

---

**abracadabra**

**Cameron MacLeod**

**Feb 19, 2022**



**CONTENTS:**

<b>1</b>	<b>What can you use it for?</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Usage as a script</b>	<b>7</b>
<b>4</b>	<b>Usage as a library</b>	<b>9</b>
<b>5</b>	<b>Issues and contributing</b>	<b>11</b>
5.1	Getting Started . . . . .	11
5.1.1	Installation . . . . .	11
5.1.2	Basic usage . . . . .	12
5.2	abracadabra . . . . .	12
5.2.1	abracadabra package . . . . .	12
5.3	Documentation . . . . .	12
5.3.1	Pre-requisites . . . . .	12
5.3.2	Building the docs . . . . .	13
<b>6</b>	<b>Indices and tables</b>	<b>15</b>



*abracadabra à la Shazam*

abracadabra is sound recogniser written in Python. It is an implementation of the Shazam paper: [An Industrial Strength Audio Search Algorithm](#).

Read the [docs here](#) or read [an explanation of how it works](#).



## **WHAT CAN YOU USE IT FOR?**

abracadabra works like Shazam. You register songs in advance and then later you can use your computer's microphone to identify what song is playing. It could be used (as part of another system) to:

- Align multiple videos of the same event using the audio
- De-duplicate your music library





## INSTALLATION

First, clone or download this repository:

```
git clone https://github.com/notexactlyawe/abracadabra.git
```

Next, install the dependencies abracadabra relies on. On Ubuntu you can install them with the following line:

```
sudo apt-get install gcc portaudio19-dev python3-dev ffmpeg
```

Now you can use pip to install the project:

```
cd abracadabra  
pip install .
```



## USAGE AS A SCRIPT

Installing the project through pip will install the `song_recogniser` script. To see all the options you can pass to the script, run the following:

```
song_recogniser --help
```

Below is an example of how you can use `song_recogniser`:

```
$ song_recogniser initialise
Initialised DB
$ song_recogniser register ~/Music/CoolArtist/AwesomeAlbum
$ song_recogniser recognise --listen # records a 10 second clip for recognition
ALSA ...
* recording
* done recording
('CoolArtist', 'AwesomeAlbum', 'SweetTrack')
```



## USAGE AS A LIBRARY

You can use abracadabra as part of your own project by using it as a library. The main modules you'll be interested in are the [recognise](#) and [settings](#) modules.

Most functions in the library are documented. If you want to use lower-level components in your project, please take a look at the [docs](#).



## ISSUES AND CONTRIBUTING

If you encounter an issue with abracadabra or have a suggestion for improving the project, please [create an issue](#)!

Pull requests are welcome, but please create an issue first to discuss what you intend to do.

---

This project is maintained by [Cameron MacLeod](#).

---

### 5.1 Getting Started

#### 5.1.1 Installation

First, create a virtual environment to install the dependencies and activate it:

```
virtualenv -p python3 venv  
source venv/bin/activate
```

Next, install the pre-requisites for the Python packages. On Ubuntu, you can do this using:

```
sudo apt-get install gcc portaudio19-dev python3-dev ffmpeg
```

Now install the project with:

```
pip install .
```

If you have problems with dependency clashes, then try running the following:

```
pip install -r requirements.txt  
pip install .
```

`requirements.txt` contains a set of dependencies that are known to work together.

## 5.1.2 Basic usage

abracadabra is best used as a library in other projects, but a simple script for song recognition is included. Once you've installed abracadabra, then the script will be available by running the following command:

```
song_recogniser --help
```

song\_recogniser provides three commands that you can use.

- **initialise**: Creates the database in which fingerprints are stored.
- **register**: Takes a file or a directory and registers it to the database.
- **recognise**: Recognises a song. Takes a filename, or listens using your computer's microphone.

Below is an example where song\_recogniser is used:

```
$ song_recogniser initialise
Initialised DB
$ song_recogniser register ~/Music/CoolArtist/AwesomeAlbum
$ song_recogniser recognise --listen # records a 10 second clip for recognition
ALSA ...
* recording
* done recording
('CoolArtist', 'AwesomeAlbum', 'SweetTrack')
```

The source code for this simple application is under abracadabra/scripts/song\_recogniser.py.

## 5.2 abracadabra

### 5.2.1 abracadabra package

#### Submodules

**abracadabra.fingerprint module**

**abracadabra.recognise module**

**abracadabra.record module**

**abracadabra.settings module**

**abracadabra.storage module**

#### Module contents

## 5.3 Documentation

### 5.3.1 Pre-requisites

The abracadabra docs are written using [Sphinx](#). Everything you need to work on the docs is described in docs/requirements.txt.



```
cd docs
virtualenv -p python3 venv
source venv/bin/activate
pip install -r requirements.txt
```

### 5.3.2 Building the docs

The docs come with a Makefile that builds the docs for you. The easiest way to get started writing the docs is to run the `livehtml` command.

```
make livehtml
```

This will start a live-reloading server that is accessible from [localhost:8000](http://localhost:8000).



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`