RESOURCE 3

Computer Content Analysis Software

Paul D. Skalski

This appendix provides information about quantitative computer text analysis software. Table R3.1 lists quantitative text analysis programs and highlights key features of each. Additional information about each program is included in Part I, which follows. This list owes much to the work of Popping (1997), Evans (1996), Alexa and Zuell (1999), and numerous Web site authors who have previously compiled lists of quantitative text analysis software. Part II of this Resource focuses on one basic text analysis software program, VBPro.

Although Table R3.1 contains a sample of some of the best programs currently available, it is not comprehensive; for an exhaustive list of quantitative text analysis programs, we recommend visiting Harald Klein's text analysis resources page, which can be accessed through *The Content Analysis Guidebook Online*. This companion site to the book also contains a more up-to-date list of programs, including qualitative and audio and video content analysis programs and links to the program Web sites whenever possible. The site can be accessed at the following URL: **http://academic.csuohio.edu/kneuendorf/content**.

Part I. Quantitative Computer Text Analysis Programs

The programs listed in Table R3.1 were identified in a search of published sources and Web sites. The table presents the following information about each program (numbers correspond to table columns, from left):

Table R3.1 Computer Text Analysis Software

Program	*Operating System*	*Freeware*	*Demo*[a]	*Cases Analyzed*	*Frequency Output*	*Alphabetical Output*	*Multiunit Output*	*KWIC or Concondance*	*Standard Dictionary*	*Custom Dictionary*	*Special Analyses*
CATPAC	Windows	No	No	Single	Yes	Yes	No	No	No	Yes	Yes
CPTA	DOS	NS	No	NS	Yes	Yes	NS	Yes	NS	NS	Yes
Concordance 2.0	Windows	No	Yes	Multiple	Yes	Yes	No	Yes	No	No	No
Diction 5.0	Windows	No	Yes	Multiple	Yes	Yes	Yes	No	Yes	Yes	Yes
DIMAP-3	Windows	No	Yes	Multiple	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Gen. Inquirer	Internet	Yes	NA	Single	Yes	No	No	No	Yes	No	No
INTEXT 4.1	DOS	No	Yes	Multiple	Yes	Yes	NS	Yes	No	Yes	No
Lexa	DOS	NS	Yes	Multiple	Yes	NS	NS	NS	NS	NS	Yes
LIWC	Windows and Mac.	No	No	Multiple	No	No	NS	No	Yes	Yes	No
MCCAlite	Windows	No	Yes	Multiple	Yes	Yes	Yes	Yes	Yes	No	Yes
MECA	DOS	NS	No	Multiple	No	No	Yes	No	No	Yes	Yes
PCAD 2000	Windows	No	No	Multiple	No	No	No	No	Yes	No	Yes
SALT	Windows and Mac.	No	Yes	Single	Yes	Yes	Yes	No	Yes	Yes	Yes
SWIFT 3.0	DOS	Yes	NA	Multiple	Yes	Yes	NS	Yes	No	Yes	No
TextAnalyst	Windows	No	Yes	NS	No	No	NS	No	No	Yes	Yes
TEXTPACK 7.0	Windows	No	Yes	Multiple	Yes	Yes	Yes	Yes	No	Yes	Yes
TextQuest 1.05	Windows	No	Yes	Multiple	Yes	Yes	NS	Yes	No	Yes	No
TextSmart	Windows	No	No	Multiple	Yes	Yes	NS	Yes	No	Yes	Yes
VBPro	DOS	Yes	NA	Multiple	Yes	Yes	Yes	Yes	No	Yes	Yes
WordStat 3.03	Windows	No	Yes	Multiple	Yes	Yes	Yes	Yes	No	Yes	Yes

a. Some demos allow for program testing with any user-provided data, whereas others allow testing using only data provided by the authors.
NS = not specified; NA = not applicable.

1. Program name
2. Operating system
3. Freeware (yes or no)
4. Demo availability (yes or no)
5. Cases analyzed at once (single or multiple)

The table also indicates (yes or no) whether the programs can perform the following common text analysis functions:

6. Frequency list
7. Alphabetical list
8. Multi-unit data file output (in case-by-variable form)
9. Key word in context (KWIC) or concordance
10. Coding with a built-in (standard) dictionary
11. Coding with a user-created (custom) dictionary
12. Specialty analyses

The functions are explained in the VBPro section to follow. An elaboration on each column in Table R3.1 can be found in Chapter 6.

The annotated list to follow provides a capsule description for each program listed in Table R3.1. The list contains (a) the name of the program, (b) a brief description, and (c) at least one published reference (if available), either reporting on the program or about research in which the program was used. Following the reference(s), the designation "**Web site**" indicates that there is a program Web site from which additional information can be obtained; visit *The Content Analysis Guidebook Online* for a link to the site.

A research team also evaluated several of the programs from Table R3.1. The results of our tests, along with sample output from each program, can also be viewed at *The Content Analysis Guidebook Online.*

The VBPro program receives special attention in this Resource for several reasons. First, it performs all key computer text analysis functions, making it a good vehicle through which to explain the typical process and principal functions of a computer text analysis. Second, the program is available free online. If you are a beginner to computer text analysis, we recommend trying VBPro first to get a feel for the technique. Our step-by-step guide in Part II of this Resource is designed to make the process of using VBPro as simple as possible.

VBPro

Description: VBPro outputs frequency and alphabetical word lists, key words in context (KWIC), and coded strings of word-occurrence data based on user-defined dictionaries. In addition, it includes a multidimensional concept-mapping sub-program called VBMap that measures the degree to which words co-occur in a text or series of texts. Miller, Andsager and Riechert (1998) used the program to compare the press releases sent by 1996 GOP presidential candidates to the coverage the candidates received in the press. The program helped the researchers (a) generate a list of key words appearing in the text and (b) generate a map showing the relative positions of candidates, in both press releases and media coverage, to each other and on key issues in the election (e.g., family values, education). The program runs under DOS and is available for free from the software author's Web site.

Criteria From Table

Unit of Analysis: VBPro can analyze sentences, paragraphs or cases. Users can combine all of their text into a single text file to simplify the analysis process.

Frequency and Alphabetical Output: Alphabetical output includes the name of the word and frequency of occurrence (e.g., *conservative, 12*). Word frequency output (listed from most frequent to least occurring) includes the word name, number of occurrences, and percentage of tokens (or total words) that the word represents (e.g., *liberal, 11, 0.750)*.The frequency command also produces a type/token frequency and ratio and the occurrence rates of tokens.

Multi-Unit Data File Output: Refers to grid-like frequency output suitable for import into a statistical analysis package (e.g., SPSS), typically with case or unit numbers on the vertical axis and dictionary terms on the horizontal, as in this partial example.

	Dictionary Counts by Term			
Case	*Liberal*	*Political*	*Blame*	*Other*
1	5	12	3	—
2	2	5	4	—
and so on . . .	—	—	—	—

Through the CODE command, VBPro can give multiunit data file output, displaying frequency information about each dictionary category for each case (e.g., each news story in a text file of 100 stories).

KWIC or Concordance: KWIC can be generated through the SEARCH function of VBPro. Each time a dictionary term occurs, the entire unit of analysis (e.g., sentence) will appear in the SEARCH output with the key word designated by marks (e.g., . . . due to the leftist principles of . . .). After the KWIC analyses, the SEARCH command outputs summary information, including number of cases, sentences, and paragraphs containing the dictionary terms.

Coding With Dictionary: Once a dictionary is prepared, the VBPro CODE command can code a text file based on the dictionary terms. This type of coding is done on a case-by-case basis within a single text file. The comma-delimited text output file from CODE can then be imported into a statistics program (e.g., SPSS) for further analysis.

Specialty Analyses: VBPro can also perform concept mapping through a subprogram called VBMap (see earlier "Description" section and Chapter 6 for more information).

Developer: M. Mark Miller

References: Dyer (1994a, 1994b); Dyer, Miller, and Boone (1991); Miller et al. (1992); Miller et al. (1998); Miller and Denham (1994); Web site

CATPAC

Description: CATPAC reads text files and produces a variety of outputs ranging from simple diagnostics (e.g., word and alphabetical frequencies) to a summary of the main ideas in a text. It uncovers patterns of word usage and performs analyses such as simple word counts, cluster analysis (with icicle plots), and interactive neural cluster analysis. A nifty add-on program, called Thought View, can generate two and three-dimensional concept maps based on the results of CATPAC analyses. One especially neat feature of Thought View allows users to look at the results through 3-D glasses and experience MDS-style output like never before!

Developer: Joseph Woelfel

References: Doerfel and Barnett (1999), Salisbury (in press); Web site

Computer Programs for Text Analysis

Description: This is not a single computer program but rather a series of separate programs. Each performs one or two basic functions, including analyzing appearances of characters in a play (ACTORS program), getting KWIC (CONCORD program), computing the amount of quotation in texts (DIALOG program), and comparing the vocabulary of two texts (IDENT program). The programs seem ideal for literary-type analyses.

Developer: Eric Johnson

Reference: Web site

Concordance 2.0

Description: This program allows users to make concordances on texts of any size. Program functions include (a) making one concordance from multiple input files; (b) sorting output according to length, frequency, alphabetical listing, and other criteria; and (c) making Web concordances, which allow users (and Web surfers) to point to a word in an online text and retrieve concordance information instantly, through the magic of hypertextuality. A demo of Web concordances can be found on the program Web site.

Developer: R. J. C. Watt

Reference: Web site

In addition to the Concordance, several other programs are available, primarily for KWIC analyses, such as MonoConc and ParaConc. Information on these programs can be found at *The Content Analysis Guidebook Online*.

Diction 5.0

Description: Diction 5.0 contains a series of built-in dictionaries that search text documents for five main semantic features (activity, optimism, certainty, realism and commonality) and 35 subfeatures (including tenacity, blame, ambivalence, motion, and communication). After the user's text is analyzed, Diction compares the results for each of the 40 dictionary categories to a provided normal range of scores established by running more than 20,000 texts through the program. Users can compare their text to either a general normative profile of all 20,000-plus texts or to any of six specific subcategories of texts (business, daily life, entertainment, journalism, literature, politics, and scholarship) that can be further divided into 36 distinct types (e.g., financial reports, computer chat lines, music lyrics, newspaper editorials, novels and short stories, political debates, social science scholarship). In addition, Diction outputs raw frequencies (in alphabetical order), percentages, and standardized scores; custom dictionaries can be created for additional analyses.

Developer: Roderick P. Hart

References: Hart (1985, 2000a); Hart and Jarvis (1997); Web site

DIMAP-3

Description: DIMAP stands for DIctionary MAintenance Programs, and its primary purpose is text content analysis dictionary development. The program includes a variety of tools for lexicon building rooted in computational linguistics and natural language processing (Litkowski, 1992). With DIMAP, users can build, manage, edit, maintain, search, and compare custom and established dictionaries. The program also includes a text analysis module called MCCA (the "lite" version of which is described later), providing printing of output, additional co-occurrence analyses, and multidimensional scaling of matrices measuring the distance of texts. A new version of the program, DIMAP-4, is now available in a beta version.

Developer: Ken Litkowski

References: Litkowski (1978); Litkowski (1992); Web site

General Inquirer (Internet version)

Description: This venerable, still widely used program has found new life on the World Wide Web. The online version of the General Inquirer gets our vote for the simplest and quickest way to do a small-scale computer text analysis—simply visit the Internet General Inquirer site, type or paste some text into a box, click "submit," and your text will be analyzed. The Internet General Inquirer codes and classifies text using the Harvard IV-4 dictionary, which assesses such features as valence, Osgood's three semantic dimensions, language reflecting particular institutions, emotion-laden words, cognitive orientation, and more. The program also returns cumulative statistics (e.g., simple frequencies for words appearing in the text) at the end of each analysis. A stand-alone computer-based version of the program, which can process multiple cases, is now available free to academics and nonprofit organizations. See the program Web site for more information.

Developers: Phillip J. Stone, The Gallup Organization, and Vanja Buvac (Web version)

References: Kelly and Stone (1975); Stone et al. (1966); Web site

INTEXT 4.1

Description: INTEXT is a program designed for the analysis of texts in the humanities and the social sciences. It performs content analysis, text analysis, indexing, concordance, KWIC, KWOC (key word out of context), readability analysis, personality structure analysis, word lists, word sequence, word permutation, stylistics, and more. The content analysis module allows interactive coding and can handle ambiguous or negated search patterns. In dictionaries, search patterns can be single words, phrases, or co-occurences.

Developer: Harald Klein

References: Klein (1991); Web site

Lexa (Version 7)

Description: Designed with linguists in mind, Lexa Corpus Processing Software is a suite of programs for tagging, lemmatization, providing type/token frequency counts, and conducting several other forms of computer text analysis.

Developer: Raymond Hickey

Reference: Web site

LIWC (Lingustic Inquiry and Word Count)

Description: LIWC was developed for researchers interested in emotional, cognitive, social, or other psychological processes. Using internal dictionaries, the program analyzes individual or multiple text samples along 72 dimensions, including psychological constructs (e.g, affect and cognition), personal-concern categories (e.g, work, home, and leisure activities), and standard linguistic dimensions (e.g., percentages of pronouns and articles). The various dictionaries have good psychometric properties. LIWC can also analyze up to 100 dimensions with custom dictionaries. Individual files can be automatically divided and analyzed in units set by the user.

Developers: James W. Pennebaker and Martha E. Francis

References: Gunsch, Brownlow, Haynes, and Mabe (2000); Pennebaker and Francis (1999); Pennebaker, Francis, and Booth (2001); Web site

MCCALite

Description: Though somewhat hampered by quirks such as limited function availability, the "lite" version of MCCA analyzes text by producing frequencies, alphabetical lists, KWIC, and coding with built-in dictionaries. The built-in dictionaries search for traditional, practical, emotional, and analytic contextual dimensions and emphases among 116 idea categories, such as physical descriptions; role, time, emotion, traditional and analytic nouns; positive and negative adjectives; control, analysis, deviance, activity, pressure, and negative and positive reaction verbs. The program's window-driven output makes sorting and switching between results easy. MCCALite also includes a multiple-person transcript analysis function suitable for examining plays, focus groups, interviews, hearings, TV scripts, and other such texts.

Developers: Donald G. McTavish and Ellen B. Pirro

References: McTavish and Pirro (1990); Web site

MECA

Description: MECA, which stands for Map Extraction Comparison and Analysis, contains 15 routines for text analysis. Many of these routines are for doing cognitive mapping and focus on both concepts and the relations between them. There are also routines for doing more classic content analyses, including a multiunit data file output routine that shows the number of times each concept appears in each map.

Developer: Kathleen Carley

References: Carley (1993, 1997a, 1997b)

PCAD 2000

Description: PCAD 2000 applies the Gottschalk-Gleser content analysis scales, which measure the magnitude of clearly defined and categorized mental or emotional states, to transcriptions of speech samples and other machine-readable texts. In addition to deriving scores on a variety of scales, including anxiety, hostility, social alienation, cognitive impairment, hope, and depression, the program compares scores on each scale to norms for the demographic groups of subjects. It can also explain the significance and clinical implications of scores and place subjects into clinical diagnostic classifications derived from the Diagnostic and Statistical Manual of Mental Disorders, Fourth Edition (DSM-IV), developed by the American Psychiatric Association.

Developers: Louis Gottschalk and Bob Bechtel

References: Gottschalk (1995); Gottschalk & Bechtel (1993); Gottschalk & Gleser (1969); Gottschalk, Stein, & Shapiro (1997); Web site

SALT (Systematic Analysis of Language Transcripts)

Description: This program is mainly designed to help clinicians identify and document specific language use problems. It executes myriad analyses, including types of utterances (e.g., incomplete, unintelligible, nonverbal), mean length of utterances, number and length of pauses and rate of speaking, and frequencies for sets of words (e.g., negatives, conjunctions, and custom dictionaries). The Salt Reference Database, described online, allows users to compare the results of their SALT analyses to normative language measures collected via a sample of more than 250 children of various ages, economic backgrounds, and abilities in the Madison, Wisconsin area plus rural northern Wisconsin.

Developers: Jon F. Miller, Robin S. Chapman, and Ann Nockerts

Reference: Web site

SWIFT 3.0

Description: SWIFT (Structured Word Identification and Frequency Table) is a program designed for the analysis of many short passages of text (e.g., abstracts, open-ended questionnaire responses). The program outputs word occurrence frequencies based on user-defined dictionary terms and the output can be imported into statistical software for further analysis. According to the program Web site, SWIFT now

> serves mainly as a introductory-level program that does a few things very well but is not up to snuff as a general purpose CATA program. . . . It is still very good (the best in the author's humble opinion) for analyzing a large set of short responses to open-ended questions.

Developer: Ronald B. Heady

Reference: Web site

TextAnalyst

Description: TextAnalyst seems to be designed primarily for managing texts, but it has some functions of use to content analysts. It contains neural network technology for the structural processing of texts. This technology helps manage large amounts of textual information and can be applied effectively to perform the following tasks: creation of knowledge bases expressed in a natural language; creation of hypertext, searchable, and expert systems; and automated indexing, topic assignment, and abstracting of texts.

Developer: Megaputer Intelligence Inc.

Reference: Web site

TEXTPACK 7.0

Description: The TEXTPACK program, which was originally designed for the analysis of open-ended survey responses, has been broadened in recent years to include many aspects of CATA and most of content analysis. It produces word frequencies, alphabetical lists, KWIC and KWOC searches, cross-references, word comparisons between two texts, and automatic coding based on user-created dictionaries. This multiunit data file output can be imported in statistical analysis software. A Windows version of the program is now available.

Developers: Peter Ph. Mohler and Cornelia Zuell

References: Olsen (1989); Web site

TextQuest 1.05

Description: This is the Windows version of Intext, described earlier. It performs most of the Intext analyses in the current version but through an easier-to-use

Windows interface. A new add-on program, TextGrab, captures whole Web sites, copies them to a hard drive, and formats them for TextQuest analyses.

Developer: Harald Klein

Reference: Web site

TextSmart by SPSS

Description: This software, designed primarily for the analysis of open-ended survey responses, uses cluster analysis and multidimensional scaling techniques to automatically analyze key words and group texts into categories. Thus, it can code, so to speak, without the use of a user-created dictionary. TextSmart has a pleasant, easy-to-use Windows interface that allows for quick sorting of words into frequency and alphabetical lists. It also produces colorful, rich-looking graphics, such as bar charts and two-dimensional MDS plots.

Developer: SPSS

Reference: Web site

WordStat v3.03

Description: This add-on to the SimStat statistical analysis program includes several exploratory tools, such as cluster analysis and multidimensional scaling, for the analysis of open-ended survey responses and other texts. It also codes based on user-supplied dictionaries and generates word frequency and alphabetical lists, KWIC, multi-unit data file output, and bivariate comparisons between subgroups. The differences between subgroups can be displayed visually in high-resolution line and bar charts and through 2-D and 3-D correspondence analysis biplots. One particularly noteworthy feature of the program is a dictionary-building tool that uses the WordNet lexical database and other dictionaries (in English and five other languages) to help users build a comprehensive categorization system.

Developer: Provalis Research

Reference: Web site

Part II: VBPro How-To Guide and Executional Flowchart

This how-to guide takes you through a typical VBPro session. (See also the flowchart that follows.) For information about how to get a free copy of

VBPro, visit *The Content Analysis Guidebook Online*. The VBPro program includes a copy of the official, complete VBPro User's Guide, which contains in-depth information on all of the topics mentioned here.

Preparation: Before starting VBPro, there are a few preparatory steps that may need to be taken. First, if you plan to analyze several cases of data (e.g., single news stories in a file of news stories), each case should be identified (ID'd) with a "#" sign followed by numbers, as in "#001." Second, if you plan to code with a set of custom dictionaries, they should be created at this point. A user-created dictionary file in VBPro consists of search categories of interest (i.e., dictionaries) separated by >><< marks above each entry (e.g., >> colors <<, with red, blue, green, etc., on their own separate lines underneath).

Tips: Here are some additional preparatory steps that we've come up with, based on our own experience with the program:

1. Delete all # signs before adding IDs to your text file.
2. Make sure all IDs are left justified and each is located on its *own* line.
3. Bracket all material that you do *not* wish to have included in your analysis, using the characters [and]. All other brackets should be removed, as the material enclosed by them will be ignored by VBPro.
4. Change ellipses with embedded spaces to strings with no spaces (" . . . " becomes "...").
5. Change hyphens to spaces ("ticket-holder" becomes "ticket holder").

Note that most of these changes can easily be done with the "Find and Replace" tool on popular word-processing packages.

Step 1: Save your text and dictionary (if used) in the VBPro program directory (e.g., C:\VBPro) in ASCII-DOS format. If done properly, both files should now have an .asc extension (e.g., JOKES.ASC, DICT.ASC). As you move through the process of using VBPro, you will notice that the dictionary file (DICT.ASC) is required for several functions: CODE, SEARCH, and MAP. Your dictionary file should be the *same* for all three applications in any given text analysis.

Step 2: Open VBPro and *format* your ASCII text using the FORMAT function. Note that VBPro does not support the use of a mouse. All commands must be executed with a keyboard. To select functions from the main screen, press the control (Ctrl) key and the first letter of the function simultaneously (e.g., "Ctrl + F" for FORMAT). Follow the directions in the format menu. After you do so, the format window should say "Formatted File Saved: [filename].frm" (e.g., JOKES.FRM).

Once your text is formatted, you have three immediate choices:

Step 3a: One choice is to have the text words sorted in ***alphabetical order*** along with each word's frequency of occurrence (e.g., ***conservative,*** *12*). The ALPHABETIZE function does this. Enter the ALPHABETIZE menu and follow the instructions. The alphabetical words file will be saved in your VBPro directory with an .alf extension (e.g., JOKES.ALF).

Step 3b: Another choice is to ***code*** your text using custom dictionaries. The VBPro CODE command produces multiunit data file output (illustrated earlier in this Resource) based on user-created dictionaries. To run CODE, enter the code menu and follow the instructions. Say "yes" to "Ignore Caps?" and be sure to input the name of your dictionary file (e.g., DICT.ASC)and output file. The coded data file will be saved in your VBPro directory with the name and extension of your choice (e.g., JOKES.COD).

Step 3c: A third choice is to have VBPro perform a KWIC analysis of your text. The SEARCH command does this. Like CODE, SEARCH requires user-created dictionaries. KWIC output will be returned for all dictionary terms. To run SEARCH, enter the menu and follow the on-screen instructions. Be sure to input the dictionary name and an output file name of your choice. The KWIC analysis will be saved in the VBPro directory with the name you specify (although again, we recommend a standard .src extension, e.g., JOKES.SRC).

Step 4: The saved alphabetical file can be used to create a ***frequency*** output file. Though the ALPHABETIZE command gives a simple frequency figure, the RANK command gives some additional statistics, including percentage of occurrence for each term compared to all terms and a variety of type/token measures. To produce frequency information, select RANK from the main VBPro menu and simply indicate the name of your alphabetical file. The frequency file will be saved in the VBPro directory with an .frq extension (e.g., JOKES.FRQ).

Step 5: Using a separate, optional program available at the developer's Web site, the SELECT function compares two different alphabetized lists (i.e., two alf. files). The top 200 word frequency differences between the lists are identified and analyzed using a chi-square test. The output of this analysis is written to a file with a .sct extension (e.g., JOKES.SCT).

Step 6: A final option, ***concept mapping,*** can be performed through the VBMap subprogram (vbmap.exe in your VBPro directory). Concept mapping generates a matrix of dimensional coordinates that can be used to construct a three-dimensional map of concepts (dictionaries), in which the proximities between terms are indicative of the degree to which they tend to co-occur. To run VBMap, you first need coded output, which can be obtained through the CODE function. The coded output must then be entered into VBMap, which, using the dictionary file (e.g., DICT.ASC), produces the dimensional coordinates—but *not* the actua three-dimensional representation of concepts. The dimensional coordinate alone are written to a file with a .uns extension automatically provided (e.g. JOKES.UNS), and output, including eigenvalues, coordinates, percentage tota variance per eigenvector, and cumulative percentage total variance, are provided in a file with a .map automatic extension (e.g., JOKES.MAP). To subsequently obtain a visual map of the results, you need to import the output from VBMap into a program with graphics capabilities (e.g., SPSS or Thought View). If SPSS is used, the concepts can be visually mapped with the "Scatterplot" selection under the "Interactive graphs" menu. See Figure 8.7 for an example of a VBPro 3-D concept map.

Figure R3.1 presents a flowchart indicating the range of analyses that VBPro performs and the steps that must be taken to execute and access each type of analysis. The file name JOKES.ASC refers to a sample file of jokes, and the file name DICT.ASC refers to a multiconcept dictionary file. Examples of dictionary files and VBPro output can be found at *The Content Analysis Guidebook Online.*

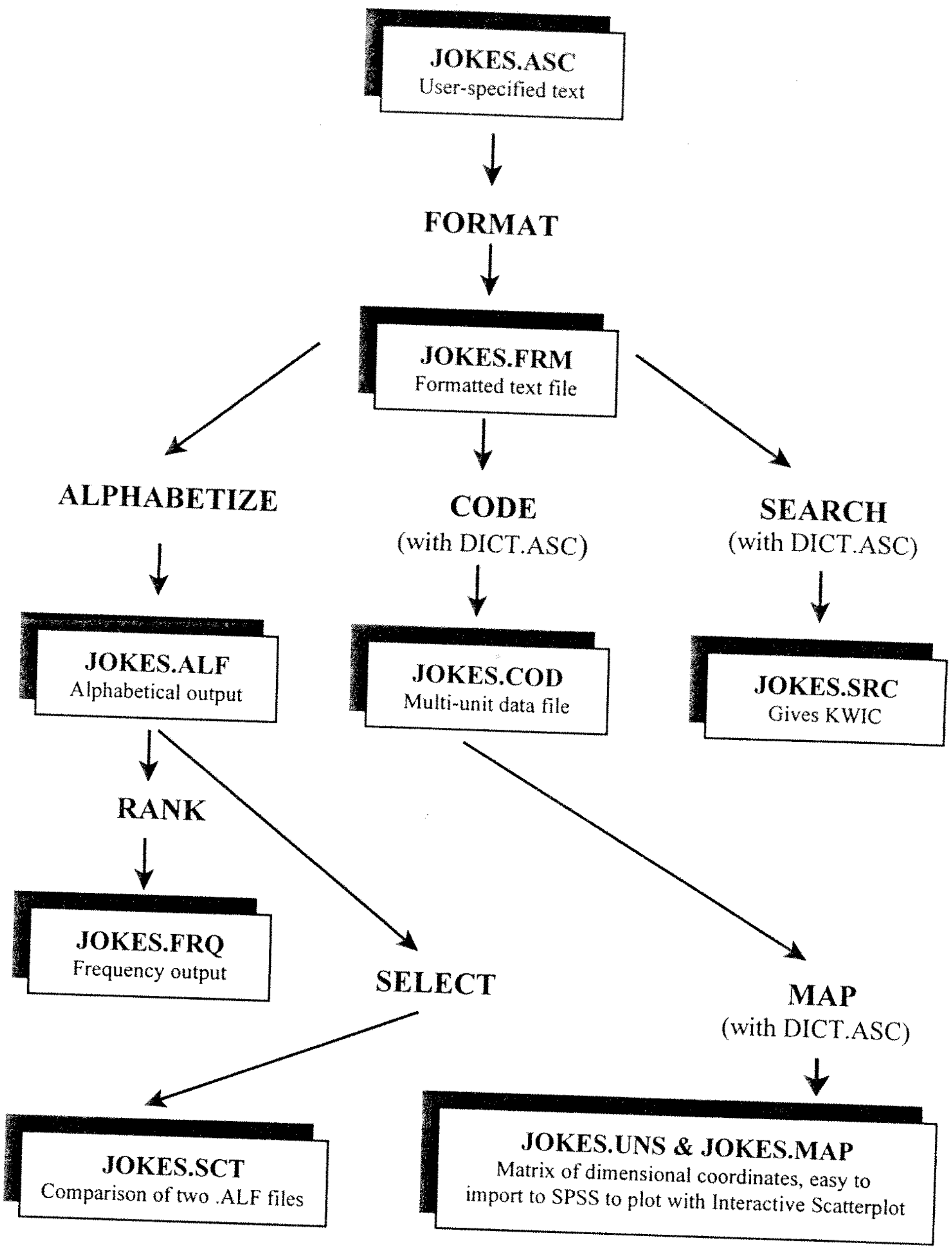

Figure R3.1. VBPro Flowchart