Automatic subgrouping of multitrack audio

Ronan, DM; Moffat, D; Gunes, H; Reiss, JD; 18th Int. Conference on Digital Audio Effects (DAFx-15)

AUTOMATIC SUBGROUPING OF MULTITRACK AUDIO

David Ronan[*]*, Hatice Gunes*

Centre for Intelligent Sensing,
Queen Mary University of London
London, UK
`{d.m.ronan, h.gunes}@qmul.ac.uk`

David Moffat, Joshua D. Reiss

Centre for Digital Music,
Queen Mary University of London
London, UK
`{d.j.moffat, joshua.reiss}@qmulac.uk`

ABSTRACT

Subgrouping is a mixing technique where the outputs of a subset of audio tracks in a multitrack are summed to a single audio bus. This is done so that the mix engineer can apply signal processing to an entire subgroup, speed up the mix work flow and manipulate a number of audio tracks at once. In this work, we investigate which audio features from a set of 159 can be used to automatically subgroup multitrack audio. We determine a subset of audio features from the original 159 audio features to use for automatic subgrouping, by performing feature selection using a Random Forest classifier on a dataset of 54 individual multitracks. We show that by using agglomerative clustering on 5 test multitracks, the entire set of audio features incorrectly clusters 35.08% of the audio tracks, while the subset of audio features incorrectly clusters only 7.89% of the audio tracks. Furthermore, we also show that using the entire set of audio features, ten incorrect subgroups are created. However, when using the subset of audio features, only five incorrect subgroups are created. This indicates that our reduced set of audio features provides a significant increase in classification accuracy for the creation of subgroups automatically.

1. INTRODUCTION

At the early stages of the mixing and editing process of a multitrack mix, the mix engineer will typically group audio tracks into subgroups. An example of this would be grouping guitar tracks with other guitar tracks or vocal tracks with other vocal tracks. Subgrouping can speed up the mix work flow by allowing the mix engineer to manipulate a number of audio tracks at once, for example by changing the level of all drums with one fader movement, instead of changing the level of each drum track individually. It also allows for processing that is not possible to achieve by manipulation of individual audio tracks. For instance, when non-linear processing such as dynamic range compression or harmonic distortion is applied to a subgroup, the processor will affect the sum of the sources differently than when it would be applied separately to every audio track. A Voltage Controlled Amplifier (VCA) group is another type of subgroup by which the individual faders of the group can be moved in unison, but each channel is not summed together and no subgroup processing can occur. An example of a typical subgrouping setup can be seen in Figure 1. Due to the varying amount of instrumentation used in recordings, there seems to be no clearly defined rules on how these subgroups are created, but it was found that the most commonly used approach is to group by instrument family [1].

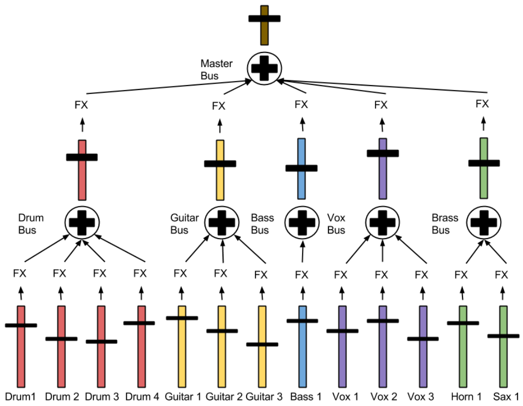

Figure 1: Typical mixing desk

In the literature reviewed, there is currently no proposals or discussions of a system that attempts to automate the subgrouping process [2, 3, 4, 5]. In this paper, we suggest that this can be done autonomously using machine learning techniques. The motivation is two-fold. Firstly, not only would it be possible to subgroup the audio tracks in the conventional sense, but through analysis of each audio track's spectro-temporal audio features, we may discover in this study that there are more intelligent ways to create subgroups.

Secondly, the audio features that are determined to be important can be used to answer the research question are we putting the instruments in the correct subgroups? Whereby, if we have good audio features to determine subgroups, this may inform us that a certain audio track or even certain sections of an audio track should be subgrouped differently from how they would be typically subgrouped [1]. An example of how this may work would be when we find over time that an audio track changes and may become more similar to another audio track in another subgroup. This could occur if the bass player suddenly switched from picking the bass guitar to playing in the style of slap bass. The audio track that was once in the bass subgroup could now be subgrouped with the percussive instrument audio tracks. At this point, it would make sense to split the single bass guitar audio track into two individual audio tracks and have them designated to their appropriate subgroups.

In light of the above discussion, the subgroup classification problem can been seen as somewhat similar to musical instrument identification, which has been done before for orchestral style instruments [6, 7, 8, 9]. However, in subgrouping classification we are not trying to classify traditional instrument families, but defined groups of instrumentation that would be used for the mix-

[*] This work was supported by the EPSRC

ing of audio from a specific genre. For example, in rock music the drum subgroup would consist of hi-hats, kicks and snares etc. while the percussion subgroup may contain tambourines, shakers and bongos. In practice, the genre of the music will dictate the type of instrumentation being used, the style in which the instrumentation will be played and what subgroup the instrument belongs to. It is also worth noting that typical subgroups such as vocals or guitars can be further broken down into smaller subgroups. In the case of vocals the two smaller subgroups might be lead vocals and background vocals. Furthermore, we can never assume that the multitrack recordings being used are good quality recordings. They may contain background noise, microphone bleed interference or other recording artefacts. All of these factors can affect the accuracy of a classification algorithm.

The purpose of this study is to determine the best set of audio features that can be extracted from multitrack audio in order to perform automatic subgrouping. In our particular case, we looked at multitracks that would be considered as Rock, Pop, Indie, Blues, Reggae, Metal and Punk genres, where the subgroups would typically be drums, bass, guitars, vocals etc. The rest of the paper is organised as follows. Section 2 describes the dataset used for feature selection and testing. Section 3 provides a list of features used and describes how they were extracted. Section 4 explains how the experiments, classification and clustering were performed. Section 5 presents the results obtained. Section 6 discusses the results and then finally the paper is concluded in section 7.

2. DATASET

The amount of data available for multitrack research is limited due to a multitrack being an important asset of a record label and the copyright issues that come with distributing them. A subset was selected from a larger multitrack test bed consisting of roughly 1.3 TB of data, including raw audio recordings, stems, mixdowns and recording session files. The Open Multitrack Testbed was used because it is one of the largest of its kind and contains data that is available for public use [10].

The subset used for feature selection consists of 54 separate multitracks and 1467 audio tracks in total once all duplicate audio tracks were removed. The multitracks that were used span a wide variety of musical genres such as Pop, Rock, Blues, Indie, Reggae, Metal, and Punk. We annotated each track by referring to its filename and then listening to each file for a brief moment to confirm its instrument type. The labels used for each audio file were based on commonly used subgroup instrument types [1]. These were drums, vocals, guitars, keys, bass, percussion, strings and brass. Table 1 shows the breakdown of all the multitrack data used for feature selection relative to what subgroup each audio track would normally belong to. It is worth noting the imbalance of label types in our dataset. This is because the most common instruments in our multitrack dataset are drums, vocals and guitars. Furthermore, the drum subgroup consists of many different types of drums such as kicks, snares, hi-hats etc. meaning it tends be the largest subgroup.

The subset used to test if the selected features were useful or not consists of five unseen multitracks. The breakdown of the different types of audio tracks for each test multitrack can be seen in Table 2.

Table 1: *Details of the subset used for feature selection*

Subgroup type	No. of tracks	Percentage of subset
Drums	436	29.72%
Guitars	365	24.88%
Vocals	363	24.74%
Keys	103	7.02%
Bass	93	6.34%
Percussion	80	5.45%
Strings	19	1.30%
Brass	8	0.55%

3. EXTRACTED AUDIO FETAURES

Each audio track in the dataset was downsampled to 22050 Hz and summed to mono using batch audio resampling software. The audio features were then extracted from the 30 secs of highest energy in each audio track. This was done to speed up the feature extraction process as we did not see the need to extract features from long periods of silence that occur in multitrack recordings. We extracted 159 low level audio features in total with a window size of 1024 samples and a hop size of 512 samples. A list of the audio features and the relevant references are in Table 3. Overall we have 42 different low level audio feature types and the majority of these are window based. Only three audio features were whole audio track features and not window based. Since the whole track audio features were not windowed like the others, no pooling was required [11]. We took the mean, standard deviation, maximum and minimum values of each windowed audio feature over the 30 secs of audio used for feature extraction. This allowed us to pool the windowed features over the 30 secs of audio and is the reason why we have 159 audio features in total [11].

4. EXPERIMENT

Two experiments were conducted. The first experiment determined a reduced set of audio features from the 159 audio features that we extracted previously. This was done by performing feature selection. The goal of this experiment was to determine the best subset of the 159 original audio features that could be used for automatic subgrouping. A second experiment was conducted where five test multitracks were agglomeratively clustered using all of the 159 audio features extracted and then agglomeratively clustered using

Table 2: *Details of the subset used for testing*

Subgroup type	MT 1	MT 2	MT 3	MT 4	MT 5
Drums	11	8	9	10	1
Vocals	17	11	6	9	3
Guitars	12	2	6	2	0
Keys	1	4	2	4	3
Bass	1	1	1	1	1
Percussion	1	0	1	0	0
Strings	0	0	0	0	6
Brass	0	0	0	0	0

Table 3: *Audio features*

Category	Feature	Reference
Dynamic	RMS	
	Peak Amplitude	
	Crest Factor	
	Periodicity	[12]
	Entropy of Energy	[13]
	Low Energy	[14]
Spectral	Zero Crossing Rate	[15]
	Centroid	.
	Spread	.
	Skewness	.
	Kurtosis	.
	Brightness	.
	Flatness	.
	Roll-Off (.85 and .95)	.
	Entropy	.
	Flux	.
	MFCC's 1-12	.
	Delta-MFCC's 1-12	[15]
	Spectral Crest Factor	[16]

the reduced feature set for comparison. This was done to investigate how well the reduced audio feature set compared to the entire audio feature set when performing automatic subgrouping.

4.1. Feature Selection

Random Forest is a particular type of Ensemble Learning method based on growing decision trees. This can be used for either classification or regression problems, but can also be used for feature selection. Random Forest is based on the idea of bootstrap aggregating or more commonly know as bagging. After training has occurred on a dataset each decision tree that is grown predicts an outcome. For regression decision trees, the output is the average value predicted by all of the decision trees grown. For classification decision trees it is the classification outcome that was voted most popular by all of the decision trees grown [17]. Random Forest was chosen because it has been proven to work very well for feature selection in other fields such as bioinformatics and medicine [18, 19].

Determining the most salient features using the Random Forest classifier was performed as follows. 100 decision trees were grown arbitrarily and a feature importance index was calculated. It will be seen further on in Section 5 that this was an appropriate amount of decision trees to grow. Random Forest feature importance can be defined for X^i, where the vector $X = (X^1, ...X^p)$, contains feature values and where p is the number of audio features used. For each tree t in the Random Forest, consider the associated OOB_t sample (this is the out-of-bag data that is not used to construct t). $errOOB_t$ denotes the error of a single tree t using the OOB_t sample. The error being a measure of the Random Forest classifier's accuracy. If the values of X^i are randomly permuted in OOB_t to get a different sample denoted by $\widetilde{OOB_t^j}$ and we compute $err\widetilde{OOB_t^j}$. $err\widetilde{OOB_t^j}$ being the error of t because of the different sample. The feature importance of X^i is equal to:

$$FI(X^i) = \frac{1}{ntree}\sum_t (err\widetilde{OOB_t^j} - \widetilde{OOB_t^j}) \qquad (1)$$

where the sum is over all trees t of the Random Forest and $ntree$ is the number of trees in the Random Forest [20].

The feature importance index was calculated for each of the 159 audio features. The average feature index was then calculated and the audio features that performed below the average were eliminated. The use of the average importance index was found to give us the most satisfactory set of audio features.

We also tried eliminating the 20% worst performing audio features, then retraining on the new audio feature set and repeating the 20% worst performing audio feature elimination process. This process would then stop once the out-of-bag error began to rise. However, we found that this was found to give us an unsatisfactory set of audio features. They were unsatisfactory because when we used these audio features to automatically create subgroups, the subgroups created were mostly incorrect e.g. drums in the same subgroup as guitars. This was the search method that was used in [20].

It should also be noted that when using the Random Forest classifier we set prior probabilities for each class based on our imbalanced dataset. The prior probabilities were set using the data in the *Percentage of subset* column in Table 1

4.2. Agglomerative clustering

Agglomerative clustering is a type of Hierarchical clustering. Generally in Hierarchical clustering a cluster hierarchy or a tree of clusters, also known as a dendrogram is constructed. This is not to be confused with the decision trees used in Section 4.1. An example of a dendrogram can be seen in Figure 4. Hierarchical clustering methods are categorised into agglomerative and divisive. The agglomerative clustering method is what we use in this experiment. The idea is that the algorithm starts with singular clusters and recursively merges two or more of the most similar clusters [21]. The reason why we chose agglomerative clustering is because the algorithmic process is similar to how a human would create subgroups in a multitrack. Initially, a human would find two audio tracks that belong together in a subgroup and then keep adding audio tracks until a subgroup is formed. An example would be pairing a kick track with a snare track and then pairing them with a hi-hat track to create a drum subgroup [1]. It is also worth noting that Figure 1 which is a typical subgrouping setup can be likened to a tree structure, so it would make sense to attempt to cluster audio tracks in a tree like fashion.

The agglomerative clustering algorithm can be described as thus [22]. Given a set of N audio feature vectors to be clustered.

1. Assign each audio feature vector N to its own singleton cluster and number the clusters 1 through c.
2. Compute the between cluster distance $d(r, s)$ as the between object distance of the two objects in r and s respectively, $r, s = 1, 2, ..., c$. Where $d(r, s) = \sqrt{\sum_c (r_c - s_c)^2}$ is the euclidean distance function and let the square matrix $D = (d(r, s))$.
3. Find the most similar pair of clusters r and s, such that the distance, $D(r, s)$, is minimum among all the pairwise distances, $d(c_i, c_j) = min\ \{d(r, s) : r \in c_i, s \in c_j\}$. This is what is known as the linkage function. A similar pair of clusters could be a snare track and a hi-hat track.

4. Merge r and s to a new cluster t and compute the between-cluster distance $d(t, k)$ for any existing cluster $k \neq r, s$. Once the distances are obtained, remove the rows and columns corresponding to the old cluster r and s in D, since r and s do not exist any more. Then add a new row and column in D corresponding to cluster t. Merging two clusters is like grouping two audio tracks together or else adding an audio track to an existing subgroup.
5. Iteratively repeat steps 3 to 5 a total of $c - 1$ times until all the data items are merged into one cluster.

In our case the similarity is found between every pair of audio feature vectors that represent the audio tracks in our dataset. This is normally calculated using a distance function such as euclidean, manhattan or mahalanobis distance. We used euclidean distance as we found it gave us more realistic clusters. It is also worth noting that we normalised each instance in our dataset using L2-normalisation, while each audio feature value was normalised between zero and one. This was done due to the euclidean distance function being used. We then linked together audio feature vectors into binary pairs that were in close proximity to each other using a linkage function. We used the shortest distance measure as our linkage function, as this would make the most sense in our case as we are trying subgroup similar audio tracks based on instrumentation. The newly formed clusters created through the linkage function were then used to create even larger clusters with other audio feature vectors. Once linkage has occurred between all the audio feature vector clusters, all the branches of the tree below a specified cut-off are pruned. This cut-off can be specified as an arbitrary height in the tree or else the maximum amount of clusters to create. A maximum number of eight clusters was specified in our case. This was due to there only being eight labels in the original dataset used for feature selection.

Figure 4 depicts that any two audio tracks in the dataset become linked together at some level of the dendrogram. The height of the link is known as the cophenetic distance and represents the distance between the two clusters that contain those two audio tracks. If the agglomerative clustering is suited to a dataset, the linking of audio tracks in the dendrogram should have a strong correlation with the distances between audio tracks generated by the distance function. A cophenetic correlation coefficient can be calculated to measure this relationship. The cophenetic correlation coefficient is measured from -1 to 1 and the closer the value is to 1 the more accurately the dendrogram reflects the dataset. Suppose that the previous example dataset N_i has been modelled using the above cluster method to produce a dendrogram T_i. The cophenetic correlation coefficient is calculated as such

$$c = \frac{\sum_{i<j}(d(i,j) - \bar{d})(t(i,j) - \bar{t})}{\sqrt{\left[\sum_{i<j}(d(i,j) - \bar{d})^2\right]\left[\sum_{i<j}(t(i,j) - \bar{t})^2\right]}} \qquad (2)$$

where $d(i, j)$ is the ordinary euclidean distance between the ith and jth observations of the dataset and $t(i, j)$ is the cophenetic distance between the dendrogram points T_i and T_j. $\bar{d}$ is the average of the $d(i, j)$ and $\bar{t}$ is the average of the $t(i, j)$.

5. RESULTS

In this section we present the results of the experiments conducted. We firstly show the results of the feature selection performed and then show the results of the agglomerative clustering. We also present the resulting dendrograms from the clustering.

5.1. Selected Features

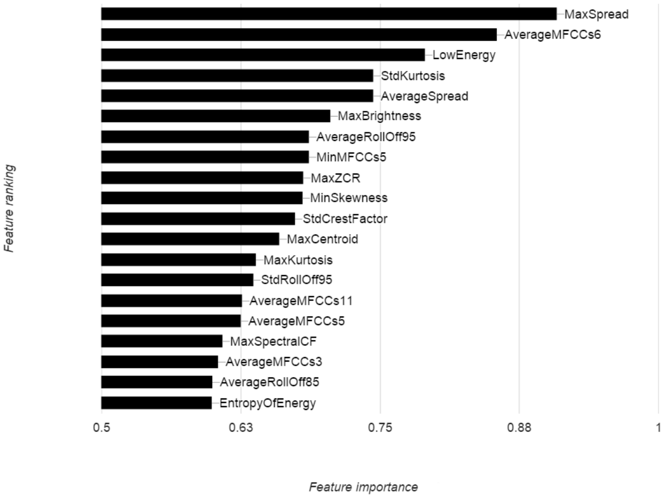

Figure 2: The 20 most important features

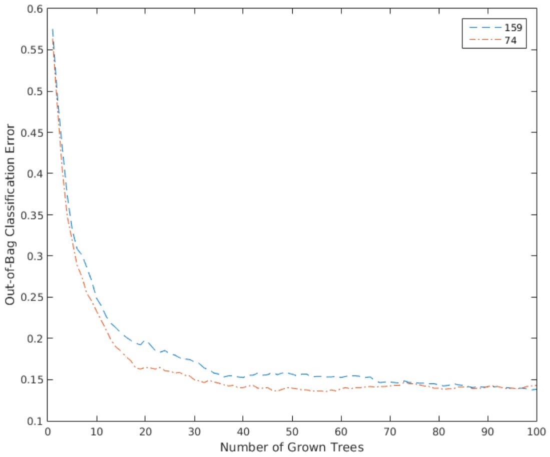

Figure 3: Cumulative out-of-bag classification errors for both feature sets

Using the feature selection method mentioned in Section 4.1 we determined a subset of 74 audio features from the original 159. The average feature importance index was 0.421 with a standard deviation of 0.1569. The maximum value for feature importance index was 0.9086 and the minimum was -0.0135. The 20 most important features are depicted in Figure 2. This illustrates some of the audio features that would occur in an audio feature vector used during agglomerative clustering.

The cumulative out-of-bag error having grown 100 trees with the full audio feature set was 0.1384. Using the reduced feature set and growing 100 trees the cumulative out-of-bag error was 0.1431.

Table 4: *Agglomerative clustering results using all features and the reduced feature set*

159 Features	**MT 1**	**MT 2**	**MT 3**	**MT 4**	**MT 5**
Cophenetic C.C.	0.799	0.844	0.751	0.894	0.814
Audio tracks	43	26	25	26	14
Incorrect subgroups	3	1	3	1	2
Incorrect audio tracks	19	7	5	7	2
Percentage incorrect audio tracks	44%	26.9%	20%	26.9%	14%
74 Features	**MT 1**	**MT 2**	**MT 3**	**MT 4**	**MT 5**
Cophenetic C.C.	0.771	0.887	0.806	0.924	0.956
Audio tracks	43	26	25	26	14
Incorrect groups	2	0	1	0	2
Incorrect audio tracks	6	0	1	0	2
Percentage incorrect audio tracks	13%	0%	4%	0%	14%

Table 5: *Agglomerative clustering results for all multitracks*

	159 Features	**74 Features**
Avg. Cophenetic C.C.	0.8203	0.8642
Total no. audio tracks	114	114
Avg. no. audio tracks	26.1	26.1
Total incorrect subgroups	10	5
Total incorrect audio tracks	40	9
Percentage incorrect audio tracks	35.08%	7.89%

Figure 3 shows that these results converge and start becoming very close after about 70 trees. This also supports our original choice to arbitrarily grow 100 decision trees for feature selection.

5.2. Agglomerative clustering

In Table 4 we present the results for each of the five multitracks that were agglomeratively clustered using the entire audio feature set and the reduced audio feature set. Also, we give the cophenetic correlation coefficients as described in Section 4.2. Also, we give the number of audio tracks in each multitrack as well as how many incorrect subgroups were created to show how well the clustering is at creating meaningful subgroups [1]. An incorrect subgroup would be where at least two different audio tracks with different instrument types are subgrouped together. An example of an incorrect subgroup would be if a subgroup consisted of drums, guitars and vocals. These three instrument types would normally be separate. There will always be eight subgroups due to the labels used in the training dataset, but these eight subgroups may not always be constructed correctly using agglomerative clustering. The number of incorrect audio tracks is measured by how many audio tracks were placed in a cluster where the majority of the instrument types where incompatible. An example being if we had a cluster of six guitars and two vocals. The guitars are the majority, so the incorrect audio tracks would be the vocal tracks. We also show this measure as a percentage of all the audio tracks in each multitrack.

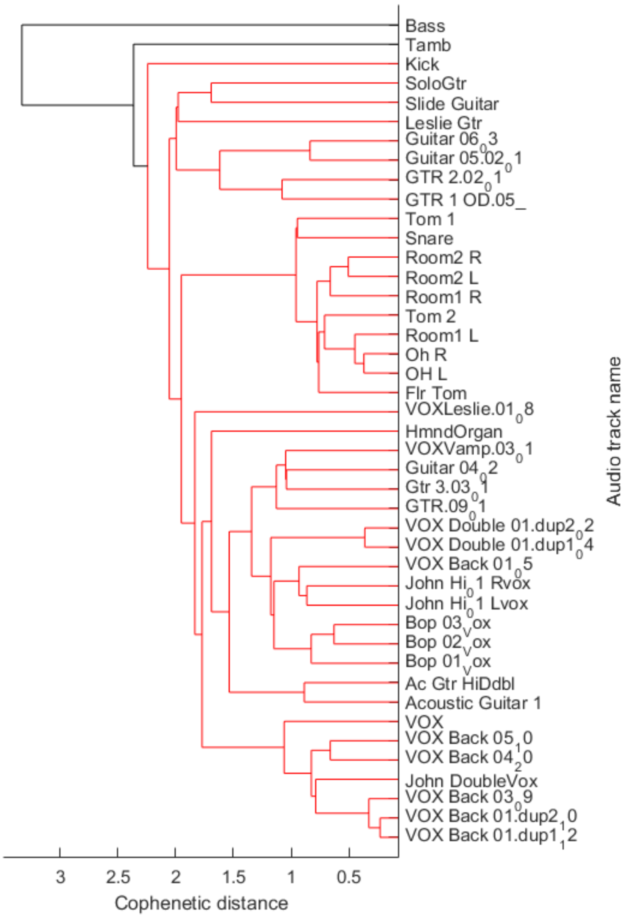

Figure 4: Dendrogram of MT 1 using the reduced feature set

6. ANALYSIS AND DISCUSSION

6.1. Selected Features

Looking at Figure 2 we can see the list is dominated by spectral features and has only three features related to dynamics. We were not surprised to see MFCC's in the 74 selected audio features as they have been proven before to perform quite well in speech recognition and audio classification tasks [23, 24, 25]. The Low Energy audio feature also plays a very significant role in classification. The Low Energy audio feature can be defined as the percentage of frames showing less than average RMS energy [14]. Vocals with silences or drum hits would have a high low energy rate compared to say a bowed string, so this may be one of the reasons it was so successful.

The maximum and average spectral spread as well as the standard deviation of kurtosis are also placed in top five ranked audio features. This suggests that the shape of the audio spectrum for each audio track was one of the most important factors. The spectral centroid, brightness and roll off 95% also featured in the top

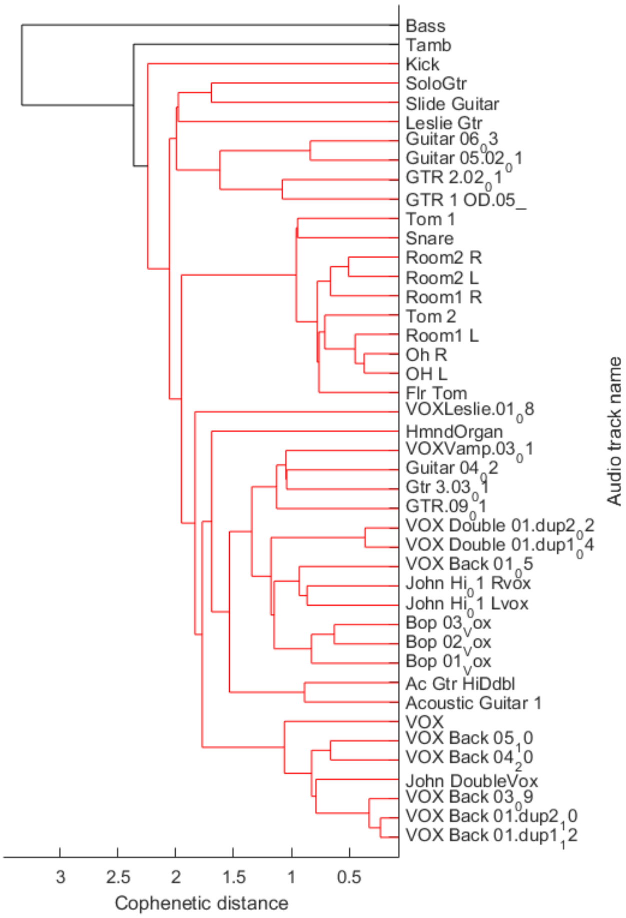

Figure 5: Dendrogram of MT 2 using the reduced feature set

20, which are all spectral features.

We were expecting the Periodicity feature to perform much better, but it did not even make it into the subset of 74 audio features. We expected this to be important for drum and percussion classification. Ideally, this would be predictably high for drums, but low for vocals.

6.2. Agglomerative clustering

If we compare the results from agglomerative clustering using the entire audio feature set and the reduced audio feature set we can clearly see that the reduced audio feature set achieved a higher performance. The overall percentage of incorrectly clustered audio tracks changes from 35.08% for the entire audio feature set to 7.89% for the reduced audio feature set. We also found that the reduced audio feature set has a slightly higher average cophenetic correlation coefficient than the entire audio feature set. This suggests the clustering better fits the reduced audio feature dataset. Furthermore, the total number of incorrectly created subgroups was halved when using the reduced audio feature set. Table 5 shows these results.

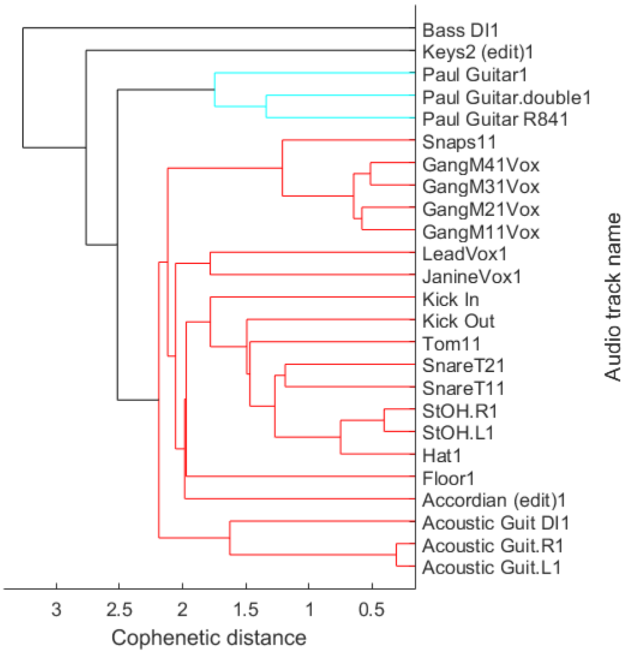

Figure 6: Dendrogram of MT 3 using the reduced feature set

There is also an overall trend of higher performance for the reduced audio feature set when we examine each multitrack separately. MT 1 was the worst performing multitrack for both the entire audio feature set and the reduced audio feature set. MT 1 when using the reduced audio feature set, had a lower misclassification measure than MT 1 using the entire audio feature set, but surprisingly has a slightly lower cophenetic correlation coefficient. Overall, MT 1 had the lowest cophenetic correlation coefficient for both sets of audio features and this maybe because it also had the most amount of audio tracks to cluster. This may have been improved by using a varying maximum amount of clusters based on how many audio tracks are present. It is also worth mentioning that once the experiment was finished we listened back to the incorrectly subgrouped audio tracks for the reduced audio feature set and we found that these audio tracks suffered badly from microphone bleed. This is most likely the cause of the poor classification accuracy as two different instrument types can be heard on the recording. This problem could be addressed by using an automatic noise gate to reduce the microphone bleed [26].

The four other multitracks had greater success than MT 1 when clustered, but this may be due to them having less audio tracks to cluster. When we compare the results of the entire audio feature set versus the reduced audio feature set we can see a big improvement in results. Especially in MT 2 and MT 4 where the misclassification measure dropped to 0% in both cases. In MT 3, when using the reduced audio feature set we see that we had only one misclassification. This was the ’Snaps’ audio track being subgrouped with the ’GangM’ vocal tracks and is depicted in Figure 6. There is a small amount of microphone bleed on the ’GangM’ vocal tracks, so this may be the reason why we are seeing this misclassifica-

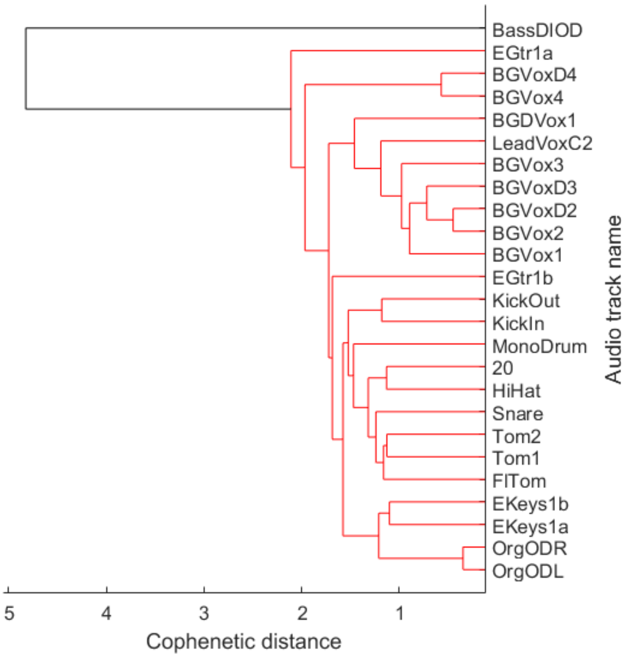

Figure 7: Dendrogram of MT 4 using the reduced feature set

tion. In MT 5 the misclassification is more difficult to explain as there does not seem to be any audible microphone bleed. This may be because the synthesiser has a similar timbre to the lead vocalist. Figure 8 shows that 'Synth21' is further away from the violins than the 'Synth11' is from the vocals, suggesting that 'Synth11' is similar to the vocal audio tracks.

When looking at Figure 4, Figure 5, Figure 6, Figure 7 and Figure 8 generally the lower parts of the trees tend to cluster the audio tracks together correctly. It is very easy to pick out drum, vocal and guitar clusters especially. The best examples are shown in Figure 5, Figure 6 and Figure 7. Interestingly, the 'Bass' audio track is the furthest distance from any other audio track in each of the multitracks. This most likely has to do with this instrument occupying the lower frequency bands and the rest of the instruments tending to be in mid and upper frequency ranges.

7. CONCLUSION

In this paper, we determined a set of audio features that could be used to automatically subgroup multitrack audio using a Random Forest for feature selection. We took a set of 159 low level audio features and reduced this to 74 low level audio features using feature selection. We selected these features from a dataset of 54 individual multitrack recordings of varying musical genre. We also showed that the most important audio features tended to be spectral features. We used the reduced audio feature set to agglomeratively cluster five unseen multitrack recordings. We then compared the results of the agglomerative clustering using the entire audio feature set to the agglomerative clustering using the reduced audio feature set. We were able to show that the overall misclassification measure went from 35.08% using the entire audio feature set to 7.89% using the reduced audio feature set. Thus indicating that our reduced set of audio features provides a significant increase in classification accuracy for the creation of automatic subgroups. Part of the novelty of this approach was that we were trying to classify audio tracks of entire multitrack recordings. Whereby, multitracks have the issue where recordings may contain artefacts such as microphone bleed. This did cause us problems in some cases, but we were easily able to identify the cause by listening to the problematic audio tracks.

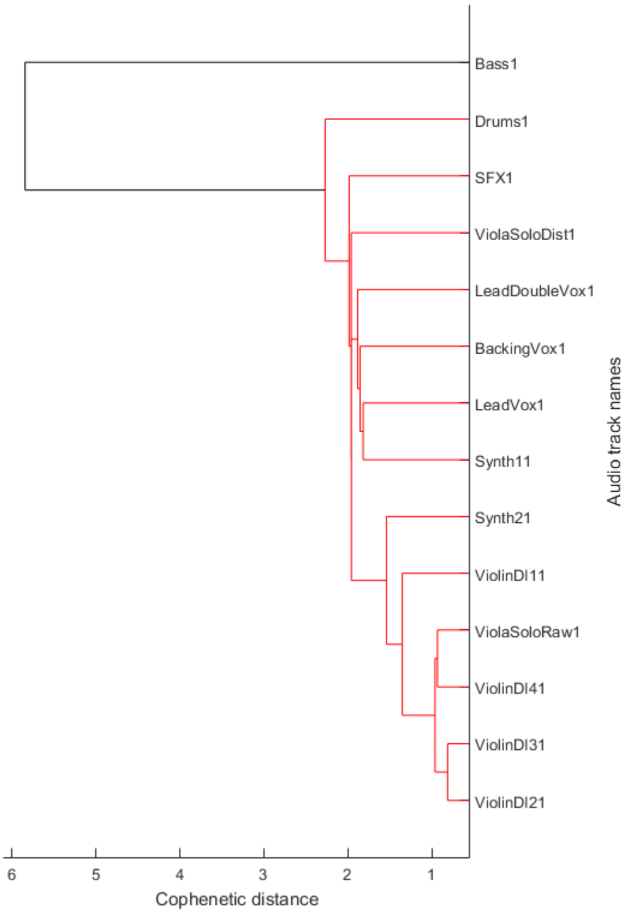

Figure 8: Dendrogram of MT 5 using the reduced feature set

In future work, automatic subgrouping could be applied to music from the Dance or Jazz music genres. In this case we only applied automatic subgrouping to Pop, Rock, Indie etc. However, it would seem that currently the subgroups for the Dance or Jazz music genres are not very well defined, so further research would be needed on best practices in subgrouping for music production of this kind. It would also be interesting to see how automatic subgrouping could be used in current automatic mixing systems like [27, 28, 29], where each automatic mixing algorithm is used on each subgroup of instruments individually to create a submix. Then once all the subgroups are automatically mixed, the automatic mixing algorithm would be used to mix each individual subgroup. In this work we inspected the correctness of the automatically generated subgroups manually, in further work we would like to test the validity of this technique automatically by using cross validation.

8. REFERENCES

[1] David Ronan, Brecht De Man, Hatice Gunes, and Joshua D. Reiss, "The impact of subgrouping practices on the perception of multitrack mixes," in *139th Convention of the Audio Engineering Society*, October 2015.

[2] Jeffrey Scott and Youngmoo E Kim, "Instrument identification informed multi-track mixing.," in *14th International Society for Music Information Retrieval Conference (ISMIR 2013)*, 2013, pp. 305–310.

[3] Jeffrey Scott, Matthew Prockup, Erik M Schmidt, and Youngmoo E Kim, "Automatic multi-track mixing using linear dynamical systems," in *Proceedings of the 8th Sound and Music Computing Conference, Padova, Italy*, 2011.

[4] Joshua D Reiss, "Intelligent systems for mixing multichannel audio," in *17th International Conference on Digital Signal Processing (DSP), 2011*. IEEE, 2011, pp. 1–6.

[5] Enrique Perez-Gonzalez and Joshua D Reiss, "Automatic mixing," *DAFX: Digital Audio Effects, Second Edition*, pp. 523–549, 2011.

[6] Philippe Hamel, Sean Wood, and Douglas Eck, "Automatic identification of instrument classes in polyphonic and polyinstrument audio.," in *10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, 2009, pp. 399–404.

[7] Judith C Brown, Olivier Houix, and Stephen McAdams, "Feature dependence in the automatic identification of musical woodwind instruments," *The Journal of the Acoustical Society of America*, vol. 109, no. 3, pp. 1064–1072, 2001.

[8] Antti Eronen and Anssi Klapuri, "Musical instrument recognition using cepstral coefficients and temporal features," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000. ICASSP'00*. IEEE, 2000, vol. 2, pp. II753–II756.

[9] Keith D Martin and Youngmoo E Kim, "Musical instrument identification: A pattern-recognition approach," *The Journal of the Acoustical Society of America*, vol. 104, no. 3, pp. 1768–1768, 1998.

[10] Brecht De Man, Brett Leonard, Richard King, and Joshua D. Reiss, "An analysis and evaluation of audio features for multitrack music mixtures," in *15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, October 2014.

[11] Philippe Hamel, Simon Lemieux, Yoshua Bengio, and Douglas Eck, "Temporal pooling and multiscale learning for automatic annotation and ranking of music audio.," in *12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, 2011, pp. 729–734.

[12] Christian Uhle, "Tempo induction by investigating the metrical structure of music using a periodicity signal that relates to the tatum period," in *6th International Society for Music Information Retrieval Conference (ISMIR 2005)*, 2005.

[13] Theodoros Giannakopoulos and Aggelos Pikrakis, *Introduction to Audio Analysis: A MATLAB® Approach*, Academic Press, 2014.

[14] George Tzanetakis and Perry Cook, "Musical genre classification of audio signals," *IEEE transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[15] Olivier Lartillot and Petri Toiviainen, "A matlab toolbox for musical feature extraction from audio," in *International Conference on Digital Audio Effects*, 2007, pp. 237–244.

[16] Geoffroy Peeters, "A large set of audio features for sound description (similarity and classification) in the cuidado project," 2004.

[17] Leo Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[18] Carolin Strobl, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn, "Bias in random forest variable importance measures: Illustrations, sources and a solution," *BMC bioinformatics*, vol. 8, no. 1, pp. 25, 2007.

[19] Carolin Strobl, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis, "Conditional variable importance for random forests," *BMC bioinformatics*, vol. 9, no. 1, pp. 307, 2008.

[20] Robin Genuer, Jean-Michel Poggi, and Christine Tuleau-Malot, "Variable selection using random forests," *Pattern Recognition Letters*, vol. 31, no. 14, pp. 2225–2236, 2010.

[21] Pavel Berkhin, "A survey of clustering data mining techniques," in *Grouping multidimensional data*, pp. 25–71. Springer, 2006.

[22] Stephen P Borgatti, "How to explain hierarchical clustering," 1994.

[23] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang, "A survey of audio-based music classification and annotation," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 303–319, 2011.

[24] Tim Pohle, Elias Pampalk, and Gerhard Widmer, "Evaluation of frequently used audio features for classification of music into perceptual categories," in *Proceedings of the Fourth International Workshop on Content-Based Multimedia Indexing*. Citeseer, 2005, vol. 162.

[25] Todor Ganchev, Nikos Fakotakis, and George Kokkinakis, "Comparative evaluation of various mfcc implementations on the speaker verification task," in *Proceedings of the SPECOM*, 2005, vol. 1, pp. 191–194.

[26] Michael Terrell, Joshua D Reiss, and Mark Sandler, "Automatic noise gate settings for drum recordings containing bleed from secondary sources," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, pp. 10, 2010.

[27] Enrique Perez-Gonzalez and Joshua Reiss, "Automatic equalization of multichannel audio using cross-adaptive methods," in *127th Convention of the Audio Engineering Society*. Audio Engineering Society, 2009.

[28] Michael Terrell, Andrew Simpson, and Mark Sandler, "The mathematics of mixing," *Journal of the Audio Engineering Society*, vol. 62, no. 1/2, pp. 4–13, 2014.

[29] Zheng Ma, Brecht De Man, Pedro Duarte Pestana, Dawn A. A. Black, and Joshua D. Reiss, "Intelligent multitrack dynamic range compression," *Journal of the Audio Engineering Society*, 2015.