Research programs in the pharmaceutical industry require that very similar studies be performed on many different compounds. If the experimental designs and the variables measured and compared are sufficiently similar, it is possible to use a set of standard programs that differ only in the names of the data files and in the variables to be reported. This paper is a description of a set of interactive program generators that use PROC FSEDIT to specify the information that varies from one study to another. The program generators use a common data set to store the values of this variable information. The common data set also automatically includes audit trail information, such as, the date and programmer's name each time a program generator is executed. The design and use of the program generators are illustrated with an example from bioequivalence testing.

I. INTRODUCTION

Crow and Malicz (1984) described a set of SAS programs for analyzing and reporting data from bioavailability and bioequivalence clinical trials. The programs were coded so that by changing the values of a few MACRO variables and a few format specifications at the beginning of each program, the programs could be adapted from one clinical trial to another. This approach was made possible by using experimental design and Case Report Forms (CRFs) that were consistent from trial to trial. The consistency among CRFs made it possible to build SAS data sets that used the same variable names and that used a convention for naming data sets. The consistency of experimental design reduces the number of choices that must be made in producing data tables and standard analyses. With the availability of SAS 82 (SAS Institute, Inc., 1982) and the Full Screen Product (SAS/FSP) for the CMS operating system, we devised a different approach to this system of programs. Rather than using a text editor to change the values of MACRO variables, a program generator invokes PROC FSEDIT, and the variable information is entered into a SAS data set. The program generator then accesses the data set and writes a program based on the information provided. This generated program can be executed separately, or it can be executed in the same interactive session by using %INC. The design and use of these program generators are illustrated with an example from bioequivalence testing.

II. EXAMPLE

As an example consider a typical bioequivalence trial in which a new formulation of a compound is compared with a reference formulation. The experimental design is a 2-treatment, 2-period crossover (Cochran and Cox, 1957). During the first treatment period, each subject receives 1 of the 2 formulations (e.g., a single oral dose of drug). After a "wash-out" interval long enough to ensure complete elimination of the drug from the body, the second treatment period begins. At the beginning of the second period, each subject receives the drug formulation that was not administered during the first period. An equal number of subjects are randomly assigned to each of the 2 sequences of drug administration.

During the 24 to 48 hours after each administration of drug, several blood samples are drawn from each subject, and the samples are assayed later for drug concentration. The sampling intervals are chosen to characterize the absorption and elimination of the particular compound being studied. The variables of greatest interest for establishing equivalence of the 2 formulations are the largest observed concentration (Cmax), the time interval between drug administration and Cmax (Tmax), and area under the drug concentration-time curve (AUC). AUC is computed by using the trapezoidal rule. We may also wish to calculate elimination rates and/or absorption rates. Other data recorded include the results of clinical laboratory tests (hematology, blood chemistry, and urinalysis), physical examinations, and the results of electrocardiography performed before and after the study to monitor the safety of the subjects. The symptoms and severity of any possible side-effects are also recorded. All of the data from the clinical trial are put into a data library that consists of separate SAS datasets for each category of data; such as, clinical laboratory determinations, drug concentration assay values, adverse experiences, etc. These data sets always have the same CMS file names; the CMS file types are determined for each study by using the naming convention described below.

III. THE PROGRAM GENERATORS

A. General Features

Each of the 5 program generators has several common features. Each program generator uses the SAS MACRO language and is invoked in interactive SAS by using %INC. First the generator asks that a code identifying the project and protocol (study) be entered. The code consists of 3 letters representing the project and a 3-digit number representing the protocol. This code represents the CMS file type of a SAS data set that is used to store all of the variable information that must be
passed to the programs to be created. The CMS file name is always SYSINFO. This SYSINFO data set also includes date-time variables that automatically record the most recent execution of each program generator. This information documents the progress of the analysis.

Four of the 5 program generators can produce more than 1 program per execution. If the product being tested has more than 1 component or if metabolites are also assayed, a separate program is created for each component. These programs are differentiated by an index number in the file name. Also, different specifications can be entered into the program generators for the different components.

Figure 1 is a flow chart that depicts the flow of information among the program generators, their data sources, and the programs that they create. The dotted lines represent the flow of information, and the solid lines represent the creation of CMS files that contain programs, data, or print files.

B. PLOT Programs

The first thing that the data analyst wants to see are plots of drug concentration vs. time. The plots not only help to identify data problems or outliers, they also indicate how to proceed with the analysis. For example, elimination rates are calculated by a linear regression of log concentration vs. time over the tail of the drug concentration-time curve. If elimination rates are to be estimated, the analyst must examine the log concentration-time curve to be able to determine when absorption is complete and therefore when the tail of the curve begins.

After invoking the plot program generator and specifying the project and protocol, the generator automatically invokes PROC FSEDIT on the SYSINFO data set. The screen allows one to specify plotting information such as, treatment labels, length of the horizontal (time) axis, axis labels, and concentration units. One can choose among the available devices (graphics terminals, flatbed plotter, etc.) for producing the plots. One can also choose an arithmetic or a logarithmic scale for drug concentration. Drug concentration-time curves for each patient or plots of mean concentrations can be produced. If more than 1 program is being created, the specifications may be different for each program.

C. The BUILD Program Generator

The BUILD program is used to calculate functions of the drug concentration-time curve (such as Cmax, Tmax, AUC, etc.) for each patient and treatment. This program "builds" a new SAS data set that contains the drug concentrations, sampling times, and all of the calculated variables.
forms of data display (plots and tables) and different statistical analyses. Nevertheless, the consistency of this approach should prove to be a much less costly way to process large numbers of medical research studies.

V. REFERENCES


James T. Crowe, Jr.
A. H. Robins Company
P. O. Box 26609
Richmond, VA 23261-6609
(804) 257-2854
FIGURE 1
SYSTEM FLOW CHART

LEGEND:

INFORMATION FLOW
---
PROGRAM OUTPUTS

PLOTS

PROG GLM EUMINATION RATE REGRESSION

PROG GLM DATA SET

ABSORPTION RATE DATA

BUILD PROGRAM

SUMMARY PROGRAM

BIOAVAILABILITY DATA SET

ABSorption RATE DATA

BUILD PROGRAM

SUMMARY PROGRAM

STATISTICAL ANALYSES

SCREENING AND SAFETY TABLES

STATISTICS PROGRAM GENERATOR

LIST PROGRAM

SCREENING AND SAFETY DATA

SYSINFO DATA SET

LIMITS DATA SET

PROGRAM OUTPUTS