CMAP: Contemporary Music Analysis Package

by Craig R. Harris and Alexander Brinkman

Reviewed by John Wm. Schaffer

I. INTRODUCTION

The past several decades have witnessed not only a vast increase in the utilization of set-theoretical techniques, but also an exciting proliferation of analytical tools suited to the serial and free atonal repertoire. While these techniques have contributed greatly to our understanding of this literature, they have also considerably complicated the task of the music analyst. Even though the specific mathematical tasks involved in wielding the theories are not always that difficult in themselves, the sheer number of computations required to complete an analysis of even modest proportions can be so time-consuming and error-prone that the analysis often becomes overwhelmingly difficult and, not infrequently, inaccurate. In addition, the meaningful task of making significant analytical observations is all too often eclipsed by the mathematics themselves.

With the large influx of computer usage over the last half decade, many theorists bitten with the computer bug have attempted to program software tools designed to aid them in their analytical quest. Without negating the occasional success, few have shown the fortitude necessary to pursue the task through to its logical conclusion. Although the world has seen many twelve-tone matrix generators, few programs can claim any degree of comprehensiveness. Fortunately, CMAP responds by providing a relatively complete collection of analytical tools capable of handling nearly all the needs of the contemporary music analyst.

II. OVERVIEW

Harris and Brinkman have drawn on research originally conceived as part of Harris' dissertation work, expanding and transforming it into an admirable package for the analysis and modeling of ordered and unordered pitch materials. Their aim is to present a set of analytical tools capable of being flexibly and creatively inter-linked by the user, allowing both specific tailoring of information retrieval, as well as increased explorative flexibility. The authors stress, however, that the programs are intended primarily as tools to assist the analyst, they are not meant to supply interpretive higher-level analytical assertions.

The key design strategy behind CMAP is power through flexibility. The tools, consisting of twenty-four individual programs originally implemented in the UNIX environment[1] and later ported to run on MS-DOS[2] machines, are designed so that the output of one program can be selectively masked, redirected, or "piped" as

[1] UNIX is a registered trademark of Bell Laboratories.

[2] MS-DOS is a registered trademark of Microsoft Corporation. The authors state that the programs will run on any IBM PC/XT/AT or clone running PC/MS DOS 3.0 or higher. All the testing for this review was done on either an IBM PC-AT operating under PC-DOS 3.2, or a Zenith PC-XT clone running MS-DOS 3.01. No problems were encountered on either machine.

input to a printer, file, or other programs. All the programs are accessed through the command-line structure inherent to both the UNIX and MS-DOS environments. In other words, a program name is typed at the operating system prompt followed by one or more "arguments" specifying the set(s) or set class(es) upon which the program is to operate, along with any operation "qualifiers" desired by the user. Unfortunately, the command structures in CMAP are not always intuitive and, in fact, are sometimes even contradictory (e.g., the use of "-t" to mnemonically specify transposition, but also to indicate other non-related operations in other programs). Thus, the inherent power of the programs comes at the expense of intuitive learning ease. Initial work with the programs can be laborious until the various command syntaxes and options are committed to memory.

All pitch information is entered either as numeric pitch-class representations (i.e., 0123456789ab) or as set names as defined by Allen Forte[3] or John Rahn.[4] For example, figure 1 shows a command line sequence to obtain the default information on three different sets, two given as pitch-class collections (0437845 and 9743) and one by its prime form name (5-3).

Figure 1

```
C> GETSET 0437845 9743 5-3
```

Due to differences in the manner of computing prime forms, seven primes work out differently depending on whether they

[3]Allen Forte, *The Structure of Atonal Music* (New Haven: Yale University Press, 1973).

[4]John Rahn, *Basic Atonal Theory* (New York: Longman, 1980).

are derived using Forte's or Rahn's method (figure 2). CMAP provides a method for pre-specifying which method of computation is to be used throughout the programs, thus satisfying both camps.[5]

Figure 2

Set Class	Forte	Rahn
5-20	01378	01568
6-29	013689	023679
6-31	013589	014579
7-18	0123589	0145679
7-20	0124789	0125679
8-26	0124579a	0134578a

Even the best theorists are occasionally prone to error. In CMAP if a command line is entered incorrectly the user is prompted with a syntax diagram and brief description of the particular program in question. This is a rather nice feature since it enables one to work without the manual constantly at hand. If a program syntax is ever forgotten, typing a Usage Statement (the command without any qualifiers) will cause the same information to be displayed on the screen. A bit more awkward is the manner in which the program handles qualified searches. If no set classes satisfy a particular search specification, only the program headings print out without any accompanying message relevant to the failed search. To

[5]The default computational method used with CMAP is that defined by Allen Forte. Changing the default setting is accomplished by setting the system variable PRIMETYPE. In MS-DOS this can be accomplished either from within a batch file or by entering a command-line sequence at the DOS prompt.

compound the problem, this practice is only explained within the body of certain command summaries. Certainly a more user-friendly approach could have been implemented, or at least the authors could have done more to apprise the user of the practices being used.

In addition to the twenty-four pre-compiled programs, the authors have included the complete CMAP library of C-language[6] functions used to generate the programs, thus giving an avid C-language programmer the power to create his own personalized programs to amend or extend the current package, or simply to create something completely new.

The extensive manual which accompanies CMAP is divided into four sections: 1) a user's reference containing a description of each program's operation and syntax, 2) a programmer's section containing a C-language reference to the CMAP function library, 3) a technical reference describing the data structures used in the programs,[7] and 4) a tutorial introduction presenting a less-thorough, yet command-comprehensive "walk-through" of each of the program tools, as well as a glossary of relevant theoretical terms.

In an effort to create a comprehensive library of analytical tools it was necessary for the authors to draw upon numerous sources. Harris and Brinkman cite the following theorists as primary

[6]The MS-DOS versions of the programs were written using Microsoft C, version 4.0.

[7]Of particular interest is the binary representational scheme devised by Brinkman in which all prime forms are represented by a twelve-digit binary number. If a particular pitch class is present then its corresponding digit is equal to one, otherwise it remains at zero. Collectively, all the pitch classes forming a particular set can be represented by one number. For example, the set 014 (3-3) is represented in binary form as 000000010011, or as decimal 19 (1+2+16).

sources of information related to CMAP: Allen Forte and John Rahn (general theoretical information including prime forms, set-class names, normal orders, interval vectors, and set relationships),[8] Daniel Starr (the bitwise prime-form algorithm utilized throughout the programs),[9] Richard Chrisman (adjacent interval arrays),[10] Hubert Howe (multiplicative operators),[11] Robert Morris (invariance vectors, multiplicative operators),[12] Bo Alphonce (invariance matrices),[13] and John Rogers (rotational arrays).[14]

At this point it will be helpful to examine briefly each of the twenty-four programs. The order of presentation will follow that of the tutorial.

[8]Forte and Rahn.

[9]Daniel Starr, "Sets, Invariance and Partitions," *Journal of Music Theory* 22 (1978):1-42.

[10]Richard Chrisman, "The Identification and Correlation of Pitch-Sets," *Journal of Music Theory* 15 (1971):58-83.

[11]Hubert Howe, "Some Combinatorial Properties of Pitch Structure," *Perspectives of New Music* 4/1 (1965):45-61.

[12]Robert Morris, *Composition with Pitch-Classes: A Theory of Compositional Design* (New Haven: Yale University Press, 1987).

[13]Bo Alphonce, "The Invariance Matrix" (Ph.D. Dissertation, Yale University, 1974).

[14]John Rogers, "Some Properties of Non-duplicating Rotational Arrays," *Perspectives of New Music* 7 (1968):80-102.

Tools for Set Identification

GETSET: represents one of two general purpose programs designed primarily for identifying pitch-class sets and presenting pertinent data associated with them. The program responds to a search by listing the collection's set-class name (Forte or Rahn), Z-related set (if any), multiplicatively related set, interval-class vector, invariance vector, prime form, and cyclic interval pattern. Additional parameters can be appended to limit the output choices (see FILTERS below), and to investigate the content of the interval vector and the invariance vector.

PRIME: is similar to GETSET except that it continuously recycles, allowing for repeated queries without re-running the program. It can also function as a minimal program shell capable of accessing all the other programs from within its loop. Since all twenty-four programs reside on disk and must be loaded and run with every query, disk access is intense, particularly for floppy disk users. The ability of PRIME to at least partially circumvent this problem will probably make this tool the program of choice for most users. PRIME presents the user with a slightly different set of information than GETSET. Included are the normal order, twelve-tone operators (TTO)[15] relating the normal order to the prime, set-class name, prime form, M-related set class, Z-related set class (if any), interval vector, a two-digit octal (base eight) number representing the interval vector,[16] invariance vector, and cyclic interval pattern.

[15]The primary TTO operators used throughout CMAP are Tn(P), Tn(I(P)), Tn(M(P)), and Tn(MI(P)) where 'n' ranges from 0-11.

[16]The authors state that this two-digit number, used internally by the programs to aid computation, can also abe used to indicate at a glance which interval classes have

Tools for Operating on Sets

TRANSPOSE: recalculates any given pitch collection or set class based on one of the standard TTO operators.

COMPLEMENT: determines the complement of a given pitch collection or set class followed by an optional TTO operation.

INTERSECT: calculates the set class represented by those pitch classes forming the intersection between two or more set classes or pitch collections.

UNION: calculates the set class represented by those pitch classes forming the union between two or more set classes or pitch collections.

SETDIF: calculates the set class represented by those pitch classes found in set A that do not intersect with set B.

SYMDIF: calculates the set class represented by those pitch classes found in both sets A and B that do not intersect with each other (i.e., the symmetrical difference).

Tools for Set and Set-class Relations

SETMAP: is used to map a user-specified set class or pitch collection onto itself under any or all of the TTO operations and to show the intersection of the original and resultant sets. Additionally, the user is informed if a resultant set 1) remains totally invariant with the original set, 2) maps onto the complement of the original

a non-zero entry. Since the actual interval vector appears adjacent to it in the listing and is equally (if not more) easy to interpret, the inclusion of this number in the output display seems redundant and often uninformative (e.g., all six note sets display the octal number 77).

set, or 3) lies in the relationship Rp with the original set.[17] One problem associated with this and several other programs is the need for the large amount of information presented to be scrolled across several screens. Scrolling is automatic and can only be halted and restarted with a Control-S (^S) command entered from the keyboard. It is unfortunate that the authors do not mention this command anywhere in the manual, since it is probably not common knowledge among all users.

SUBSET: searches a given superset for all possible mappings of a given subset.

KH: generates a K/Kh[18] chart based on a collection of user- supplied sets. (Since the set-complex premise is itself highly questionable, this particular feature of CMAP is of dubious utility.)

Tools for Operating on Ordered Sets

ROWOP: is designed to perform any combination of the standard TTO operations (T,I,R,M,MI) on an ordered collection of pitches. (This program is essentially the same as TRANSPOSE except that it operates on ordered sets.)

MATRIX: generates a standard twelve-tone matrix. The user has the option of normalizing the row to begin on pitch-class zero or leaving it untransposed. All matrices are accompanied by an order number matrix as well.

IMATRIX: generates an invariance matrix[19] together with a compiled listing of invariance for each of the twelve pitch classes.

[17]Forte.
[18]Forte.
[19]Alphonce.

SEARCH: constructs a standard P, I, M, or MI matrix for a given ordered collection of pitches (either normalized or non-normalized) and allows for repeated searches of unordered user-specified subsets. Either contiguous or non-contiguous occurrences may be selected for the search. The program graphically displays the results of the search by highlighting all occurrences on the matrix. (NOTE: This program requires the user to have access to the DOS file ANSI.SYS which must be included as part of the user's CONFIG.SYS file.)

ROTARRAY: constructs and displays a rotational array matrix[20] for any ordered collection of pitches, as well as a complete listing of vertical set types. Row rotations may be normalized (Type II) or non-normalized (Type I) prior to each rotation.

ROWSTRUCT: displays a complete list of imbricated subsets of cardinality 'n' for any given ordered collection of pitches.

PARTITION: displays a complete mapping of all subsets of a specified set-class or cardinality upon a given superset. (Unlike most of the other programs, this command unfortunately works only with set classes and not with pitch-class collections.)

ROWCOMB: is used to examine the combinatorial properties of a given ordered collection of pitches. The user may specify the number of rotational levels together with the partition size governing the rotation. The program visually displays the overlapping rotations, as well as identifies all set types formed by the combinations.

[20]Rogers.

Miscellaneous Tools

PCMAP: uses the standard TTO operators to transform a given pitch collection, displaying both the resultant set type as well as the actual transformation for each pitch class. (Unfortunately, this command works only with pitch-class collections and not with set types.)

PCCOUNT: simply counts the total number of discrete pitch classes found in a user-supplied pitch collection.

PRINTTAB: displays (or prints) the entire set-class table used by CMAP programs. The table covers the complete range from sets of cardinality 1-12, and for each entry displays all the information supplied with PRIME and GETSET (see above).

TESTTYPE: allows the user to determine whether the Forte or Rahn prime forms are currently in use.

FILTERS: are used to filter (or extract) the output of a particular program so that only a selected amount of information is retrieved. FILTERS can also be used to gather relevant information to be fed into other programs. They include:

no -	normal order
ic -	interval-class vector
inv -	invariance vector
z -	Z-related set class
m -	M-related set class
cint -	cyclic interval array.

The command line for GETSET shown in figure 1 (above) is shown with the filter for normal orders in figure 3.

Figure 3

```
C> GETSET 0437845 9743 5-3 |no
```

III. SUMMARY

Overall, CMAP represents a thorough, well-executed, and comprehensive set of analytical programs. Its command-line structure, while occasionally cumbersome, allows for a great deal of flexibility through the use of pipes, filters, and multiple entries to enhance the extraction and re-channeling of relevant information. CMAP includes most of the analytical tools one might use during normal inquiry and also usually offers the flexibility to create additional tools either through programming or by taking advantage of the inherent design flexibility. The program code also appears to be streamlined, enhancing the operating speed of the individual programs.[21]

Despite the overall impressive nature of the programs, CMAP is not without a few problems. Fortunately, most of these do not seriously mar the package, since the bulk of them reside in the manual and not in the programs themselves. Two problems in the reference section bear mention. First, explanations often seem

[21]Using a hard disk on an IBM PC-AT, the program GETSET took no longer than one second to execute. The same command on an IBM PC-XT using a floppy disk took about five seconds.

overly sketchy, particularly regarding program parameters and options. The directions for executing Interval Count Searches and Invariance Mapping Relationships within the GETSET programs are especially scanty. Second, command-line synopses occasionally prove problematic. For example, the square brackets which appear throughout the manual are never discussed in the reference section; it is only in the opening of the tutorial that the authors explain that bracketed information identifies "options," yet within at least one program (ROWCOMB) a minimum of one "option" is required for the program to function properly. Fortunately, such shortcomings can be overcome by working through the entire tutorial, since most explanations in this section are clear and straightforward. However, the courageous user who decides to jump in cold can probably expect to refer to the tutorial with an exasperating degree of regularity. Unfortunately, the tutorial is relegated to the end of the manual, even though it really should be required reading for anyone new to these programs. In spite of the fact that several errors appear in the text, the bulk of the manual nonetheless appears to be quite accurate.[22]

One glaring omission from the manual is perhaps more problematic. The authors state that "CMAP filters are designed to take as their standard input the output of one of the other programs

[22]Only three errors were actually encountered during the process of reviewing the programs: 1) in the ROWSTRUCT program the "card" parameter was omitted from the command-line summary, 2) in the SEARCH command the manual states the "-s" search option is the default when in reality the "-c" option appears to be the true default, and 3) the "-t" option in the same command does not default when the "-p," "-i," "-m," or "-w" options are supplied on the command line. Actually the reverse appears to happen in that they specifically negate the "-t" option even when it is specified in the command line.

and to extract specific information from this data."[23] What is not stated is that, for the most part, the filters only work with the output of one specific program: GETSET. If the filters are applied to the output of other programs (e.g., COMPLEMENT) erroneous results usually occur.

Despite these noted shortcomings, the twenty-four programs in the CMAP package should more than adequately meet the needs of even the most ardent atonalist, and they offer plenty of power to spare. Despite being a bit lackluster in appearance, CMAP is a well-designed, thorough, flexible, powerful, and accurate set of software tools. Without a doubt, CMAP should be an indispensable tool for assisting the user in wielding the often cumbersome techniques which have become the mainstay of atonal analysis, thus freeing him to concentrate on higher analytic endeavors.

[23] CMAP Reference Manual Main Programs section, FILTERS, p. 1.