Meertens Online Reports

Meertens Instituut, Amsterdam, Netherlands
www.meertens.knaw.nl

ISSN 2352-2135

The Meertens Tune Collections: MTC-FS-INST 2.0

Peter van Kranenburg, Martine de Bruin

2019-1

The Meertens Tune Collections: MTC-FS-INST 2.0

Peter van Kranenburg, Martine de Bruin
Meertens Institute, Amsterdam
{peter.van.kranenburg,martine.de.bruin}@meertens.knaw.nl

Contents

1 Introduction

This report describes in detail the data-set MTC-FS-INST 2.0, which is part of the Meertens Tune Collections (Van Kranenburg et al., 2014). MTC-FS-INST 2.0 contains more than 18 thousand melodies in various formats, including **kern, MIDI and pdf. Syllabized lyrics are included as well. The melodies were collected from Dutch sources from five centuries, including prints, manuscripts, and field recordings. Metadata include: source, tune family membership, composer (if known), segmentation at phrase level, key, meter, dating, geotags (for field recordings).

2 Background

The Meertens Institute of the Royal Netherlands Academy of Arts and Sciences (Amsterdam, Netherlands) hosts a large collection of Dutch songs. These are accessible via the online interface of the Dutch Song Database *(Nederlandse Liederenbank)*,[1] which contains metadata of c. 175 thousand songs, full texts of c. 50 thousand songs, and music notation for more than 18 thousand songs.[2] The online interface offers search functionality and access to songs on an individual basis. Each song record contains a rich collection of metadata fields.

With the *Meertens Tune Collections* (MTC), the Meertens Institute releases several collections of digitized songs. The core of these collections consists of digitized melodies. The MTC are aimed to serve those who wish to use entire data-sets instead of individual songs.

The availability of digitized musical material is important for several disciplines. From a musicological perspective, the study of these melodies is relevant for understanding transmission and dissemination of folk tunes. For research in the field of Music Information Retrieval (Schedl et al., 2014), the availability of large annotated data sets is of crucial importance to test algorithms and tools, such as segmentation algorithms, melodic similarity measures, and pattern discovery algorithms. In the area of music cognition, analysis of the songs may yield insight in melodic segmentation, melodic similarity and the stability and variability of motifs, intervals and rhythms.

3 History of the Data Set

The data set MTC-FS-INST 2.0 is successor of both MTC-FS 1.0 and MTC-INST 1.0 (Van Kranenburg et al., 2014). In an ongoing digitization effort, thousands of melodies have been newly digitized since the release of both version 1.0 data sets, and many corrections and improvements have been made to the existing data. Most of these additions and changes are included in the current release, version 2.0. In the version 1.0 releases, we had reasons to split the collection into folk songs (FS) and instrumental pieces (INST). These represented two well separable sub-collections, FS mainly including 19th and 20th century vocal folk songs, and INST mainly including 18th century instrumental tunes. It, however, proved difficult to maintain the strict separation between these two types in the 2.0 release. One consequence would have been that pieces from the same source would end up in different data sets, since several sources include both vocal and instrumental music. Therefore, we decided to combine both types in the current release into one big set, MTC-FS-INST, while maintaining annotation of the type (vocal/instrumental) in the metadata.

[1] Dutch Song Database: `http://www.liederenbank.nl`

[2] These are the numbers at the end of 2018. Data and metadata are continually added.

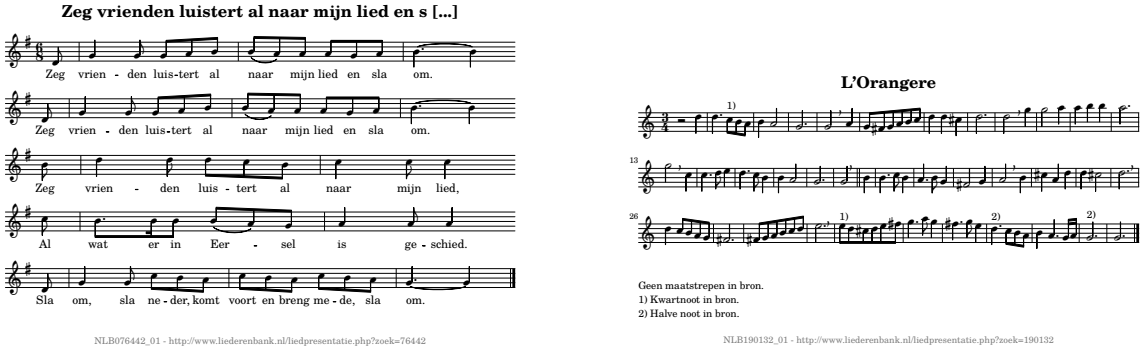

Figure 4.1: Examples of a vocal song (left), and an instrumental piece (right).

4 Origins of the Melodies

The melodies have been transcribed from over 200 sources dating from the 16th up to the 21st century. The two main sub-corpora in the data set are: 1) instrumental pieces mainly from the 17th and 18th centuries, and 2) vocal songs, mainly from the 19th and 20th centuries. Figure 4.1 shows representative examples.

The instrumental sub-corpus includes popular melodies from mainly 17th and 18th century Dutch sources. Most of the tunes have been printed or written in monophonic form, without accompaniment. The core of the collection consists of tunes from the printed series *Oude en nieuwe Hollantse Boeren Lietjes en Contredansen* (Old and New Dutch Peasant Songs and Countrydances, Amsterdam: Estienne Roger 1701–1714, 13 vols.), *De Hollantsche Schouburgh* (Amsterdam: Estienne Roger and Michel-Charles Le Cène, 1714–c.1730, 7 vols.), and *De Nieuwe Hollandsche Schouwburg* (Amsterdam: Johannes Smit, 1751–1771, 9 vols.). Although many of these dance and song tunes may originate in low culture, others have been identified as melodies from operas by French court composers such as J.B. Lully (1632–1687). With his 'Peasant Songs' Roger must have aimed at well-to-do amateurs, playing the violin, flute or oboe. The same tunes are found in Dutch music manuscripts from the same era, but in slightly varied form. Many of those have been included in the data set as well. In general, the tunes are short and simple. Characteristic forms are menuet and marche. This repertoire may be regarded as the Dutch equivalent of John Playford's *The English Dancing Master* (1651, many reprints up to 1728).

The vocal sub-corpus consists of digitized melodies both from the *Onder de groene linde* sound recordings, and from printed songbooks. For his radio program *Onder de groene linde* (Under the Green Linden; 1957–1994), Dutch song collector Ate Doornbosch (1926–2010) made c. 5,000 audio recordings of songs mainly sung by elderly people who knew the songs from their youth. These songs were used during work on the fields or in workshops, or at home. Around 3,750 of these recordings were transcribed into music notation in a major project at the Meertens Institute during the 1990s. These transcriptions consist mainly of ballads, but also include a minority of simple folk songs and children's songs. In MTC-FS-INST 2.0 virtually all of these are included. In addition to the transcriptions of sound recordings, a large amount of melodies are included that are taken from printed songbooks such as Jan Frans Willems, *Oude Vlaemsche Liederen* (1848), Florimond Van Duyse, *Het oude Nederlandsche lied* (1908), and Jop Pollmann and Piet Tiggers, *Nederland's volkslied* (1941). Initially, we digitized those songs that also appear among the recordings of *Onder de groene linde*. In a later stage, we also digitized many melodies from unrelated sources, including 16th–18th century songbooks.

5 Transcription Principles

Preparing a transcription of a score often is a challenging process. Especially in written sources, one could encounter (systematic) errors, idiomatic symbols, difficult to read handwriting, etc. But, also printed sources often pose the transcriber for problems. In those cases, the transcription partly is an act of interpretation, in which, inevitably, subjective decisions are involved. To help the transcribers as much as possible, some guiding principles have been established. The primary reference for the transcription is the notation as

it is in the source. The transcribers stick to the source as close as possible. Changes are only made in cases where the requirements of modern notation or the limitations of the digital encoding prevents a literal transcription of the source (e.g., incomplete measures, problems with the key signature). In principle, every such change with respect to the source is accounted for in a footnote. Generally, from polyphonic sources (e.g., keyboard manuscripts), only the melody is transcribed without further notification in a footnote.

6 Detailed Description of MTC-FS-INST 2.0

6.1 General Remarks

6.1.1 Entering and encoding

The melodies have been entered using an online editor that has been developed by the Meertens Institute.[3] This editor offers an integrated online environment including a scan of the source (if available), input fields for metadata, a textual input field for entering the melody and a textual input field for entering the lyrics. The textual encoding that is used to encode a melody is a subset of the input encoding for music typesetting software LilyPond (see Section 6.2.2 for details).

6.1.2 Songs, Stanzas and Voices

Most of the vocal items contain a transcription of the first stanza of the song. In some cases, mainly those concerning transcription from field recordings, the transcriber has opted to transcribe another stanza as well, or even to skip the first stanza altogether. Possible reasons are the singer having initial problems recalling the melody correctly, or considerable melodic variation between the stanzas.

In the instrumental sources, occasionally a second voice — often a bass part — is digitized as well.

Since each file only contains one stanza or one voice, there are more files than songs in the data set.

6.1.3 File Naming

The file names of content files have the following format: `NLBxxxxxx_yy.ext`, where `xxxxxx` is the record number of the song in the Dutch Song Database, preceded with leading zeros if necessary, `yy` is a serial number which has different meanings for the different types of songs, and `ext` indicates the file format. For vocal songs `yy` indicates the number of the stanza that is encoded in the file. For instrumental melodies `yy` indicates the voice: `01` for first voice, `02` for second voice, and so on. For example, `NLB074344_01.mid` designates a midi file, containing the first stanza of the song with record number 74344.

To quickly retrieve the full song description from the Dutch Song Database, just put the record number in the search field on the front page of the online interface at `http://www.liederenbank.nl`.

6.1.4 Hard Breaks and Soft Breaks

All melodies have been segmented into phrases. In most cases, the positions of phrase boundaries have been determined by the transcriber or encoder, which implies a certain level of subjectivity. The phrase boundaries were encoded by inserting a *hard break* (i.e., a line ending) in the manual input in our online editor.

For *vocal songs*, sometimes the source provides a partitioning of the song into phrases. But in many cases, decisions had to be made by the transcribers and encoders. For these decisions both musical and textual cues such as rhyme and closed syntactical units have been used.

For the *instrumental pieces*, there are no textual cues. Therefore, encoders were instructed only to insert hard breaks at clear structural divisions such as repeat bar lines, which, however, often results in relatively long segments. To have a finer segmentation as well, an additional level of phrasing was added by introducing *soft breaks*, encoded as `\sb`. These were intended to indicate points of perceived melodic closure according to the musical intuition of the encoder. As cue the encoders were instructed to take

[3] `http://speelmuziek.liederenbank.nl`

those places in the melody that serve as a 'natural' breathing point during singing or playing. However, in practice, the various encoders that contributed to the MTC did not apply these instructions consistently, or not at all. Therefore, a clear distinction in meaning between encoded soft breaks and hard breaks should not be expected, and hard breaks may occur at positions other than clear structural divisions. Furthermore, because of the inherent ambiguity of melodic segmentation, the breaks should be regarded as one possible, musically valid, way to divide the melody into smaller units. Often, segmentation is not consistent among members of the same tune family.

In the provided pdf renderings of the scores, the visualization of the breaks depends on the type. For vocal songs, a hard break results in a system break, while a soft break is visualized with a breathing sign. For instrumental pieces, both hard and soft breaks are visualized with a breathing sign.

6.1.5 The Concepts of Tune Family and Text Family

Two concepts are used to group songs: *tune family*, and *text family*. Because of the variability due to the process of (oral) transmission, almost no two instances of the same song are exactly identical in a literal sense. In our collections, we rather have groups of variants, both for texts and melodies. For variants of the same melody, we use the concept of *tune family*.[4] By analogy, to indicate a group of songs with the same textual contents, we use the concept of *text family*. These songs are designated in the metadata to belong to the same text family. The categorization in both tune and text families has been done over the course of many years by the collection specialists at the Meertens Institute, and is subject to constant revision. Therefore, tune family labels are not guaranteed to be consistent across different versions of MTC data sets. Three levels of confidence are used to assign a tune family label to a melody: `strong`, `neutral`, and `weak`. Sometimes a melody is clearly related to a tune family, but only partially, or with specific differences. In that case it is marked as *variant* of the tune family, and the confidence is designated as `weak`.

6.1.6 File Types

The following file types are used in MTC-FS-INST 2.0:

pdf portable document format
`https://www.adobe.com/devnet/pdf/`

krn Humdrum **kern representation
`http://www.humdrum.org/`

ly LilyPond source file
`http://www.lilypond.org/`

wce witchcraft editor file
See section 6.2.2.

txt UTF-8 plain text file
`http://www.unicode.org/`

mid, midi midi sequence
`http://www.midi.org/`

csv UTF-8 text file containing comma-separated values
`http://tools.ietf.org/html/rfc4180`

All textual files (.krn, .ly, .wce, .txt, .csv) have UNIX line endings and character encoding UTF-8.

[4] The concept of *tune family* has been introduced and defined by Samuel Bayard (1950) and has been used in American folklore studies. In Bayard's definition it is the (presumed) 'genetic' relation between the melodies that establish a tune family. In Europe, the concept of *type* (Typus) has been used more often, be it with a multitude of definitions. An important example is the edition of melody types of German folk song by Suppan & Stief (1976).

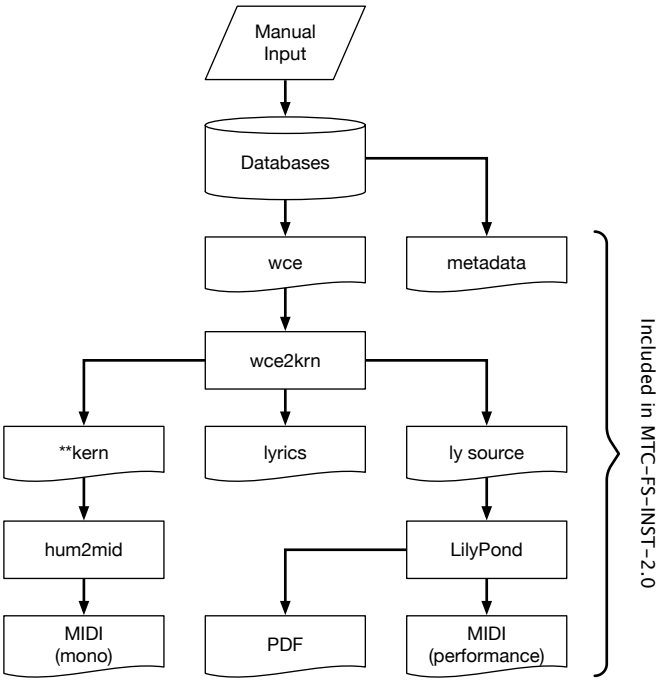

Figure 6.1: Conversion flow.

6.2 Contents

The package contains the following directory structure:

MTC-FS-INST-2.0
- COPYRIGHT.txt
- README.txt
- VERSION.txt
- **krn**
- **ly**
- **lyrics**
- **metadata**
- **midi_mono**
- **midi_performance**
- **pdf**
- **wce**

The **metadata** directory contains the tables with metadata. The other directories contain various representations of the songs.

The manual input of the melodies is contained in the witchcraft editor (wce) files (see Section 6.2.2). The Humdrum **kern files, the LilyPond source files, and the lyrics files have been generated from the wce files using the `wce2krn` converter (see Section 6.2.2.3). The midi files in directory `midi_mono` are generated from the **kern files using humdrum extra command `hum2mid`,[5]. The midi files in directory `midi_performance` are generated by LilyPond 2.19.80 from the LilyPond source files. The pdf renderings of the scores have been generated from the LilyPond source files. The conversion flow is depicted in Figure 6.1.

[5] `http://extras.humdrum.org/`

6.2.1 Metadata

Metadata are provided in text-files that can be found in the directory `metadata`. The character encoding is UTF-8, and the line endings are in UNIX style for all text files. The metadata are contained in tables that are formatted as comma-separated-values (csv). Most values are enclosed in double quotes. Field-headings are provided in separate files. Sometimes a field contains a list, for example if there is more than one singer in a recording, or if there is more than one author of a song book. In that case, the pipe symbol, `|`, is used as separation marker. Names are formatted as: `Last Name, First name`. As an example:

```
Wolsey, Boy|Waldorp, Jan
```

means Boy Wolsey and Jan Waldorp.

6.2.1.1 Metadata Table MTC-FS-INST-2.0

Metadata table MTC-FS-INST-2.0 contains metadata for the individual melodies. Not all fields contain values for each song. There are, for example, fields that relate to audio recordings, such as recording date, or singer. These are empty for songs that have been transcribed from written sources.

filename the basename of the file the metadata record refers to (string).

songid the record number of the song in the Database of Dutch Songs (integer).

source_id identifier of the source of the song - provides a cross-link with table MTC-FS-INST-2.0-sources (integer).

serial_number the serial number of the song in the source (string).

page the page number of the song in the source (string).

singer_id_s identifiers of one or more singers - provides a cross-link with table MTC-FS-INST-2.0-singers (list of integers).

date_of_recording date of recording (DD-MM-YYYY).

place_of_recording place of recording, in most cases the name of the municipality (string).

latitude latitude of place of recording (float).

longitude longitude of place of recording (float).

textfamily_id identifier of the text family the song belongs to (integer).

title title of the song (string).

firstline first line of the lyrics (string).

tunefamily_id short identifier of the tune family (ASCII string).

tunefamily full name of the tune family to which the song belongs (string).

type type of the song ({`vocal`, `instrumental`}).

voice_stanza_number serial number of the stanza or voice that is in the file (integer).

voice_stanza whether the value of voice_stanza_number is a voice (in case of instrumental music) or a stanza (in case of vocal songs) ({`voice`, `stanza`}).

image_filename_s list of filenames of images in MTC-OGLSCAN that belong to the song (list of strings).

audio_filename filename of the audio recording in MTC_OGLAUDIO that belongs to the song (string).

variation whether the tune should be considered a more distant variant within the tune family (`variant`) or a full member (∅).

confidence the confidence with which the melody was assigned to the tune family ({`weak`, `neutral`, `strong`, ∅}).

comment extra information for some of the field recordings (string).

MTC_tite title as shown in the pdf renderings, an assemblance of various fields (string).

author composer(s) of the melody (list of strings). The name of a composer is followed by either `(composer)` or `(original composer)`. In the second case, the melody has been arranged. If the name of the composer is not explicitly mentioned in the source, it is included in square brackets.

6.2.1.2 Metadata Table MTC-FS-INST-2.0-sources

source_id identifier of the source (integer).

title title of the source (string).

author author(s) of the source (list of string).

place_publisher place(s) and publisher(s) of the source (list of strings). Place and publisher are separated by a colon (":").

dating (approximate) dating of the source (string).

sorting_year year that can be used for sorting (integer).

type type of source ({`manuscript`, `print`, `audio`}).

copy_used library siglum of the copy that was used for digitization (string).

scan_url url of scanned images of the source, in case the source is publicly available online (string).

6.2.1.3 Metadata Table MTC-FS-INST-2.0-singers

singer_id identifier of the singer (integer).

year_of_birth year of birth of the singer (integer).

place_of_birth place of birth of the singer, in most cases the name of the municipality (string).

latitude latitude of place of birth.

longitude longitude of place of birth.

6.2.2 WCE Files

6.2.2.1 Contents The wce-files contain the manual input of the encoders. The file format is inherited from the predecessor of the online editor, the Witchcraft-editor, which was developed during the WITCHCRAFT-project (Wiering et al., 2009) as a Mac OS X application to enter melodies and lyrics. Other file types such as **kern, LilyPond, lyrics, score (pdf), and midi files are derived from the wce files by using various converters. The various data and metadata fields are stored in the wce-files according to the Apple property-list XML format.[6] The following fields are included in the wce-files:

NLBController-recordIDTextField Record number of the song in the Database of Dutch Songs.

SignatureController-stropheNumberTextField Number of the stanza/voice (first is 1).

[6] `http://www.apple.com/DTDs/PropertyList-1.0.dtd`

SignatureController-titleTextField Title of the song.

SignatureController-footerTextView If a LilyPond source file is assembled from the contents of the wce-fields, the contents of this field should be appended at the end of the resulting LilyPond source file. Currently, the only use is to add footnotes to a score.

SignatureController-isMeterInvisibleSwitch True if the melody has a free meter.

SignatureController-keyTextField Key in LilyPond encoding (e.g., `c \major`, `d \dorian`, `a \minor`, etc).

SignatureController-partialTextField Length of upbeat (if any) in LilyPond duration encoding (e.g., `8*3`).

SignatureController-tempoTextField Tempo in LilyPond encoding (e.g., `4=120`).

SignatureController-clefTextField Initial clef (`treble` or `bass`).

SignatureController-relativeTextField The pitch the first note is related to according to the LilyPond `\relative` command (e.g., `g'`).

SignatureController-timeTextField The (initial) meter of the song in LilyPond encoding (e.g., `9/8`). A meter is always provided in this field, also in the case of a melody in free meter. In those cases the meter can be ignored in processing.

TextProcessingDefaults-inputTextView A text field containing the melody and lyrics. Melody and lyrics are interwoven. The melody of each phrase is directly followed by the lyrics of the phrase (if any). Phrases are separated by a blank line. For the encoding of the melody a subset of LilyPond's music encoding is used. The encoding of the lyrics is also according to the LilyPond format.

6.2.2.2 Encoding The exact subset of the LilyPond encoding that is used is defined in the input files for the lexical analyzers as included in the source distribution of `wce2krn`: `lilylexer.ll` and `textlexer.ll` (see Section 6.2.2.3). The encoding of the bar lines is according to LilyPond versions $\geqq$ 2.18. The following LilyPond definitions are used in the melody encodings:

```
sb = {\breathe}
mBreak = { \bar "" \break }
bBreak = { \break }
x = {\once\override NoteHead #'style = #'cross }
gl=\glissando app = #(define-music-function (parser location notes)
    (ly:music?) #{ \appoggiatura $notes #} )
kvs = #(define-music-function (parser location notes) (ly:music?)
    #{ \slashedGrace $notes #} )
itime = #(define-music-function (parser location timesig)
    (fraction?) #{ \once\override Staff.TimeSignature #'stencil = ##f
    \time $timesig #} )
ficta = {\once\set suggestAccidentals = ##t}
fine = {\once\override Score.RehearsalMark #'self-alignment-X = #1
    \mark \markup {\italic{Fine}}}
dc = {\once\override Score.RehearsalMark #'self-alignment-X = #1
    \mark \markup {\italic{D.C.}}}
dcf = {\once\override Score.RehearsalMark #'self-alignment-X = #1
    \mark \markup {\italic{D.C. al Fine}}}
dcc = {\once\override Score.RehearsalMark #'self-alignment-X = #1
    \mark \markup {\italic{D.C. al Coda}}}
ds = {\once\override Score.RehearsalMark #'self-alignment-X = #1
    \mark \markup {\italic{D.S.}}}
```

```
dsf = {\once\override Score.RehearsalMark #'self-alignment-X = #1
    \mark \markup {\italic{D.S. al Fine}}}
dsc = {\once\override Score.RehearsalMark #'self-alignment-X = #1
    \mark \markup {\italic{D.S. al Coda}}}
pv = {\set Score.repeatCommands = #'((volta "1"))}
sv = {\set Score.repeatCommands = #'((volta #f) (volta "2"))}
tv = {\set Score.repeatCommands = #'((volta #f) (volta "3"))}
qv = {\set Score.repeatCommands = #'((volta #f) (volta "4"))}
xv = {\set Score.repeatCommands = #'((volta #f))}
```

The following settings are needed in the LilyPond source:

```
\set melismaBusyProperties = #'()
\override Score.MetronomeMark #'transparent = ##t
\override Score.RehearsalMark #'break-visibility = #(vector #t #t #f)
```

Normally in LilyPond if notes are slurred, one lyrics syllable is aligned with the entire group of slurred notes. The first setting (melismaBusyProperties) disables this behavior. This means that for each subsequent note in a melisma, an underscore is needed in the lyrics input.

6.2.2.3 wce2krn The **kern files, LilyPond source files, and lyrics files have been generated from the wce source files using `wce2krn`, a converter for the wce format that has been developed by the Meertens Institute. The source code is available at github.[7] For MTC-FS-INST 2.0, `wce2krn` version 1.82 was used.

6.2.3 **kern Files

The **kern files have been generated from the manual input by the `wce2krn` converter (see Section 6.2.2.3). `wce2krn` is able to convert most of the input information into the **kern representation. In principle the generated **kern files adhere to the **kern representation as specified in the *Humdrum Users Guide*. However, there are some caveats. These issues will be discussed in the following sections.

6.2.3.1 Segmentation During digitization, all melodies have been segmented into phrases. As explained in Section 6.1.4, there are two kinds of segment bounary markers: hard breaks and soft breaks. In early stages of the digitization of the collection, these were meant to represent two hierarchical levels in segmentation. It proved, however, not possible to maintain this distinction. Nevertheless, the distinction between hard and soft breaks has been made visible in the **kern representation. For each segment, the start is indicated in **kern with a global comment, preceding the first note of the segment:

```
!!  segment x.y
```

where `x` denotes the level of hard breaks, and `y` denotes the level of soft breaks. Furthermore, each segment is indicated with **kern phrase markers `{` and `}`, respectively at the first and last note of the segment.

6.2.3.2 Structural indicators Structural markers, such as first and second endings, segno marks, coda marks, and indications such as *da capo*, or *da capo al fine* have not been properly converted to into **kern. The proper way to indicate structure in **kern is to use section labels and expansion lists (c.f. *The Humdrum User Guide*, Chapter 20). Given the variability of structure indications in the we input encoding, it proved not possible to do so in the current release.

In case of first and second endings (*prima* and *secunda volti*), omitting the structural indication results in melodies that cannot be processed linearly from the beginning to the end. For that reason, all melodies containing first and second endings have not been included in the conversion to **kern. For this reason, there are less **kern files than there are wce files in MTC-FS-INST 2.0. If desired, one could obtain the **kern representations of the omitted files by running `wce2krn`. The volti indications will be included as local comments in **kern.

[7] `https://github.com/pvankranenburg/wce2krn`

Lilypond	**kern
\trill	Tt
\prall	Mm
\prallprall	Tt
\mordent	Ww
"//"	O
\turn	S
\staccato	'
\staccatissimo	`
\- (tenuto)	~
\accent	^

Table 1: Conversion of Lilypond ornamentations and articulations to **kern signifiers.

Other structural indicators have been included in the **kern representation as local spine comment. E.g., a *da capo al fine* results in:

```
!  D.C.
```

6.2.3.3 Dynamics Dynamics have not been coverted to **kern.

6.2.3.4 Ornaments Various ornaments and articulations are included in the manual input. These are represented by LilyPond commands and have been converted to **kern signifiers according to Table 1.

In **kern, it is required to indicate whether an ornament involves a semitone or a whole tone below or above the main note. Upper case T, M, and W respectively denote a whole tone trill, mordent, and inverted mordent, while lower case t, m, and w respectively denote a semitone trill, mordent, and inverted mordent. Since the interpretation of ornaments as they occur in the sources is not deterministic in terms of semitone or whole tone intervals, the choice for upper or lower case **kern signifier cannot automatically be made. A way to represent this ambiguity in **kern is provided in Chapter 2 of the *Humdrum User Guide*.[8] The offered solution is to include both the upper and lower-case letter. We follow this solution.

The "double-slash" ornament is found quite often in the instrumental sources. There is no LilyPond command for it. Therefore, it has been entered as a TextScript. Because **kern does not present a signifier for it as well, the general ornamentation signifier O has been taken.

6.2.3.5 Chords In the manual input, simultaneous notes are represented using the LilyPond chord notation. In most cases, these concern double stops in violin parts. The encoders were instructed to put the main note at the first position in the chord. This main note is not necessarily the highest note in the chord. The chords are converted to a single note in **kern by wce2krn. Because of the convention, always the first note of a chord is included in the **kern representation of the melody.

6.2.3.6 Encoding of grace notes The manual input contains four types of grace notes: grace note with slashed stem (LilyPond macro: \kvs), cue sized notes before the main note without slur (LilyPond macro: \grace), cue sized notes before the main note with slur (LilyPond macro: \app), and cue sized notes after the main note (LilyPond macro: \afterGrace). The \app macro is currently not used anymore for new input since it is also possible for the encoder to explicitly insert a slur, but the macro still occurs in the data. The *Humdrum User Guide* specifies the signifier q for slashed grace notes (*acciaccaturas*) and the signifier Q for groups of cue sized notes (*grupettos*). We follow this specification in MTC-FS-INST-2.0. Table 2 shows the conversion of the four types of grace notes.

6.2.3.7 Footnotes and Free Text The encoders are able to add free text below or above a note or rest using LilyPond TextScript. This allows them to include information from the source for which not an explicit

[8] http://www.humdrum.org/guide/ch02/

LilyPond	**kern
`\kvs { a8 } g4`	`8aq` `4g`

`\grace { g16 a } g4`	`16gQ`
	`16aQ`
	`4g`

`\app { g16 a } g4`	`(16gQ`
	`16aQ`
	`4g)`

`\grace { g16( a } g4)`	`(16gQ`
	`16aQ`
	`4g)`

`\afterGrace g4 { a16 g }`	`4g`
	`16aQ`
	`16gQ`

Table 2: Examples of conversion of grace notes to **kern, assuming `\relative g'` for the LilyPond pitches.

LilyPond command or macro is available. This is used for ornaments, character indiciations (“allegro”, “andante”, “un poco Largo”, etc.), structural indicators, which occur in a wide variety of spellings in the sources (“1mo Allegro Da Capo”, “4 Fois”, “4 maal”, “4 mal”, “Da Capo of van t begin”, “da capo toute la piece”, “Da cap.”, “na de herhaling op segno”, etc. etc.), and for other kinds of free text as notated in the source.

Free text is included in **kern as a local comment directly after the note to which the text belongs, and a `??` signifier at the note to which the text belongs.

The same mechanism is used for editorial footnotes. In this case the free text contains the serial number of the footnote, while the text of the footnotes (in Dutch) is included in a global comment at the end of the **kern file. E.g., as follows:

```
[...]
=16
2ff#
4r
4ee??
!  1)
4r
4aa
[...]
=:|!
*-
!!  1) Halve noot in bron.
[...]
```

6.2.3.8 Compatibility with music21 All **kern files can be parsed by Python module music21 (Cuthbert & Ariza, 2010). There is, however, one caveat. Because of an apparantly ambiguous passage on note durations in Chapter 6 of the *Humdrum User Guide*, music21 does not allow dotted notes in tuplets.[9] In fact, a dotted note in a triplet, has the same duration as a non-triplet note without a dot. For example, in **kern, the durations `12.` and `8` would mathematically imply exactly the same duration in clock time. However, to keep a faithful representation of the music notation as found in the original sources, we opted to allow dotted triple-notes. For music21 to parse these, a flag needs to be set, as follows:

[9] `http://www.humdrum.org/guide/ch06/#tuplets`
`https://groups.google.com/d/msg/music21list/hXoJO0c4scc/STtScCJyq5cJ`

```
import music21 as m21
m21.humdrum.spineParser.flavors['JRP'] = True
```

To be sure, always set this flag when parsing **kern files from MTC-FS-INST-2.0.

6.2.4 LilyPond Files

The included LilyPond source files have have been generated by `wce2krn` 1.82 from the manual input as contained in the wce-files. The encoding of the bar lines is according to LilyPond versions $\geqq$ 2.18.

6.2.5 MIDI Files (mono)

The midi files in directory `midi_mono` have been generated with the Humdrum Extras tool `hum2mid` (version 30 August 2018).[10]. Therefore, all limitations that apply to the **kern files, also apply to these midi files (see Section 6.2.3). Notably, chords have been replaced by the main note of the chord, and all melodies with volti have been omitted. Furthermore, all grace notes have been removed before converting to midi.

6.2.6 MIDI Files (performance)

The midi files in directory `midi_performance` have been generated from the LilyPond source files by LilyPond 2.19.82. These contain full chords, and all grace notes. The fraction of the duration of the main note that is used for the grace notes is determined by LilyPond's default algorithm.

6.2.7 Lyrics Files

The `lyrics` directory contains the syllabized lyrics for each vocal song as produced by `wce2krn` from the wce source files.

6.2.8 PDF Files

The `pdf` directory contains the pdf renderings as produced by LilyPond 2.19.82.

7 Using MTC-FS-INST 2.0 as a background corpus for MTC-ANN 2.0.1

It could be desirable to use the large set of melodies that is provided by MTC-FS-INST as background corpus for the melodies in MTC-ANN (Van Kranenburg et al., 2016). For that purpose, it is necessary to remove from MTC-FS-INST all melodies that are somehow related to the melodies in MTC-ANN. This could be accomplished by the following procedure:

1. Identify all melodies in MTC-FS that are also in MTC-ANN.
2. For each of these melodies, obtain the tunefamily_id from the metadata of MTC-FS-INST (metadata table MTC-FS-INST-2.0).
3. From the tunefamily_ids remove the part after the underscore (the *sub-identifier*) to retain the *main-identifier*. E.g. `9744_1` becomes `9744`.
4. Remove all 633 melodies from MTC-FS-INST that are in tune families that have the main-identifiers that are in the list that results from step 3. See below for a full list.
5. Remove all 5,856 melodies from MTC-FS that do not have a tune family identifier at all.

[10] `https://github.com/craigsapp/humextra`

The reason to disregard the sub-identifier of tunefamily_id is that there might be relations between tune families with the same main-identifier, but different sub-identifiers. E.g., tune family `9744_1` (Er reed er eens een ruiter 1) might be related to `9744_2` (Er reed er eens een ruiter 2). This is not always the case, but to be absolutely sure to remove all melodies that are related to melodies in MTC-ANN it is better to remove all tune families with the same main-identifier. For the same reason, it is necessary to remove all melodies without tune family label. Among these might be members of one of the MTC-ANN tune families that have not been identified yet.

This is the full list of tune families that need to be removed in step 4 from MTC-FS-INST 2.0 to serve as a background corpus for MTC-ANN 2.0.1: ∅, 1419_1, 1419_2, 1419_3, 1419_4, 1419_5, 1507_1, 1507_2, 1507_3, 1507_4, 1507_5, 1507_6, 2694_1, 2694_2, 2694_3, 2774_0, 3676_1, 3676_2, 5301_0, 5301_1, 5301_2, 5301_3, 5301_4, 5301_5, 5301_6, 6382_0, 8592_0, 9664_1, 9664_2, 9665_1, 9665_2, 9665_3, 9665_4, 9665_5, 9668_1, 9668_2, 9668_3, 9668_4, 9668_5, 9673_1, 9673_2, 9673_3, 9673_4, 9673_5, 9673_6, 9673_8, 9675_0, 9676_0, 9677_1, 9677_2, 9686_1, 9686_2, 9686_3, 9693_0, 9700_1, 9715_1, 9715_2, 9722_1, 9722_2, 9722_3, 9722_4, 9743_1, 9743_2, 9744_1, 9744_2, 9744_3, 9744_4, 9744_5, 9749_0, 10043_0.

The resulting background corpus consists of 12,099 melodies that are unrelated to the melodies in MTC-ANN 2.0.1.

8 Distribution and Availability

MTC-FS-INST 2.0 is provided as a gzipped tar file. The tar file expands into a directory with name and version number of the collection. This top-level directory contains subdirectories for each of the representations (pdf, krn, etc.), and for the metadata. The tar archive can be downloaded from `http://www.liederenbank.nl/mtc`.

Long-term access is guaranteed by the Meertens Institute. As trusted data repository, the Meertens Institute obtained the CoreTrustSeal.[11]

9 License and Attribution

If you use the data for a research paper, please cite this report:

> Van Kranenburg, Peter, Martine de Bruin (2019). The Meertens Tune Collections: MTC-FS-INST 2.0. *Meertens Online Reports* 2019-1, Meertens Institute, Amsterdam.

To employ this data for commercial purposes, explicit permission should be obtained from the Meertens Institute.

Acknowledgements

We thank collection specialist Ellen van der Grijn for sharing her extensive knowledge of the contents of the data sets, Hester Drost, Jasper van den Bergh, and several students for entering many songs.

References

Bayard, S. (1950). Prolegomena to a study of the principal melodic families of british-american folk song. *Journal of American Folklore*, 63(247), 1–44.

[11] `https://www.coretrustseal.org/about/`

Cuthbert, M. S. & Ariza, C. (2010). Music21: A toolkit for computer-aided musicology and symbolic music data. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR 2010)* (pp. 637–642).

Pollmann, J. & Tiggers, P. (1941). *Nederland's volkslied*. Haarlem: De Toorts.

Schedl, M., Gomez, E., & Urbano, J. (2014). Music information retrieval: Recent developments and applications. *Foundations and Trends in Information Retrieval*, 8(2-3), 127–261.

Suppan, W. & Stief, W., Eds. (1976). *Melodietypen des Deutschen Volksgesanges*. Tutzing: Hans Schneider.

Van Duyse, F. (1903–1908). *Het oude Nederlandsche lied*. Den Haag / Antwerpen: Martinus Nijhoff / De Nederlandsche Boekhandel.

Van Kranenburg, P., De Bruin, M., Grijp, L. P., & Wiering, F. (2014). *The Meertens Tune Collections*. Meertens Online Reports 2014-1, Meertens Institute, Amsterdam.

Van Kranenburg, P., Janssen, B., & Volk, A. (2016). *The Meertens Tune Collections: The Annotated Corpus (MTC-ANN) Versions 1.1 and 2.0.1*. Meertens Online Reports 2016-1, Meertens Institute, Amsterdam.

Wiering, F., Veltkamp, R. C., Garbers, J., Volk, A., & Van Kranenburg, P. (2009). Modelling folksong melodies. *Interdisciplinary Science Reviews*, 34(2–3), 154–171.

Willems, J. F. (1848). *Oude Vlaemsche liederen*. Gent: F. en E. Gyselynck.