descriptives_parsed

December 2, 2020

1 Descriptives

1.1 Segmentation strategy

Descriptives of features f were calculated by collapsing each album in 4 segments. The number of tracks within each each segment is equal to the nearest integer to $\frac{length\ of\ album}{4}$.

The size of the last segment is determined according to the following:

- If we round up, last chunk is smaller;
- If we round down, last chunk larger;
- If $\frac{length\ of\ album}{4} = x.5$, section length is rounded up, which leaves section 4 smaller.

Note: after averaging features across sections, each album contributes exactly with 4 observation for each feature (one for each section). The distribution of tracks throughout segments is:

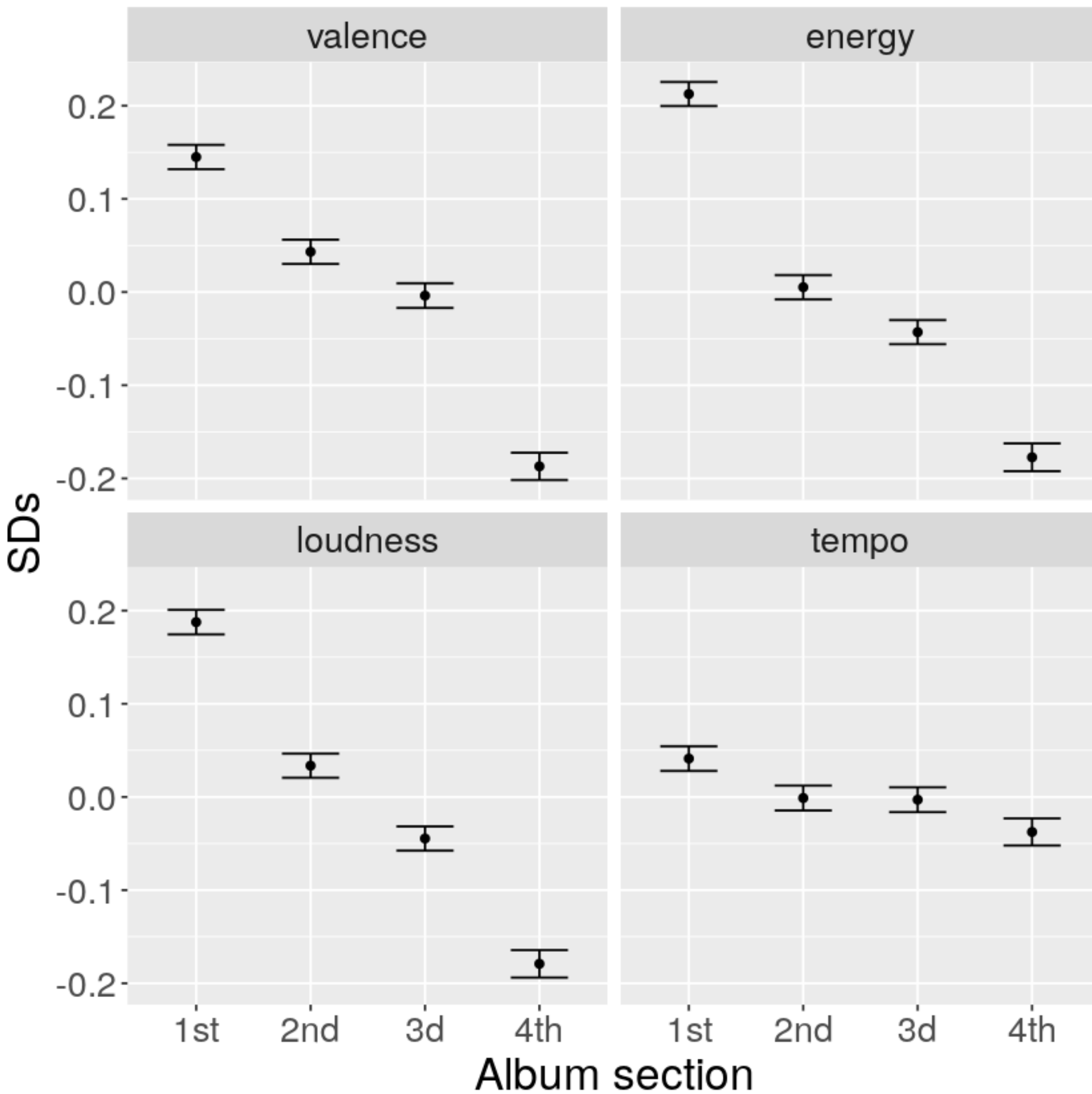
- 1st 12652
- 2nd 12652
- 3d 12652
- 4th 10129

1.2 Metric

For each feature and each one of the 4 sections, I computed the average feature f throughout all tracks Tn, and then took an overall average for each album k:

$$\frac{1}{kTn}\sum_{j=1}^{k}\sum_{i=1}^{Tn} f_{ji} \tag{1}$$

Features were converted to z-scores within each album, so the results are expressed in terms of SDs.



2 Dissimilarity matrix

Dissimilarities were computed between all pairwise combinations of tracks (A, B) within the same album, and summed across all features f. Then the dissimilarities were averaged across all albums k:

$$d(A, B) = \frac{1}{k} \sum_{j=1}^{k} \sqrt{\sum_{i=1}^{f} (A_{ji} - B_{ji})^2} \tag{2}$$

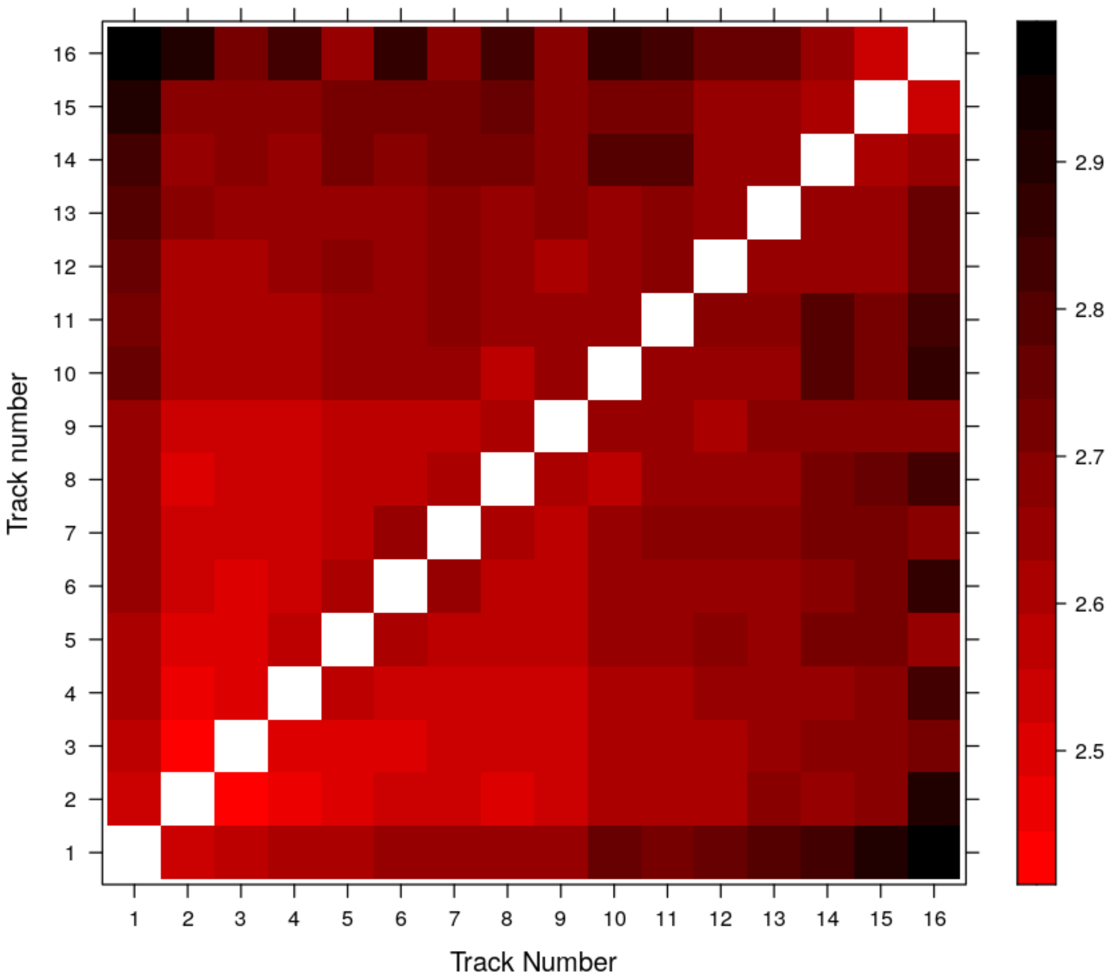
The output of $d(A, B)$ constitutes the mean dissimilarity between two tracks in positions A and B.

2.1 Sanity check

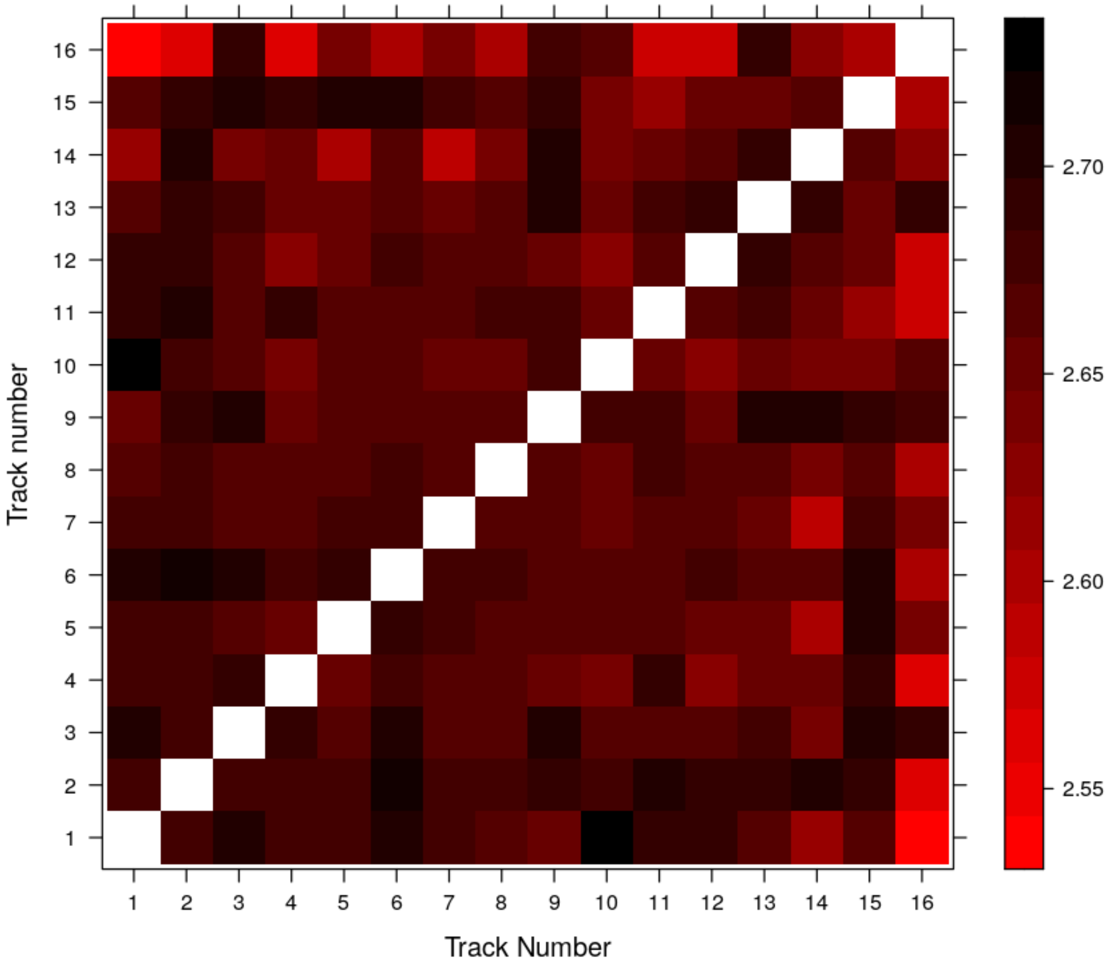
Dissimilarity measurements were taken 1) from the original dataset, and 2) from a shuffled version of the data, where track numbers were randomly shuffled within each ablum. I did this because:

If the pattern of dissimilarities found in the original dataset is really due track ordering factors, these patterns should disapear in a shuffled version of albums.

2.2 Results - Original dataset



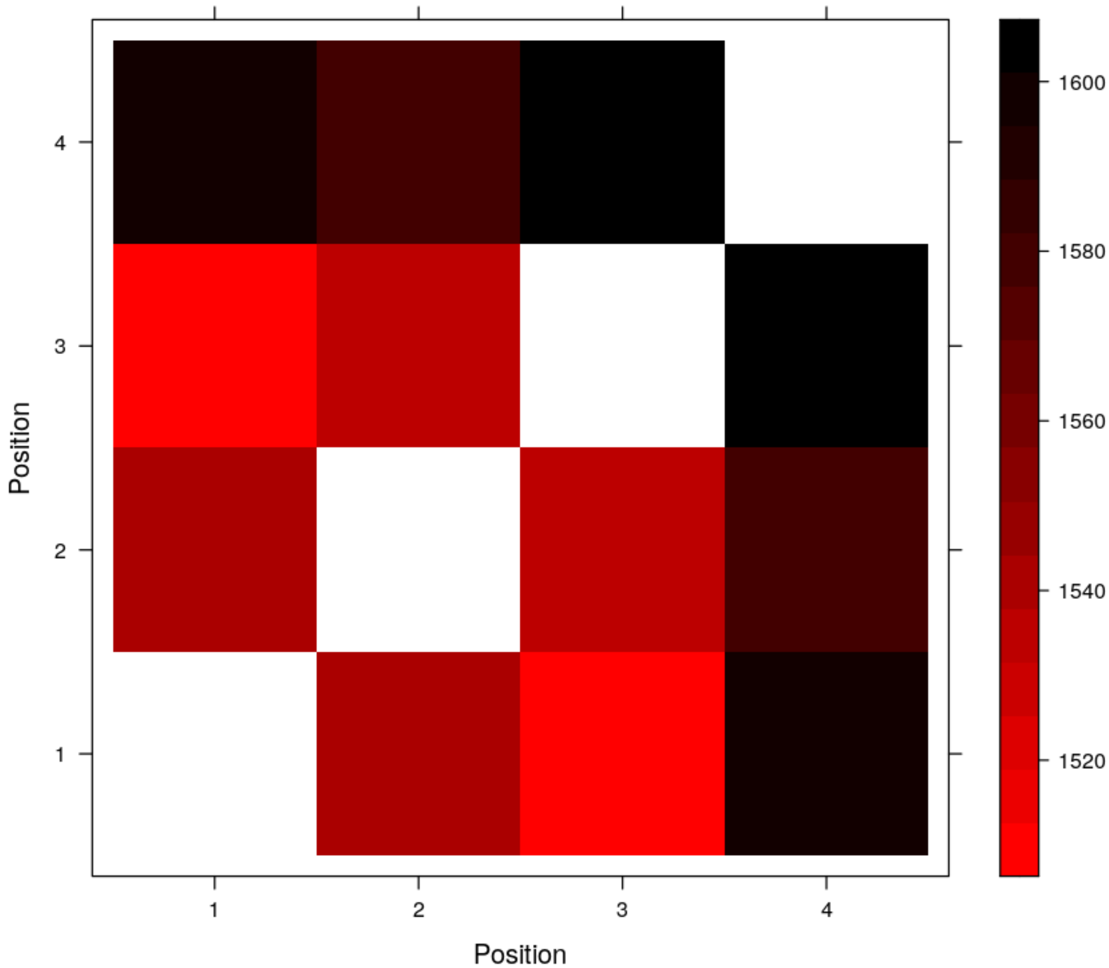
2.3 Results - Shuffled dataset



3 Dissimilarity by album segment

Next we collapsed - again - each album in 4 different segments see segmentation strategy. Then we computed the average feature for each album section. Dissimilarity measurements were taken using Eq. 1, but with the tuple (A, B) meaning all pairwise combinations of album sections, instead of album tracks.

Note: even though section 4 is constituted of fewer tracks, this does not affect - directly - the computation of dissimilarity ratings.



4 Upsampling

Each album has a certain length, and this might affect some of the descriptive statistics that we are computing.

In order to normalize the length of each album, we can duplicate each track $\frac{n}{Al}$ times, where n is the Least Common Multiplier LCM of all album lengths Al.

4.1 Problem

We have Al from 5 to 17. The LCM of this array is 720720, which will result in a total of 3 billion lines of data. I tried processing it, but I don't have enough memory.

4.2 Alternative

We can exclude some albums in order to minimize LCM, while also minimizing the loss of albums from our dataset. This was implemented and shown below.

4.3 Procedure

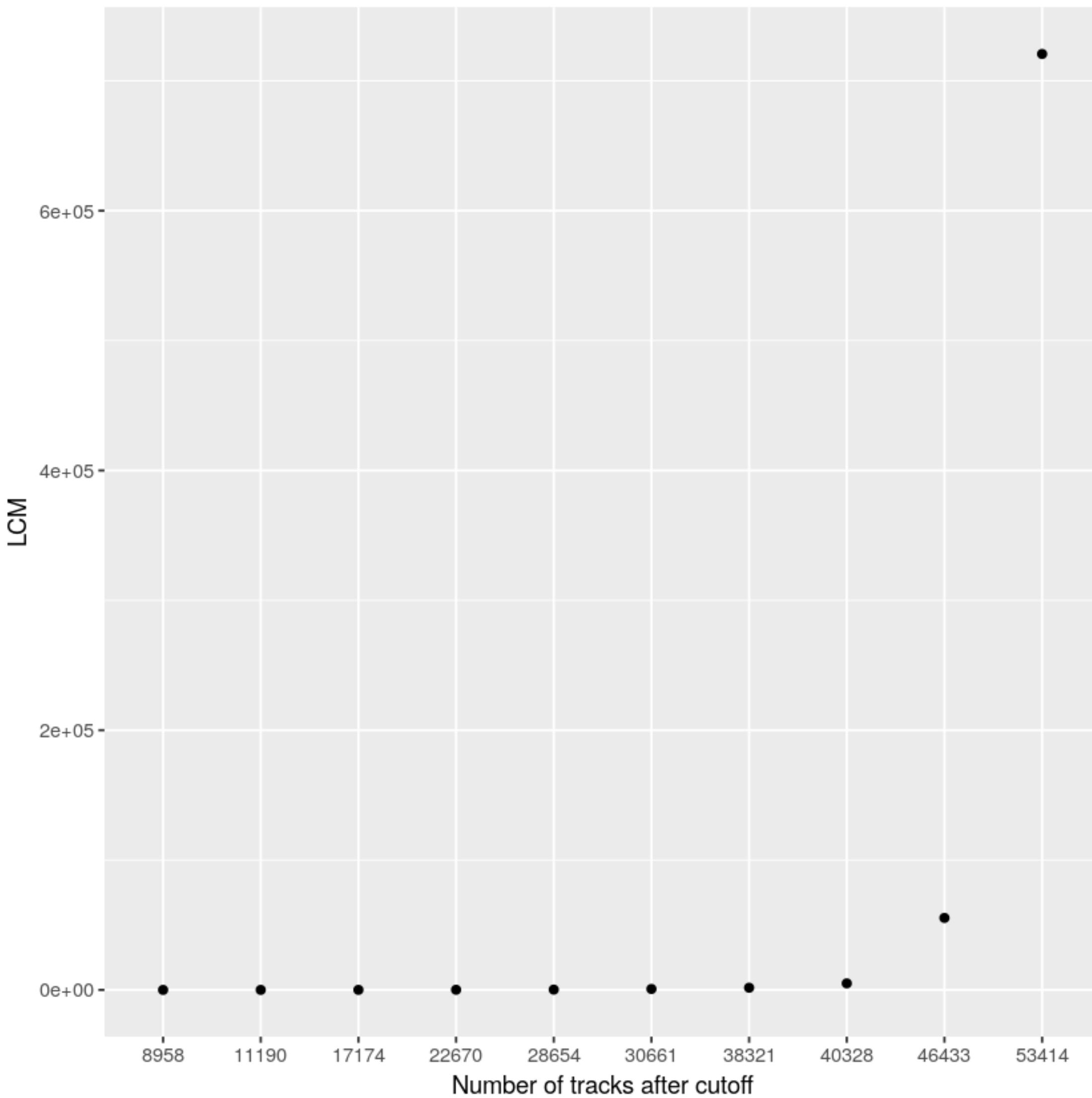
- I created subsets S of 2 to 11 different Als. For instance, two subsets of Als could be $S_2 = \{6, 7\}$ or $S_3 = \{6, 8, 9\}$, where each element within the set represents an Al, and the subscript to S means the number of Al within the subset.
- From S_2 to S_{11}, I tested all possible subset of Als and calculated LCM for each combination. For instance, $LCM(\{6, 7\}) = 42$ and $LCM(\{6, 8, 9\}) = 72$.
- Then, for each subset of Als, I kept only the combination which yealded the lowest LCM. For instance, if $S_3 = \{6, 7, 10\}$ and $S'_3 = \{6, 8, 10\}$, I kept only the S_3.
- I visually chose a cut off to maximize the number of albuns within the dataset, while minimizing the LCM.

4.4 Result

Table and graph of LCM (y axis) by number of remaining tracks (x axis).

A tibble: 10 × 2

LCMs	Size of dataset after cut
<int>	<int>
12	8958
24	11190
48	17174
120	22670
240	28654
720	30661
1680	38321
5040	40328
55440	46433
720720	53414



4.5 Conclusion

If we keep albums of size 6, 7, 8, 9, 10, 12, 14, 15 and 16, we decrease LCM to 5040 (third point from right to left on the x axis), and the number of tracks in our data to 36636. We loose around 12 thousand tracks, but our dataset decreases from 3 billion to 25 million datapoints.

Next I'm implementing the upsampling by the factor of 5040

5 Remaking the heatmaps