6. SUPPLEMENTARY MATERIAL

The full list of all 106 features is presented below. Where the name of an entry in this list contains a slash (/), it describes more than one feature. Every symbolic music feature in jSymbolic2 has associated with it a short identifier code, for ease of reference within McKay's doctoral dissertation [5] and within the jSymbolic2 documentation. Where a feature was taken from or adapted from a feature in jSymbolic2, its identifier code is listed.

Highest / lowest note: MIDI note value of the highest and lowest notes in the input.

Number of notes: The number of note onsets in the input (counting tied notes) [2].

Range: The difference in semitones between the highest and lowest notes of the input (jSymbolic P-13).

Unique pitch classes: The number of unique pitches represented in the input (jSymbolic P-12).

Pitch mean: The average of all MIDI note numbers of the input (jSymbolic P-15).

Pitch standard deviation The standard deviation of all MIDI note numbers in the input.

Maximum / minimum interval: The signed values of the size in semitones of the largest and smallest intervals in the input, where descending intervals are negative and ascending intervals are positive.

Largest ascending / descending interval: The absolute value of the size in semitones of the largest ascending interval and the largest descending interval.

Proportion small intervals: The proportion of intervals in the input with size less than or equal to two semitones (jSymbolic M-11).

Proportion large intervals: The proportion of intervals in the input with size greater than or equal to five semitones. Adapted from (jSymbolic M-19), which specified a size threshold of an octave; jumps of an octave or more are rare in the songs of MTC-ANN.

Interval size mean: The average of the absolute size of all intervals in the input (jSymbolic M-3).

Interval size standard deviation: The standard deviation of the absolute size of all intervals in the input.

Position of highest / lowest note: The index of the first occurrence of the highest / lowest pitch, divided by the number of notes in the input.

Ascending or descending: Takes a value of 1 if the final note is higher in pitch than the first, -1 if lower, and 0 if the same.

Strictly ascending or descending: Takes a value of 1 if all intervals in the input are strictly ascending, -1 if all are strictly descending, and 0 otherwise.

Average interval signs: The proportion of intervals in the input that are strictly ascending.

Duration: The total duration of the input, in beats (jSymbolic R-36).

Longest / shortest note: The duration in beats of the longest and shortest notes of the input (jSymbolic R-19, R-20) .

Rhythmic density: The average duration of all notes in the input, in beats (jSymbolic R-17).

Rhythmic variability: Takes the standard deviation of the durations of all notes in the input, and saves the logarithm of that value [4].

Last note duration: The duration of the final note in the input.

Starts on downbeat: Takes a value of 1 if the first note of the input occurs at the beginning of a measure, and 0 if it does not.

Crosses measure: Takes a value of 1 if any two notes of the input occur in different measures, and 0 if the entire input occurs within a single measure or if there are no bar lines in the input's parent song.

Start / end beat strength: The "strength" of the beat on which the first / last note of the input lies. Beat strength is a measure of metrical accent, defined in music21's beatStrength note property [1]. Takes a value of 0 if no time signature is associated with the input.

Polynomial fit of order 1 / 2 / 3: Interprets the input as a point set of (onset time, pitch) pairs, normalized to have a total duration and range of 1, and then fits an n-order polynomial to this data for $n = 1, 2, 3$. The coefficient of the leading term of the polynomial fit is saved. When there are too few notes to specify a unique polynomial for the given order, takes the value 0.

Polynomial fit of order 1 / 2 / 3 residual: The residuals from the polynomial fits computed in the previous step. When there are too few notes to specify a unique polynomial or when the data fits a polynomial exactly, takes the value 0.

Pitch expected occurrences: Disregards rhythmic information. Assumes that each pitch of the song containing the input was generated using an order-0 Markov model, and computes the log probability of the input's pitches being generated from that model [3].

Interval expected occurrences: Disregards rhythmic information. Assumes that each interval of the song containing the input was generated using an order-0 Markov model, and computes the log probability of the input's intervals being generated from that model. Adapted from Conklin's [3] measure of expected occurrences for pitch data.

Rhythmic expected occurrences: Disregards pitch information. Assumes that the duration of each note in the song containing the input was generated using an order-0 Markov model, and computes the log probability of the input's note durations being generated from that model. Adapted from Conklin's [3] measure of expected occurrences for pitch data.

Difference in pitch mean: Takes the signed difference of the pitch means of the occurrence and the song containing the occurrence.

Difference in pitch standard deviation: Takes the signed difference of the pitch standard deviations of the occurrence and the song containing the occurrence.

Difference in rhythmic density Takes the signed difference of the rhythmic densities of the occurrence and the song containing the occurrence.

Difference in rhythmic variability Takes the signed difference of the rhythmic variabilities of the occurrence and the song containing the occurrence.

Difference in proportion of small intervals Takes the signed difference of the number of small intervals in the occurrence and the song containing the occurrence.

Difference in proportion of large intervals Takes the signed difference of the number of large intervals in the occurrence and the song containing the occurrence.

Difference in average interval signs Takes the signed difference of the average interval sign in the occurrence and the song containing the occurrence.

Pitch Class Histogram: Takes 12 values, each counting the number of times a particular pitch class appears in the occurrence. Normalized by dividing by the number of notes.

Interval Histogram: Takes 13 values, each counting the number of times a particular interval appears in the occurrence. Intervals of an octave or higher are all counted in the 13th bin. Normalized by dividing by the number of notes.

Rhythmic Histogram: Takes 5 values, each counting the number of times a note of a particular duration appears in the occurrence. Durations counted are: sixteenth-note, eighth note, quarter note, half note, whole note. Notes that are not exactly one of these durations are rounded down to the closest duration on the list. Normalized by dividing by the number of notes.

Pitch Class Histogram Difference: The signed difference of each bin between the pitch class histograms of the occurrence and the song containing the occurrence.

Interval Histogram Difference: The signed difference of each bin between the interval histograms of the occurrence and the song containing the occurrence.

Rhythmic Histogram Difference: The signed difference of each bin between the rhythmic histograms of the occurrence and the song containing the occurrence.

7. REFERENCES

[1] Christopher Ariza and Michael Scott Cuthbert. Modeling beats, accents, beams, and time signatures hierarchically with Music21 meter objects. In *Proc. of the International Computer Music Conference*, San Francisco, CA, 2010.

[2] Tom Collins, Robin Laney, Alistair Willis, and Paul H. Garthwaite. Modeling pattern importance in Chopin's mazurkas. *Music Perception: An Interdisciplinary Journal*, 28(4):387–414, 2011.

[3] Darrell Conklin and Mathieu Bergeron. Feature set patterns in music. *Computer Music Journal*, 32(1):60–70, 2008.

[4] Tuomas Eerola and Adrian C. North. Expectancy-based model of melodic complexity. In *Proc. of the Sixth International Conference on Music Perception and Cognition.*, 2000.

[5] Cory McKay. *Automatic music classification with jMIR*. Doctoral Thesis, McGill University, Montreal, Canada, 2010.

	Num. Clusters Ratio	Median Cluster Size	All Points Homogeneity	All Points Completeness	Significant Points Homogeneity	Significant Points Completeness
Embedding						
All Features	6.68 ± 1.37	**6.00 ± 0.28**	0.53	**0.13**	0.69	0.66
Exclude Pitch	8.19 ± 1.03	**6.00 ± 0.28**	**0.58**	0.11	0.70	0.65
Exclude Rhythm	7.15 ± 1.06	5.60 ± 0.21	0.57	**0.13**	**0.71**	0.65
Exclude Context	8.82 ± 1.18	5.40 ± 0.21	0.56	0.12	0.70	0.64
Exclude Histogram	6.51 ± 1.08	6.40 ± 0.45	0.53	**0.13**	0.70	0.67
Exclude Contour	**5.70 ± 0.87**	5.70 ± 0.18	0.47	**0.13**	0.63	**0.69**
PCA						
All Features	6.94 ± 1.39	4.40 ± 0.22	0.37	0.14	0.47	0.60
Exclude Pitch	**6.04 ± 0.91**	4.90 ± 0.09	0.37	0.13	0.47	0.62
Exclude Rhythm	6.32 ± 1.34	4.40 ± 0.21	0.39	**0.15**	0.48	0.62
Exclude Context	6.61 ± 1.05	4.60 ± 0.22	0.38	0.13	0.47	0.61
Exclude Histogram	6.42 ± 1.18	4.60 ± 0.22	0.38	0.14	0.49	0.63
Exclude Contour	6.26 ± 0.97	**5.90 ± 0.29**	**0.55**	0.13	**0.67**	**0.67**

Table 2. An extension of the results described in Table 1, excluding different parts of the feature set, using a value of ϵ_{15} for all runs of DBSCAN.

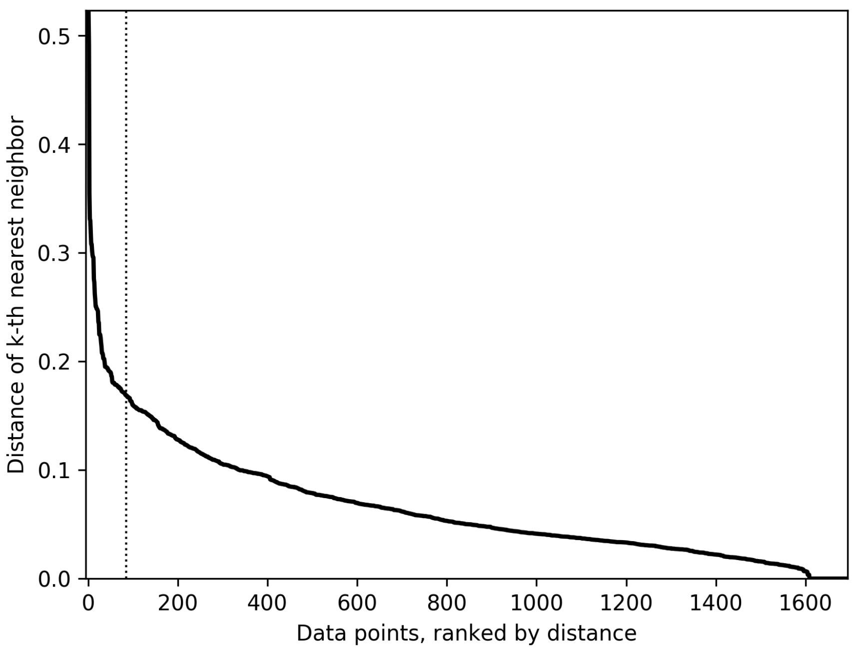

Figure 2. The k-dist graph on the data points used in one of the runs of the experiments in Table 1, after embedding into the 5-dimensional learned subspace, for $k = 3$. The position of the 5th percentile is marked with a vertical dotted line; the value at the intersection would be this run's value for ϵ_5.

Parameter & Value	Comment
`r` = 1	Specifies the number of superdiagonals to search in the similarity matrix built on the input point-set data.
`compactThresh` = 1	The minimum compactness, on a scale from 0 to 1, of each returned occurrence.
`cardinaThresh` = 3	The minimum number of notes in each occurrence. Note that when allowing variation between occurrences, two occurrences with different numbers of notes may found to be similar, and this means some 2-note occurrences are discovered with this parameter setting.
`regionType` = `'lexicographic'`	Defines which definition of region should be used when calculating occurrence compactness. Takes a value of either `'lexicographic'` or `'convex hull'`. Pattern occurrences in MTC-ANN are maximally compact by the lexicographic definition of region, so it is used to discover trivial examples as well.
`similarThresh` = 0.6	Takes a value from 0 to 1. If two patterns are judged to be more similar than this value, one is categorized as a variation on the other.
`similarFunc` = `'cardinality score'`	Determines the measure used to compute similarity between patterns. Takes a value of either `'cardinality score'` or `'normalized matching score'`. The latter uses a symbolic fingerprinting approach that in practice requires more than three notes per occurrence, so the former measure is used.
`similarParam` = 1	A parameter whose purpose changes depending on the choice of `similarFunc`. A value of 1 here tells the cardinality score matching algorithm not to penalize translations in pitch or time.

Table 3. The values used as parameters of SIARCT-C to create a set of trivial patterns.