

Contour clustering: manual for Praat-script and R-script

The following sections provide usage instructions to run the Praat-script and R-script in order to apply cluster analysis to f0 contours as introduced in the accompanying article. The scripts, a tutorial, and other supplementary materials can be downloaded here:

<https://constantijnkaland.github.io/contourclustering/>

1. Requirements

First, the three minimal requirements are discussed, which need to be fulfilled before the clustering approach can be applied. Although the required scripts are commented in a detailed way and provide references to the corresponding sections in this manual, a basic understanding of scripting in both Praat (Boersma & Weenink, 2019) and R (R Core Team, 2019; R Studio Team, 2019) is helpful.

1.1. Unit of analysis

The researcher is required to decide which linguistic unit of analysis needs to be investigated. The unit of analysis depends on which level of prosody is subject to the interest of the researcher (e.g. word or phrase level). For example, the procedure can be applied to investigate tonal contrasts at the lexical level as well as contours at the phrase level. In the former, the unit of analysis would typically be the syllable or word, whereas in the latter the unit of analysis would typically be the phrase. Note that some understanding of the prosodic relevance of these units in the language under investigation needs to be present. It is beyond the scope of this manual to provide fieldwork paradigms that help to identify the units that are relevant in the prosody of certain language. Studies have shown that cross-linguistic validation of prosodic units is crucial. With respect to intonation units, instructions to indicate “distinct units perceivable by means of a coherent melody” (Himmelman et al., 2018, p. 6) lead to considerable agreement among native *and* non-native annotators in different languages. This would hint at the universal nature of intonation unit perception, possibly cued by boundary tones.

Selecting the unit of analysis offers an opportunity to restrict the collected data according to the demands of the research question. For example, one could select only phrase final syllables/words, only question phrases, or only phrases occurring between two silent intervals (i.e. pauses) of a

specific length. In principle there is no restriction as to which unit can be analysed and the chosen unit can be further restricted by specifying a time-range (e.g. syllables/words/phrases with a certain duration). However, the decision for a unit has a couple of implications for later steps in the procedure. For this reason, it is advisable to keep the unit of analysis the same throughout one analysis, i.e. not comparing syllable-level f0 movements with those found in phrases.

1.2. Dataset

The researcher should have access to a collection of audio-recordings and corresponding annotations in Praat textgrids (Section 1.3 for annotation requirements), consisting of sufficient observations. What a sufficient amount of observations is, will depend on the quality of the recordings and a number of later steps in the analysis. For example, when only four clusters need to be identified, less observations are needed than when eight clusters need to be identified (see Section 2.2.4 for instructions on the number of clusters). It is likely that some data needs to be discarded due to f0 measurement errors. It is therefore recommended to collect more units than strictly speaking needed. In the phrase contour example (Article section 3), 321 contours were used for initial analysis, on the basis of which at least one clearly defined follow-up hypothesis could be formulated concerning potential functional differences between in the contours. For the tonal analysis (Article section 4) the contours of 213 syllables were analysed and again provided insightful results for further testing. Another indication of whether the dataset is large enough for cluster analysis can be obtained from the standard errors within each cluster (provided in the output, section 2.2.5). Large standard errors indicate large variation within the cluster, which could be the result of (too) few observations. It is important to note that in general more observations will often lead to more representative results and that the ideal size of the dataset largely depends on decisions taken during the analysis (see Section 2.1 and 2.2). Unbalanced datasets are not necessarily problematic for the cluster analysis. For example, a dataset with 5% questions and 95% statements could be accurately distinguished assuming that their difference is marked by means of distinct f0 contours.

1.3. Segmentation

Once the unit of analysis and the dataset are decided upon, the researcher needs to segment all units of analysis on an interval tier in a Praat textgrid (Boersma & Weenink, 2019). This means that the left and right boundary of each unit (an interval) need to be indicated. It is highly recommended to label the units using glosses in the language under investigation or another common language (e.g. English), or ideally both. The more information is available in the textgrid for each unit, the more

solid the interpretation of the results can be after analysis. This information can include other linguistic aspects that are not necessary included in the initial intonation analysis (e.g. segments, syllables, words, phrases, discourse units, grammatical annotations). For a typical dataset, the researcher has access to a number of audio recordings and corresponding textgrids, which are ideally, but not necessarily, equal to the number of speakers or sessions sampled in the field or lab.

2. Workflow

In what follows, two essential components of f0 contour clustering are discussed in the form of hands-on guidelines that follow the steps a researcher is suggested to take in the initial stages of the research. Each component corresponds to a script; first, a Praat-script for the collection of f0 measurements, and second, an R-script for the statistical analysis of the f0 measurements and for generating a cluster-annotated Praat textgrid. For researchers who prefer to do the entire workflow in R, an equivalent R-script is provided that could replace the Praat-script (using PraatR; Albin, 2014), although slower processing times are expected when measuring f0 using PraatR. Note that the use of Praat is highly recommended, not just for processing effort, but also for merging the generated textgrids with the original ones and for visual inspection of the outcome in the Praat editor window.

2.1. Praat-script: obtaining f0 measurements

The first part of the analysis takes f0 measures for each unit of analysis based on a combination of sound files and textgrids. Note that filenames need to be identical for the script to be able to link the textgrid annotation to the correct wavefile; only the extensions should differ (e.g. Recording_1.wav and Recording_1.TextGrid). Prior to running the script, the researcher is required to set a number of parameters that determine the way the measurements are taken. Apart from the directories that need to be specified for the respective audio files, corresponding textgrids and output-files, the parameters concerning the f0 analysis are explained in detail below.

2.1.1. Tier

The researcher needs to provide the name of the tier in the textgrid that contains the units of analysis. The measurements will be taken from the tier specified here. When multiple files and corresponding textgrids are used, the tier names containing the units of analysis should be identical across the textgrids.

2.1.2. Duration

The researcher has the option of restricting the analysis to units that have a length within a specified range (in seconds). This range needs to be specified prior to analysis. Setting the duration range is a means of making the set of analysed units more homogeneous. For example, if phrase-level contours are the target of analysis, the researcher can decide to measure only the most common phrase lengths (Article section 3 for an example). Note that the number of measures (Section 2.1.3) is set once and is therefore identical for all contours, thus automatically time-warping the measured contours. This provides the opportunity to compare compressed and uncompressed versions of the same contour (see Ladd, 2008, p. 180 for a discussion). Note that truncated contours are not accounted for by the duration settings. Setting the duration range may therefore be mostly needed when additional control over the unit length is desired, e.g. to exclude outlying lengths. Note that the largest degree of control in selecting a homogeneous dataset is obtained by selecting the unit of analysis (section 1.1). The script requires the duration range to be set in all cases. When no duration restrictions are needed, the range still needs to be defined and should be set using a very small number (zero is not accepted) as the minimum and a very high number as the maximum. For example, with a duration range between 0.00001 and 1000 seconds, most - if not all - intervals are taken into account for f0 measuring.

2.1.3. Number of measures

The researcher needs to decide how many f0 measurements are needed per contour. This largely depends on the unit of analysis and the desired accuracy of the measures. Taking too few measures could lead to missing essential f0 movements within the contour, whereas too many measures could give too much importance to insignificantly small f0 measures. Note that the stylization process (Section 2.1.6) could be responsible for over-estimation of small f0 movements. The standard setting of 20 measures appeared accurate for both phrase-level and syllable-level units of analysis (Article section 3 and 4). It is recommended to change the number of measure only in accordance with the stylisation resolution and the output of the cluster analysis (Section 2.2). If the output of the cluster analysis does not provide an indication of inaccurate measurements, there is no need to change the standard setting. It is important to note that inaccurate f0 measurements are likely to occur for each dataset. The cluster analysis provides an automatic detection of erroneous and outlying contours, which can then be removed from the data (Section 2.2.6). Changing the number of measures to account for a small number of erroneous contours is not recommended, as this affects also the correct measures. Only if the same type of f0 errors occur repeatedly, changing the number of measures could be considered.

2.1.4. Time-step

The time-step setting refers to the frame duration used by Praat to calculate f0. The standard setting used in Praat depends on the pitch minimum (or *pitch floor*; see Praat Manual “Sound: To Pitch...”; Boersma & Weenink, 2019) and defaults to 0.01 with a minimum of 75 Hz. However, manually specified time-step provides the researcher another way to configure the resolution of the f0 measures. This might be needed in addition to number of measures and stylization resolution, however the researcher is recommended to only change the time-step if necessary and otherwise default to 0.01.

2.1.5. Pitch minimum and maximum

The upper and lower boundaries of f0 calculation in Praat can be specified by the researcher. The default minimum of 75 Hz and default maximum of 500 Hz may be able to capture most f0 movements produced by the human voice. However, additional control is often desired to account for the gender of the speaker(s) or the type of phonation (Gordon & Ladefoged, 2001).

2.1.6. Stylization resolution

The stylisation process is handled in Praat and requires a resolution: i.e. a number indicating the minimum f0 difference between two adjacent f0 points to be taken into account in the stylised contour. The stylisation has two objectives. First, stylisation abstracts over f0 differences with a size below the specified resolution and is therefore able to discard insignificant f0 movements. Second, cluster analysis is best applied to non-missing data. In natural speech, f0 contours are frequently interrupted due to voiceless segments. It is known however, that listeners interpolate (fill in the voiceless gaps when perceiving the f0 contour) to some extent (Mixdorff & Niebuhr, 2013). Thus, by means of stylisation the researcher obtains f0 measurements that span the entire unit of analysis. The stylisation resolution is specified in semitones and a standard setting of 2 ST is recommended for initial analysis. Lower resolution values allow for more local variation in the f0 contours and could be needed to reveal phenomena that require more subtle f0 movements.

Note that with smaller numbers of measures higher degrees of stylisation are already obtained in the measurements. That is, taking 5 measures per unit of analysis could easily abstract over f0 movements that would be captured when taking 20 measures. However, with many measures, the risk increases that insignificant f0 perturbations fall within the stylisation resolution and become part of essential turning points in the stylised contour. It is highly recommended to test several combinations of number of measures and stylisation resolutions, to obtain the most accurate stylisations of the original contours. This can be done by running several rounds of measuring and

cluster analyses (Section 2.2), and evaluating the accuracy of the output each time before doing any interpretation of the results. The accuracy can best be evaluated by comparing the clustering output to the individual contours in the Praat editor window. When analysing spontaneous field data, it is likely that f0 measurement errors cannot be entirely avoided, even if the number of measures and the stylisation resolution have been set to optimal values. Therefore, erroneous cases are likely to be revealed by the cluster analysis and option to discard them are provided in Section 2.2.6.

2.1.7. Kill octave jumps

This setting allows the researcher to correct octave jumps in the units of analysis. When selected, octave jumps are removed before the stylisation process. These jumps can be the result of inaccurate f0 calculation in Praat and can be a challenge to fully correct for. This problem particularly applies to spontaneous speech and recordings made in acoustically challenging environments. When octave jumps are not handled correctly, extreme f0 movements end up in the stylised contour and will make the cluster analysis much less accurate. For example, a single contour with octave jumps might be different to all other contours in the data to such an extent that the contour forms a cluster on its own. Such an error could have an undesired effect on the outcome or at least hinders the interpretation of the results.

Although killing octave jumps is therefore recommended by default, some contours might be corrected for jumps even when there are none in the data (e.g. for instances where speakers use a wide pitch range). Furthermore, it is important to note that octave jumps are handled by assuming a correct “baseline” f0. Any jumps from this baseline will thus be lowered or heightened to this baseline. Deviations from a wrongly assumed baseline could therefore be treated erroneously as octave jumps. This could result in Praat adding or subtracting an octave to/from a large portion of the f0 contour, which would have otherwise been measured correctly. In order to obtain some insight into the octave jump handling, the pitch analysis calculates a change ratio by dividing the mean f0 of the contour after handling the octave jumps from the mean f0 of the contour before handling the octave jumps. In this way the researcher obtains a rough estimate of the overall effect of the octave jump handling on the f0 contour (written in the output file under “Jumpkilleffect”). If the change ratio decreases or increases the mean f0 to a large extent, it is recommended to inspect these cases for errors in measuring and/or octave jump handling. The option to specify the size of the allowed change is given when preparing the data for cluster analysis (Section 2.2.2). Obtaining a high number of extreme change ratios can also be a sign of inaccurate (advanced) pitch settings in Praat or problems with f0 calculation due to recording quality.

2.2. R-script: cluster (re-)analysis

The optimal settings used for clustering depend largely on the unit of analysis and the nature of the dataset. Some essential recommendations are given in this manual, however data-specific demands require careful consideration of each (combination of) setting(s). The main guidelines provided here help to recognize common problems as f0 related (Section 2.1) or cluster analysis related (this section). However, it is crucial to note that settings used in either part of the analysis should correspond as they depend on each other for a useful outcome.

2.2.1. Optional user interface

The R-script is provided in two versions; one without graphical user interface (no-gui) and one with graphical user interface (gui). The core elements are the same in either version of the R-script. In the no-gui version, the analysis is run directly from specific (combinations of) codeblocks in the script. Most of the settings required for usage of the no-gui version can be set in the code block “set variables” occurring at the beginning of that script. In this way, settings can be changed centrally and are intuitively visualized in the environment provided by the recommended software R-Studio (R Core Team, 2019). Note that the subsetting variables are not set centrally, but rather in their respective code blocks (section 2.2.6). The no-gui script provides additional comments explaining its code-block structure and reference to the relevant sections in this manual.

The gui version is provided using the Shiny package (Chang et al., 2020) and offers a possibility to execute the analysis without need to refer to the code. This is recommended for users who do not intend to alter the functionality of the script or do not master R scripting. The gui version also stores the most recently loaded datafile, dendrogram, table and plot automatically in the working directory until the script is stopped. The datafile is stored in two versions. A long version (‘data_long.csv’) essentially resembles the output of the Praat script which was uploaded in the gui (and optionally cleaned or subset). In the long version, one row represents a single f0 value. A wide version (‘data_wide.csv’) is stored with the most recent cluster analysis results written in an additional column (‘cluster’) in the dataframe. This provides a readable outcome of the cluster analysis for checking individual contours. In the wide version, one row represents a single contour, i.e. consisting of all f0 measurements that make up this contour (equal to the number of measures, Section 2.1.3). The gui version provides an option to save the current analysis (data_long, data_wide, dendrogram, table and plot) for later use (see also Section 2.2.5). If the files are saved by the user, they are cleaned upon closing the gui version.

2.2.2. *Selecting data for cluster analysis*

The f0 movements obtained from following the procedure outlined in Section 2.1 need to be read in R. By default, the no-gui script assumes that the output file with the f0 movements is stored in the same directory as that script. Reading the dataset should be done using the correct arguments for the command ‘read.csv()’ in R, depending on the formatting of datafile. All the available formatting arguments needed to read the data can be specified in the no-gui script. In the gui version, four arguments can be set that could overcome the most common issues; *separator* (default: comma, corresponding to the default output of the Praat-script), *stringsAsFactors* (default: false), *fileEncoding* (default: utf-8, change to utf-16 for more complex orthographic notations such as IPA), and *SkipNul* (default: false, set to true to omit nul bytes in the datafile). Note that when using the output file as directly generated by the Praat-script, the default values are likely to be the correct ones.

Prior to running the cluster analysis, it is important to obtain a dataset without missing values. Several common erroneous cases can be removed automatically from the data. The status tab in the gui version provides an overview of automatically generated statistics on the datafile. These include the number of contours in the data, the number of measurements taken (should correspond to the setting used in the Praat-script), number of speakers (equals number of different wave files), number of rows with empty filenames or labels, number of unused levels, number of f0 measurement errors. Whenever these statistics are inaccurate or reveal errors, the option ‘clean data’ should be selected and applied to the data. In most cases, this resolves the errors, as can then be checked from the updated statistics in the status tab. If errors remain, the datafile should be corrected manually or re-generated from the Praat-script.

In addition an option is provided to correct cases for which the octave jump handling resulted in deviations from the mean f0 (Section 2.1.7) that fell beyond the range of accepted values (default: 0.90-1.10, corresponding to a maximum allowed decrease or increase of 10%). It is likely that additional data cleanup is needed before the cluster-analysis can be interpreted usefully. How the remaining cleanup could be handled by the clustering technique is described in Section 2.2.6. Furthermore, the option to convert the f0 values as obtained in Hertz from Praat to semitones is provided. The semitone scale is a logarithmic one which approximates the perception of pitch more accurately than the Hertz scale (Nolan, 2003). Conversion to semitones is selected by default to obtain perceptually more intuitive contour representations.

2.2.3. *Controlling for speaker differences*

Six options are provided for handling speaker differences: 1) no correction, 2) subtracting the speaker's mean f0 from each measure, 3) standardizing the f0 measures, 4) min-max normalizing the f0 measures, 5) robust re-scaling the f0 measures, or 6) octave-median re-scaling the f0 measures. The correction methods are listed in Table 1.

Subtracting the mean f0 controls to some extent for speaker differences such as gender and speaking style. After applying this correction method, the dataset contains corrected f0 values centered around 0. Therefore, the corrected values could be negative. Negative values indicate that the speaker produced an f0 which is below the mean f0 of that speaker. Note that subtracting the mean f0 does not take into account speaker differences that concern f0 range. For example, a monotonous speaker will remain to show a flatter contour after subtracting the mean than a less monotonous speaker. These unaccounted differences could affect the cluster analysis. For example, consider a final rise that might be used to signal a question. This rise is likely to be shallow for monotonous speakers and steep for speakers with a more variable f0. As a result, the two types of speakers are likely to end up in different clusters. In that case, the cluster analysis will fail to correctly group a functional difference (question signaling) due to speaker differences. Or worse, all the contours of the monotonous speaker end up in a single cluster, because the larger f0 differences between contours of speakers with a more variable f0 will be clustered first (due to complete linkage, Article section 1.6). To reduce the influence of speaker differences on the cluster analysis due to f0 range, three additional methods are provided.

Standardizing the f0 subtracts the mean f0 a speaker from each f0 measure of that speaker and divides the outcome by the f0 standard deviation of that speaker. After standardizing, the mean f0 centers around zero and the standard deviation is one. The minimum and maximum f0 values still reflect the original distribution and outlying f0 values are therefore not accounted for. Note that if outliers are the result of f0 measurement errors or stylization inaccuracies, they can still effectively be handled by means of subsetting (section 2.2.6). Standardizing is a recommended method to preserve functional f0 differences such as tone contrasts when comparing different speakers (e.g. Rose, 1987).

Min-max normalization forces all f0 measures in a scale ranging from zero to one. This is done by subtracting the minimum f0 of a speaker from each f0 measure of that speaker and divides the outcome by the f0 range of that speaker (maximum-minimum). Thus, both monotonous speakers and speakers with a more variable f0 will show comparable f0 ranges after rescaling. However, the original minimum and maximum f0 values still affect the scaling process, which might give to much weight to outlying f0 values.

To reduce undesired effects of the outliers, robust scaling could be applied. In this method the median f0 of a speaker is subtracted from each f0 measure of that speaker and divided by the interquartile range of that speaker's f0 distribution (75th quantile – 25th quantile). In this way, outliers remain outlying and can still be accounted for in the subsetting procedures. The robust scaling method does not maintain a range of values between zero and one and might therefore offer a less strict comparison between speakers with varying f0 ranges.

Another method to scale the f0 range is based on the octave scale, taking into account the speaker's median f0. This makes the f0 measures more robust to outliers and is proposed as more representative method for speech melody compared to reference values based on the mean and expressed in semitones (De Looze & Hirst, 2014) . Octave-median scaling appeared particularly accurate for estimating acoustic emphasis as a result of focus, topic change or turn-taking.

By default, both the no-gui and gui scripts assume that each audio file (Section 2.1) corresponds to a unique speaker. If this is not the case or if all audio files are obtained from the same speaker, it is recommended to either not apply any corrections or to adjust the datafile and/or script such that speaker differences can be taken into account.

Table 1. Overview of speaker correction methods and their formulas. f0 = a single f0 measure (of a particular speaker), f0.spk = all f0 measures of that particular speaker.

Correction method	Formula
Subtract mean	$f0 - \text{mean}(f0.spk)$
Standardize	$f0 - \text{mean}(f0.spk) / \text{sd}(f0.spk)$
Min-max normalize	$f0 - \min(f0.spk) / \max(f0.spk) - \min(f0.spk)$
Robust rescale	$f0 - \text{median}(f0.spk) / 75q(f0.spk) - 25q(f0.spk)$
Octave-median rescale	$\log_2(f0.spk) / \text{median}(f0.spk)$

2.2.4. Cluster analysis / number of clusters

The type of cluster analysis – hierarchical clustering using complete linkage - is discussed in detail in the accompanying article (Article section 1.6). It goes beyond the scope of this manual to discuss the outcomes under different linkage criteria. Note, however, that other clustering techniques and linkage criteria are available and that the scripts could be changed to apply them accordingly.

Hierarchical clustering provides the researcher with a dendrogram, a tree-structure showing the outcome of each merge. The dendrogram provides a first visualisation of the clustering process. On the basis of the dendrogram the researcher can decide on the amount of clusters. Finding the ideal

number of clusters is the essential component in the approach outlined in the article. The dendrogram also provides insight in the scale of the f0 differences between the clusters. Since the largest numerical differences between clusters are found at the top of the dendrogram, an initial analysis might reveal only the differences involving a larger f0 range. In such an initial outcome, the dendrogram is likely to show asymmetry. This asymmetry is the result of late adjoining of major f0 excursions (e.g. large boundary tones) with a cluster containing all smaller scale (e.g. phrase-internal) f0 excursions. Asymmetry can be avoided to some extent by choosing a speaker correction method that accounts for outliers. Initial clustering outcomes thus provide insight into contour differences of the largest scale. Smaller scale differences, such as f0 peak height, are more likely to be successfully revealed when increasing the number of clusters. However, large numbers of clusters might result in clusters consisting of few contours, compromising the prototypicality of these contours. To accurately reveal small scale f0 differences, it is recommended to analyse a more homogeneous (controlled) dataset, or a subset of the data leaving out large-scale f0 excursions. The dendrogram provides useful directions for the degree and type of subsetting needed, as further discussed in Section 2.2.6.

Before deciding on the number of clusters, it is recommended for the researcher to have a theoretically motivated estimation of this number, i.e. before obtaining any result from the analysis. For example, if the aim is to find a basic set of different lexical tone contours from words, around four to six clusters could be sufficient to accurately capture the contrasts. However, if the aim is to find a broad set of all lexical tone contrasts in combination with phrase intonation patterns, e.g. 15 clusters could be the minimum number needed. One way of determining the number of clusters is to run several rounds of analysis, each time with an increasing number of clusters (Article section 2, 3 and 4). This is particularly useful in an exploratory context, where only a rough estimation of the number of clusters can be made. It is also recommended to run the cluster analysis with a number of clusters that exceeds the hypothesized number. In this way, the researcher can reduce the risk of missing relevant contrast that were unexpected. The dendrogram provides the initial guidance in finding the ideal number of clusters for a certain analysis. Obtaining the dendrogram for a given dataset does not require any prior decisions on the number of clusters. That is, due to the hierarchical clustering technique used here, a dendrogram can be obtained *before* the number of clusters is set. Commonly, the dendrogram shows asymmetry as the result of a cluster with a single observation being merged with a cluster with many other observations. If the aim is to find prototypical f0 contours, it is questionable whether single-observation-clusters contribute to this aim (see also Section 2.2.5). They might constitute either outliers that fail to represent the dataset or unique cases which the researcher could consider for further individual analysis. It is not

recommended to remove small clusters from the dataset without inspection. The subsetting procedures described in Section 2.2.6 can reduce the asymmetry in the dendrogram. An indication of the accuracy of the number of clusters can also be derived from plotting the mean contours per cluster, as further explained below. Apart from the general guidelines outlined here, there are statistical methods to obtain an estimation of the ideal number of clusters in an analysis, which are left for the user to explore (e.g. Charrad et al., 2014).

2.2.5. Interpreting the output

Once the researcher has settled on a number of clusters for initial analysis, the output of the cluster analysis can be used to create a number of essential data summaries that guide the interpretation. Apart from the dendrogram, the current approach provides three summaries; a table, a plot and an automatic annotation.

The table provides 1) the number of contours in each cluster, 2) the mean standard error per cluster, as calculated by adding up the standard errors for each measurement point and divide the outcome by the number of measurement points, and 3) a tentative indication (flagging) whether a cluster should be treated with caution. The flagging is given by a zero (no caution) or a one (caution advised) based on the data obtained in (1) and (2). That is, caution is advised when a cluster contains only one contour or when the mean standard error is more than two times the median of the mean standard errors of all clusters. The logic behind the latter threshold is to obtain a single criterion that can be applied to f_0 values regardless of the speaker correction method or number of clusters chosen (Section 2.2.3). In particular, standard errors might be affected by some of these correction methods. By taking the median as center value, the distance from zero (no deviation) is known. Thus, mean standard error values that lie further away from median than zero does, are advised to be treated with caution. The criterion is a rough estimate of how deviant the mean contour in a particular cluster is. Individual inspection is still advised to be able to quantify the standard errors on a known f_0 scale (e.g. Hertz or semitones). Therefore, the inspection should be done before applying speaker correction. For example, a single cluster with a mean standard error of 10 Hz might already be enough to overlook potentially meaningful differences between the contours. That is, the f_0 range for human speech can be taken as 75 to 500 Hz (standard setting in Praat; Boersma & Weenink, 2019). Thus, allowing for an average deviation of up to 10 Hz for the entire contour corresponds to a semitone difference that lies between 0.3 (480-500 Hz) and 2.2 (75-85 Hz). It has been shown that in Dutch, for example, 1.5 ST excursions (locally!) can be enough to perceive a linguistically meaningful prominence shift (Rietveld & Gussenhoven, 1985). Although the generalizability of this finding to other languages is open to further research, it is crucial to note

that within cluster variation of 10 Hz might not fully avoid overlooking important f_0 movements in the contour. The double-median threshold for the mean standard errors could therefore be too crude without consideration of the f_0 scale. While smaller standard errors are generally preferred, increasing the number of assumed clusters can help to gain insight into this type of variability and is recommended prior to any subsetting procedures. The standard errors can thus be taken as an indication of how well the contours fit in the cluster. As a rule of thumb: if homogeneously sized clusters with a low mean standard error can be achieved with subsetting applied only to discard erroneous cases, the analysis is likely to provide an optimal outcome.

The plot provides a line through the mean f_0 values for each measurement in each cluster and draws a shaded area around the line indicating the standard deviation. In addition, the number of observations in each cluster, as also given in the distribution table, are plotted for reference. The plot can then be used to link a prototypical contour onto an individual observation. It is recommended to check individual observations in order to inspect the small clusters, and to obtain immediate feedback on the accuracy of the f_0 measures and the chosen number of clusters. If only minimal differences between the clusters are found, the f_0 contours might have been too much stylised and/or the number of clusters too high. If the plotted mean contours show a large number of local f_0 changes (i.e. not smooth, large standard deviations), the f_0 measurements could have been too fine-grained, possibly magnifying the differences between the clusters and failing to accurately approach the prototypical shape of a contour (see Section 2.1 for a discussion on the relevant settings).

It is plausible that some of the clusters in the plot show highly similar contours. This is to be expected with a (recommended) high number of clusters for initial analysis. It is recommended to not solely rely on the plot of mean contours to inspect the potential similarity between two clusters. It might not be immediately clear why two similar contours end up in separate clusters. Inspection of the actual acoustic differences between the contours is recommended in such a case. This can be done by reading the f_0 values from the plot. These values should be compared to more obvious differences between other clusters obtained from the same analysis. For example, rising phrase final boundary tones could have a large f_0 range and therefore be visually easy to detect. This type of “landmark” in the contour can provide an indication of the scale of the differences between clusters. Thus, visually similar contours in separate clusters might also be the result of small numerical differences which could be more accurately clustered after subsetting. Decreasing the number of assumed clusters could also be considered as a means to obtain visually more distinguished contours (an example of such a consideration is given in Article section 3). However, this method

makes the cluster analysis more course-grained, introducing a higher risk of overlooking relevant contour differences.

For more thorough interpretation of the clustering analysis, two functions are provided: saving the dendrogram, data files, table and plot (gui version) and generating textgrids. Saving the files as the output of a particular cluster analysis is useful for future reference and further inspection and avoids redoing the cluster analysis once the number(s) of clusters are settled on. Generating textgrids essentially indicates for each contour to which cluster it belongs. The generated output is a textgrid file with one tier that can be merged with the existing textgrid that contains the interval tier from which the unit of analysis was taken for f0 measures (Section 2.1). With the plotted mean contours as a reference, the researcher can now go through the wave file and textgrid in Praat, and explore the potential meanings of each prototypical intonation contour. It is crucial to note that the cluster labelling is likely to differ when the cluster analysis is re-run with different settings (i.e. number of clusters, cluster technique, different ways of analysing f0 etc.). It is recommended for the researcher to try different settings and repeatedly go back and forth between running the analysis and interpreting the outcome. There is no a priori indication of how many times the analysis needs to be run for the researcher to obtain an informed interpretation of the data, and to ultimately be able to attribute specific functions to specific contour shapes. A typical analysis could consist of an initial run on all data and subsequent runs on informed subsets, as a means of “zooming” into cluster differences that would otherwise remain masked due to asymmetry (see the article; Section 1.6 for a theoretical motivation and Section 2 and Section 3 for examples). However, much depends on the type of speech data or contours under analysis, and the language at hand.

The automatic annotation allows the researcher to compare the original contour in Praat with the mean contour representing a cluster that was generated by the analysis. Interpreting the automatic annotations could therefore reveal f0 measurement errors. It is plausible that a contour is assigned to a cluster without actually sharing similarities with the plotted mean contour of that cluster. In this case, f0 measurement errors are the most likely cause of the mismatch. Although these errors cannot be avoided in all cases, it is recommended to inspect them individually and improve f0 measurement settings or change the number of clusters.

2.2.6. Subsetting

A procedure for subsetting the data is provided in the no-gui and gui version. This procedure has two purposes. Its initial use can be applied to a high number of clusters (Article section 3 and 4 for examples) in order to remove erroneous or outlying contours. For this purpose, a high number of clusters (e.g. 25 or more) is recommended to obtain small sized clusters. If the data consists of

erroneous or outlying contours, these will then be revealed in small sized clusters, potentially single-contour clusters. Thus, when removing small sized clusters, the risk of discarding error-free contours remains low. This way of “pruning” the data can be done using the automatic flagging of erroneous/outlying contours (Section 2.2.5).

A second purpose of the subsetting is to “zoom in” into a specific subset of the data, typically after initial round(s) of clustering. For example, initial clustering could reveal a small number of rising contours among an overall majority of falling contours. It could be useful to separate the rises and falls in subsets and perform subsequent clustering on either subset. This has the advantage of revealing smaller scale differences between the contours (e.g. different types of rises or falls), without their differences being affected by contours of the other category. Such an application of the subsetting procedure is particularly useful when there is clear indication or supportive evidence from distinguishing these two types as categories.

In the non-gui version, an additional automatic subsetting procedure is provided. This procedure is based on setting a minimum portion of the data to be left after subsetting. Then, the smallest cluster is removed in each round of cluster analysis until the minimum portion is reached. This subsetting procedure provides a crude means of removing non-prototypical contours, particularly useful for large datasets with a high degree of asymmetry in the dendrogram. Note that this procedure should only be run in the Rstudio environment (Rstudio Team, 2020), in order to keep track of the outcome. In particular, as the risk exists that non-erroneous contours will be removed from the data. A general word of caution should be given here, as subsetting essentially ignores (potentially large) parts of the collected data. This procedure compromises the representativeness of the empirical investigation and re-introduces the risk of giving researchers’ intuitions a decisive role in the analysis-process. Although these disadvantages cannot be entirely avoided, it is crucial to keep the ultimate goal of the investigation in mind. The procedures outlined here are designed to reveal prototypical contours, for which some deviant instances can be naturally expected in spontaneous speech (see further discussion in the Article section 3).

3. References

- Albin, A. L. (2014). PraatR: An architecture for controlling the phonetics software “Praat” with the R programming language. *The Journal of the Acoustical Society of America*, 135(4), 2198–2199. <https://doi.org/10.1121/1.4877175>
- Boersma, Paul, & Weenink, David. (2019). *Praat: Doing Phonetics by Computer* (6.0.56) [Computer software]. <http://www.praat.org/>
- Chang, W., Cheng, J., Allaire, J. J., Xie, Y., & McPherson, J. (2020). *shiny: Web Application Framework for R*. <https://CRAN.R-project.org/package=shiny>
- Charrad, M., Ghazzali, N., Boiteau, V., & Niknafs, A. (2014). NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. *Journal of Statistical Software*, 61(6). <https://doi.org/10.18637/jss.v061.i06>
- De Looze, C., & Hirst, D. (2014). The OMe (Octave-Median) scale: A natural scale for speech melody. *7th International Conference on Speech Prosody 2014*, 910–914. <https://doi.org/10.21437/SpeechProsody.2014-170>
- Gordon, M., & Ladefoged, P. (2001). Phonation types: A cross-linguistic overview. *Journal of Phonetics*, 29(4), 383–406. <https://doi.org/10.1006/jpho.2001.0147>
- Himmelman, N. P., Sandler, M., Strunk, J., & Unterladstetter, V. (2018). On the universality of intonational phrases: A cross-linguistic interrater study. *Phonology*, 35(2), 207–245. <https://doi.org/10.1017/S0952675718000039>
- Mixdorff, H., & Niebuhr, O. (2013, August 26). *The Influence of F0 Contour Continuity on Prominence Perception*. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH.
- Nolan, F. (2003). Intonational equivalence: An experimental evaluation of pitch scales. *Proceedings of the 15th International Congress of Phonetic Sciences*, 39, 771–774.
- R Core Team. (2019). *R: The R Project for Statistical Computing* (3.5.3) [Computer software]. <https://www.r-project.org/>
- R Studio Team. (2019). *RStudio: Integrated Development for R* (1.0.143) [Computer software]. RStudio, Inc. <https://www.rstudio.com/>
- Rietveld, A. C. M., & Gussenhoven, C. (1985). On the relation between pitch excursion size and prominence. *Journal of Phonetics*, 13(3), 299–308. [https://doi.org/10.1016/S0095-4470\(19\)30761-2](https://doi.org/10.1016/S0095-4470(19)30761-2)
- Rose, P. (1987). Considerations in the normalisation of the fundamental frequency of linguistic tone. *Speech Communication*, 6(4), 343–352. [https://doi.org/10.1016/0167-6393\(87\)90009-4](https://doi.org/10.1016/0167-6393(87)90009-4)