# Part 1: From Raw Audio to Source Separated Audio

For Durham singers (AG, CC, SCh) we used **Spleeter source separation** (4 stem model)

For Pune singers (AK, AP, MG, MP, NM, RV, SM, SS) we used **Audacity Noise Removal** (called ANR hereupon) – the parameters are mentioned in the following explanation

This choice was made based on some trial and error. We had three choices to do the source separation:

1. Using Spleeter Only
2. Using ANR only
3. Using ANR followed by Spleeter (ANR+Spleeter)

The drawback of using Spleeter was that some portion of the vocals was getting lost because of aggressive source separation. The drawback of using ANR was that it was not as effective as Spleeter in removing the accompaniment (Tanpura). So we had a tradeoff.

We noticed the following:

**Pune Singers:**

For SM, the voice was loud enough so ANR was working well. But it was not working better than Spleeter. For other Pune Singers ANR is working worse than Spleeter. So for new singers, using Spleeter only was the best. Audacity noise removal is not giving any improvement, irrespective of whether we use Spleeter or not. Because of the loud tanpura, we need the aggressive splitting of Spleeter to get the separated vocals.

**Durham Singers:**

For Durham singers, low pass filtering followed by noise removal was working well. After than we could use Spleeter in order to be extra sure or we could do without it as well (the difference was negligible). For CC and AG, ANR only and ANR+Spleeter were working much better than Spleeter only. For SCh the difference is not so significant. The tanpura is soft enough so that all methods give decent pitch contours but ANR is still better than Spleeter.

Steps in ANR:

1. Low pass filter (2400 Hz, 48 dB roll off) using Audacity:
2. Noise removal:

The noise profiles and parameter values chosen are mentioned later. For details on how to use Audacity Noise Removal, check this page: <https://manual.audacityteam.org/man/noise_reduction.html>

For AG and CC, ANR only worked the best. For SCh, all three were similar. So for old singers we used ANR only, to not risk losing the vocals because of Spleeter.

So for Durham singers the pipeline is:

1. Filtering using lowpass filter of 2400 Hz cutoff and 48 dB rolloff (in Audacity: Effects → EQ and Filters → Low-pass filter)

For steps 2 and 3: (Effects → Noise Reduction and Repair → Noise Reduction)

1. Choose noise profile: (note: noise profile is chosen after filtering, not before)
2. Noise removal

Based on trial-error and tuning the parameters, the noise profiles and parameters chosen finally, for the Durham singers were:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Singer** | **Noise Reduction (dB)** | **Sensitivity** | **Frequency smoothing (bands)** | **Noise Profile (for ragas except Bageshri)** | **Noise profile (for Bageshri) \*** |
| **AG** | 12 | 4 | 0 | AG\_Aalap2\_MM – 0 to 4.6s | AG\_Aalap2\_Bag – 0 to 2.9s |
| **CC** | 18 | 4 | 0 | CC\_Aalap2\_Bahar – 0 to 4.8s | CC\_Aalap1\_Bag – 0 to 5.2s |
| **SCh** | 12 | 4 | 0 | SCh\_Aalap1\_Shree – 0 to 4.5s | SCh\_Aalap2\_Bag – 3 min 7.0 s to 3 min 12.3 s |

**\*** The reason for using different noise profiles for Bageshri was that in Bageshri the tanpura is played in cycles of Sa-Ma instead of Sa-Pa. Hence, the spectra of tanpura in raga Bageshri is different than that in other ragas.

Using this, we got the source separated audios, which are stored in the folder Source\_Separated\_Audios under two subfolders:

* Old\_Singers\_ANR: Durham singer audios separated using Audacity Noise Removal
* New\_Singers\_Spleeter: Pune singer audios separated using Spleeter

# 

# Part 2: From Source Separated Audio to Pitch Contours

Once, we have the source separated audio, we use the Python API for Praat software, namely Parselmouth to derive the pitch contour from the audio using filtered autocorrelation method. Parselmouth has several parameters that can be tuned for increased accuracy of the pitch contour. Based on extraction performance and comparison with the audio done by manual listening, we found that the following values worked best on our audios: (The tonic is fixed for a given singer, other parameters are tuned). The tuning was done mainly with the intention to avoid octave errors and false silences.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Singer** | **Pitch Min (Hz)** | **Tonic** | **Pitch Max (Hz)** | **Silence Threshold** | **Voicing Threshold** | **Octave Cost** | **Octave Jump Cost** | **Voiced/Unvoiced Cost** |
| (Praat Standard) | 50 | **-** | 800 | 0.09 | 0.50 | 0.055 | 0.35 | 0.14 |
| AK/MP/ NM/MG | 80 | **146.8** | 600 | 0.001 | 0.01 | 0.01 | 20 | 10 |
| CC | 70 | **138.6** | 560 | 0.01 | 0.01 | 0.1 | 20 | 10 |
| AG | 125 | **207.7** | 880 | 0.01 | 0.01 | 0.1 | 10 | 10 |
| SCh | 150 | **246.9** | 900 | 0.001 | 0.01 | 0.1 | 10 | 10 |
| SS | 150 | **220.0** | 800 | 0.01 | 0.001 | 0.1 | 20 | 10 |
| SM/RV/AP | 150 | **220.0** | 800 | 0.0001 | 0.0001 | 0.1 | 20 | 10 |

Attenuation at ceiling: 0.03 for all

The main steps in pitch extraction were:

1. Parselmouth pitch extraction on source separated audio using above parameter values
2. Linear Interpolation of silences less than 400 ms

The code for these steps is present in the file extract\_pitch\_contours.py

Instructions to run the script:

1. In the file extract\_pitch\_contours.py, change the variables INPUT\_FOLDER, OUTPUT\_FOLDER and FILE according to your directory structure. INPUT\_FOLDER should contain the audio file that is to be pitch-extracted, and OUTPUT\_FOLDER should be a folder where the csv file of pitch contour is to be stored.
2. Run the script using the command:

**python extract\_pitch\_contours.py**

The interpolated pitch contour will be saved as a csv file in OUTPUT\_FOLDER