

Universidad Nacional de Quilmes
Taller de Tesis

Presentación de artículo referente

Docente: **Esteban Calcagno**

Alumno: **Lisandro Fernández**

Septiembre, 2017

Reseña sobre “Humdrum Toolkit”:
Herramientas UNIX para Investigación Musical, creadas
por David Huron

**A Review of the Humdrum Toolkit: UNIX Tools for Musical
Research, created by David Huron**

Jonathan Wild

Music Theory Online, Volume 2, Number 7, November 1996

Copyright © 1996 Society for Music Theory

Abstract: David Huron's Humdrum Toolkit is a set of software tools created for computer-assisted musical research. This review describes and evaluates the Toolkit and the music representation schemes that support it.

1. Introducción

[1.1] Justificacion

Music-theoretical endeavors vary greatly in their breadth. At one end of the spectrum are accounts of a particular composer's technique within a particular piece; at the other, attempts to answer the most general questions about how music works. As an investigator's aims tend toward the latter end of the continuum, he or she must explore a larger and larger repertoire. Compounded with the level of intimacy with musical details that a systematic inquiry can require, a large body of music can quickly exceed the capacity of one scholarly lifetime. An indispensable step in pursuing this broader type of research is thus the accumulation and maintenance of a large, searchable computer database of music. Even the study of repertoires of manageable size is more flexibly undertaken when digital representation and automated manipulation of the music is available.

[1.2] Contexto

The Center for Computer Assisted Research in the HGumanities (CCARH) at Stanford has an ongoing musical database project; among other items virtually all of J. S. Bach's oeuvre has been electronically encoded. Of vital importance to such a task's usefulness is the format of musical representation within the database. As the success of computer-assisted musicology becomes more widely apparent, musical representation schemes are likely to proliferate, jockeying for position until such time as some sort of standard is agreed upon. The evaluation of representation schemes has been a hot topic in the not-so-distant past;⁽¹⁾ however, most of the schemes discussed were designed to aid composers of computer music, whose requirements do not necessarily intersect with those of investigative scholars. In CCARH's database, several formats for encoding music are employed concurrently, all subsets of CCARH's master representation MuseData. The structure of the database and the aims of the various schemes have been described elsewhere; most promising among the schemes is the Humdrum representation, designed by David Huron.

2. Humdrum

[2.1]

Humdrum comprises two distinct aspects, corresponding to the representation and manipulation of musical data. First, there is the syntax defined for representing discrete information as a series of records in a computer file. Its definition allows for many kinds of information to be encoded; the scheme used in the CCARH database for essential musical pitch and duration information is just one of an open-ended set—some other members of which I mention below—that may be augmented by user-defined schemes for specific research tasks. Second, there is the set of commands, the Humdrum Toolkit, devised to manipulate files that conform to the Humdrum syntax in the name of machine-aided research in music. The emphasis here should be on “aided”: Humdrum possesses no higher-level analytical faculties per se. Rather, its power derives from the flexibility of its kit of elemental commands, which the user must learn to use in combination to exploit fully the system’s potential.

3. Humdrum Representations

[3.1]

Before discussing the ways in which the Toolkit can be used it will be helpful to inspect the conventions of one particular Humdrum representation, by means of a musical fragment. Example 1 (below) contains the first two phrases of the Bach chorale “Nun danket alle Gott” in Humdrum’s core pitch/duration representation scheme, `**kern.^`(2) In this example, the first three lines, each tagged by a triple exclamation mark, are reference records, each of whose meaning should be readily apparent. Humdrum specifies over 80 types of reference records for electronic scores, providing for information as diverse as the catalogue number, name of the dedicatee, manuscript location or translated title in over forty languages.

EXAMPLE 1

[3.2]

A Humdrum file is comprised of a series of records, each taking up one line. The four columns that follow the reference records are termed spines; they are separated by tab characters. Events in the same data record occur simultaneously, while events in the same spine but in distinct data records occur sequentially. The heading ****kern**, which is the exclusive interpretation of each spine here, signifies that the information within concerns pitch and duration. A file may contain any number of spines with various exclusive representations; in this example each spine represents one voice of a four-part chorale. The fifth data record of Example 1 contains a tandem interpretation for each spine which denotes the key of the excerpt; other tandem interpretations within ****kern** contain information about meter signature, tempo, instrument etc

[3.3]

A ****kern** representation may be thought of as a rotation, 90-clockwise, of the musical score. Thus the leftmost spine corresponds to the lowest voice, and time runs down the page instead of left to right. Additional spines, possibly running concurrently with ****kern**, could contain either fairly straightforward things, such as vocal text (****text** representation) or a Roman-numeral harmonic analysis (****harm** representation), or relatively abstruse information such as cochlear coordinates (****cocho**), sensory dissonance (****diss**) or acoustic spectral data (****spect**). Some of these other representations can be generated as output by commands operating on a ****kern** spine or set of spines. For example, the **pcset** command would automatically append to the chorale in example 1 a ****pcset** spine that labeled each simultaneity by its Fortean set-class name, prime-form or interval vector. The data for a ****harm** spine, on the other hand, must be entered manually (until a dependable machine algorithm for harmonic analysis is developed).

[3.4]

Each encoded note in example 1 has at least two parts, corresponding respectively to the note's duration and pitch: an integer, and one or more upper- or lower-case letters. The integer represents duration via the reciprocal of the North-American note-value; for example, a quarter-note by "4" and a sixteenth-note by "16." A breve (length of two whole-notes) takes the special value 0. Eighth-note triplets would each be allotted the integer 12—since they each span a third of a quarter-note, logically they are twelfth-notes. A note participating in a septuplet in the time of a dotted quarter-note would receive the value 28 (one seventh of a quarter-note, or a 28th-note) augmented by a dot; a further subdivision of this note into three parts would yield a dotted 84th-note (!). This convention may at first appear opaque, but manages succinctly and consistently to express almost all durations. Exceptions that spring to mind include the admittedly obscure odd subdivisions of a breve, such as whole-note triplets (three to a breve), and such Ferneyhoughian atrocities as "four in the time of three notes of a quintuplet," which require the use of superfluous

[3.5]

The lower- or upper-case letters denote the pitch-class and octave of a note; chromatic alterations are of course also provided for, as are a large number of articulation markings (such as the tie in the alto part of Example 1, shown by the square brackets), ornaments, editorialisms, and stem and beam indicators. It is important to note that each of the components of a note's representation is optional: a file containing durations only could be compiled were one's object to investigate solely the rhythmic character of a set of melodies.

Rests are indicated by combining a reciprocal duration with the letter "r"; the frequent period characters are merely place markers, "null tokens" in a data record with activity in another spine. Barlines appear as records with a leading equals sign; an elaborate system of visual correspondences distinguishes between various forms of double barlines and/or repeat signs. Dynamics are conspicuously absent in ****kern**, and indeed do not appear in any of the pre-defined representation schemes documented in the manual⁽³⁾—should they be required for a specific task the researcher may invent his or her own scheme. Guidelines for this are given in a chapter of the

[3.6]

As may be surmised, thoroughly encoding a piece of music can be a fairly laborious process. Unless electronic editions of the works under investigation happen to be available, either commercially or in the public domain, this is likely to be the biggest impediment to a researcher's making use of Humdrum. While there are functions designed for capturing data from a MIDI controller (encode and record), it may often be simpler to enter ****kern** data manually once one is familiar with the encoding format—especially when reduced information suffices for the task at hand. With the advent of a reliable Optical Music Recognition system, under development at McGill University and elsewhere, a program could certainly be written to scan a monophonic score directly into a Humdrum file. But with more complex input such as found in, say, a late Romantic piano work, it will be some time before a computer can correctly parse the music into meaningful strands suitable for Humdrum spines—an inner part crossing between staves is apt to cause confusion even in music students, let alone in a segmentation program. Automated translation of a standard MIDI file into a

[3.7]

While here is not the place for a detailed examination of all the various pre-defined Humdrum formats, a couple more will serve as additional examples to ****kern**. The tablature representation for fretted instruments, ****fret**, is particularly well developed, and can accommodate non-standard and non-Western instruments. The tuning of each string and the exact position of each fret are specified in tandem interpretations, and detailed information is encoded in each data record about which strings are plucked, how they are plucked, and at which fret. The ****IPA** scheme phonetically represents vocal texts via an ASCII alphabet, and includes stress and intonation information. The “formant” command can be used to generate a spine with the two strongest formant frequencies of each vowel encoded in the ****IPA** representation.

4. Humdrum Commands

[4.1]

The “active” component of Humdrum, as compared to the “passive” syntax and representation schemes, is the kit of tools for manipulating data files that conform to the Humdrum syntax. Once the Toolkit is installed, commands are invoked via a standard UNIX terminal, and the bulk of a researcher’s work will probably rely on ordinary UNIX commands such as “grep” or “sort.” A modicum of familiarity with this operating system is required for the aspiring Humdrummer to embark on a research project, and the more at ease he or she is in a UNIX environment the sooner Humdrum will be mastered. There are roughly a dozen basic UNIX commands that a user should be able to negotiate, but no introduction to this basic knowledge within the current Humdrum documentation.^ (5)

[4.2]

A glance at the names of most Humdrum commands reveals very little about their purpose: only a small minority of them, such as “tran” (transpose), “reihe” (give variant row form), “iv” (determine interval vectors) and “freq” (return frequency in Hz), have any obvious musical significance. And, as noted in the reference manual, these tend to be the least useful commands, since they are so specialized. The nature of many of the basic commands taken individually is quite abstract. To give an idea of their scope in combination, as well as of the working environment of Humdrum, here follows a modest sample problem we might wish to answer.⁶

5. Sample Problem

[5.1]

In what proportion are leading-tones in Bach chorale melodies approached from beneath, and in what proportion from above? If we are unaided by a computer, the question will pose no special problem as long as we restrict the number of chorales we wish to consider: for some purposes a few will suffice and the appropriate notes can easily be counted by hand. But should we wish to consider a large number of chorales, our task will be considerably easier—and will provide more reliable answers—if we let the computer do the counting. Let us work through the steps involved, assuming the existence of a file containing encoded versions of several Bach chorales.⁷ First, we are interested only in the melody, which is in the soprano part, or the fourth spine of the file. For this we may use the Humdrum command “extract,” which selects certain data from a file. The full command would be

```
extract -p 4 chorales > temp1
```

where the “-p 4” indicates the fourth spine, and “temp1” is the name of the temporary file we are creating for the extracted spine.

[5.2]

The original file represents notes by their pitch. We need scale-degree information instead, in order to locate leading-tones. The command “deg” will convert a file of ****kern** data records into a file of ****deg** records, where each note appears as a number from 1 to 7, representing its scale degree. The calculation is relative to the current key, as specified in key tandem interpretation records in the file. We also require contour information; happily the ****deg** representation includes a direction indicator: the lower-case letter **v** signifies “lower than previous note” and the caret **^** signifies “higher than previous note.” The next step in our sample problem is thus:

```
deg -x temp1 > temp2
```

where the “-x” removes certain extraneous data. The new file, temp2, will begin something like this:

```
!!!COM: Bach, Johann Sebastian  
!!!XEN: J. S. Bach Chorale Harmonizations  
!!!OTL: Ermuntre dich, mein schwacher Geist  
!!!ONM: No. 9
```

[5.3]

This is quite a messy file, for it includes comments (records with leading exclamation marks), null tokens (records containing only the period punctuation sign) and bar-lines (records with leading equals signs). We could invoke the Humdrum “rid” command to get “rid” of the items superfluous to our inquiry, but it is not strictly necessary. For our purposes the only records that matter are those which contain the digit 7, for the leading-tone, and start with an up or down sign (^ or v). We can use the UNIX command “grep” (Get Regular ExPression) to comb the file under scrutiny for a target string, and output only those lines containing at least one occurrence of the string. The target-string syntax will no doubt be obscure to those unfamiliar with UNIX regular expressions; leaving that aside for now, the following command (working on the file previously generated):

```
grep ^[v^]7 temp2 > temp3
```

will yield the following file, called temp3:

```
^7
```


[5.4]

Notice that a few records with a minus sign appear in this file. These represent lowered seventh scale-degrees such as may occur in a tonicization of IV, or in the mediant area of a minor-key chorale.⁸ To ignore these records we can make use of `grep`'s `-v` option, which removes any lines containing the target string and returns only those lines without it:

```
grep -v [-] temp3 > temp4
```

[5.5]

The original file has been whittled down to a series of records that contain exclusively unlowered leading-tones along with directional information. The final step will be to compare the frequency of occurrence of ^7 to that of v7. This requires the standard UNIX compound command `sort | uniq -c | sort`, which removes all duplicate lines from a file, recording the number of occurrences of each, and returns the contents in increasing order of frequency. ^ (9)
The last step in our problem, then, is:

```
sort temp4 | uniq -c | sort
```

[5.6]

The sequence of commands given above generated a chain of intermediate, “junk” files—Humdrum and UNIX commands generally create new files, leaving those they worked on intact. Instead the whole process, once designed and tested on a shorter file, could be amalgamated in a “pipeline” of commands, each passing its result to the next through the UNIX “pipe” (typed as |):

```
extract -p 4 chorales | deg -x | grep ^[v^]7 | grep -v [-]  
sort | uniq -c | sort
```

The final output will look something like this:

```
15 ^7  
22 v7
```

indicating that, within the sample of sixteen chorales, Bach approached the leading tone from above roughly 50% more often than from below.

[5.7]

As we can gather from the above, the difficulty in implementing the machine-aided solution does not lie in understanding the functioning of the individual commands, but in devising an appropriate way to combine them. The advantages of a machine-aided solution to this problem are threefold. First, extremely large amounts of music could be processed with no extra work on the part of the researcher, and the task would remain tractable within a reasonable length of time.¹⁰ Secondly, we obtain a concrete and objective result, reproducible by other workers with access to the same data. And thirdly, if we decide afterwards that we wish to vary the question by, say, discounting leading tones approached by leap instead of step, we would not have to return to the scores from scratch, but simply add a couple of steps to our pipeline of commands.

6. The Toolkit

[6.1]

The manual divides the Toolkit into the categories of specialized and basic tools. Specialized tools work only on a particular subset of exclusive representations. Two that crop up frequently in standard analytical applications derive harmonic and melodic intervals from ****kern**-type spines; these are respectively invoked by “hint” and “mint.” Alternative representations of pitch are generated from a ****kern** spine by the commands “cents” (distance in hundredths of a semitone from a reference point), “freq” (frequency in Hz), “semit” (semitones away from a reference point), and several others, which could all be useful depending on the nature of one’s research. (I personally can’t imagine an application for the ****solfg** representation, which encodes pitch as French solfege syllables, but should anyone dream one up Humdrum is prepared.) There are several advanced commands that calculate numerical results from spines already present, such as “spect,” which assembles sonorities’ spectral content for acoustical tasks, “synco,” which gives a measure of metric syncopation for each note, and “diss,” which implements the Kameoka/Kuriyagawa measure of dissonance. The algorithms

[6.2]

The “perform” command plays back the contents of a Humdrum MIDI file, converted from a **kern file by the midi command (and further convertible to a standard MIDI file by the smf command, for exporting to a sequencer or other external application program).⁽¹¹⁾ Play-back is via an external MIDI synthesizer and control card (not included!), and allows the user to proof-listen interactively, changing the tempo, repeating sections and skipping forwards or backwards to the desired bar number. An extremely promising feature from a pedagogical point of view is that “perform” relays comments encoded alongside the music onto the screen. Thus the encoding of a sonata could include labels for the various formal sections or particular harmonic devices, which would appear on the screen at the appropriate moment during playback. Another application would be to use the “pattern” command to locate tone rows in a twelve-tone piece and tag them with a comment, letting “perform” skip straight to each occurrence of the row and provide on-screen commentary about the row-form.⁽¹²⁾

[6.3]

Basic tools, contrary to specialized tools, do their work on any representation scheme. Many of them combine or dissociate spines, files or parts of files as intermediate steps in a problem's solution; these include "assemble," "extract," "rend," "yank," and "cleave" (this last command is used in the older sense, of "join", as in "a man shall cleave unto his wife," rather than its opposite, as in "rock of ages, cleft for me"). There are also several tools useful for statistical and information-theoretical measures; these include "correl," which provides a coefficient of correlation between the numeric data in two spines, "simil," which measures the similarity between two spines based on typographical "edit distance," "infot," which describes the flow of information, and "scramble" which randomizes data records for control purposes.

[6.4]

Two very powerful basic tools deserve special mention: “patt” and “pattern,” which differ only slightly in their functioning. Each allows the user to define a pattern in a template that should match one or a sequence of data records in the file, not necessarily contiguous. Humdrum will proceed exhaustively to locate patterns in the encoded selection that match the template. Patterns can be horizontal (simultaneous), vertical (sequential within one spine), or even diagonal (useful for identifying Klangfarbenmelodie or other “threaded” patterns), and can be defined as explicitly or as “fuzzily” as is needed. The researcher can use other representations than ****kern** in conjunction with these tools—for example to discover idiomatic fingering patterns from a ****fret** representation, or particular harmonic progressions from a ****harm** representation. There is a considerable art involved in correctly defining the target template; readers familiar with UNIX regular expression syntax, crucial here in the creation of a template, will appreciate the niceties involved in fine-tuning an expression that is to be sought. The manual justly includes an tutorial in regular-expression syntax, and

[6.5]

The demonstration program bundled with the Toolkit solves ten problems on screen, explaining each step as it is implemented. To work through the problems myself I found I had to refer to the shell scripts containing the commands that actually perform the manipulations, rather than the onscreen versions, which are somewhat simplified. Half of the sample problems incorporate audio illustrations; they include locating all occurrences of “B-A-C-H” in a Brandenburg Concerto and a comparison of syncopation in George Gershwin and Stephen Foster. The reference manual gives several other examples of questions that Humdrum could be used to answer; they include: Are crescendos in Wagner more strongly associated with rising pitch than is the case with other composers? In Bartok, are dissonances more common in strong metric positions than in weak metric positions? In Urdu folk songs, how common is the so-called “melodic arch”—where phrases tend to ascend and then descend in pitch? While the demonstration software provides some help in appreciating how problems can be solved using connected Humdrum tools, and while annotated solutions to sample

[6.6]

The Toolkit and representation schemes are not intended to meet the requirements of a researcher's every contingency, nor could they be. Creating "home-made" task-specific representations is a simple job, given some discussion in the manual. The creation of new commands, on the other hand, is likely to remain the domain of those already familiar with a programming language. A skeleton program to aid in software development is provided, written in the AWK programming language and incorporating several useful functions.

7. Reference Manual

8. Humdrum Now and in the Future

[8.1]

The Toolkit can currently be downloaded from the server at the University of Waterloo. The FTP site is , and Humdrum resides in the directory called / uw-data. Although the software is free, users must license their copy after ten days by sending their vital statistics to David Huron at Waterloo; this information is used to demonstrate to granting agencies that the software is being used. It runs in UNIX, DOS, Windows95 or OS/2 environments, but the last three require the presence of some UNIX utilities—the AWK language and a shell command interpreter—before Humdrum can be installed. On UNIX machines the perform, encode and record functions are not implemented as of yet, and several other commands must be compiled locally before they can run—I had difficulty getting the simil command to compile, and eventually gave up. All instructions are given very clearly in the Installation Guide, which is included with the software in PostScript format, as is the Reference Manual. A very clear FAQ (Frequently Asked Questions) file is included with the release, too; Huron recommends that potential users read this document carefully before deciding to

[8.2]

Scores that accompany the current release of Humdrum include a selection of sixteen Bach chorales, in both ****kern** and ****harm** representations; all 48 fugues from the Well-Tempered Clavier; vocal melodies from Schubert Lieder and a collection of Scottish and English folk tunes. A ****kern** thematic database of some ten thousand well-known classical themes (melody only) is expected to be accessible via the World Wide Web early in the new year, and is being considered for release as part of the Humdrum Toolkit in the near future. Such thematic catalogues exist in books already, but the flexibility of searching that Humdrum offers far surpasses that possible in the printed medium. For instance, one could very quickly find all tunes in the database whose first four notes have increasing durations; or whose range is exactly a minor sixth; or which are in Eb minor and triple metre. This could be very useful in a music library, among other contexts. A user might wish to find the source of a tune that started with, say, three notes descending stepwise, followed by upwards motion for one note then downwards motion for one note, the last of these five notes being longer than the others.

[8.3]

Huron is nearing completion of a Users' Guide to Humdrum that "takes novice users by the hand, and teaches both Humdrum and the pertinent Unix utilities."⁽¹⁴⁾ Many sample problems will be given annotated solutions, filling a gap in Humdrum documentation as it currently stands. To master the system will still require hard work, but the frustration of learning to navigate the unfamiliar environment of UNIX will hopefully be lessened. An alternative way of learning Humdrum is through a seminar: Professor Huron taught a two-week summer course on the use of the Toolkit at Stanford this year,⁽¹⁵⁾ and at McGill the previous summer.

[8.4]

All this proselytizing may still fail to convince, given the reluctance and inertia many of us are likely to experience when faced with as new and as vast a resource as Humdrum. So, software developers have turned their attention to Humdrum-based graphic user interfaces, which may well prove more appetizing to those who feel claustrophobic when restricted to a bare UNIX shell. Two third-party graphic interfaces to Humdrum have been developed, in Ireland and in Germany.⁽¹⁶⁾ At the time of writing I have been unable to inspect them, though Andreas Kornstaedt's is apparently described in a recent publication.⁽¹⁷⁾ Also in the works is a third-party translator from Humdrum to a notation-based program, so that Humdrum files can be viewed as scores or printed in traditional notation.

[8.5]

Humdrum provides concrete answers to well-posed questions. Surprising or counterintuitive answers could be genuine, or they could be the result of a malformed question or incorrect implementation of Humdrum commands. It thus becomes important, when using machine-aided methods to prove or disprove a hypothesis, to document one's methods. Full documentation of Humdrum processes could quickly become distracting, tedious, or unfeasible in a printed article where the ends, and not the means, constitute the main point. It would however be a simple matter to place the relevant Humdrum scripts and encoded scores on an internet site, referenced by the article, and available to interested parties for retrieval.

9. Conclusions

[9.1]

Appreciation of the full potential of Humdrum is definitely a hands-on experience. In the words of David Huron, from the reference manual:

[a]ny set of tools requires the development of a concomitant expertise, and the Humdrum Toolkit is no exception. It is my hope that the investment of time required to learn how to use Humdrum will be more than offset by the subsequent scholarly gains.

Huron has certainly proved the last portion of this excerpt through his own research, much of which has relied on Humdrum. His work in music perception accounts for the inclusion in the Toolkit of some specialized tools relevant to that area—in fact Humdrum users have so far tended to work in music perception or ethnomusicology, while theorists and historical musicologists have been slow to recognize the system's potential. Huron offers the following sampling of projects he is aware of that relied on Humdrum:⁽¹⁸⁾ a study of canonic compositional rules by Zarlino, a study of Korean court music, a study of text / melody relationships in Gregorian

[9.2]

Humdrum or other systems like it offer the wherewithal for a brand new paradigm for music research. The tedium of collating hard evidence for one's theories can be greatly alleviated by automation, and the more music examined the greater the rigor of proof for one's hypotheses. However the unfortunate possibility is that many of today's musicologists and theorists who could benefit from a little machine-aided insight are likely to be put off by Humdrum's completely text-based interface. Although in the final analysis UNIX-style commands are surely more flexible and efficient than a "user-friendly," graphic interface, they can appear daunting to non-programmers, many of whom stand to be deterred from making use of an otherwise valuable tool. Whether or not music theorists will choose to augment their invaluable musical intuition with equally invaluable empirically verified results, based on maximum amounts of relevant data, will be a most important factor in the evolution of our discipline.

Jonathan Wild McGill University Faculty of Music, Dept. of Theory
555 Sherbrooke West Montreal, Quebec, H3A 1A3

Footnotes