# Query by Humming: How good can it get?

Bryan Pardo
EECS Dept, University of Michigan
153 ATL, 1101 Beal Avenue
Ann Arbor, MI 48109-2110
+1 (734) 369-3207
bryanp@umich.edu

William P. Birmingham
EECS Dept, University of Michigan
110 ATL, 1101 Beal Avenue
Ann Arbor, MI 48109-2110
+1 (734) 936-1590
wpb@umich.edu

## ABSTRACT
When explaining the Query-by-humming (QBH) task, it is typical to describe it in terms of a musical question posed to a human expert, such as a music-store clerk. An evaluation of human performance on the task can shed light on how well one can reasonably expect an automated QBH system to perform. This paper describes a simple example experiment comparing three QBH systems to three human listeners. The systems compared depend on either a dynamic-programming implementation of probabilistic string matching, or hidden Markov models. While results are preliminary, they indicate existing string matching and Markov model performance does not currently achieve human-level performance.

## 1. INTRODUCTION
Our research group is interested in Query-by-humming (QBH) systems that allow users to pose queries by singing or humming them. QBH systems search musical content. This is in contrast to approaches to music retrieval based on searching metadata, such as song title, genre, and so forth.

The Music Information Retrieval (MIR) community has, of late, been rightly concerned with finding meaningful empirical measures for the quality of a MIR system that searches musical content. Those interested in such systems have focused on building large databases of monophonic songs. Experimental queries are often synthetic, and are generated from elements of the database [1-5]. Unfortunately, differences in database composition, representation, query transcription methods, ranking methods, and methodology in evaluation of results makes comparison between systems difficult, if not impossible. Further, we are unaware of any direct comparisons between automated MIR system performance and human performance.

When describing the QBH task, it is typical to describe it in terms of a musical question posed to a human expert, such as a music-store clerk. An evaluation of human performance on the task can shed light on how well one can reasonably expect an automated QBH system to perform. How well, then, could one expect a human to perform? One can compare human and algorithmic performance, if database is limited to a set of pieces known to the human and the task is limited to that of "name that tune," rather than ranking the full database.

This paper describes a simple example experiment to compare three QBH systems against three human listeners. While the results are preliminary, it shows how to establish a human performance baseline for QBH systems.



## 2. THE SEQUENCE MATCHERS USED
A *string* is any sequence of characters drawn from an alphabet, such as a sequence of notes in a written musical score, or notes transcribed from a sung query. String matchers find the best alignment between string $Q$ and string $T$ by finding the lowest cost (or, equivalently, highest reward) transformation of $Q$ into $T$ in terms of operations (matching or skipping characters). The score of the best alignment can be used as a measure of the similarity of two strings.

Dynamic-programming based implementations that search for a good alignment of two strings have been used for over 30 years to align gene sequences based on a common ancestor [6], and have also been used in musical score following [7, 8, 9 ] and query matching [10, 4]. For this experiment, we chose to use the probabilistic string matcher described in [11] using both the global alignment algorithm and the local alignment algorithm, as described in [12].

Hidden Markov models, or HMMs, have been used relatively infrequently in the MIR literature [2, 13] to do melody recognition, but seem promising. We chose to represent targets using the HMM architecture described in Shifrin et al. The Forward algorithm [14] measures the similarity of a target string, represented as an HMM, to a query string by generating the probability the target generated the query.

Given a query, $Q$, and a set of targets, $\{T_1 \ldots T_n\}$, an order may be imposed on the set of targets by running the same scoring algorithm (global, local, or Forward) between Q and each target, $T_i$, and then ordering the set by the value returned, placing higher values before lower. We take this rank order to be a direct measure of the relative similarity between a theme and a query. The $i$th target in the ordered set is then the $i$th most like the query. Thus, the first target is the one most similar to the query, according to the given scoring algorithm.

## 3. EXPERIMENTAL SETUP
Figure 1 outlines the experimental setup used to compare our system to the three human listeners. As can be seen from the figure, humans had the advantage in that they listened directly to recorded queries, rather than a sonification (such as a MIDI performance) of the transcribed queries. This was because, for this experiment, we were interested in the maximal performance that could be achieved by a human, without introducing error from pitch tracking, pitch quantization, and note segmentation. We note that the algorithms had to deal with errors introduced from these processing steps, which can be substantial.

The remainder of this section describes our simple experiment to compare the performance of our complete string matcher and HMM systems against human performance.

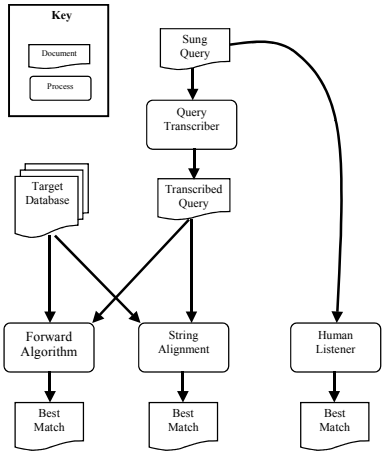


**Figure 1. Experimental Setup**

## 3.1 Transcribed Target Database

We used a corpus of 260 pieces of music encoded as MIDI from public domain sites on the Internet. The corpus is composed entirely of pieces that have been recorded by the Beatles. This includes all pieces recorded on albums for U.S. release and a number of "covers" they performed that were originally composed by other artists, such as "Roll Over Beethoven." For a full list of pieces in the database, please consult the MusEn website at http://musen.engin.umich.edu/.

We selected music performed by the Beatles because their music tends to be well known, the salient information to identify pieces tends to be melodic and relatively easy to sing, and the pieces are readily accessible, both as audio and as MIDI. Each piece in the corpus was represented in a database by a set of themes, or representative monophonic melodic fragments. The number of distinct "catchy hooks" decided the number of themes chosen to represent each piece. Of the pieces, 238 were represented by a single theme, 20 by two themes, and two pieces were represented by three themes, resulting in a database of 284 monophonic themes. These themes constitute the set of targets in the database.

A sequence of <pitch-interval, InterOnsetInterval-ratio> {Pardo, 2002 #227} pairs was created and stored in the database for each MIDI theme. Themes were quantized to 25 pitch intervals and five log-spaced InterOnsetInterval-ratio intervals. Each theme was indexed by the piece from which it was derived. An HMM for each theme was then generated automatically from the theme sequence and placed in the database.

## 3.2 Query Corpus

A query is a monophonic melody sung by a single person. Singers were asked to select one syllable, such as "ta" or "la", and use it consistently for the duration of a single query. The consistent use of a single consonant-vowel pairing was intended to minimize pitch-tracker error by providing a clear starting point for each note, as well as reducing error caused by dipthongs and vocalic variation.

Three male singers generated queries for the experiment. Singer 1 was a twenty-two year old male with no musical training beyond private instrumental lessons as a child. Singer 2 was a twenty-seven year old male with a graduate degree in cello performance. Singer 3 was a thirty-five year old male with a graduate degree in saxophone performance. None are trained vocalists. All are North American native speakers of English.

Sung queries were recorded in 8 bit, 22.5 kHz mono using an Audio-Technica AT822 microphone from a distance of roughly six inches. Recordings were made directly to an IBM ThinkPad T21 laptop using its built-in audio recording hardware and were stored as uncompressed PCM .wav files.

Each singer was allowed a trial recording to get a feel for the process, where the recorded melody was played back to the singer. This trial was not used in the experimental data. Subsequent recordings were not played back to the singer. Once the trial recording was finished, each singer was presented with a list containing the title of each of the 260 Beatles recordings in our database. Each singer was then asked to sing a portion of every song on the list that he could. Singer 1 sang 28 songs; Singer 2 sang 17; and, Singer 3 sang 28 songs. The result was a corpus of 73 queries covering forty-one of the Beatles' songs, or roughly 1/6 of the songs in our database. Singers 1 and 3 sang 24 songs in common. Singer 2 had 8 songs in common with the other two. These 8 songs were common to all 3 singers. These songs are 'A Hard Days Night,' 'All You Need Is Love,' 'Here Comes The Sun,' 'Hey Jude,' 'Lucy In The Sky With Diamonds,' 'Ob-La-Di Ob-La-Da,' 'Penny Lane,' and 'Sgt. Peppers Lonely Hearts Club Band.'

These queries were then automatically pitch tracked, segmented and quantized to 25 pitch intervals and five IOI ratio intervals. This resulted in 73 query strings. These were used as the query set for all experiments. Mean query length was 17.8 intervals. The median length was 16. The longest query had 49 intervals, and the shortest had only two. The median number of unique elements per query sequence was nine.

## 3.3 Experimental Results

We define the recognition rate for a system to be the percentage of queries where the correct target was chosen as the top pick. The highest recognition rate achieved by any of our systems was 71% by the local-string matcher on Singer 3. The lowest rate was 21% by the Forward algorithm on Singer 1. We created a baseline by presenting the sung queries to the singers who generated the query set to see how many of them would be recognized.

Two months after the queries were made, the three singers were gathered into a room and presented the original recordings of the queries in a random order. Each recording was presented once, in its entirety. Singers were told that each one was a sung excerpt of a piece of music performed by the Beatles and that the task was to write down the name of the Beatles song the person in the recording was trying to sing. Only one answer was allowed per song and singers were given a few (no more than 15) seconds after each query to write down an answer.

Once all queries had been heard, responses were graded. Recall that queries were sung with nonsense syllables and that lyrics were not used. Because of this, we judged any response that

contained a portion of the correct title or a quote of the lyrics of the song as a correct answer. All other answers were considered wrong.

Table 1 contains the results of the human trials, along with the results for the automated QBH systems. As with the human trials, the automated algorithms were judged correct if the right answer was ranked first and incorrect otherwise. Each column represents the results for a query set. Each row in the table contains the recognition rates achieved by a particular listener or QBH system. The row labeled "Other 2 Singers" contains the average recognition rates of the two singers who did NOT sing a particular set of queries. Thus, for Singer 2's queries, the "Other 2 Singers" value is the average of how well Singer 1 and Singer 3 recognized Singer 2's queries.

**Table 1. Human Performance vs. Machine Performance**

| | **Singer 1** | **Singer 2** | **Singer 3** | **Mean** |
|---|---|---|---|---|
| Singer 1 | 96% | 71% | 79% | 82% |
| Singer 2 | 50% | 82% | 46% | 59% |
| Singer 3 | 71% | 76% | 89% | 79% |
| Other2 Singers | 61% | 74% | 63% | 66% |
| String Matcher (Global) | 29% | 24% | 39% | 31% |
| String Matcher (Local) | 36% | 41% | 71% | 49% |
| HMM (Forward) | 21% | 35% | 68% | 41% |
| N | 28 | 17 | 28 | |

It is interesting to note that the human listeners achieved an average recognition rate of 66%, when presented with queries sung by another person. This figure was lower than expected and may provide a rough estimate to how well one can expect a machine system to do. Even more interesting was the inability of Singers 2 and 3, both with graduate degrees in music performance, to achieve even a 90% recognition rate on their own queries, while Singer 1 achieved a much higher recognition rate on his own queries.

# 4. CONCLUSIONS

This paper describes an experiment comparing three QBH systems to three human listeners. The systems compared depend on either a dynamic-programming implementations of probabilistic string matching, or hidden Markov models. While results are preliminary, they indicate existing string matching and Markov model performance does not currently achieve human-level performance. Future work in this project includes collecting more queries and listeners and having listeners attempt to recognize pieces from audio generated from query transcriptions.

# 5. ACKNOWLEDGMENTS

We gratefully acknowledge the support of the National Science Foundation under grant IIS-0085945, and The University of Michigan College of Engineering seed grant to the MusEn project. The opinions in this paper are solely those of the authors and do not necessarily reflect the opinions of the funding agencies.

# 6. REFERENCES

[1] McNab, R., et al. Towards the Digital Music Library: Tune Retrieval from Acoustic Input. in First ACM International Conference on Digital Libraries. 1996. Bethesda, MD.

[2] Meek, C. and W.P. Birmingham. Johnny Can't Sing: A Comprehensive Error Model for Sung Music Queries. in ISMIR 2002. 2002. Paris, France.

[3] Pickens, J. A Comparison of Language Modeling and Probabilistic Text Information Retrieval. in International Symposium on Music Information Retrieval. 2000. Plymouth, Massachusetts.

[4] Uitdenbogerd, A. and J. Zobel. Melodic Matching Techniques for Large Music Databases. in the Seventh ACM International Conference on Multimedia. 1999. Orlando, FL.

[5] Downie, S. and M. Nelson. Evaluation of a Simple and Effective Music Information Retrieval Method. in the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2000. Athens, Greece.

[6] Needleman, S.B. and C.D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology, 1970. 48: p. 443-453.

[7] Dannenberg, R. An On-Line Algorithm for Real-Time Accompaniment. in International Computer Music Conference. 1984: International Computer Music Association.

[8] Puckette, M. and C. Lippe. Score Following In Practice. in International Computer Music Conference. 1992: International Computer Music Association.

[9] Pardo, B. and W.P. Birmingham. Following a musical performance from a partially specified score. in Multimedia Technology Applications Conference. 2001. Irvine, CA.

[10] Hu, N., R. Dannenberg, and A. Lewis. A Probabilistic Model of Melodic Similarity. in International Computer Music Conference (ICMC). 2002. Goteborg, Sweden: The International Computer Music Association.

[11] Pardo, B. and W. Birmingham. Improved Score Following for Acoustic Performances. in International Computer Music Conference (ICMC). 2002. Goteborg, Sweden: The International Computer Music Association.

[12] Durbin, R., et al., Biological Sequence Analysis, Probabilistic models of proteins and nucleic acids. 1998, Cambridge, U.K.: Cambridge University Press.

[13] Shifrin, J., B. Pardo, and W. Birmingham. HMM-Based Musical Query Retrieval. in Joint Conference on Digital Libraries. 2002. Portland, Oregon.

[14] Rabiner, L. and B.-H. Juang, Fundamentals of Speech Recognition. 1993, Englewood Cliffs, New Jersey: Prentice-Hall.