

# Chordify: Chord transcription for the masses

W. Bas de Haas<sup>1,3</sup>, José Pedro Magalhães<sup>2,3</sup>,

Dion ten Heggeler<sup>3</sup>, Gijs Bekenkamp<sup>3</sup>, Tijmen Ruizendaal<sup>3</sup>

<sup>1</sup>Department of Information and Computing Sciences, Utrecht University

<sup>2</sup>Department of Computer Science, University of Oxford

<sup>3</sup>Chordify BV

{bas, pedro, dion, gijs, tijmen}@chordify.net

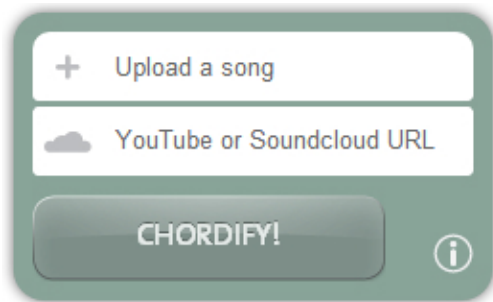


Figure 1: Musical source submission dialogue at the Chordify homepage.

## 1 Introduction

Chordify is a web-based music player that strives to help people play any song on a (harmony based) musical instrument. Our web-service automatically transcribes the chord labels from an arbitrary audio source, and presents the chords to the user in an intuitive manner. Next, the chords are used as an interface to the music: during playback, a cursor highlights the currently played chord, and users can select the chords in the sequence to play or loop any fragment of the audio source.

Increasingly many music students mainly use online streaming services like SoundCloud or YouTube to practice songs they would like to play[5]. Therefore, a user can not only upload a personal audio file to Chordify, but also point to a YouTube or SoundCloud source. Chordify has been built on existing Music Information Retrieval (MIR) technology, and aims at making this technology available to a less technically oriented audience.

One year after its initial release, Chordify has analysed the chords of about one million songs. Hence, it offers typical *big data* challenge to digital humanities researchers that are interested in studying tonal harmony in large.

## 2 Interface

If you visit the Chordify<sup>1</sup> webpage, you will be prompted with the dialogue displayed in Figure 1. This dialogue allows you to do two things: upload a music file from your local computer, or paste a URL from YouTube<sup>2</sup> or SoundCloud<sup>3</sup>. Chordify will then automatically transcribe the chords from the supplied audio source and display them to the user (see Figure 2 for an example). The chord interface is designed to be intuitive: every beat is represented by a square, and chord changes are denoted with chord labels. When the song is played, a cursor highlights the current beat position, telling the user which chord to play at that position in the piece.

Users can navigate through the song by clicking any of the squares. The music player will jump to the clicked position in the song, and start or continue playback from that beat on. The audio or video playback can also be controlled by the keyboard, and it is possible to select and loop certain sections of the musical piece. Recently, we have extended Chordify by adding premium features that allow the user to transpose the chords, listen to the extracted chords, speed up or slow down the playback of audio, manage collection of *chordified* songs, and download the chords in PDF or MIDI format.

## 3 Technology

The aim of Chordify is to make state-of-the-art music technology accessible to a broader audience. Under the hood, the Haskell program HarmTrace [3, 1] uses spectral features and downbeat positions to compute chord candidates at every beat position. Next, a model of Western tonal harmony is used to select the final chords[2, 4]: at beat positions where the audio matches a particular chord well, that chord is selected for the final transcription. However, when this is not the case, the HarmTrace harmony model will select a

<sup>1</sup><http://chordify.net>

<sup>2</sup><http://youtube.com>

<sup>3</sup><http://soundcloud.com>

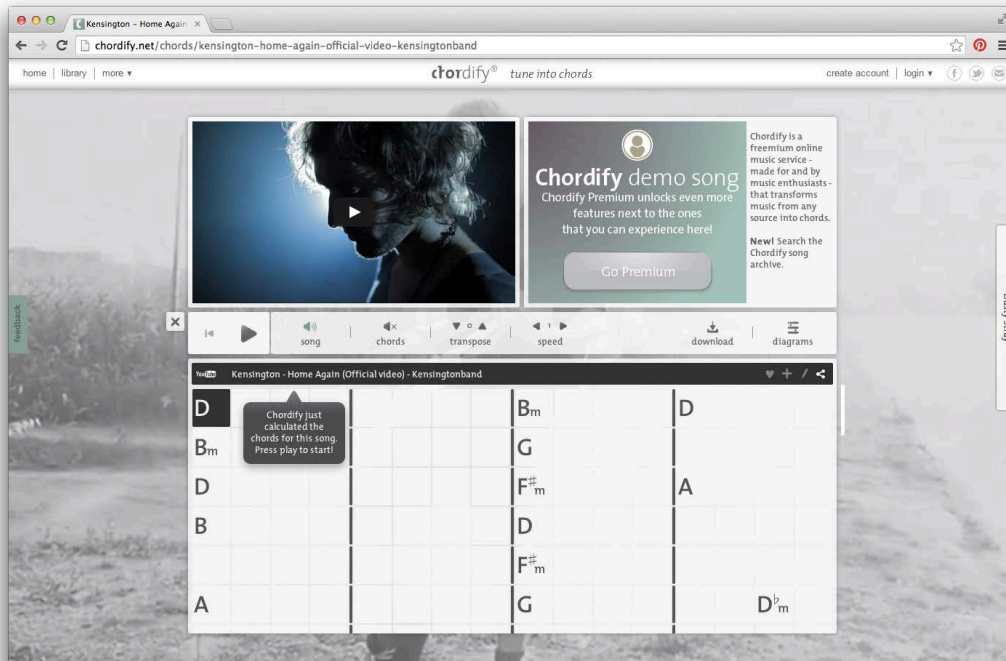


Figure 2: A screenshot of the Chordify interface displaying the chords and YouTube video of the song *Home Again* by Kensington. The video is displayed in the top pane, the chords are displayed in the bottom pane, and in between the two panes you find the playback controls. Within the chord sequence, the squares represent beat positions, the black bars are bar lines, and chord labels denote chord changes. The dark square is the cursor, highlighting the playback position within the song.

chord based on how well it fits the rules of tonal harmony.

school music classes. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 347–352, 2011.

## References

- [1] W. B. de Haas, J. P. Magalhães, and F. Wiering. Improving audio chord transcription by exploiting harmonic and metric knowledge. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, 2012.
- [2] W. B. de Haas, J. P. Magalhães, and R.C. Wiering, F. and Veltkamp. Automatic functional harmonic analysis. *Computer Music Journal*, 37(4):37–53, 2014.
- [3] J. P. Magalhães and W. B. de Haas. Functional modelling of musical harmony—an experience report. In *Proceedings of the International Conference on Functional Programming*, pages 156–162, 2011.
- [4] M. Rohrmeier. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*, 5(1):35–53, 2011.
- [5] D. Stowell and S. Dixon. MIR in school? Lessons from ethnographic observation of secondary