidden Markov model, where each state corresponds to one of the major and minor triads. The parameters of the model, initialized using simple musical knowledge, are trained in an unsupervised fashion using the Expectation-Maximization (EM) algorithm. During training, state-to-observation parameters are clamped, thus resulting on a 'semi-blind' optimization. The final sequence of triads is obtained by decoding the model using the Viterbi algorithm.

TP: 69.64	PAR: 3.44	REL: 5.81	V: 4.00
IV: 2.21	III: 2.42	OTH: 7.69	NR: 4.79

Table 1. Chord estimation results using frame-based chromas. Values are in percentage of total detections.

While chord estimation results are available on the original paper, it is more relevant to this work to discuss a more recent evaluation of the system. Table 1 depicts results and confusion on a chord recognition test performed against 110 manually-annotated chord sequences of recordings by the Beatles (see [7] for more details about this dataset). For the test we assume enharmonic equivalence and map complex chords, e.g. 6^{ths}, 7^{ths}, to their base triad (e.g. Em7 = Em). Numbers in the table indicate the percentage of total detections for the following categories: true positives (TP), parallel major/minor confusions (PAR), relative major/minor confusions (REL), dominant confusions (V), sub-dominant confusions (IV), third or sixth confusions (III), confusions not in the above categories (OTH) and chords which are not recognized by the system and counted as errors (NR, e.g. diminished, augmented, silences). The results are revealing in that they show that nearly half the errors that the system makes (REL + V + IV + III) are in the immediate vicinity of the true positive in the doubly-nested circle of 5^{ths} of major and minor triads [2]. Assuming that these results can be generalized to the larger set we use for retrieval, then true positives and these closely-related errors account for 85% of total sequence content. Beyond the obvious relation between these results and our choice of initialization for the HMM's state-transition probability matrix, lies the fact that the ordering of chords in the circle provides a good model for the scoring of character substitutions, an issue at the heart of sequence alignment methodologies.

3 SEQUENCE ALIGNMENT

Finding the globally-optimal alignment between strings is an extensively researched topic, notably in bioinformatics [5]. The idea is to find the best possible path between the strings by allowing inexact character matches (i.e. substitutions or swaps) and the introduction of gaps in either of the sequences. In this context, the best path is the one that maximizes a score function, usually the sum of individual scores for aligned pairs of characters, under the consideration that both gap insertions and substitutions imply a penalty, to be respectively known as γ and $\hat{S}$. Because the number of substitutions and gaps is expected to be low between similar sequences, the resulting score is a good measure of similarity. In our application, measuring similarity using string matching provides the added

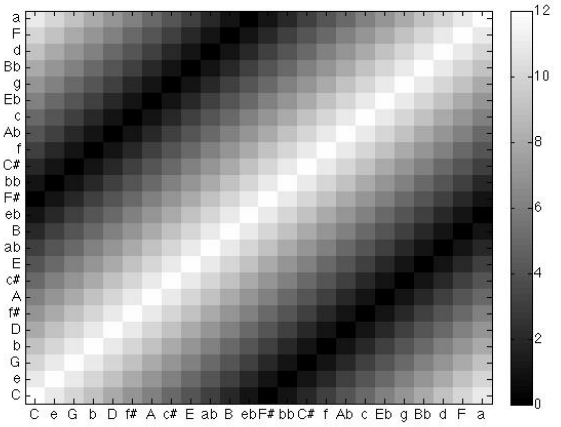

Figure 1. Matrix S based on unitary distances on the doubly-nested circle of 5^{ths}

benefit of taking the sequence ordering into consideration, and thus the temporal structure of the musical piece. This stands in contraposition to the common "bag-of-features" approach where feature ordering is mostly, or totally, ignored. In the present system we use a standard solution to globally-optimal string alignment, based in Dynamic Programming, and known as the Needleman-Wunsch-Sellers (NWS) algorithm (see [5] for a detailed explanation). We use the implementation in *NeoBio*, and open-source library of bio-informatics algorithms in Java [4].

3.1 Substitution Matrix

Some string alignment implementations use a uniform penalty for all substitutions (e.g. string-edit distance). However, our chord sequences are inaccurate and, more importantly, they follow an non-uniform error pattern that can be predicted from the data on Table 1. Hence, it is best to use a score function, i.e. a substitution matrix, that is able to favor certain chord swaps above others. The matrix is defined such that a positive/negative value on the matrix results on an increase/decrease of the global score. For our experiments we use the substitution matrix $\hat{S} = (S - \alpha) \times \beta$, where S, in Figure 1, is derived from the ordering of chords in the doubly-nested circle of 5^{ths}; α is an offset that changes the distribution of positive and negative values in the matrix; and β is a scaling factor (= 10 in the rest of this paper). Values in the main diagonal of S (characterizing perfect matches) are equal to 12. In any given column, going a step up or down from the main diagonal results on a unitary decrease on the substitution value. This pattern is repeated until we reach zero at the opposite end of the circle (e.g. for a C/F# substitution). From that point on, values start to increase again until we reach full circle. As can be seen in the figure, the matrix favors substitutions between harmonically-related triads (e.g. between C and e/a or F/G), i.e. between those triads that, according to results in Table 1, are more likely to be confused.

3.2 Key-Invariant Alignment

The characterization of similarity using chord sequence alignment is key dependent. Except perhaps for a few cases, e.g. when the key shift is a relative minor or a dominant, the scoring of the alignment will be badly affected by variations on the key context. Even in those cases, key dependency increases the probability that non-relevant songs that happen to be on the same key as the query will be scored higher. As we cannot assume that different versions of a song will all be in the same key, we propose a simple mechanism for key matching between sequences before alignment. Let us define x and y as two integer sequences (of any length) such that their elements $x_i, y_i \in 0..23$. This integer range corresponds to the 24 major and minor triads organized from C to A minor, following the ordering in the axes of Figure 1. Let us also define X and Y as the normalized histograms of sequences x and y respectively. We propose that the score is maximized for the alignment between x and $\hat{y}_\phi$, a key-shifted version of y defined as $\hat{y}_\phi = (y + \phi)\ mod\ 24$, where $\phi = \text{argmax}_m(X \cdot \hat{Y}_m)$, $\hat{Y}_m = Y[(n - m)\ mod\ 24]$, $\forall n \in 0..23$ and $m \in 0 : 2 : 22$ is defined such that only major/major or minor/minor shifts are allowed. This very simple approach is only bound to be effective when histogram shapes are similar, as we hope to be the case between cover songs. The latter assumption is not necessarily true when the structure of the songs being compared is significantly different.

4 EXPERIMENTS

A collection of 3208 mp3 files of commercially-available music is used for testing. It contains songs on a wide variety of genres with an emphasis on Anglo-American Pop and Rock. Within that collection there is a cover song sub-set of 157 songs representing 36 different pieces of music. This averages to 4.36 versions per piece, although actual numbers oscillate between 2 and 16 versions per piece. This sub-set is quite heterogeneous, ranging from 22 studio-live pairs by the same band (out of 391 cover-song pairs) such as Nirvana's "Come as you Are" in *Nevermind* and in *MTV Unplugged in New York*, to radical interpretations such as Rancid's remake of Bob Marley's "No Woman No Cry" or REM's remake of Gloria Gaynor's "I Will Survive". Most versions are by different artists and usually involve changes on instrumentation.

Performance is evaluated by using all 157 cover songs as queries and measuring precision and recall based on the ranking of the other versions of each query. The queries, which are always retrieved at rank 1, are removed before evaluation. Since this is a standard IR evaluation, where the number of relevant documents is known, we use common performance measures such as the average R-Precision (Precision at rank R, where R is the total number of relevant items), the average Mean Reciprocal Rank (MRR - 1/rank of the first relevant item) and average 11-point Precision/Recall graphs for visualization.

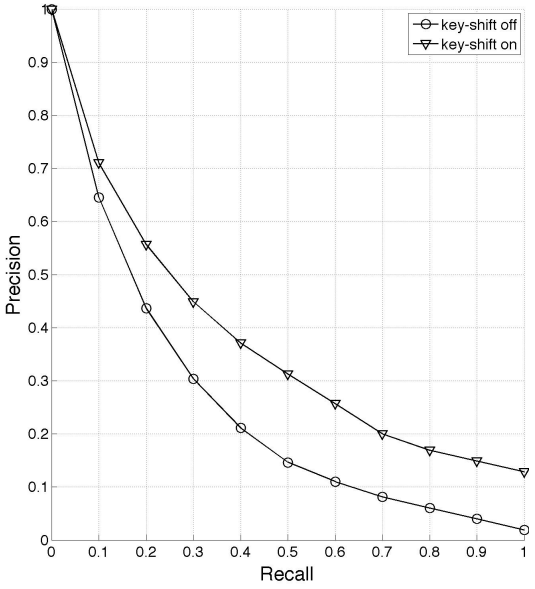

Figure 2. 11-point P/R graphs for retrieval with and without key shifting.

Parameters					Results (0-1)	
Key-shift	γ	α	Scope	δ	R-P	MRR
on	-10	10	frame	12	0.221	0.395
off	-10	10	frame	12	0.089	0.223
on	0	10	frame	12	0.116	0.220
on	-20	10	frame	12	0.182	0.337
on	-10	2	frame	12	0.122	0.265
on	-10	6	frame	12	0.149	0.317
on	-10	10	beat	1	0.168	0.320
on	-10	10	beat	2	0.170	0.323
on	-10	10	beat	4	0.145	0.285
on	**-10**	**10**	**frame**	**6**	**0.245**	**0.401**
on	-10	10	frame	20	0.208	0.377

Table 2. Results for various model parameters (Best set in bold).

In our experiments we aim at measuring the impact of the following actions: (i) using key-shifting - in Section 4.1; (ii) varying the gap penalty γ used by the scoring algorithm - see Section 4.2; (iii) changing the distribution of positive and negative values on the substitution matrix $\hat{S}$ by varying the offset value α - in Section 4.3; and (iv) using beat synchronous instead of frame-based chroma features for the sequence estimation - see Section 4.4. Since our sequences are highly redundant and the NWS algorithm is computationally very expensive ($O(n^2)$ for sequence length n), sequences are downsampled by a factor δ. The impact of this resampling in the performance of the system is also measured on the last experiment. For the same reason, we avoid the testing of all combinations in the parameter space by assuming that parameters are independent from each other. This is an arguable assumption but a necessary one. Table 2 shows averages of R-Precision (R-P) and Mean Reciprocal Rank (MRR) for all the combinations of parameters tested in our experiments. Values range from 0 to 1, with 1 being the best possible value. The low values in the table hint at the difficulties of the task of cover song identification. Since an open test collection is not available, comparisons cannot be made with existing approaches. However, by the time of publication, results of this method in the context of the MIREX 2007 Cover-song ID task will be available for comparison. To get an idea of what the numbers in this paper mean in practice, the reader is encouraged to look at the full list of music and test results on the author's website [1] .

4.1 Testing Shifts

In the first experiment, we test the impact of key shifting on the system's performance. Figure 2 shows the average 11-point P/R graph with and without the key-shifting algorithm described in Section 3.2. For this experiment: $\gamma = -10$, $\alpha = 10$, $\delta = 12$, and feature scope = frames. Results on Table 2 and Figure 2 show how key-shifting brings about significant improvement on retrieval results. Precision increases for all recall rates showing that relevant items are consistently ranked higher using this approach. This increase is particularly acute ($> 15\%$) for recall rates between 0.3 and 0.6. These results are no surprise as they corroborate the intuition that key-shifting is a solution to the known key independence of cover songs. However they cannot be taken for granted, since key shifting also increases the risk that non-relevant songs with similar chord progressions, a common occurrence in pop music, will be ranked higher than relevant songs. It is possible that our simple key-shifting approach might help decrease this risk by favoring alignment between songs with very similar chord distributions. On the other hand, this approach might be precluding covers which are significantly different in structure, and thus bound to have dissimilar chord distributions, to be ranked higher.

4.2 Testing Gaps

Experiment 2 is aimed at testing the sensitivity of the system to changes on the gap penalty γ. For this experiment we use γ = 0, -10 and -20, while $\alpha = 10$, $\delta = 12$, key-shifting is on and feature scope = frames. Results in Figure 3 show how worst performance is achieved for the case when no penalty is used, i.e. $\gamma = 0$. This indicates that, if allowed to time-scale at no cost, many a non-relevant chord sequence can be matched to a query. Again, this is related to the constant use of similar chord progressions in popular music, where harmonic palettes are often less varied that in orchestral music, for instance. However, the fact that results are consistently better for $\gamma = -10$ than for $\gamma = -20$ indicates that over-penalizing for gap insertions also has a negative effect on performance. This is intuitive since large gap penalties do not allow the flexibility needed to match similar songs with different tempi or with slight changes of form.

[1] http://homepages.nyu.edu/∼jb2843/Publications/ismir07.html

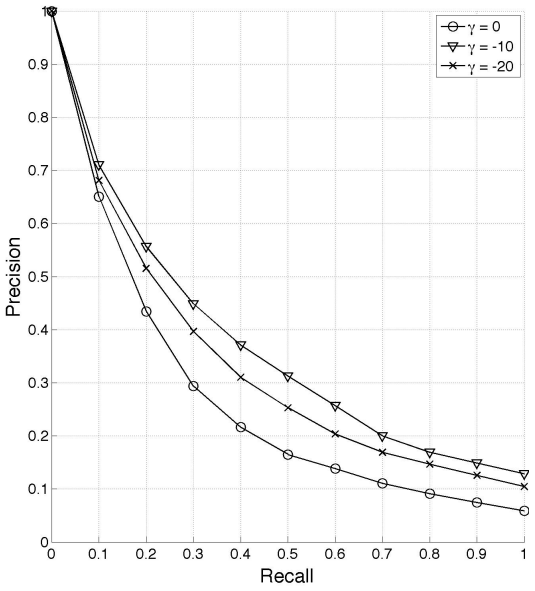

Figure 3. 11-point P/R graphs for variations of the gap penalty γ

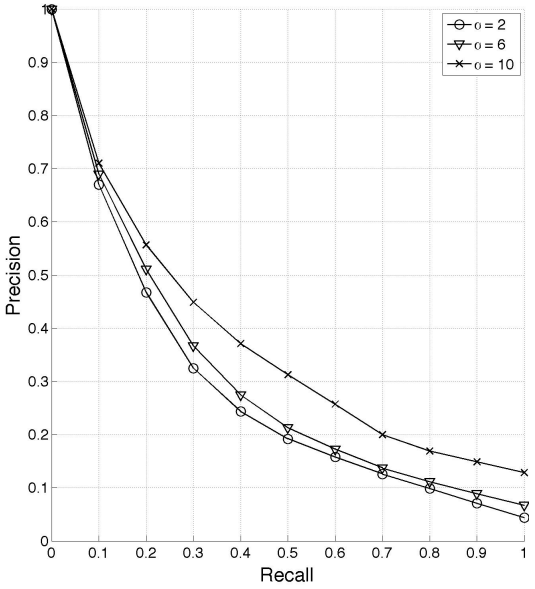

Figure 4. 11-point P/R graphs for variations of α

4.3 Testing Swaps

Although the order of preference of chord swaps is predefined by the values in matrix S, changes in the offset value α, used to define the substitution matrix $\hat{S}$, signify which swaps have a positive or negative impact on the score function. Experiment 3 is aimed at testing how performance is affected by these changes. For this test we use α = 2, 6 and 10, with $\gamma = -10$, $\delta = 12$, key-shifting on and feature scope = frames. The range of α was selected to be symmetrical with respect to the center of the circle of 5^{ths} (corresponding to $\alpha = 6$), while avoiding values that will render $\hat{S}$ completely positive or negative, i.e. $\alpha \leq 0$ and $\alpha \geq 12$. Figure 4 shows that results are worse when scoring for swaps is too permissive, e.g. for $\alpha = 2$ when most values in $\hat{S}$ are positive. Results are slightly better for $\alpha = 6$ and much better for $\alpha = 10$.This is an important observation as the increase of α is the same between 2 and 6 as it is between 6 and 10, while the rate of improvement is notably different. This difference highlights the suitability of using positive scoring only for those swaps which are close in the circle of 5^{ths}, as suggested by the information on Table 1. These results strongly support the view that an adequate choice of substitution matrix can help offset the negative impact that chord estimation errors can have on the retrieval of similar songs.

4.4 Testing Beats

The final experiment tests: (a) the impact of using beat-synchronous instead of frame-based chroma features, and (b) the effect of downsampling sequences by a factor δ before alignment. These tests are grouped together because both these parameters affect the length of the sequences to be aligned, and thus the computational expense of querying the system. In fact, beat-synchronous estimation reduces the average sequence length to one-sixth of the frame-based length. As a result we test frame-based features with δ = 6, 12 and 20, against beat-based features with δ = 1, 2 and 4. The other parameters are set to: γ = -10, $\alpha = 10$ and key-shifting on. Figure 5 shows results for this experiment. Because of the density of this graph, the figure only depicts a detail of the 11-point P/R curves for Precision $\in [0.05, 0.7]$ and Recall $\in [0.1, 1]$.

Against our expectations, frame-based analysis consistently outperforms beat-synchronous analysis. The difference is further emphasized when comparing parameter combinations with similar computational complexity (e.g. [beat, $\delta = 1$] with [frame, $\delta = 6$]). Perhaps this is an indication of the difficulties in performing robust onset detection and beat-tracking on a large collection of music with many different styles and instrumentations. If beat-tracking is noisy, e.g. if beat segments include chord transitions, then our chord labels will be prone to errors. Furthermore, it is very unlikely that the error distribution will correspond to values in Table 1, thus rendering our swap scoring strategy useless. This is by no means a reflection on all beat-tracking strategies. These results could very well be due to the shortcomings of our beat-based analysis (see a description in [2]). However they do highlight the risks taken when segmenting prior to sequence estimation.

The results for the various values of δ are more predictable. As expected, an increase of δ, implying a lossy compression of the sequence, entails a decrease in performance. This can be seen for both frame and beat-based analysis. As a result, best performance overall is for frame-based analysis with the smallest downsampling factor ($\delta = 6$). It is also logical to expect that frame-based analysis without downsampling ($\delta = 1$) would perform even better, but this experiment takes too long to run under our current configuration.

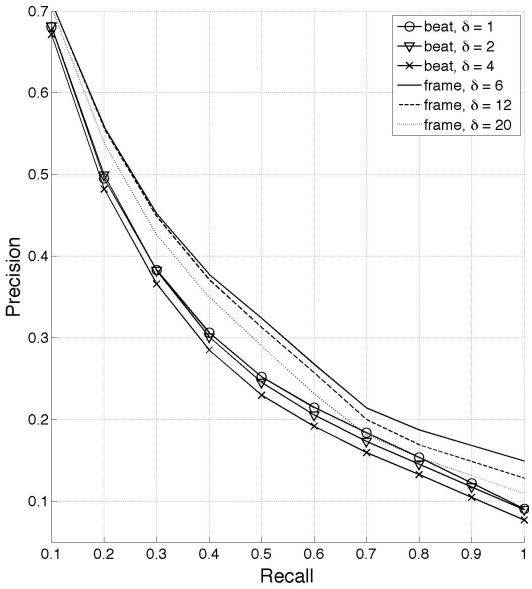

Figure 5. Detail of 11-point P/R graphs for variations of feature scope and downsampling factor δ

5 CONCLUSIONS

We present a solution to cover-song identification using approximate chord sequences and string alignment. More so than the approach itself, the emphasis is on the choice of a parameter set that: (i) helps us characterize the *essence* of cover songs independently of key, tempo or instrumentation; while (ii) taking into account the error-prone nature of chord sequences estimated from audio. Specifically, the paper contributes a systematic evaluation of key shifting, the cost of gap insertions and character swaps in string alignment, and the use of a beat-synchronous feature set. Results show that frame-based analysis consistently outperforms beat-synchronous segmentation, contradicting our intuition that such pre-processing could help overcome tempo differences between covers. We speculate, in the absence of a full evaluation, that this is due to the inability of our beat-based analysis to generalize to music of different styles and instrumentation. This negative result could be reversed in future implementations by the use of a more sophisticated beat-tracking system, such as the one used in [6]. Results also show that considerable improvement is brought about by pairwise key matching, moderately penalizing gaps and positively emphasizing swaps that are related to common confusions of our chord estimation algorithm. These results support the view that string alignment, as a mechanism for audio-based retrieval, cannot be oblivious to the problems of robustly estimating musically-meaningful information from audio. Future research will concentrate on overcoming the limitations imposed by the high computational cost of the implemented approach (in excess of 100ms per pairwise comparison, resulting in 5+ minutes of computation per query). Possible solutions to this problem could include the use of efficient search methodologies such as *iterative deepening* [1], or the use of representative parts of a song (e.g. chorus) for comparison.

6 ACKNOWLEDGMENTS

The author would like to thank Tim Crawford, Jeremy Pickens, Matija Marolt, Agnieszka Rogniska and Ernest Li for their ideas and support during the development of this work.

7 REFERENCES

[1] N. Adams, D. Marquez, and G. Wakefield. Iterative deepening for melody alignment and retrieval. In *Proceedings of ISMIR-05, London, UK*, 2005.

[2] J.P. Bello and J. Pickens. A robust mid-level representation for harmonic content in music signals. In *Proceedings of ISMIR-05, London, UK.*, 2005.

[3] M. Casey and M. Slaney. The importance of sequences in music similarity. In *Proceedings of ICASSP-06, Toulouse, France*, 2006.

[4] S.A. de Carvalho. NeoBio: Bio-informatics algorithms in Java. http://neobio.sourceforge.net.

[5] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis*. Cambridge UP, 1998.

[6] D. Ellis and G. Poliner. Identifying cover songs with chroma features and dynamic programming beat tracking. In *Proceedings of ICASSP-07, Hawai'i, USA*, 2007.

[7] C. Harte, M.B. Sandler, S.A. Abdallah, and E. Gómez. Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of ISMIR-05, London, UK*, 2005.

[8] K. Lee. Identifying cover songs from audio using harmonic representation. In *MIREX task on Audio Cover Song ID*, 2006.

[9] K. Lemström. *String Matching Techniques for Music Retrieval*. PhD thesis, University of Helsinki, Department of Computer Science, 2000.

[10] M. Marolt. A mid-level melody-based representation for calculating audio similarity. In *Proceedings of ISMIR-06, Victoria, Canada*, 2006.

[11] M. Müller, F. Kurth, and M. Clausen. Audio matching via chroma-based statistical features. In *Proceedings of ISMIR-05, London, UK*, 2005.

[12] C. Meek and W.P. Birmingham. A comprehensive trainable error model for sung music queries. *J. Artif. Intell. Res. (JAIR)*, 22:57–91, 2004.

[13] R. Typke. *Music Retrieval based on Melodic Similarity*. PhD thesis, Utrecht University, Netherlands, 2007.