Chapter 1

Humdrum: A Brief Tour

Humdrum is a general-purpose software system intended to assist music researchers. Humdrum's capabilities are quite broad, so it is difficult to describe concisely what it can do. This chapter provides a brief tour through Humdrum: the goal is to give you a quick glimpse of Humdrum's capabilities and to help sketch some of the big picture. Don't worry if you don't understand everything in this chapter. All of the topics mentioned here will be covered more thoroughly in later chapters.

What Can Humdrum Do?

Although Humdrum facilitates exploratory investigations, it is best used when the user has a clear problem or question in mind. For example, Humdrum allows users to pose and answer questions such as the following:

- In the music of Stravinsky, 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?
- What are the most common fret-board patterns in guitar riffs by Jimi Hendrix?
- Which of the Brandenburg Concertos contains the B-A-C-H motif?
- Which of two different English translations of Schubert lyrics best preserves the vowel coloration of the original German?
- Did George Gershwin tend to use more syncopation in his later works?
- After the V-I progression, which harmonic progression is most apt to employ a suspension?
- How do chord voicings in barbershop quartets differ from chord voicings in other repertories?
- In what harmonic contexts does Händel double the leading-tone?

Although the Humdrum tools may take just seconds to compute an answer for any of the above problems, only a few of these questions are *easy* to answer using Humdrum. The primary impediment to a quick solution is the user's skill in interconnecting the right tools for the task at hand. The purpose of this book is to help you develop that skill.

The Humdrum Syntax and the Humdrum Toolkit

Humdrum consists of two distinct components: the *Humdrum Syntax* and the *Humdrum Toolkit.* The *Humdrum Syntax* is a grammar for representing information. Over the course of this book we'll encounter a variety of representation schemes that conform to this syntax. The syntax itself doesn't represent anything: it merely provides a common framework for representing all sorts of information. Within the syntax an endless number of representation schemes can be defined. Theoretically, any type of sequential symbolic data may be accommodated — such as medieval square notation, MIDI data, acoustic spectra, piano fingerings, changes of emotional states, Indian tabla notations, dance steps, Schenkerian graphs, concert programs — or even industrial chemical processes.

A number of representation schemes are pre-defined in Humdrum; however, users are free to concoct their own task-specific representations as necessary. For example, if you are interested in Telugu notation or Dagomba dance, you can design your own pertinent representation scheme within the Humdrum syntax. Humdrum representations may be meticulously crafted, or they may be invented in a matter of seconds. It is common to generate intermediate or "throw-away" representations that are used only for a single research task.

The *Humdrum Toolkit* is a set of more than 70 inter-related software tools. These tools manipulate ASCII data (text) conforming to the Humdrum syntax. If the ASCII data represents music-related information, then we can say that the Humdrum tools manipulate music-related information.

What kinds of manipulations can be done with Humdrum? The various Humdrum tools can be grouped roughly into the following sixteen types of operations.

1. **Visual display.** E.g., display a score beginning at measure 128; output the libretto from Act II, Scene 5; print the string parts for the Coda — including the Roman numeral harmonic analysis.
2. **Dural display.** E.g., play the bass trombone part slowly beginning at measure 70; play just the opening two measures from all of the works in a given repertory.
3. **Searching.** E.g., search for instances of a motive; locate any deceptive cadences; find all of the works that are composed for a given combination of traditional Japanese instruments.
4. **Counting.** E.g., how often do augmented intervals occur in Hungarian folk songs? What proportion of phrases do not begin with a pick-up or anacrusis?
5. **Editing.** E.g., change all up-stems to down-stems in measure 88 of the second horn part.
6. **Editorializing.** E.g., add an editorial footnote to a specified note; indicate that a passage differs from the composer's autograph.
7. **Transforming or translating between representations.** E.g., transpose from one key to another; calculate the harmonic intervals between two parts; represent a score according to scale-degrees; reconstitute chords so they are represented in set *normal form.*

8. **Arithmetic transformations of representations.** E.g., calculate the semitone spacings between successive notes, or determine points where parts cross in pitch.
9. **Extracting or selecting information.** E.g., extract the second verse; exclude the development section; isolate the third phrase; grab the second chord in each measure; select the brass parts; take the second endings when repeating the trio; choose the Dresden manuscript version.
10. **Linking or joining information.** E.g., assemble instrumental parts into a full score; tag notes with their harmonic function; coordinate heart-rate data from a listener with the musical score.
11. **Generating inventories.** E.g., list all the types of embellishment (non-chordal) tones from the most common to the least common; what chord functions are absent from a work?
12. **Classifying.** E.g., classify all chords as "open" or "closed" position; identify all secondary dominants; classify all intervals as either unisons, steps or leaps; classify various piano fingerings as either easy, moderate, difficult, or impossible.
13. **Labelling.** E.g., mark musical sections; label themes; identify French, Italian and German sixth chords; mark appropriate words in a vocal text as either "passionate," "apathetic," or "neutral." Mark sonorities as falling on either strong or weak metric positions.
14. **Comparison.** E.g., determine whether the Amsterdam and Manchester manuscripts for a work have identical pitches in the third movement; determine whether motets by John Dunstable are more similar to motets by Thomas Morley or by Lionel Power.
15. **Capturing Data.** E.g., import live or recorded MIDI data; import data from a notation program.
16. **Trouble-shooting.** E.g., identify any transgressions of notational conventions; check whether a score has been tampered with; get help when you're stuck.

If the above functions sound vague, that's because the corresponding Humdrum tools are similarly broad in their function. For example, the Humdrum **patt** tool can be used to locate Landini cadences, label statements of 12-tone rows, or search for piano fingering patterns in Liszt. All three tasks involve searching for a sequential pattern that might have concurrent features as well.

When using computers in research activities, it is important to maintain a keen awareness of their limitations. For example, computers are typically not good at *interpreting* data. Because much music scholarship hinges on interpretations, this would seem to preclude computers from being of much use. However, there are some things that computers do well, and when a skilled user intervenes to make a few crucial interpretations, the resulting computer/human interaction can lead to pretty sophisticated results — as we will see.

Although computers are typically not good at interpreting data, interpretive software will become increasingly important as computational musicology continues to develop. Humdrum provides a handful of tools that *interpret* data in various ways. For example, the **key** command implements the Krumhansl and Kessler method for estimating the key of a music passage. The **melac** command implements Joseph Thomassen's model of melodic accent. Other interpretive tools characterize syncopation or implement Johnson-Laird's model of rhythmic prototypes.

The names of some of the Humdrum tools will be readily recognizable by musicians. Humdrum tools such as **key, pitch, record, tacet, trans** and **reihe** may evoke fairly accurate ideas about what they do. Ironically, the most recognizable tools are typically the least useful tools in the toolkit — because they are so specialized. The most powerful Humdrum tools have names such as **cleave, humsed, simil, recode, context, patt** and **yank.**

By itself, each individual tool in the Humdrum Toolkit is quite modest in its effect. However, the tools are not intended to be self-sufficient. They are designed to work in conjunction with each other, and with existing standard UNIX commands. Like musical instruments, their potential usefulness is greatly increased when they are combined with other tools. Musical problems are typically addressed by linking together successive Humdrum (and UNIX) commands to form one or more command *pipelines*. Although each individual tool may have only a modest effect, the tools' combined capacity for solving complex problems is legion.

Now that we've sketched an overview of Humdrum, we can consider in greater detail the two principal components of Humdrum: the *Humdrum Syntax* and the *Humdrum Toolkit.*

Humdrum Syntax

Humdrum data are organized somewhat like the tables of a spread-sheet. As with a spread-sheet, you can define or label your own types of data. As with a spread-sheet, the different *columns* can be set up to represent whatever type of data you like. The *rows* of data, however, have a fixed meaning — they represent successive moments in time; that is, time passes as you move down the page.

By way of example, consider Table 1.1. This table shows something akin to a piano-roll. The diatonic pitches are labelled in columns from C4 (middle C) to C5. Each row represents the passage of a sixteenth duration. The table entries indicate whether the note is *on* or *off.* The table encodes an ascending scale. As the table stands, it's not clear whether each note has a duration of an eighth note or whether two successive sixteenth notes are sounded for each pitch:

Table 1.1

		C4	**D4**	**E4**	**F4**	**G4**	**A4**	**B4**	**C5**
	1st 16th	ON	off	off	off	off	off	off	off
	2nd 16th	ON	off	off	off	off	off	off	off
	3rd 16th	off	ON	off	off	off	off	off	off
	4th 16th	off	ON	off	off	off	off	off	off
time	5th 16th	off	off	ON	off	off	off	off	off
	6th 16th	off	off	ON	off	off	off	off	off
↓	7th 16th	off	off	off	ON	off	off	off	off
	8th 16th	off	off	off	ON	off	off	off	off
	9th 16th	off	off	off	off	ON	off	off	off
	10th 16th	off	off	off	off	ON	off	off	off
	11th 16th	off	off	off	off	off	ON	off	off
	etc.								

Table 1.2 shows another example where different kinds of information are combined in the same table. Here the last column represents a combination of trumpet valves:

Table 1.2

	Pitch	**Duration**	**Valve Combination**
1st note	C4	quarter	0
2nd note	B3	eighth	2
3rd note	G4	eighth	0
4th note	F4	eighth	1
5th note	G4	eighth	0
6th note	A4	quarter	1-2
7th note	G4	eighth	0
8th note	Ab4	quarter	2-3

Humdrum representations can be very similar to the data shown in Tables 1.1 and 1.2. With just a few formatting changes, either table can be transformed so that it conforms to the Humdrum syntax. For example, Table 1.3 recasts Table 1.2 so that it conforms to the Humdrum syntax. Just four changes have been made: (1) the left-most column has been given a heading name so that *every* column has a label, (2) each column heading is preceded by two asterisks, (3) the columns have been left-justified so successive columns are separated by a tab, and (4) each column has been terminated with the combination of an asterisk and hyphen.

Table 1.3 A Humdrum Equivalent to Table 1.2

```
**Note      **Pitch     **Duration    **Valve Combination
1st note    C4          quarter       0
2nd note    B3          eighth        2
3rd note    G4          eighth        0
4th note    F4          eighth        1
5th note    G4          eighth        0
6th note    A4          quarter       1-2
7th note    G4          eighth        0
8th note    Ab4         quarter       2-3
*-          *-          *-            *-
```

It does not matter what characters appear in the table — numbers, letters, symbols, etc. (although there are some restrictions concerning the use of spaces and tabs). The table can have as many columns as you like, and can be as long as you like. Unlike spreadsheet columns, Humdrum "columns" can exhibit complicated "paths" through the document; columns can join together, split apart, exchange positions, stop in mid-table, or be introduced in mid-table. Humdrum also allows subsidiary column headings that can clarify the state of the data. Subsidiary headings can also be added anywhere in mid-table. Finally, Humdrum also provides ways of adding running commentaries; comments might pertain to the whole table, to a given row or column, to a given data cell, or to a particular item of information within a cell (such as a single letter or digit). Since the *columns* in Humdrum data can roam about the table in a semi-flexible way, they are referred to as **spines.** We'll see how these devices are used in later chapters.

The most common Humdrum files encode musical notes in the various cells of the table; the most common use of a spine is to represent a single musical part or instrument.

Some twenty or more representation schemes are pre-defined in Humdrum, but remember that users are always free to concoct their own representations as necessary. As in a spread-sheet, the spines or columns can be used to represent whatever you like. In the following chapter, we'll look at the most commonly used of the pre-defined Humdrum representations — the *kern* representation. This representation gets its name from the German word for *core*; it is a scheme intended to represent the basic or core musical information of notes, durations, rests, barlines, and so on.

Humdrum Tools

The Humdrum software is not a program that you invoke like a word-processor or notation editor. Humdrum is not a big program that you start-up when you want to do music research. Instead, Humdrum provides a toolbox of *utilities* — most of which can be accessed at any time from anywhere in the system.

Any data that conforms to the Humdrum syntax can be manipulated using the Humdrum software tools. Since the tools can be interconnected with each other (and can also be interconnected with non-Humdrum tools) there are a lot of ways to manipulate Humdrum data. Much of this book will deal with how the tools can be interconnected to do musically useful things.

For the remainder of this chapter, we will describe a few Humdrum tools and illustrate how they might be used. Once again, the goal in this chapter is to give you an initial taste of Humdrum. Don't worry if you don't understand everything at this point.

Some Sample Commands

One group of tools is used to extract or select sections of data. Vertical spines of data can be extracted from a Humdrum file using the **extract** command. For example, if a file encodes four musical parts, then the **extract** command might be used to isolate one or more given parts. The command

```
extract -f 1 filename
```

will extract the first or left-most column or spine of data. Often it is useful to extract material according to the encoded content without regard to the position of the spine. For example, the following command will extract all spines containing a label indicating the tenor part(s).

```
extract -i '*Itenor' filename
```

Instruments can be labelled by "instrument class" and so can be extracted accordingly. The following command extracts all of the woodwind parts:

```
extract -i '*ICww' filename
```

Any vocal text can be similarly extracted:

```
extract -i '**text' filename
```

Or if the text is available in more than one language, a specific language may be isolated:

```
extract -i '*LDeutsch' filename
```

Segments or passages of music can be extracted using the **yank** command. Segments can be defined by sections, phrases, measures, or other any user-specified marker. For example, the following command extracts the section labelled "Trio" from a minuet & trio:

```
yank -s Trio -r 1 filename
```

Or select the material in measures 114 to 183:

```
yank -n = -r 114-183 filename
```

Or select the second-last phrase in the work:

```
yank -o { -e } -r '$-1' filename
```

Don't worry about the complex syntax for these commands; the command formats will be discussed fully in the ensuing chapters. For now, it is important only that you get a feel for some of the types of operations that Humdrum users might perform.

Two or more commands can be connected into a *pipeline.* The following command will let us determine whether there are any notes in the bassoon part:

```
extract -i '*Ifagot' filename | census -k
```

The following pipeline connects together four commands: it will play (using MIDI) the first and last measures from a section marked "Coda" at half the notated tempo from a file named `Cui`:

```
yank -s '*>Coda' Cui | yank -o ^= -r 1,$ | midi | perform -t .5
```

Some tools translate from one representation to another. For example, the **mint** command generates melodic interval information. The following command locates all tritones — including compound (octave) equivalents:

```
mint -c filename | egrep -n '((d5)|(A4))'
```

Incidentally, Humdrum data can be processed by many common commands that are not part of the Humdrum Toolkit. The **egrep** command in the above pipeline is a common computer utility and is not part of the Humdrum Toolkit.

Depending on the type of translation, the resulting data can be searched for different things. The following command identifies French sixth chords:

```
solfa file | extract -i '**solfa' | ditto | grep '6-.*4+' | grep 2
```

Locate all sonorities in the music of Machaut where the seventh scale degree has been doubled:

```
deg -t machaut* | grep -n '7[^-+].*7'
```

Count the number of phrases that end on the subdominant pitch:

```
deg filename | egrep -c '(}.*4)|(4.*})'
```

The following command identifies all scores whose instrumentation includes a tuba but not a trumpet:

```
grep -sl '!!!AIN.*tuba' * | grep -v 'tromp'
```

Some tasks may require more than one command line. For example, the following three-line script locates any parallel fifths between the bass and alto voices of any input file:

```
echo P5 > P5
echo '=      *' >> P5; echo P5 >> P5
extract -i '*Ibass,*Ialto' file | hint -c | pattern -s = P5
```

More complicated scripts can be written to carry out more sophisticated musical processes. In later chapters we'll encounter some scripts that contain 10 or more lines of commands.

Reprise

In this chapter we have seen that Humdrum consists of two parts: (1) a representation *syntax* that is similar to tables in a spread-sheet, and (2) a set of utilities or *tools* that manipulate Humdrum data in various ways. The tools carry out operations such as displaying, performing, searching, counting, editing, transforming, extracting, linking, classifying, labelling and comparing. The tools can be linked together to carry out a wide variety of tasks. Even one-line commands can carry out sophisticated operations. A few lines of Humdrum can also be used to write programs of some complexity. Users can write their own programs using the Humdrum tools, or they can add new tools that augment the functioning of Humdrum.