Chapter 38

Systematic Musicology

Much of music research centers on the task of describing things. A researcher might offer a description of "House" style, or Wagner's orchestration, commonalities in themes by Respighi, or a Balinese variation technique. Good descriptions are based on the identification of *features*. A "feature" is a notable or characteristic part of something — something that helps to distinguish one thing from another thing (or one group of things from another group of things).

Not all truthful descriptions qualify as distinctive features. Imagine that you were robbed by someone and were later asked by the police to provide a description of your assailant. Suppose you began your description by saying that the robber had a nose, a mouth, and two eyes. These facts would undoubtedly be true, but the police would be rightly dismayed by your description for the simple reason that the facts fail to distinguish your assilant from billions of potential suspects.

When characterizing a musical work, genre, style, or composer, it is important, not simply to make truthful observations. It is also important to identify those characteristics that distinguish the work, genre, style, or composer from others. If the goal of an analytic description is to convey what is unique or characteristic of a given object or class of objects, then good features must embody or define some of that distinctiveness. That is, the research must guard against describing something that is commonplace.

In this chapter, we will describe several methods to help researchers determine whether a presumed feature is distinctive. In essence, we will ask the following question: "How likely is it that this is a feature of some larger class of objects — like the class of all musical works?" If the feature is commonly found in many situations, then we are not justified in claiming that the feature is unique to a particular situation.

In order to determine whether a proposed feature is distinctive of a work, we need to compare the incidence of occurrence with a body of works where we wouldn't expect the feature to be so common. In systematic musicology, we say we are looking for a comparison or *control* sample. There are four basic approaches to establishing a control sample:

- comparison repertory
- randomizing

- counter-balancing
- autophase procedure

Comparison Repertory

By a "comparison repertory" we mean a group of works, that we hypothesize do not show the same distinctive feature — although in other respects the works are similar.

Suppose a scholar has observed a particular feature that appears to occur frequently in the music of Respighi. Is this pattern a characteristic feature of Respighi's music? We cannot know this by looking only at the music of Respighi. We must choose a comparison repertory and show that the same feature is not common in the comparison group.

Ideally, the researcher should choose a comparison repertory that is as similar as possible to the target repertory. For example, if we found that Respighi's music contrasted with Rameau's music, we would not be able to dismiss the possibility that the observed differences arise because of differences in nationality, or differences in stylistic period, or differences in instrumentation, etc. Choosing a similar composer would be a better test of the proposed feature.

Of course it is almost never possible to select a perfectly matched comparison group.

When selecting a comparison repertory, the researcher should make a list of possible confounds. For example, the feature might be

In more careful studies, the researcher might attempt to match the repertories for a series of attributes. For example, one might match the number of works in both repertories that are in triple meters. Similarly, one might match the modality so both the target and comparison repertories have the same proportion of major/minor keys. One might similarly match instrumentation, date or period of composition, duration of work, tempi, and so on.

Humdrum users can use the **find** and **grep** commands to identify works that match particular controlled characteristics (see Chapter 33). For example, the following command might locate all files that contain scores in triple meter. The results are placed in a file called `control`:

```
find /scores -type f -exec grep -l '!!!AMT.*triple' "{}" ";" \
    > control
```

In some cases, the number of possible control works is excessively large. In this case one can make a random selection from the control list using the **scramble** command described later.

Note that in many instances it is difficult to establish a good comparison repertory. For example, suppose we find significant differences between Beethoven melodies and Kanartic folk melodies. How should we interpret these differences? The differences might be symptomatic of the difference between folk music and classical music. Or the differences might reflect differences between German and Indian music. Or the differences might reflect differences between early 19th century music and mid-twenthieth century music. Or the differences might be attributable to differences of instrumentation.

No matter what comparison repertory we choose, someone might be able to claim that any observed differences arise due to some other factor. For example, we might compare early and late works by Bach as a way of tracing his musical development. However, someone might claim that any differences found are not related to Bach's development as a composer, but are due to different tastes in Weimar versus Leipzig. Bach was simply showing his ability to adapt to local tastes.

Randomizing

Sometimes scholars formulate hypotheses that are intended to pertain to the whole of music — that is, the feature is thought to be a musical universal. In these cases it may be impossible to identify a comparison repertory.

Many theorists have noticed, for example, that melodies tend to be composed predominantly of small intervals. That is, there appears to be a preference for small melodic intervals in most forms of music. Unfortunately, the predominance of small intervals might simply be an artifact of a small range. Consider the case of a melody that is restricted to the range of an octave. Between any two pitches, there is only a single instance of a 12-semitone interval, however, there are 12 possible 1-semitone intervals. Even in a random ordering of notes, one would expect to see a preponderance of small intervals.

So is there any way of testing whether composers really do favor small melodic intervals? Or is this simply an artifact of range restrictions? Many such hypotheses can be tested using a randomizing procedure. We can illustrate this procedure by working through the problem of small melodic intervals.

We begin by measuring the average melodic interval size in semitones for a sample of actual melodies. We can use the **semits** command to translate data to semitone representations and then use the **xdelta** command to calculate numerical differences. The **-a** option for **xdelta** causes only absolute (unsigned) values to be calculated. The **rid** command can be used to eliminate everything but data records and the **grep** command can be used to eliminate barlines and rests. We can then calculate the average interval size by piping the output to the **stats** command. For typical folk melodies, the average interval size is roughly two semitones.

```
semits melody | xdelta -a | rid -GLId | grep -v '[=r]' | stats
```

Next, we need to determine the average melodic interval size that would result for a random re-ordering of the pitches within each melody. We can do this using the Humdrum **scramble** command.

Using the *scramble* Command

The **scramble** command is useful for randomizing the arrangement of Humdrum data. Suppose we had the following Humdrum input:

```
**numbers
1
2
3
4
5
*-
```

We can scramble the order of data records using the following command:

```
scramble -r numbers
```

The **-r** option indicates that it is the order of records which should be randomized. A possible output might look like this:

```
**numbers
3
2
5
1
4
*-
```

Notice that only data records are scrambled: comments and interpretations stay put. Each time **scramble** is invoked, it produces a different random ordering.

Returning to our melodic interval problem, we can now generate an inventory of melodic intervals for our original repertory, where the order of the notes has been randomly ordered:

```
scramble -r melody | semits | xdelta -a | rid -GLId \
    | grep -v '[=r]' | stats
```

For a typical folksong repertory, the average melodic interval size for a randomly re-ordered melody is roughly 3 semitones in size. Using common statistical tests, it is possible to prove that this difference is unlikely to occur by chance and that it likely is a symptom of real efforts to organize melodies using relatively small melodic intervals.

A similar approach can be used to address innumerable questions. For example, in Haydn's music, it seems that Haydn tends to avoid following the dominant by a subdominant chord (i.e., *V-IV*). On the other hand, Haydn's use of the *IV* chord is comparatively infrequent, so the apparent absence of this progression may simply be an artifact of the relative scarcity of subdominant chords. We can address this question by comparing Haydn's actual harmonic progressions with randomly generated progressions. First we count the total number of *V-IV* progressions:

```
extract -i '**harm' haydn  | context -n 2 -o ^= \
    | grep -c '^V IV$'
```

Next we randomly re-order his harmonies and count the number of *V-IV* progressions:

```
scramble -r haydn | extract -i '**harm' | context -n 2 -o ^= \
    | grep -c '^V IV$'
```

In some cases, problems can be addressed by randomizing one part of voice with respect to another. For example, there is strong evidence that Bach uses more augmented eleventh harmonic intervals than would occur by chance. That is, the tritone is "sought-out" rather than "avoided" in his writing. Suppose we are looking at a two-part invention. We begin by counting the number of augmented elevenths in his actual writing:

```
ditto -s = bach | hint | grep -c 'A11'
```

We can create a random comparison by extracting one of the parts, scrambling the order of notes, and then re-assembling the scrambled part with the original. The resulting harmonic intervals arise from a random juxtaposition of parts.

```
extract -f 1 bach > temp1
extract -f 2 bach > temp2
scramble -r temp1 > temp1.scr
assemble temp1.scr temp2 | ditto -s = | hint | grep -c 'A11'
```

Note that the **scramble** command also provides a **-t** option so that the order of tokens within a data record can be randomly re-arranged.

Retrograde Controls Using the *tac* Command

Suppose a theorist found an unusually large number of occurrences of the B-A-C-H pitch pattern in some repertory. Are these patterns intentional on the part of the composer? Or should we expect a fair number of such patterns to occur simply by chance?

One way of determining the chance frequency of B-A-C-H might be to randomize the order of pitches using the **scramble** command, and then use **patt** to count the number of occurrences in the reordered melodies. Unfortunately, we already know that musical lines tend to be constructed using small intervals, and the pitches B-A-C-H are very close together. Since random reordering of the pitches will reduce the proportion of small intervals, we would naturally expect fewer instances of B-A-C-H in the random musical lines. We need some way to maintain the identical interval distribution in our control repertory.

One way to do this is by *reversing* the sequencial order of the notes. If we could rearrange the notes so that the first note was last and the last note was first, then our control repertory would preserve both the frequency of occurrence of all the pitches, and also preserve the pitch-proximity distribution.

The UNIX **tac** command can be used to reverse the order of records. Suppose we had an input consisting of the number 1 through 10 on successive lines. The **tac** command would transform this input so that the output consists of the reverse ordering of numbers from 10 to 1.

If we apply **tac** to a Humdrum file, then the result will no longer conform to the Humdrum syntax — the spine-path terminators will appear at the beginning of the file and the exclusive interpretations will appear at the end of the file. If we use **tac** we could simply restore the correct syntax by hand-editing the file and moving the exclusive interpretations and the spine-path terminators to their proper locations. We now have a "retrograde" passage.

Such a retrograde passage will provide a useful control repertory to test our B-A-C-H hypothesis. If a composer is intentionally composing several instances of B-A-C-H into his/her music, then we would expect the number of occurrences to be somewhat more frequent than instances of B-A-C-H found in retrograde versions of the works.

Another way of testing the same hypothesis would be to search for the reverse pitch sequence: H-C-A-B.

Autophase Procedure

Frequently researchers are interested in the relationship between concurrent musical parts or voices. Suppose, for example, that we had reason to suspect that a particular polyphonic composer tends to actively avoid octave intervals between the bass and soprano voices. If we find that the proportion of octave intervals is 6 percent, how do we know whether this is a lot or a little?

One approach to answering this question is to use an *autophase procedure* (Huron, 1991a). The essence of this approach is to shift two spines with respect to each other.

Recall that the **reihe** command (Chapter 35) provides a **-s** option that causes a shift in the serial position of data tokens. For example, suppose we had an input consisting of the numbers 1 through 5. The following command:

```
reihe -s +1 file
```

Will cause all data tokens to be moved forward one position, and the last data token to be moved to the beginning:

```
**numbers
5
1
2
3
4
*-
```

Let's apply this technique to our problem of whether a given composer tends to avoid octaves between the soprano and bass voices. First, we extract each of the voices. Let's also eliminate barlines and use **ditto** to replicate the pitch values through null tokens.

```
extract -i '*sopran' composition | grep -v = | ditto > voice1
extract -i '*bass' composition | grep -v = | ditto > voice2
```

Now let's shift one part with respect to the other using **reihe -s**.

```
reihe -s voice1 > voice1.shifted
```

Now we reassemble the parts, determine the harmonic intervals present, and count the number of octave intervals:

```
assemble voice2 voice1.shifted | hint | grep -c 'P8'
```

In effect, we have concocted a control group, by shifting the parts with respect to each other. Of course we have utterly destroyed the *relationship between the two parts.* However, many things remain untouched. The bass voice remains identical, and the soprano voice is identical except that there is an extra melodic interval (between the first and last notes) and one melodic interval missing. In short, we have preserved the within-voice organization while destroying the between-voice organization.

Rather than using a single shifted control, it is typically better to repeat the procedure, methodically shifting the spines through a complete 360 degree rotation. We can then compare measures for the actual work with a distribution of measures for all of the shifted values.

The following script calculates the number of P8 intervals for each of the possible shifts between the first and second spines in the file `composition`. The number of data records is determined and assigned to the `LENGTH` variable. A `while` loop is used to calculate the number of octave intervals for each of the possible shifts between the parts:

```
extract -f 1 composition | grep -v = | ditto > spine1
extract -f 2 composition | grep -v = | ditto > spine2
LENGTH=`rid -GLId spine1 | wc -l | sed 's/ //g'`
X=1
while [ $X -ne $LENGTH ]
do
    reihe -s $X spine1 > temp
    assemble spine2 temp | hint | grep -c 'P8'
done
rm spine[12] temp
```

This *autophase* procedure has been used to address many differ kinds of questions pertaining to how musical parts interrelate.

Reprise

When using computers to measure or observe something it is important not to jump to conclusions from what we find. Just because something is either prevalent or rare does not mean that it is significant: it might be prevalent or rare simply by chance (e.g., von Hippel & Huron, MS). In this chapter we have illustrated a number of methods for testing hypotheses by contrasting a target repertory with various controlled data.

We have discussed four different control methods: comparison repertory, randomizing, retrograde, and the autophase procedure. In addition, one might use musical inversion as a control method. All of these methods allow us to compare how music is actually organized with how it might be organized. In certain cases, such contrasts allow us to infer aspects of musical organization that would otherwise be difficult or impossible to decipher.