Automatic Marking of Musical Dictations By Applying the Edit Distance Algorithm On a Symbolic Music Representation

Guy Tremblay
Dépt. d'informatique
Université du Québec à Montréal
C.P. 8888, Succ. Centre-Ville
Montréal, QC, H3C 3P8, Canada
+1 514 987-3000 ext. 8213#
tremblay.guy@uqam.ca

France Champagne
Dépt. d'informatique, UQAM
Montréal, QC, Canada
fc@francechampagne.org

ABSTRACT

A key practice of basic musical training is the use of *musical dictations* for ear training and training in music writing. Marking such dictations for large groups of students can be a lot of work. In this paper, we present a tool that can help automate the marking of musical dictations.

The edit distance, which computes a similarity metric between two *strings*, has been used in various areas such as string/text analysis, protein/genome matching in biocomputing, and musical applications, for example, music retrieval or musicological analysis. The tool we present can be considered an application of the edit distance to the marking of musical dictations.

Computing an edit distance on musical scores requires using an appropriate symbolic representation. We use MusicXML, an XML application for standard Western music notation. Given an appropriate Document Type Definition for MusicXML, existing Java tools can be used to obtain a MusicXML parser. Such a parser, given appropriate input files, then generates an intermediate form (DOM object) on which analyses and transformations are performed in order to compute the edit distance. In turn, the edit distance is used to give a mark as well as identify some of the key errors.

Keywords
Symbolic music representation, marking of musical dictation, edit distance and sequence comparison.

1 INTRODUCTION

A key skill in basic musical education is the knowledge of *solfège*, that is, the ability to recognize the signs and symbols of musical notation and associate them with the sounds and rhythms they represent. One important pedagogical tool for learning the basic rules of music and solfège is the use of *musical dictation*. In a musical dictation exercise, the teacher plays a music piece and the students must recognize and write down, in standard Western music notation, the piece being played. The goal is to train the students in auditory recognition of musical pieces as well as train them in basic music writing.

Marking musical dictations, like any kind of marking, can be a tedious task, especially with large groups of students as typically found in introductory classes. The goal of the project presented in this paper is to develop a tool that will help music teachers evaluate and mark their students' music dictations. A longer-term goal for such a tool is to integrate it within an intelligent tutorial system, in order to identify students' recurring weaknesses in solfège and, thus, help improve their learning.

A key element of the kind of tool we envision is a software component that can detect the similarities and differences between two music scores, both encoded in a symbolic representation appropriate for Western music notation, that is, as sequences of *notes* with specific *pitch* and *duration*. Sequence comparison algorithms can be applied to such sequences, as typically used in text searching or biocomputing, to compute an appropriate metric that can be used for marking.

The paper is organized as follows. Section 2 discusses the basic characteristics of musical dictations and the need for an appropriate symbolic representation. The next section

presents basic concepts and work related with musical sequences comparisons based on the edit distance are presented. This is then followed in Section 4 by a presentation of the heuristics to be used for marking musical dictation and a discussion of how they can be turned into an appropriate edit distance. Section 5 presents the architecture of the musical dictation marking tool which we are currently developing, while Section 6 concludes and presents future work.

2 REPRESENTING MUSICAL DICTATIONS USING XML

2.1 Key Characteristics of Musical Dictations

In a musical dictation exercise, a music teacher plays a music piece, one fragment at a time; each fragment is played several times (typically, three or four), as the piece is generally not known to the students. The student, by listening to the various fragments, must then recognize and write down the piece in standard Western music notation.

Basic music dictations are generally one voice dictation, that is, any given dictation consists of a sequence of individual notes for a single voice, each note having a specific *pitch* and *duration*. The pitch comes from a *finite* alphabet (*do, re, mi, ...*) and the duration is some fraction of the *whole*. A musical fragment can thus be represented as a sequence of pairs (p, d), with $p \in \{do, re, mi, \ldots\}$ and $d \in \{\frac{1}{16}, \frac{2}{16}, \ldots, 1\}$.[1]

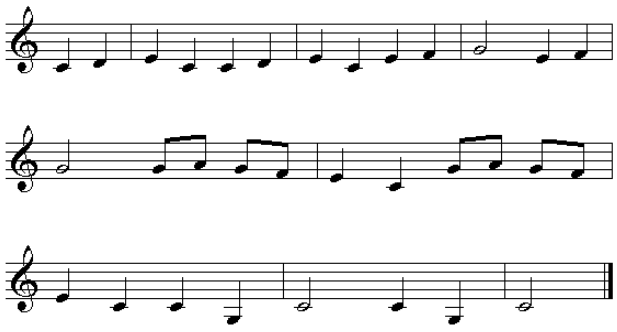

Figure 1. Musical score of *Frère Jacques*

The errors made by students while doing musical dictation exercises typically fall into a small number of categories [Beaudet, personal communication], for instance, errors regarding pitch (e.g., wrong note, missing or additional note), or errors regarding duration (e.g., wrong duration, split vs. combined note, wrong rhythm). For example, Figure 1 presents the correct score for the well-known French song "*Frère Jacques*", while Figure 2 illustrates a few typical mistakes that could have been made by a student.

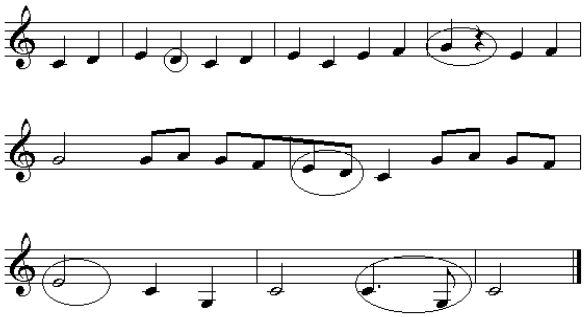

Figure 2. Typical errors for *Frère Jacques*

2.2 Symbolic Representation of Music Notation and MusicXML

In order to make the kind of comparative analysis required for marking musical dictations, an appropriate *symbolic* representation of music must be used. More precisely, both the exact pitch and duration of each note must be known, as well as the exact sequencing of the notes, and this information must be represented in a format that can be put in direct correspondence with standard Western music notation, with staff, key, notes, etc. A representation such as MIDI [23], thus, would not be appropriate: MIDI (Musical Instrument Digital Interface) is more of an exchange protocol for electronic instruments (which even attempts to describe details such as touch, attacks, nuances, etc.) than a purely symbolic notation.

Various symbolic representations for music scores have been proposed. SMDL [21] (Standard Music Description Language) is an attempt to define a general-purpose specification for music based on SGML (Standard Generalized Markup Language). However, due to its complexity, lack of reference manual and long development time, other simpler models were developed, with the result that, to our knowledge, no implementation was ever provided for SMDL. NIFF [17, 21] (Notation Interchange File Format) seems to be an interesting choice for our tool since it is "designed to allow the interchange of music notation data between and among music notation editing and publishing programs and music scanning programs." [17, p. 491] However, its adoption by industry has been very limited. The Humdrum format [11, 21] is also interesting since its goal is

[1] Note that a rest can simply be represented as a pair with a null pitch component and an appropriate duration.

to facilitate the implementation of analysis algorithms and is designed to allow multiple representations of musical fragments. Among these representations is the Kern Code which is designed for traditional Western musical notation. A Unix-based software development kit is available [17].

One important issue in the design of our tool is to eventually develop a web-based tutorial application that would include our program. A representation written in XML would be interesting and would allow us to develop a platform-independent tool with Java.

Over the recent years, interest has grown in using XML for representing various symbolic concepts. A number of XML applications have been developed in order to represent music notation, for example, 4ML [13], MML [22], and MusicXML [8]. Among these, MusicXML seems particularly interesting since it clearly aims at representing faithfully all the notions encountered in Western music notation. MusicXML is based in part on the Humdrum Kern format and describes most of its elements. Thus, it allows for the representation of all key notions, including alterations, dotted notes, triplets, lyrics, etc., as well as embodying a clean hierarchical approach (viz., score, part, measure, note). Some of the other notations appear less *symbolic* in that they seem to represent a kind of compromise between a purely symbolic notation and a digital one such as MIDI. MusicXML, at the time of this writing, also seems to be in a more stable specification state. Furthermore, a special plug-in for Finale [4], a music notation software with a graphical user interface, has recently been made available, making it possible to generate MusicXML files from music scores encoded with Finale.

With the various tools now available for XML and Java [6], using an XML representation with an appropriate DTD (Document Type Definition) makes it possible to automatically generate a parser which can read an XML representation of a music piece and can then generate a corresponding DOM (Document Object Model) object representing this music piece. This DOM object can then be manipulated using DOM-related operations, and appropriate analyses and transformations can be performed. This topic will be discussed in further detail in Section 5.

3 MUSICAL SEQUENCE COMPARISON

3.1 Detection of Similarities and the Edit Distance

Detecting similarities between two musical fragments has many applications. For instance, this can be used for identifying and retrieving musical pieces, as done in *query-by-humming* systems or in digital libraries of music [15, 2, 7, 19, 10]. Similarity detection tools can also be used for copyright infringement detection or musicological analysis [20, 12].

Given the context of our project, where a symbolic representation is a definite requirement, using a similarity measurement based on the edit distance between two sequences of notes is quite natural.

The edit distance is often used to compute a similarity metric between strings. Various forms of string edit distance have been used in various areas such as string and text analysis, protein and genome matching in bio-computing and signal processing [9, 18].

```
surgery
survery    -- Substitute g by v
survey     -- Suppress r
surveys    -- Insert s
```

Figure 3. Edit distance between "`surgery`" and "`surveys`"

The key idea behind the edit distance is to determine the minimum number of basic operations (e.g., insertions, deletions, substitutions) that can be applied to the first string in order to obtain the second. An example, presented in Figure 3, illustrates how the word "`surgery`" can be transformed into "`surveys`" using a minimal number of operations.

The general strategy for sequence comparison and edit distance computation uses dynamic programming. The cost function $C(i,j)$, which has to be minimized, can be defined as follows, where $\delta_{sup}, \delta_{ins}$ and δ_{subs} represent, respectively, the cost to suppress, insert, or $subs$titute a character:

$$
\begin{aligned}
C(0,0) &= 0 \\
C(i,0) &= C(i-1,0) + \delta_{sup}(\mathtt{A}[i]) \\
C(0,j) &= C(0,j-1) + \delta_{ins}(\mathtt{B}[j]) \\
C(i,j) &= \mathtt{min} \begin{cases} C(i-1,j-1) + \delta_{subs}(\mathtt{A}[i],\mathtt{B}[j]) \\ C(i-1,j) + \delta_{sup}(\mathtt{A}[i]) \\ C(i,j-1) + \delta_{ins}(\mathtt{B}[j]) \end{cases}
\end{aligned}
$$

More precisely, $C(i,j)$ represents the cost to go from the string $\mathtt{A}[1..i]$ to the string $\mathtt{B}[1..j]$, and the distance is the minimum cost of the operations required to make this transformation. Note that, for basic string operations, each operation is of unit cost, but it can be higher for other problems.

The asymptotic cost of this algorithm for sequences of length n is $O(n^2)$. This remains quadratic even when taking into account the special consolidation and fragmentation operations introduced for musical sequences [16], described below. For the introductory-level musical dictations we want to analyze, such a cost is clearly reasonable.

3.2 Edit Distance for Musical Sequences

Much research has been done on defining appropriate edit distances for musical fragments [16, 20, 7, 14, 1, 5]. However, to our knowledge, the present paper is the first one that tries to apply and adapt this algorithm to the specific problem of marking musical dictations.

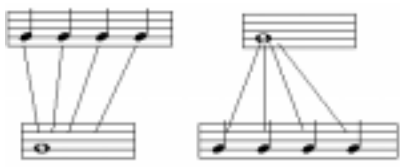

Figure 4. The consolidation and fragmentation operations of Mongeau and Sankoff [16]

Seminal work in the field of musical sequences comparison was done by Mongeau and Sankoff [16]. They used a weighted distance measure in which operations are assigned weights based on the harmonic distance between the notes and their relative *consonance*. Distinct weights are also attributed depending on whether the edit operation applies to pitch or tempo. Another interesting feature of their work is the introduction of special *consolidation* and *fragmentation* operations, as illustrated in Figure 4. Such operations are particularly interesting for our work, since these may correspond to typical student errors, where students correctly identify the pitch but do not clearly recall the exact rhythm. Consolidation would also apply when a student cannot recall a fragment and replaces it with rests or long note values.

In the context of music retrieval, where neither the exact base tonality nor the fine details of the rhythm are crucial to recognize a music piece, Lemström and Ukkonen [14] examine how the edit distance performs under tonality transposition. Thus, instead of comparing sequences of pitches, they manipulate sequences of interval differences (the distance between two consecutive notes) and conclude that an equivalent result could be obtained simply by using an appropriate distance and cost function.

Recently, Crochemore *et al.* [5] and Aucouturier and Sandler [1] have also presented work related with music retrieval. Both works deal with polyphonic melodies and do not deal directly with the specific notes and scores, working instead on the *texture* of the piece in order to recognize it under various interpretations.

Given the specific characteristics of our problem domain, the marking of musical dictations, the basic ideas introduced by Mongeau and Sankoff [16] are the most appropriate, so in the next section, we discuss how they can be adapted to our problem.

4 APPLYING THE EDIT DISTANCE TO THE MARKING OF MUSICAL DICTATIONS

4.1 A Heuristic for Marking Musical Dictations

In order to better understand how musical dictations are marked, we had a number of meetings with Ms. Luce Beaudet, a professor at the Faculty of Music (Université de Montréal) who is responsible for the music students' ear training.

Ms. Beaudet has pioneered an approach [3] to ear and solfège training where the emphasis is on understanding and analyzing a music piece based on the basic rules of harmony and common practice music. Thus, she proposes more of a macro-view to such training, where the students learn to recognize the key *patterns*, instead of simply being trained in *interval recognition* (identifying the distance between the current note and the previous one).

In basic ear training, the goal is to recognize the key patterns of tonal music, and so the rhythm is generally simple — more advanced courses put more emphasis on complex rhythm as well as atonal music. Thus, the heuristic used to mark musical dictations is that pitch errors are considered major (since they do impact the melody), whereas tempo errors are considered minor. Ms. Beaudet's approach to marking can be formalized as described below.

Let M_{pitch} and M_{tempo} be the contribution to the mark attributed to pitch errors and tempo errors, such that $M_{pitch} + M_{tempo} = 100$ (100 being the maximal mark, and with $M_{tempo} << M_{pitch}$). Let N_{notes} be the total number of notes appearing in the musical dictation and N_{beats} be the total number of beats. The weights W_{pitch} and W_{tempo} associated with each kind of error is then defined as follows:

$$\begin{aligned} W_{pitch} &= M_{pitch}/N_{notes} \\ W_{tempo} &= M_{tempo}/N_{beats} \end{aligned}$$

Typical values used by Ms. Beaudet in her introductory classes are $M_{pitch} = 88$ and $M_{tempo} = 12$. For example, a student who makes 10 pitch errors and 3 tempo errors in the "*Frère Jacques*" musical dictation (with 32 notes and 32 beats [2]) would get the following mark:

Pitch errors	$10 * (88/32)$	=	27.5
Tempo errors	$5 * (12/32)$	=	1.9
Total errors		=	29.4
Final mark		=	70.6

In our heuristic, the cost of a pitch error is constant and does not depend on the value of the note, so we define it as

[2] In general, N_{notes} and N_{beats} are distinct, although in this particular example they happen to be the same.

$$
\begin{aligned}
C(0,0) &= 0 \\
C(i,0) &= C(i-1,0) + \delta_{sup}(\mathtt{A}[i]) \\
C(0,j) &= C(0,j-1) + \delta_{ins}(\mathtt{B}[j]) \\
C(i,j) &= \mathtt{min} \begin{cases} C(i-1,j-1) + \delta_{subs}(\mathtt{A}[i],\mathtt{B}[j]) \\ C(i-1,j) + \delta_{sup}(\mathtt{A}[i]) \\ C(i,j-1) + \delta_{ins}(\mathtt{B}[j]) \\ C(i-k,j-1) + \delta_{cons}(\mathtt{A}[i-k+1..i],\mathtt{B}[j]), 2 \leq k \leq i \\ C(i-1,j-k) + \delta_{frag}(\mathtt{A}[i],\mathtt{B}[j-k+1..j]), 2 \leq k \leq j \end{cases}
\end{aligned}
$$

Figure 5. Equations for edit distance (adapted from [16])

follows:

$$C_{pitch} = W_{pitch}$$

Rhythmic errors are computed on a beat basis. However, our sequence representation of a musical fragment, as described in Section 2.1, is based on notes instead of beats. Consequently, the duration of each note must be expressed in terms of the number of beats. Let *b* be the beat unit of a musical dictation, that is, the duration of one beat. Then, we define the cost of a tempo error for a note as follows:

$$C_{tempo}(p,d) = \frac{d}{b} * W_{tempo}$$

Then let E_{pitch} be the number of pitch errors made by a student. Let E_{tempo} be the number of notes containing rhythmic errors and $\{(p_1,d_1),(p_2,d_2),\ldots,(p_{E_{tempo}},d_{E_{tempo}})\}$ be the collection of these notes (more precisely, a multiset). The total penalty associated with the pitch and tempo errors made by the student is the following:

$$E_{pitch} * C_{pitch} + \left(\sum_{i=1}^{E_{tempo}} C_{tempo}(p_i,d_i) \right)$$

The final mark attributed to S's dictation is then:

$$100 - \left(E_{pitch} * C_{pitch} + \left(\sum_{i=1}^{E_{tempo}} C_{tempo}(p_i,d_i) \right) \right)$$

4.2 Turning the Marking Heuristic Into an Edit Distance

To transform the marking heuristic described above into an appropriate edit distance, we adapt the edit distance initially proposed by Mongeau and Sankoff [16]. The basic set of equations for the edit distance is presented in Figure 5, where the weights (the δ functions) have been modified for the marking heuristic as described in the following.

First, let Eq_p be a function that returns 1 when two pitches are equal and 0 otherwise, and similarly for Eq_d, which returns 1 when two durations are equal.

The cost of both $\delta_{sup}(\mathtt{A}[i])$ and $\delta_{ins}(\mathtt{A}[i])$ is the following:

$$C_{pitch} + C_{tempo}(\mathtt{A}[i])$$

The cost of $\delta_{subs}(\mathtt{A}[i],\mathtt{B}[j])$ is defined as follows:

$$Eq_p(\mathtt{A}[i],\mathtt{B}[j]) * C_{pitch} + Eq_d(\mathtt{A}[i],\mathtt{B}[j]) * C_{tempo}(\mathtt{B}[j])$$

We also introduce two cost functions associated with consolidation and fragmentation operations: δ_{cons} and δ_{frag} represent the cost, respectively, to replace a sequence of notes with a single note, and to replace one note with a sequence of notes. We only define the consolidation operation's cost in more detail, since the other is symmetric.

In order to replace a group of short notes with one long note, both sides need to have the same duration (i.e., the duration of the long note must be equal to the total duration of the group of short notes). When this is the case, the cost $\delta_{cons}(\mathtt{A}[i-k+1..i],\mathtt{B}[j])$ can be defined as follows:

$$\left(\sum_{l=i-k+1}^{i} Eq_p(\mathtt{A}[l],\mathtt{B}[j]) * C_{pitch} \right) + C_{tempo}(\mathtt{B}[j])$$

If the durations are not equal, then the cost function simply returns an infinite value, in order to preclude the consolidation/fragmentation operation from being selected.

5 A TOOL FOR MARKING MUSICAL DICTATION

Figure 6 presents the overall architecture of the musical dictation marking tool we are currently developing. We assume that the input to the tool is a pair of MusicXML files representing the musical dictation written by the professor and the student's answer. These XML files can be obtained in different ways, for instance, the Finale tool [4] now has an option to save a music score into MusicXML format.

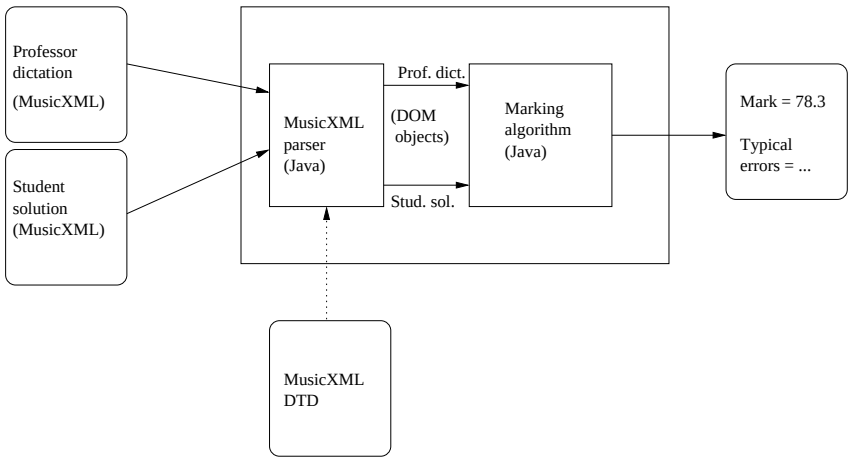

Figure 6. General architecture of our musical dictation marking tool

From a DTD for MusicXML, it is possible to generate, using an appropriate parser generation tool [6], a parser (written in Java) which can read and parse a MusicXML input file, and then construct a DOM object — essentially a syntax tree — representing the professor's dictation as well as the student's solution.

In turn, these DOM objects are analyzed and transformed into a simpler sequence representation of the musical fragments appropriate for the edit distance computation algorithm. This algorithm, still under implementation, is also being developed in Java. Based on the edit distance, the student's mark can then be computed and, from the matrix obtained through the dynamic programming implementation of the edit distance computation, the key errors made by the student can be identified.

It is important to stress that the marking strategy is indeed a *heuristic*, which means some tuning-up might be required — for example, by varying the weights and costs (see Section 4.2) associated with the various types of errors. Of course, the marks computed by the tool will also have to be validated. First, this will be done by using simple test cases (e.g., variations on *Frère Jacques*). Then, we intend to use real dictations written by students in an introductory solfège course taught at the Faculty of Music at Université de Montréal and compare the marks produced by our tool with those given by our domain expert.

6 CONCLUSION AND FUTURE WORK

In this paper, we have described a software tool to help music teachers mark their students' music dictation. Given a standard music notation tool that generates an appropriate XML representation, we are able to parse and transform this input into sequences for which an edit distance can be computed in order to give a mark.

As mentioned earlier, our short-term goal is to be able to compute fair marks for students' musical dictations based on a marking heuristic implemented using an appropriately tuned edit distance. The edit distance implementation strategy (dynamic programming) will also make it possible to provide some feedback to the students on the typical errors they have made.

A longer-term goal is to develop a more general tool and integrate it within an intelligent tutoring system. In such a context, the tool can be adapted to the students' level; for example, musical dictations of varying degrees of difficulty could be proposed to the students based on their identified weaknesses.

ACKNOWLEDGMENTS

Special thanks to Prof. Chuck Wallace (Michigan Technical University, MI) for his detailed comments on the initial version of this paper that (hopefully) helped improve the English-writing. Support for Ms. Champagne's work was provided by an NSERC's (Natural Sciences and Engineering Research Council of Canada) student scholarship.

REFERENCES

1. J.-J. Aucouturier and M. Sandler. Using long-term structure to retrieve music: Representation and matching. In *ISMIR*, June 2001.
2. D. Bainbridge and al. Towards a digital library of popular music. In E. Fox and N. Rowe, editors, *Digital Libraries*, pages 161–169, 1999.
3. L. Beaudet. Préalable à l'acte d'audition tonale. Technical report, Université de Montréal, Faculté de musique, September 1980.

4. Coda Music. Finale 2002. `www.codamusic.com/coda/fin2002.asp`, 2002.
5. M. Crochemore and al. Approximate string matching in musical sequences. In M. Baliik and M. Simanek, editors, *PSC'2001, Prague Stringoly Club*. Czech Technical University of Prague, 2001.
6. M. Daconta and A. Sagnich. *XML Development with Java 2*. Sams Publishing, 2000.
7. C. Francu and C. Nevill-Manning. Distance metrics and indexing strategies for a digital library of popular music. In *IEEE International Conference on Multimedia and Expo (ICME)*, July 2000.
8. M. Good. Music XML. `www.recordare.com`, 2001.
9. D. Gusfield. *Algorithms on strings, trees, and sequences.* Cambridge University Press, 1997.
10. G. Haus and E. Pollastri. An audio front end for query-by-humming systems. In *2nd International Symposium on Music Information Retreival (ISMIR'01)*, Bloomington, Illinois, October 2001.
11. D. Huron. *The Humdrum Toolkit: Reference Manual*. Center for Computer Assisted Research in the Humanities, 1991.
12. A. Kornstädt. The *JRing* system for computer-assisted musicological analysis. In *2nd International Symposium on Music Information Retreival (ISMIR'01)*, Bloomington, Illinois, October 2001.
13. M. Legh and L. Montgomery. 4ML. `www.4ml.org`, 2001.
14. K. Lemström and E. Ukkonen. Including interval encoding into edit distance based music comparison and retrieval. In *AISB'2000 Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science*, pages 53–60, Birmingham, United Kingkom, 2000.
15. McNab and al. The New Zealand digital library MELody inDEX. *D-Lib Magazine*, 1997.
16. M. Mongeau and D. Sankoff. Comparison of musical sequences. *Computers and the Humanities*, 24(3):161–175, 1990.
17. S. Mounce. NIFF page. `www.student.brad.ac.uk/srmounce/niff.html`, 1997.
18. G. Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88, March 2001.
19. D. Ó Maidín and M. Fernström. The best of two worlds: retrieving and browsing. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00), Verona, Italy*, December 2000.
20. K. Orpen and D. Huron. Measurement of similarity in music: A quantitative approach for non-parametric representations. *Computers in Music Research*, 4:1–44, 1992.
21. E. Selfridge-Field, editor. *Beyound (MIDI) – The Handbook of Musical Codes*. MIT Press, 1997.
22. J. Steyn. Music markup language (MML). `www.mmlxml.org`, 2001.
23. E. Tholomé. *MIDI à votre portée*. Editions Radio, 1990.