

# **Clinical Trials Simulation System Documentation**

**Peter H. Westfall<sup>1</sup>, Kuenhi Tsai<sup>2</sup>, Miles Dunn<sup>2</sup>, Stephan Ogenstad<sup>3</sup>,  
Alin Tomoiaga<sup>1</sup>, Yongang Lu<sup>1</sup>, Keyi Wang<sup>1</sup>**

<sup>1</sup> – Texas Tech University

<sup>2</sup> – Vertex Pharmaceuticals

<sup>3</sup> – Statogen Consulting LLC

# Table of Contents

## System Requirements

### I. Overview

### II. Examples

- A. Sample size allocation
- B. Choice of statistical test
- C. Choice of duration of study

### III. Using the Clinical Trials Simulation Software

- A. Installing and deploying the system
  - 1. Unpacking the files
  - 2. Setting the path
  - 3. Batch and interactive starting modes
- B. Running the application
  - 1. Starting the system
  - 2. Local or grid runs
  - 3. Input of clinical trial parameters
- C. Standard statistics collected
- D. Modifying the SAS/AF graphical interface
  - 1. Accessing the SCL code
  - 2. Inserting additional code
  - 3. Recompile and run

### IV. Customizing the Program to Collect Additional Statistics

- A. Code locations, code samples, and file names
- B. General instructions
- C. Example: From II.A. (Sample size allocation)
  - 1. Load the parameters
  - 2. Modify the code
  - 3. Data analysis
- D. Example: From II.B. (Choice of statistical test)
  - 1. Input the simulation settings via the interface
  - 2. Identify variable names
  - 3. Modify the code
- E. Example: From II.C. (Choice of duration of study)

### V. Technical Report Containing Mathematical Details

## System Requirements

The system requires at least a client (or local machine), and optionally, host machines (for grid runs).

The system requires SAS/Windows for the client (local) machine with Version 9 or higher, (the system runs with partial functionality under Version 8), including SAS/BASE, SAS/STAT, and SAS/AF for local runs. SAS/GRAPH is desirable as well, but not necessary.

For grid runs, SAS/CONNECT is also needed for client and hosts, and SAS/BASE, SAS/STAT are needed on the hosts, but can be in any operating system.

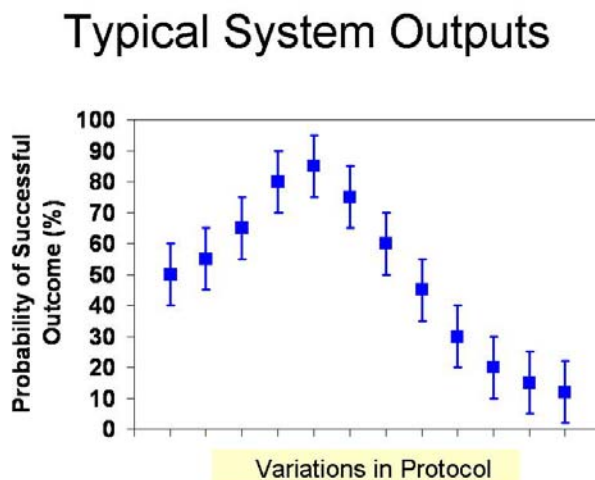
## I. Overview

This is a general trial generation/analysis system. It allows the user to input parameters of a clinical trial, and it generates many hypothetical trials resulting from such input specifications. The trials have a “typical” form, with multiple records per patient data from repeat visits to the treatment center, with multiple treatment groups and, as well as patient dropouts. These simulated trials can then be used to design trials optimally for sample size allocation, patient length on trial, choice of statistical test, etc., potentially saving costs associated with inefficient trial and protocol design.

The data sets are huge and the analyses are complex, so supercomputing is recommended. This system can be run using a grid or locally, producing “canned” and customizable output. Parameters are entered using a SAS/AF graphical user interface and can be saved for ease of future use. There are ~3,100 lines SCL code; ~2,300 lines of SAS/BASE/STAT/Macro code, and hundreds of SAS/AF instructions, all embedded in the system.

The main approach is statistical; Pharmacokinetic/Pharmacodynamic (PK/PD) models are used only used indirectly. The goal is to simulate realistic data sets with appropriate mean, covariance, and distributional structures that are needed for answering statistical questions in protocol and trial design, with emphasis on Phase II/III trials, but which can be customized for certain Phase I designs (eg crossover studies) as well. The system generates multiple timepoint/endpoint data with flexible covariance structures, flexible mean structures, including natural history and placebo effects, flexible distributions including survival, compliance effects, and informative dropout mechanisms.

The following figure shows typical use of the system.



The horizontal axis includes potential protocols with alternative sample size configurations, trial length configurations, statistical tests employed, and other design/analysis features. The vertical analysis measures success probability, which might encompass statistical power as well as other desirable outcomes (such as secondary endpoints trending in the correct directions). The vertical

bars on the graph indicate simulation error; these bars can be made smaller with more simulations. The next section provides specific examples.

## II. Examples

The following three examples were analyzed using the system, and show a sample of what is possible. The scope of applications is much broader than the small sampling shown here.

### II.A. Sample size allocation

Consider an investigation of an arthritis drug, with the binary outcome ACR20 as the primary endpoint. There will be Control, Low, Mid, and High doses. Expectations are that ACR20 response rates are 30%, 50%, 60% and 70%, respectively, and that patient dropout rates are 5%, 10%, 15%, and 20%, respectively. All comparisons will be made using Chi-Square Dose/placebo tests, using the fixed sequence multiple comparisons method (High dose first, then Mid dose, then Low dose, tested in order until one fails to achieve significance. The total number of patients is 200, and the question is, how to allocate them among the groups?

Elements that make this problem require simulation (rather than analytical results) are (a) the use of Chi-Square tests, whose mathematical distributions are asymptotic rather than exact in finite samples, (b) the dropout issue, and (c) the use of fixed sequence tests, whose power functions depend on joint distributions rather than marginal distributions.

Using the system, with 20,000 simulated clinical trials per design (using the grid implementation), we obtain the following table:

| Design       | High Dose | Med Dose | Low Dose |
|--------------|-----------|----------|----------|
| 50,50,50,50  | .973      | .816     | .465     |
| 101,33,33,33 | .966      | .800     | .448     |
| 95,30,35,40  | .981      | .822     | .426     |
| 80,40,40,40  | .977      | .835     | .480     |
| 80,35,40,45  | .985      | .837     | .452     |
| 74,42,42,42  | .976      | .834     | .484     |

Entries shown in the table are power using the fixed sequence procedure for the various tests. The bottom design seems preferable if the goal is to maximize probability of detecting Low dose significance while maintaining reasonable power for the Mid and High doses. This allocation is familiar for such designs; see e.g. Hochberg and Tamhane (*Multiple Comparison Procedures*, 1987, pp. 164-169).

### II.B. Choice of Test

ACR20 is a composite of the seven endpoints Tender Joint Count, Swollen Joint Count, Patient Global Assessment, Investigator Global Assessment, Grip Strength, Pain, and Erythrocyte

Sedimentation Rate. ACR20 = 1 if there is a 20% improvement in the first two endpoints, and a 20% improvement in at least 3 of the remaining 5 endpoints; ACR20=0 otherwise. Rather than use such a crude binary scoring of the data, which loses information and sensitivity, it has been suggested to use the O'Brien test, which more or less combines data from all endpoints in a continuous scale (Anderson, Bolognese, and Felson (2003), "Comparison of Rheumatoid Arthritis Clinical Trial Outcome Measures," *Arthritis and Rheumatism* 48, 3031-3038).

The following table compares powers of the two tests.

| Design  | O'Brien | ACR20 |
|---------|---------|-------|
| 50,50   | .60     | .41   |
| 70,70   | .86     | .40   |
| 100,100 | .98     | .58   |

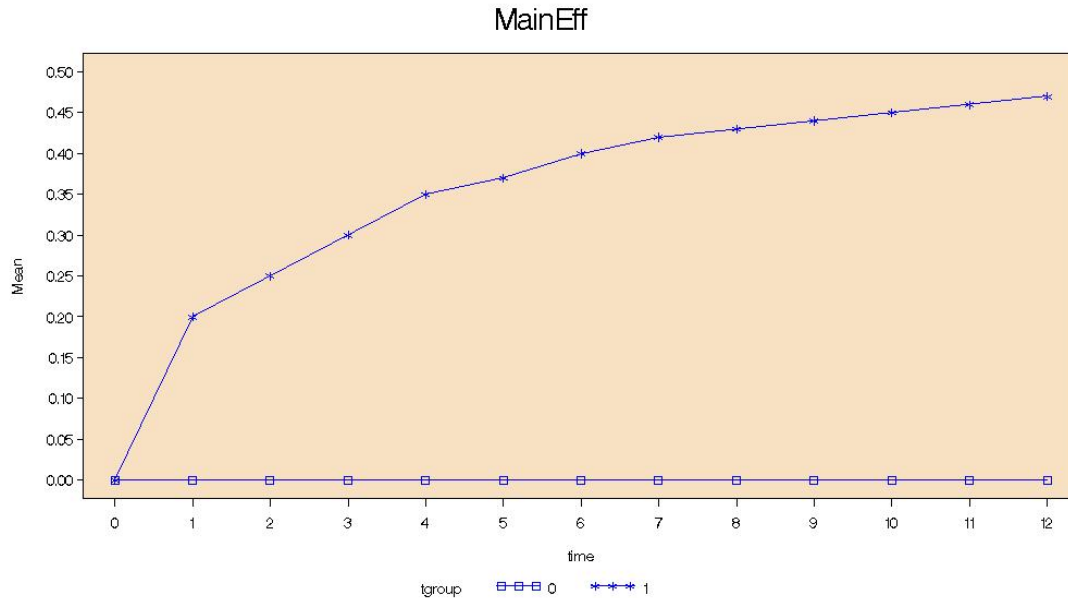
Clearly, the ACR20 test is wasteful in terms of subjects required.

This example was a "local" (non-grid) run, so a much smaller number of simulations (100) was used. The consequences are lack of accuracy as can be seen from the ACR20 power values – obviously they should be strictly increasing as the sample sizes increase.

## II.C. Choice of Design, Test, and Duration of Study

The system produces data with user-specified distributional characteristics, including outlier-prone mixture distributions, useful for modeling patient populations with "outlier" subgroups, or for modeling percentage change data, which are commonly heavy tailed. The usual statistical tests are known to lack power for these studies, so an investigation is needed to select the appropriate test, in addition to sample size allocations.

A third aspect that can be investigated is length of study. The system uses as input the responses over time (as established eg, by time-and-dose models from earlier phase studies), so one can investigate the consequences of shorter trial duration. The user input for time response in this trial looks as follows:



The functions assume a flat placebo response, but a treatment response that increases with time. The measurement standard deviation is assumed to be 1.0, so the effect size is  $(.47-0)/1 = .47$  at 12 weeks and  $(.43-0)/1 = .43$  at 8 weeks.

Dropouts are assumed, and LOCF imputations are used. Based on the inputs, the following simulated power results are obtained:

| Design             | Type of Analysis |        |        |        |        |        |      |      |
|--------------------|------------------|--------|--------|--------|--------|--------|------|------|
|                    | AOV              |        | ANCOVA |        | ANCOVA |        | K-W  | K-W  |
|                    | Mean             | Median | Mean   | Median | Mean   | Median | Diff | Diff |
| 12 wks/<br>30,30   | 0.41             | 0.55   | 0.54   | 0.48   | 0.47   | 0.58   | 0.67 | 0.65 |
| 12 wks/<br>50,50   | 0.57             | 0.73   | 0.72   | 0.67   | 0.64   | 0.80   | 0.87 | 0.86 |
| 12 wks/<br>100,100 | 0.83             | 0.94   | 0.93   | 0.90   | 0.89   | 0.97   | 0.99 | 0.99 |
| 8 wks/<br>30,30    | 0.36             | 0.49   | 0.48   | 0.43   | 0.41   | 0.51   | 0.59 | 0.57 |
| 8 wks/<br>50,50    | 0.51             | 0.67   | 0.66   | 0.59   | 0.58   | 0.73   | 0.82 | 0.80 |
| 8 wks/<br>100,100  | 0.78             | 0.90   | 0.90   | 0.86   | 0.84   | 0.95   | 0.98 | 0.98 |

There are 20,000 simulations per design, all run using the grid. The Kruskal-Wallis test using the difference from the mean of the baseline measurements is best, regardless of design.

Obviously, a longer duration will achieve higher power, but one can investigate costs of additional patients versus additional time on trial among trials with adequate (eg 80%) power to determine the optimal combination of number of patients and time on trial. For example, one can investigate further designs to find how many patients are needed per arm in a 12-week study (obviously less than 50 per arm when the K-W difference test is used), and compare costs with the 8 weeks/ 50,50 design.

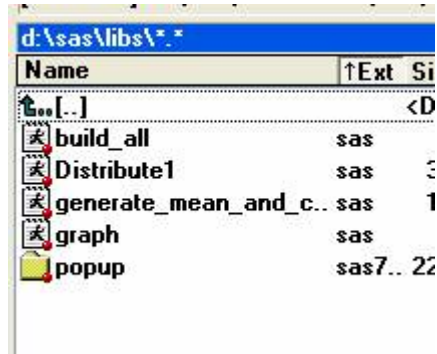
### III. Using the Clinical Trials Simulation Software

#### III.A. Installing and deploying the system

The system is available for download at the url:

[http://www.biopharmnet.com/innovation/trial\\_simulation/cts1.php](http://www.biopharmnet.com/innovation/trial_simulation/cts1.php).

III.A.1. Unpacking the files. Unzip the archived file to a directory (we will call it INTERFACE\_PATH from now on; in the screen shot below, INTERFACE\_PATH is d:\sas\local. Do not use a backslash after the path.)



III.A.2. Setting the path. Open build\_all.sas and set the macro variable path to INTERFACE\_PATH.

**%let path =D:\sas\local; ← has to be changed; no space before the semicolon; no backslash at the end**

```
proc build c=popup.popup batch;
```

```
...
```

III.A.3. Batch and interactive starting modes. First, create a SAS library called TRUELOCA that points to INTERFACE\_PATH. Then, two separate ways of continuing from this point on are provided:

a. Batch file that automatically starts a SAS session. (All other SAS sessions must be closed in order for the access to the SAS libraries to be unlocked)

a.4. In the INTERFACE\_PATH directory there are 2 files that need to be edited:

1. app.bat - sasroot needs to be set to the SAS installation directory (eg, C:\Program Files\SAS\SAS 9.1)
2. autoexec.cfg - the path parameter has to be updated to the location of the clinical trials system application directory (eg, d:\sas\libs)

a.5. Double click app.bat.

This way of running the application has the advantage that once the a.4. step has been completed once, the user only needs to apply step a.5.

b. SAS application that requires starting the SAS system (Interactive mode).

b.4. From within a SAS session, run build\_all.sas

b.5. Access the Popup catalog inside the TRUELOCA library; double click it.

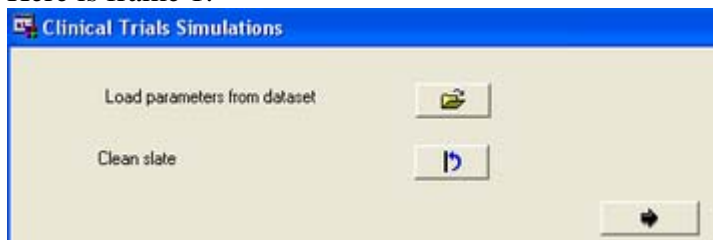
### III.B. Running the application

#### III.B.1. Starting the system.

(IMPORTANT: Close all the frames and scl files that you might have opened for editing. Save and close all of them before proceeding. Failing to do so, will cause your application not to run correctly.)

If the start is interactive, then run frame F1 by right-clicking the F1 frame and selecting “run,” and frame 1 below appears. If the batch start is used, frame 1 below appears automatically.

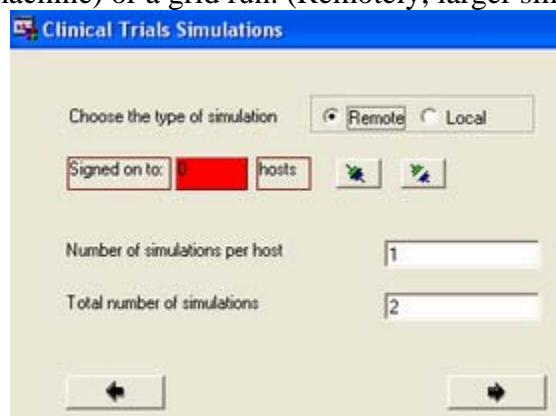
Here is frame 1:



The first frame is the place for the user to optionally upload an input parameters dataset, saved from a previous run. If the “Load parameters from dataset” button is pressed, the user is going to be presented with a file dialog, allowing them to choose the desired dataset. If the “Clean slate” button is pressed, there will be no parameters loaded and the user has to input all the parameters of the trial.

A dataset containing the input parameters may be saved after the application has been run, so that the user does not have to re-input all the trial parameters in subsequent runs.

III.B.2. Local or grid runs. The second frame is the place to choose between a local run (on the local machine) or a grid run. (Remotely, larger simulations can be run in shorter time)




For a remote (grid run), do the following:



- a. To run the file on the grid, you must first create (or have available) a SAS data set having the host computer names as well as their username and password logins. For

example,

```
data _hosts;  
  input host $7. username $10. password $11.;  
cards;  
bam237 ***** *****  
bam238 ***** *****  
...  
;
```

- b. Click the  icon and select the SAS data set containing the host computer names, userids, and passwords.

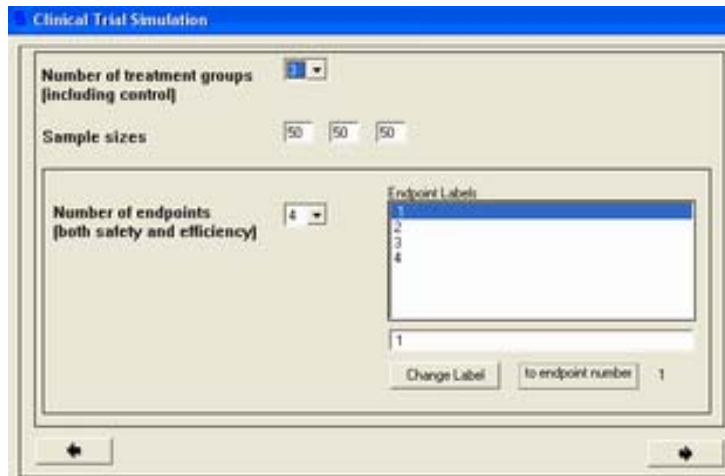
The “Number of simulations per host” should be large enough so that each run takes several seconds, perhaps as much as half a minute, on the host machines. The “Total number of simulations” should be large enough so that the success measures (usually binary proportions) are estimated with sufficient accuracy; the simulation standard error is given by  $(p(1-p)/(\text{Total number of simulations}))^{1/2}$ , where  $p$  is the success proportion. Some trial and error is often needed to see which combinations run the fastest. Contributing factors to speed (or lack thereof) are computer latency (smaller numbers of simulations per host mean greater latency) and size of data sets stored on host machines (larger numbers of simulations per host mean larger data sets). When data sets get too large, the time needed to process them increases at a higher rate than linear because there are sort operations within each run.

- c. If the user is signed on to the grid, but a remote run is desired using different machines, then the user should sign off using the  icon, then re-sign in using a different machine list by selecting the sign on icon .

For a local run, only the local computer will be used for the entire computation. In this case only the “Total number of simulations” is needed, and the “Number of simulations per host” is blanked out.

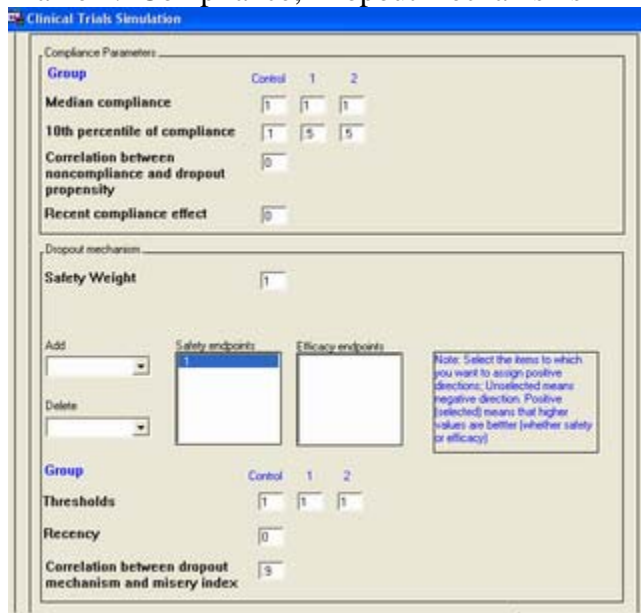
III.B.3. Input of clinical trial parameters. From this point, the user is asked to provide or modify simulation specific inputs. Details are contained in the technical documentation. Many of the fields may be left blank, and the system will supply default values when the arrow (forward or backward) button is pressed.

Frame 3: Treatment groups, Sample sizes, and endpoint labels.



This frame is largely self-explanatory, but it is worth noting that the endpoints can be given SAS labels in this frame, and future references to the endpoints will use these labels. Otherwise, the labels are generically assigned as “Endpoint1”, “Endpoint2” etc.

#### Frame 4: Compliance; Dropout mechanisms



Compliance is assumed to be on the 0 – 1 scale, with 1 denoted perfect, or 100% compliance. The first inputs are **median compliance** and **10<sup>th</sup> percentile of compliance**. Using say .9, and .3, respectively, half of the patients are 90% compliant or better, and 90% of the patients are 30% compliant or better. More details are contained in the technical report concerning generation of compliance data and effects of noncompliance. If you specify 1 for median compliance, then the system generates perfect (100%) compliance for all patients.

Noncompliance may be related to dropout propensity, if the **correlation** is high (eg .9), then noncompliance is closely related to the patient’s negative experience in the trial.

The “**Recent compliance effect**” is a number between 0 and 1 used to determine effect of noncompliance on patient outcome. If this value is “0”, then the effect of noncompliance is cumulative over the entire patient history. If the value is “1” the effect is completely determined by the most recent time interval. For values in between 0 and 1, the effect is weighted more heavily by recent compliance using exponential smoothing.

In the **Dropout Mechanism** box, the user decides which endpoints are involved in the index that determines patient dropout. The user can choose efficacy and safety endpoints, dropout is a function of both (see the technical report). Dropout rates are controlled by the **Thresholds**, which determine the proportion of patients staying per visit. The **Recency** parameter is an exponential smoothing value between 0 and 1 to determine how persistence of “misery” relates to dropout. If recency is 1, then dropout is determined solely by the most recent weeks experience. If recency is 0, then dropout is determined by cumulative misery. For values between 0 and 1, the more recent history of misery is weighted heavier, again using exponential smoothing.

Since some patients’ dropout is completely independent of their experience in the trial, the software also allows a completely random, non-informative dropout mechanism. The user can input the degree of relation of the misery index with the completely random mechanism in the field “**Correlation between dropout mechanism and misery index.**”

Finally, there are missing values that differ from dropout. A missing value is simply a missed visit; the patient will visit again in the future. The rate of such missing values can be input in the field “**Missing Value Rate.**”

Frame 5: Number of timepoints, endpoint and timepoint correlation data

|   | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 0 |
| 4 | 0 | 0 | 0 | 1 |

In this frame, the user specifies how many visits (**Number of visits** and **Visits**), and nodes at which to input the time-response functions for each dose (**Number of nodes**

for describing the response functions and Timevalues for describing the response functions). The latter two inputs may be chosen much smaller than the former to save time inputting complex functions if the piecewise linear interpolation used in the software is acceptable for defining time response functions.

The “**Include natural progression**” field is to be checked if a natural disease history will be input; in this case all patients will regress toward natural history when they are noncompliant. Otherwise they will regress toward placebo.

The “**Time Persistence**” is the first-order autocorrelation (from an AR(1) process) between time points in the subject-specific model. It must lie between -1 and 1.

The “**Subject Correlation**” parameter is the intraclass correlation between data values on a single patient for distant time lags (so carryover effects are gone, and all that is left is subject effect). This value lies between 0 (inclusive) and 1 (not inclusive).

The “**Correlation matrix**” is the correlation between endpoints at a given timepoint. A subtle but important detail is that it is the correlation between the underlying normal data, which are essentially latent when the variable’s distribution is chosen to be something other than normal. For example, if the endpoints are binary, then this matrix contains tetra choric correlations, which are known to be somewhat larger than the correlations between the raw binary data.

Next there are the “**Endpoint**” specifications. If you select one of them, you get a frame like the following:

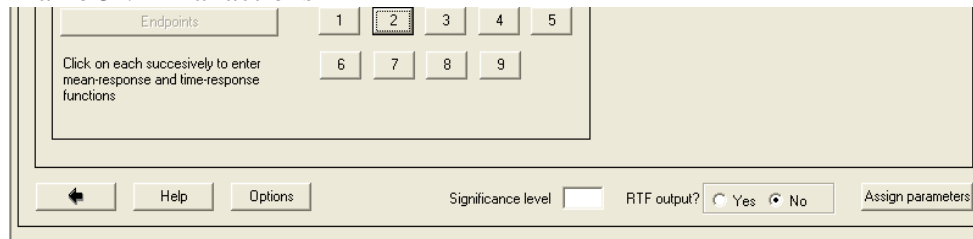
Frame 6: Endpoint specifications

The screenshot shows a software window titled "Endpoint2-SecEff1". At the top, there is a dropdown menu for "Survival" with a list of options: Normal, Binary, Ordinal, Mixture, Lognormal, and Survival. Below this is a row of 13 input boxes labeled 0 through 12. Underneath, there is a section titled "Input the mean values". It contains two rows of input boxes. The first row is labeled "Control" and has 13 boxes, all containing the value "0". The second row is labeled "Group1" and has 13 boxes containing the values: 0, .3, .3, .3, .3, .3, .4, .5, .6, .7, .8, .9, .9.

Here, you can enter the distribution type, as well as the time-and-dose response functions for that variable. There are specific differences corresponding to different distribution types; see the technical report.

After selecting distributions and time-and-dose response functions for all endpoints, the user returns to frame 5. At the bottom there are the final instructions as shown below:

#### Frame 5': Final actions



The screenshot shows a graphical user interface for Frame 5'. It features a sequence of buttons labeled 1 through 9. Button 2 is highlighted. A text box on the left contains the instruction: "Click on each successively to enter mean-response and time-response functions". At the bottom of the interface, there are several controls: a back arrow button, a "Help" button, an "Options" button, a "Significance level" dropdown menu, an "RTF output?" section with radio buttons for "Yes" and "No" (where "No" is selected), and an "Assign parameters" button.

The “significance level” will be set to the common .05 value by default, but the user can select other values as well. Based on the “**Options**” selected, the user can either simply “Assign parameters” at this point, and then use the interface for local job runs, or the user can have the system “**Execute**”, in which case the default analyses will be performed on the generated data sets.

### III.C. Standard statistics collected

If you run the job locally, the system produces two main data sets: Work.Observed, and Work.Observed\_locf. Both are data sets in “multivariate” form, with self-explanatory endpoint and timepoint variable names, along with treatment group indicator and simulation number indicator. The simulation analyses can then be performed using SAS software with “BY” variable processing. For example, one can analyze the data using PROC MIXED to perform longitudinal or Bayesian analyses; or one can use PROC MULTTEST to perform bootstrap and Resampling style analyses, both with the simulation BY variable. The data set “Work. observed” contains all the missing values, whereas “Work. observed\_locf” has missing values imputed using “last observation carried forward.”

For grid jobs, the data sets are collated over simulation runs to avoid storage problems. In this case the data sets are over-written, but summary means and p-values are collected and passed back to the client. These data sets are called “Work.sample\_means” and “Work-sample”, respectively.

When the “**Rtf output**” is selected, a report is created after the program execution. This report contains the summary of input parameters selected in the various frames, graphs of means and dropout functions, and summary rejection proportions for the various tests. The variable names are chosen to be reasonably self-explanatory, but can be discerned more specifically from the file “third.sas” which is included in the interface, and which file can be modified to collect customized statistics.

### III.D. Modifying the SAS/AF graphical interface

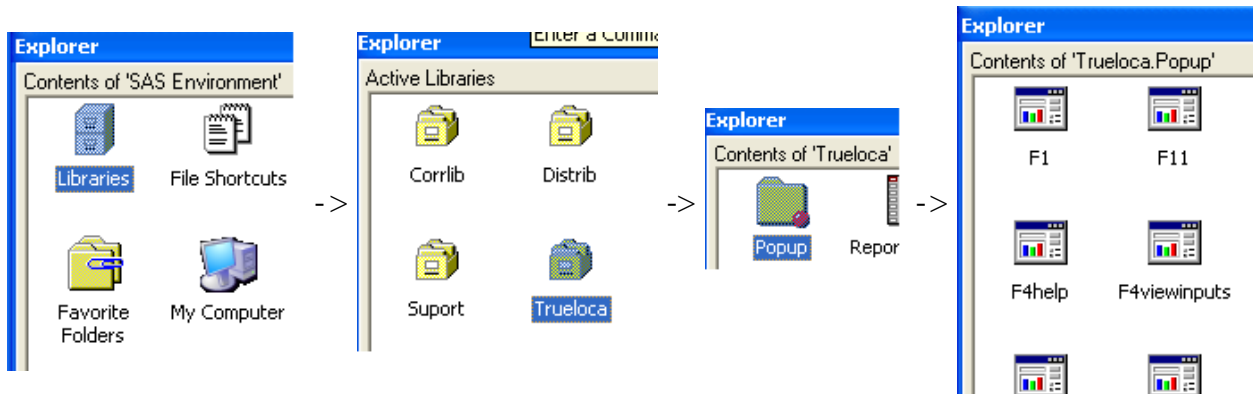
The following instructions assume that SAS/AF is licensed on your computer.

Let us consider the following example:


Add code that will display a message in the log file when we press the “Clean Slate” button on the F1 frame.

The AF code is split in separate files, each one corresponding to a GUI frame.

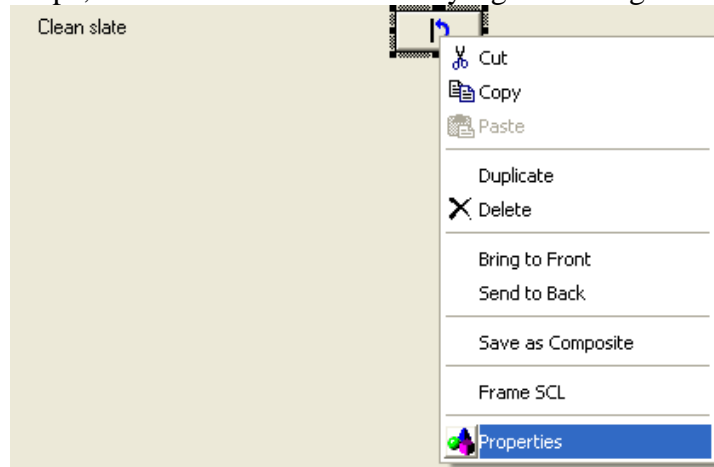
III.D.1. Accessing the SCL code. In order to access the underlying frame code, several steps need to be followed:



Once the contents of the Trueloca.popup directory are displayed in the Explorer window, there are two ways to access the SCL instructions:

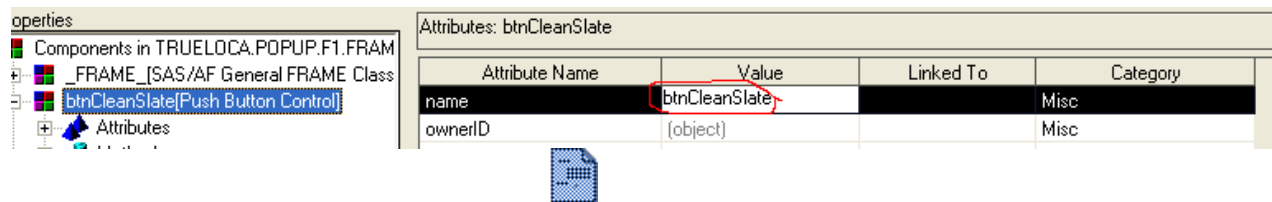
- a. Double-click on the F1 frame  .


Once at this point, the user can manipulate the frame controls: change their position, shape, add more or delete them. By right-clicking a component and selecting properties,



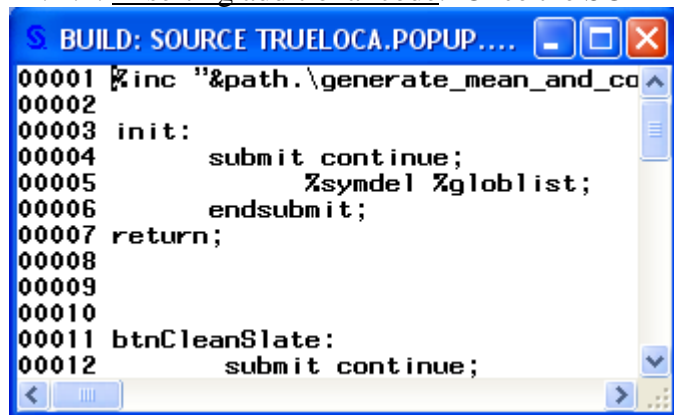
we can access and modify the properties of the selected component.

For example, by inspecting the Properties window for the button in question, we can find out that its name is “btnCleanSlate”:



b. Double click on the F1 SCL file  .

III.D.2. Inserting additional code. Once the SCL file is open,



we can search for the btnCleanSlate entry and insert additional code

```

btnCleanSlate:
    submit continue;
        %syndel %globlist;
    endsubmit;

```

```

/*NEW CDDE*/
%put ***NEW CODE***;
/*NEW CDDE*/

```

```
return;
```

that will be executed when we press the cleanSlate button.

III.D.3. Recompile and run. The last step is to recompile the F1 frame separately or run the build\_all.sas file.

The next time we run the application and press the cleanSlate button a message will be displayed in the Log window: \*\*\*NEW CODE\*\*\*.

The SUGI paper entitled "Developing SAS/AF(r) Applications Made Easy" is an introduction to SAS/AF Software. A copy of this paper can be viewed from the following URL:

<http://www2.sas.com/proceedings/sugi28/027-28.pdf>

## IV. Customizing the Program to Collect Additional Statistics

### IV.A. Code locations, code samples, and file names

| Level  | Code location                        | Code sample  | Explanation   |
|--|--------------------------------------|--|---|
| <b>Host level</b> – for every chunk of simulations | 1<br>third.sas<br>locf macro         | <pre> %macro locf; data observed_locf;   set observed;   %do i = 1 %to &amp;nendpoints;     last=.;   .... %mend; %locf; </pre>  | <p>Every host creates its own Observed_locf; This dataset contains the generated information for a chunk of simulations. (the size of the chunk can be adjusted by modifying the “Number of simulations per host” parameter.)</p> <p>It is the starting point for gathering statistics.</p> |
|  | 2<br>third.sas<br>report_stats macro | <pre> create table isample_means&amp;i as select group, avg(patient_dropout='DROPOUT') as dropout, %do i_iter_i=1 %to 3;   avg(Endpoint&amp;i._0&amp;i_iter_i ) as Endpoint_m_&amp;i._0&amp;i_iter_i ,   var(Endpoint&amp;i._0&amp;i_iter_i) as Endpoint_v_&amp;i._0&amp;i_iter_i , %end;  %do iter=1 %to &amp;n_visits;   avg(Endpoint&amp;i._&amp;&amp;visit&amp;iter ) as Endpoint_m_&amp;i._&amp;&amp;visit&amp;iter,   var(Endpoint&amp;i._&amp;&amp;visit&amp;iter ) as Endpoint_v_&amp;i._&amp;&amp;visit&amp;iter   %if &amp;iter &lt; &amp;n_visits %then , ; %end; from observed_locf group by group; </pre> | <p>Using an SQL statement, and the AVG SAS library function, the variables are selected and processed</p> <p>New datasets, isample_means&amp;i, each corresponding to a different endpoint, are created.</p>  |
|  | 3<br>third.sas<br>report_stats macro | <pre> data isample_means; merge isample_means isample_means&amp;i medians&amp;i; run; </pre>   | <p>The isample_means dataset merges all the partial isample_means&amp;i together.</p>   |
|  | 4<br>Third.sas<br>TaskRSub macro     | <pre> %macro TaskRSub;   %execute;   data Sample; set Sample iSample; run; </pre>  | <p>The sample_means merges all the partial isample_means.</p>   |

|                          |   |            |   |   |
|--------------------------|---|------------|---|---|
| <b>Local<br/>machine</b> | 5 | F4.scl     | <pre> submit continue; ...  %let ssel=CurrentParams; %save_macros;  %let Dimen = 10; %Distribute; %RCollect(Sample,Sample ); %RCollect(Sample_Means, Sample_Means ); %RCollect(Survival_All, Survival_All ); </pre>   | <p>The sample_means datasets from all of the hosts are collected on the local machine and merged into one. At the end of the simulation there will be a Sample_means dataset in the Work library.</p> |
|                          | 6 | Report.sas | <pre> create table report_means as select group, %do i=1 %to &amp;nendpoints; %do i_iter_i=1 %to 3;   avg( Endpoint_m_&amp;i._0&amp;i_iter_i ) as Endpoint_m_&amp;i._0&amp;i_iter_i ,   avg( Endpoint_med_&amp;i._0&amp;i_iter_i ) as Endpoint_med_&amp;i._0&amp;i_iter_i , %end; %do iter=1 %to &amp;n_visits;   avg(Endpoint_m_&amp;i._&amp;&amp;visit&amp;iter ) as Endpoint_m_&amp;i._&amp;&amp;visit&amp;iter ,   avg(Endpoint_med_&amp;i._&amp;&amp;visit&amp;iter ) as Endpoint_med_&amp;i._&amp;&amp;visit&amp;iter %if &amp;iter &lt; &amp;n_visits %then , ; %end; %if &amp;i &lt; &amp;nendpoints %then , ; %end; from %if "%trim(%left(&amp;RLS))"="Remote" %then sample_means; %else isample_means; group by group; </pre> | <p>The Sample_means dataset is processed and graphs and summary data are created as an rtf output.</p>  |

#### IV.B. General instructions

In order to collect more statistics one would have to add more code corresponding to steps 2-5 in the above table. There are two ways of achieving this:

- create, using a method that would best suit the purpose – sql statement, data step, sas procedure – a dataset DS&i , where the i index stands for the endpoint’s index;
- **alternatively you can create new variables in the isample/isample\_means dataset;** all the other steps below are already taken care of and the extra variable will be found locally, at the end of the run, in the Sample or Sample\_means dataset.

This code will be added below the code in step 2.

- merge all the DS&i smaller datasets into one DS dataset that contains the information for all datasets.

This code will be added below the code in step 3.

- create a DS\_ALL dataset in the Third.sas file and use it to merge each of the chunks’ DS datasets into it.

This code will be added below the code in step 4.

- add an Rcollect(DS\_ALL, DS\_ALL) statement to download all the DS\_ALL datasets from the remote hosts on the local machine

#### IV.C. Example: From II.A. (Sample size allocation)

Here we consider a grid application using the multiple group ACR20 input. This corresponds to section II.A., “Sample Size Allocation.”

These changes allow the application to collect pairwise Chi-Square p-values for comparisons with a control.

IV.C.1. Load the parameters. Assume the settings file is ACR20\_pres.sas7bdat, the actual file can be obtained from the web page [http://westfall.ba.ttu.edu/acr20\\_pres.sas7bdat](http://westfall.ba.ttu.edu/acr20_pres.sas7bdat) .

IV.C.2. Modify the code. The file “third.sas” (included with the zip file containing the system) must be modified to collect statistics specific to the settings in ACR20\_pres.sas7bdat. See changes in bold.

a. Inside the “report\_stats” macro, initialize the new data set:

```
%macro report_stats;
/*proc freq data=observed_locf;
  tables patient_dropout*group/norow nopercnt;
run;
*/
data jttest; if (0); run;
data jttest_diff_m; if (0); run;
data jttest_diff_md; if (0); run;
data aov; if (0); run;
data ANCOV_M; if (0); run;
data ANCOV_MD; if (0); run;
data diff_m; if (0); run;
data diff_md; if (0); run;
data dunnett; if (0); run;
data chisq_c; if (0); run; ← Here
data isample; if (0); run;
```

b. Collect new pairwise Chi-Square statistics:

```
proc freq data=observed_locf noprint;
  tables Diff_Base_Median&i*group/jt;
  output out=jttest_diff_md1(rename=(p2_jt=P2_jt_diff_md_E&i) keep=simu P2_jt) jt;
  by simu;
run; quit;

ods listing close; ← Starting Here.
```

**/\* The following code is not general. It is only for this application, using  
“Endpoint1\_1” instead of a generic macro term. \*/**

```
proc freq data=observed_locf(where=(group in (1,2)));  
  tables Endpoint1_1*group/chisq;  
  ods output chisq=chisq_l(rename=(prob=p_l));  
  by simu;  
run;  
proc freq data=observed_locf(where=(group in (1,3)));  
  tables Endpoint1_1*group/chisq;  
  ods output chisq=chisq_m(rename=(prob=p_m));  
  by simu;  
run;  
proc freq data=observed_locf(where=(group in (1,4)));  
  tables Endpoint1_1*group/chisq;  
  ods output chisq=chisq_h(rename=(prob=p_h));  
  by simu;  
run;  
ods listing;
```

```
data chisq_c1;  
  merge chisq_l chisq_m chisq_h;  
  if statistic="Chi-Square";  
  keep simu p_l p_m p_h;  
run; ← To Here
```

```
proc glm noprint outstat=AOV1(where=( _TYPE_="SS3")  
rename=(PROB=P_AOV_E&i) keep= simu Prob _TYPE_)  
  data=observed_locf;  
  class group;  
  model Endpoint&i._&&visit&n_visits = group;  
  by simu;  
run; quit;
```

c. Collate data to be collected by the %Distribute macro:

```
/*  
%if %eval(&i-1)=0 %then %do;  
data jttest; set jttest1; run;  
data AOV; set AOV1; run;  
data ANCOV_M; set ANCOV_M1; run;  
data ANCOV_MD; set ANCOV_MD1; run; %end;  
  
%else %do; */
```

```

data jttest; merge jttest jttest1; run;
data jttest_diff_m; merge jttest_diff_m jttest_diff_m1; run;
data jttest_diff_md; merge jttest_diff_md jttest_diff_md1; run;
data AOV; merge AOV AOV1; run;
data ANCOV_M; merge ANCOV_M ANCOV_M1; run;
data ANCOV_MD; merge ANCOV_MD ANCOV_MD1; run;
data dunnett; merge dunnett dunnett1; run;
data diff_m; merge diff_m diff_m1; run;
data chisq_c; merge chisq_c chisq_c1; run; ← Here
data diff_md; merge diff_md diff_md1; run;
%end;

data isample(drop=test effect _type_ _source_ _name_ _label_);
merge isample jttest jttest_diff_m jttest_diff_md aov
      ancov_M ancov_md diff_m diff_md dunnett chisq_c; run; ← Here

```

IV.C.3. Data analysis. After the grid run, there is a file called “work.sample” on the client machine. This file contains many p-values from various tests that are computed automatically, but also includes the new p-values called p\_l, p\_m, and p\_h. The following code computes the power of the tests as well as the power of the fixed-sequence tests. This code is to be run separate from the interface; ie, paste it into the program editor window and run it after the grid job is done.

```

proc sql;
  title "prop. significant for Comparisons with a control";
  title2 " Sample Sizes &Samplesizes ";
  select
mean(p_h<=.05) as PowHigh,

mean((p_m<=.05)) as PowMed,
mean((p_m<=.05)*(p_h<=.05)) as PowMed_s,

mean((p_l<=.05)) as PowLow
mean((p_m<=.05)*(p_h<=.05)*(p_l<=.05)) as PowLow_s,

from sample      ;

quit;

```

The interface must be re-compiled for these changes to take effect.

#### IV.D. Example: From II.B. (Choice of statistical test)

Here we consider the choice of statistical test: ACR20 or O'Brien?

IV.D.1. Input the simulation settings via the interface. This will set the values of the macro variables. The simulation settings are given in Anderson, Bolognese, and Felson (2003), "Comparison of Rheumatoid Arthritis Clinical Trial Outcome Measures," *Arthritis and Rheumatism* 48, 3031-3038. The resulting SAS file containing all these input parameters is [http://westfall.ba.ttu.edu/ra\\_anderson\\_et\\_al.sas7bdat](http://westfall.ba.ttu.edu/ra_anderson_et_al.sas7bdat).

IV.D.2. Identify variable names. Run the file called "local\_run\_after\_macro\_assign.sas" provided with the interface. This file uses the specific values of the macro variables input via the simulation to create simulated clinical trials data sets. The data sets are in work.observed and work.observed\_locf. You can also see the variable names that you need to use in the hard code.

IV.D.3. Modify the code. Edit the code in "local\_run\_after\_macro\_assign.sas" as needed. For example, using the input values from the Anderson et al study one can create the variables ACR20 by adding the following code to the first "observed\_locf" data step within the "locf" macro:

```
pch1=diff_base_mean1/baseline_Mean1;
pch2=diff_base_mean2/baseline_Mean2;
pch3=diff_base_mean3/baseline_Mean3;
pch4=diff_base_mean4/baseline_Mean4;
pch5=diff_base_mean5/baseline_Mean5;
pch6=diff_base_mean6/baseline_Mean6;
pch7=diff_base_mean7/baseline_Mean7;
acr20 = (pch1>=.2)*(pch2>=.2)*
( ((pch3>=.2)+(pch4>=.2)+(pch5>=.2)+(pch6>=.2)+(pch7>=.2)) >2 );
keep group simu patient_dropout acr20;
```

Also, the following code was used to create the data needed for the O'Brien test; this is inserted immediately after the "Proc sort" following the locf macro:

```
proc rank data=observed_locf out=observed_locf;
  var Diff_Base_Mean1-Diff_Base_Mean7 ;
  ranks rDiff_Base_Mean1-rDiff_Base_Mean7 ;
  by simu;
run;

data observed_locf;
  set observed_locf;
  OBrien = sum(of rDiff_Base_Mean1-rDiff_Base_Mean7) ;
run;
```

Finally, to obtain power for ACR20 tests (Chi-Square test) and O'Brien (F test) the following code was added at the end of the file where statistics are collected:

```
proc freq data=observed_locf noprint;
  tables acr20*group/chisq;
  output out=acr20(rename=(p_pchi=p_acr20) keep=simu p_pchi) chisq;
  by simu;
run; quit;

proc sql;
select mean(p_acr20 <=.05) as power_acr20 from acr20;
quit;

proc glm noprint outstat=OBrien(where=( _TYPE_="SS3") rename=(PROB=P_OBRIEN) keep=
simu Prob _TYPE_)
data=observed_locf;
class group;
model OBrien = group;
by simu;
run; quit;

proc sql;
select mean(p_OBRIEN <=.05) as power_OBRIEN from OBrien;
quit;
```

#### IV.E. Example: From II.C. (Choice of duration of study)

No code modification is needed, but analyses different from the default analyses are performed on the collected statistics.

This example is described in the technical document, having a heavy tailed primary endpoint and multiple secondary endpoints including a survival secondary endpoint.

The simulated data sets are huge, so a grid run is needed. After the grid run, the p-values from various tests are automatically collated into the client machine in a file called "work.sample".

The p-values are processed to obtain the summary stats shown in the paper as follows:

```
/* The following are reported automatically, but this
   code puts all the results in the same place */
proc sql;
  select mean(P_AOV_E1<=.05) as AOV,
         mean(p2_jt_E1<=.05) as KW,
         mean(P2_jt_diff_m_E1<=.05) as KW_diffm,
         mean(P2_jt_diff_md_E1<=.05) as KW_diffmd,
         mean(P_ancov_m_E1<=.05) as ancov_m,
```

```

        mean(P_ancov_md_E1<=.05) as ancov_md,
        mean(P_diff_m_E1<=.05) as aov_diff_m,
        mean(P_diff_md_E1<=.05) as aov_diff_md
    from sample; quit;

/* The following code is used to (a) subset the data upon a significant
primary analysis, and (b) compute the Hochberg adjusted p-values, and (c)
summarize significance of the Hochberg-adjusted results, given that the
primary endpoint was significant. */

data trym;
    set sample;
    if p2_jt_diff_m_e1<=.05;
    p1 = p_surv_E2;
    p2 = p2_jt_E3;
    p3 = p2_jt_E4;
    pmin = min(of p1-p3);
    pmax =max(of p1-p3);
    pmed = median(of p1-p3);
    apmin = 3*pmin;
    apmed = 2*pmed;
    apmax = pmax;
    if apmax < apmed then apmed = apmax;
    if apmed < apmin then apmin = apmed;
    if pmin = p1 then ap1 = apmin;
    if pmed = p1 then ap1 = apmed;
    if pmax = p1 then ap1 = apmax;
    if pmin = p2 then ap2 = apmin;
    if pmed = p2 then ap2 = apmed;
    if pmax = p2 then ap2 = apmax;
    if pmin = p3 then ap3 = apmin;
    if pmed = p3 then ap3 = apmed;
    if pmax = p3 then ap3 = apmax;
    Elsig = (ap1<=.05);
    E2sig = (ap2<=.05);
    E3sig = (ap3<=.05);
    anysig = (Elsig+E2sig+E3Sig>0);
    allsig = Elsig*E2sig*E3sig; run;

proc means;
    var Elsig E2Sig E3sig anysig allsig;
run;

```

## V. Technical Report Containing Mathematical Details

The following document contains technical details concerning the model. The formal reference is as follows:

Westfall P.H., Tsai K., Ogenstad S., Tomoiaga A., Moseley S., and Lu Y. (2008). Clinical Trials Simulation: A Statistical Approach. *Journal of Biopharmaceutical Statistics* 18, 611-630.

# **Clinical Trials Simulation: A Statistical Approach**

**Peter H. Westfall<sup>1</sup>, Kuenhi Tsai<sup>4</sup>, Stephan Ogenstad<sup>3</sup>, Alin**

**Tomoiaga<sup>1</sup>, Scott Mosely<sup>2</sup>, Yonggang Lu<sup>1</sup>**

**<sup>1</sup>Texas Tech University**

**<sup>2</sup>Vertex Pharmaceuticals**

**<sup>3</sup>Statogen Consulting, LLC**

**<sup>4</sup>Merck & Co.**

## **Abstract**

A generic template for clinical trials simulations that are typically required by statisticians is developed. Realistic clinical trials data sets are created using a unifying model that allows general correlation structures for endpoint\*timepoint data and nonnormal distributions (including time-to-event), and computationally efficient algorithms are presented. The model allows for patient dropout and noncompliance. A grid-enabled SAS-based system has been developed to implement this model; details are presented summarizing the system development. An example illustrating use of the system is given.

## **1 Introduction**

With increasing costs and intensified competition in the pharmaceutical industry, there is ever-growing interest in the optimization of clinical trial design.

Because trials and data resulting therefrom are often too complex to allow simple decision-theoretic solutions, interest in clinical trials simulation (CTS) has recently exploded in popularity among statisticians, clinicians, and pharmacokineticists. CTS involves drug/disease, trial design, and probabilistic data models, utilizing pharmaco-statistical methods. In addition to the optimization of trial design, applications include protocol optimization (e.g., choice of optimal models and test statistics), estimation of operating characteristics of nonstandard and computationally intensive procedures (including Bayesian and adaptive designs), and development of "mock up" trials for training review committees. Peck, Rubin and Shiner (2003) suggest that CTS might (in some cases) replace the second Phase III trial, so that only a single trial is necessary.

Figure 1 displays the essential idea of what we mean when we refer to CTS; others may emphasize different aspects. Often, "variations in study design" refers simply to different sample sizes, but the idea is much broader, encompassing length of trial, measurement of endpoints (continuous, time-to-event, categorized, binary), and analysis methods (baseline covariate-adjusted vs. percentage change, use of compliance data as covariates, parametric vs. nonparametric etc.). Similarly, while "probability of successful outcome" often means "power," the possibilities are much broader, encompassing combination rules involving both safety and efficacy, or complex rules like "3 out of 4 significant" for multiple co-primary endpoints (U.S. Department of Health and Human Services, Food and Drug Administration, 1999, p. 3), and rules that include economic considerations (Poland and Wadda, 2001), and rules involving patient quality

of life. In the more general case, the vertical axis of Figure will be replaced by "Expected Benefit."

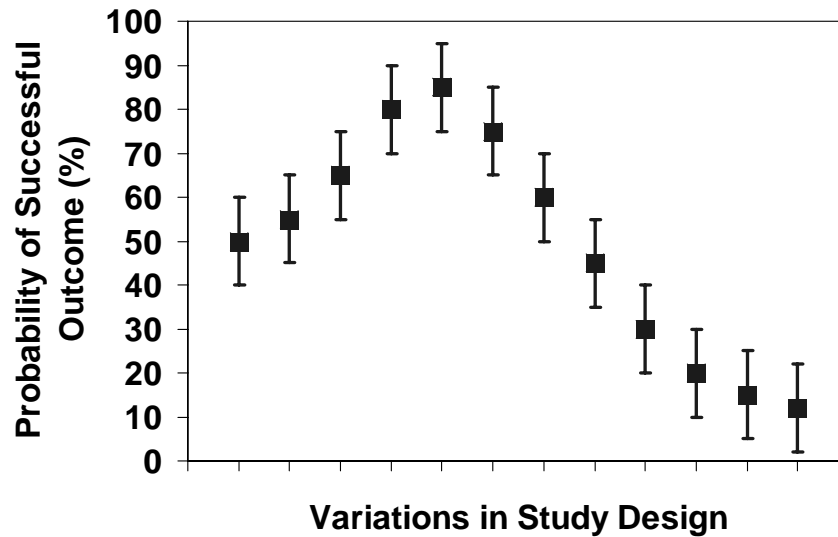


Figure 1: Simulation Methodology to Optimize Trial Design

Some commercial software (e.g., Pharsight, <http://www.pharsight.com/main.php>) require pharmacokinetic/pharmacodynamic (PK/PD) inputs as drivers for the simulation output. Several references on clinical trials simulation using PK/PD exist, see Kimko and Duffull (2003) for an overview and further cites. We have taken an alternative, more "statistical" approach for simplicity, to produce data with realistic characteristics that are useful for decisions that statisticians typically must make. An example of a similarly "statistical" approach is given by Anderson, Bolognese, and Felson (2003). Our model starts with a rich probabilistic structure to account for typical scenarios, using historical data where

possible to validate the inputs and outputs, with specific emphasis on the parsimonious yet flexible input of correlation structures. The output data sets are massive, and the analyses are allowed to be computationally challenging, often requiring grid computing for feasibility.

Our framework for multivariate simulation is reasonably simple to code using a variety of software, yet flexible, retaining the realism the doubly-multivariate endpoint/timepoint correlation structures, informative dropout mechanisms, non-normal distributions, survival endpoints, and noncompliance effects. This research is based on the development of a real, currently existing CTS software system, and simplifying, albeit somewhat questionable assumptions are made at various places. Such assumptions reflect a necessary trade-off between ease of use of the system on one hand, and realism and flexibility of its outputs on the other hand. The simulation algorithms are not entirely new, but we hope that pharmaceutical statisticians will find it useful to have them all in one place, for convenient reference. We also hope that the models we present for compliance and dropout effects will stimulate research into this area, whether to instantiate the models we suggest using parameter estimates from designed experiments, or to propose alternative models that may be more appropriate for specific diseases. Section 2 provides an overview of the system, Sections 3–6 provide technical details regarding data generation and inputs, Section 7 provides an application, and Section 8 concludes.

# Overview of System

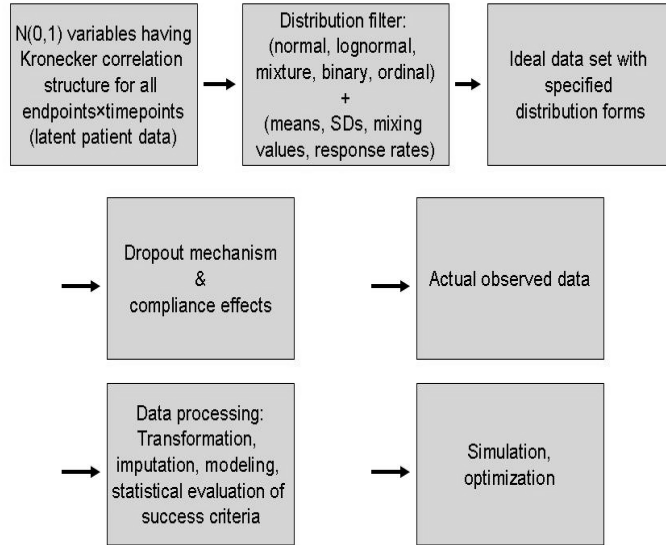


Figure 2: Overview of Clinical Trials Simulation System

## 2 Overview of a Statistical Clinical Trials Simulation System

An overview of the system is shown in Figure 2.

As shown in Figure 2, patient responses are functions of underlying correlated  $N(0,1)$  clinical quantities. All distributional forms, dropout and non-compliance effects, are determined from these underlying values. Evaluation of trial success then follows from the analysis of the simulated data sets. Details follow.

### 3 The Fundamental Correlated Quantities

Our goal is to generate realistic data sets having typical correlation structures for multiple endpoint/timepoint data with  $p$  endpoints (safety, efficacy or both) indexed by  $j = 1, \dots, p$ , and  $T + 1$  timepoints indexed by  $t = 0, \dots, T$  ( $t = 0$  denotes baseline). To start, we generate for "patient  $i$ " a  $p(T + 1)$  -vector of correlated  $N(0,1)$  variates  $Z_{ijt}$ , each of which which may be thought of as a latent indicator of the patient's health relative to a population of similar patients, for endpoint  $j$  and timepoint  $t$  (see the upper left panel of Figure 2). Observations  $Z_{ijt}$  and  $Z_{i'j't'}$  will always be considered independent when  $i \neq i'$ ; random center effects violate the assumption of independence between patients; it is possible to include such effects, but this is not pursued here. To simplify notation, we frequently drop subscripts; meanings will be clear from context.

In some cases, code for simulating the data  $Z_{ijt}$  can utilize a high-level matrix language such as MAPLE or SAS/IML. Our intent is to avoid such software for two reasons: First, it is desirable that the simulation system be entirely self-contained within a single software system, from data generation through analysis, SAS in our case. However, in our experience, the SAS/IML component is frequently not licensed, thus there is a need to keep the code at as "native" a level as possible, using SAS/BASE and SAS/STAT only. Second, we exploit special structures to obtain more efficient algorithms by operating at a more native level. Nevertheless, we also provide instructions for use with matrix-intensive software.

### 3.1 Repeated Measures Data for Patient\*Endpoint

For a given patient and endpoint, the timepoint data  $Z_0, \dots, Z_T$  are correlated because of subject and carryover effects. Frison and Pocock (1992) argue for generic use of the compound symmetry (CS) covariance structure, which accommodates subject effects only, and not time-series carryover effects, but also note "allowance in design for alternative non-equal correlation structures can and should be made when necessary." The CS model can be expanded easily to accommodate time-series carryover effects in addition to subject effects as

$$Z_t = \theta^{1/2}S + (1 - \theta)^{1/2}\epsilon_t, \quad (1)$$

where  $S \sim N(0, 1)$  is the subject effect and  $\epsilon_0, \dots, \epsilon_T$  is a realization of a unit variance AR(1) process with parameter  $\rho$ . Since repeated-measures software such as PROC MIXED of SAS/STAT have become available, the routine use of correlation structures other than CS is now commonplace. The model can be estimated using existing data using *both* the 'RANDOM' and 'REPEATED' statements:

```
random subject_id; repeated /subject=subject_id type=ar(1);
```

For simulation purposes, the parameters  $\theta$  and  $\rho$  must be specified (typically suggested by early Phase data and/or similar studies); it may be helpful to note that  $\theta$  is the within-subject correlation for large time lags. The data may be generated easily using (1), where the  $\{\epsilon_t\}$  are generated recursively as

$$\epsilon_t = \rho\epsilon_{t-1} + (1 - \rho^2)^{1/2}u_t, \quad (2)$$

with the variates  $S, \epsilon_{-1}, u_0, \dots, u_T$  generated as i.i.d.  $N(0, 1)$ . The resulting covariance structure within patient\*endpoint combination is

$$\text{Cov}(Z_0, \dots, Z_T) = \mathbf{\Sigma} = \theta \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} + (1 - \theta) \begin{bmatrix} 1 & \rho & \dots & \rho^T \\ \rho & 1 & \dots & \rho^{T-1} \\ \vdots & \vdots & \ddots & \vdots \\ \rho^T & \rho^{T-1} & \dots & 1 \end{bmatrix}. \quad (3)$$

It is also possible to generate  $\mathbf{Z}' = (Z_0, \dots, Z_T)$  using matrix decomposition. Factoring  $\mathbf{\Sigma}$  as  $\mathbf{\Sigma} = \mathbf{U}'\mathbf{U}$  (e.g., using the Cholesky decomposition available in matrix languages such as SAS/IML, or using outputs from PROC PRINCOMP of SAS/STAT), and letting  $\mathbf{W} = \{W_j\}$  denote a  $(T + 1)$ -vector of independent  $N(0, 1)$  variates, we may simply take  $\mathbf{Z} = \mathbf{U}'\mathbf{W}$ . However, it is worth noting that (1) and (2) are computationally more efficient and do not require specialized matrix functions.

### 3.2 Multiple Endpoint Data for Patient\*Timepoint

For a given patient and timepoint  $t$ , the correlation between endpoints  $Z_{1t}, \dots, Z_{pt}$  is best left unstructured (UN), rather than assumed as CS, AR(1) or some other form. For estimation purposes it is often desirable to parameterize a UN correlation using a more parsimonious form such as factor-analytic approximation (e.g. FA(1) or FA(2) in PROC MIXED). However, the actual endpoint correlation matrix (e.g., from early phase or similar studies) is a more convenient input to the system. Denote the covariance matrix for the multiple endpoints

as

$$\text{Cov}(Z_{1t}, \dots, Z_{pt}) = \mathbf{\Gamma} = \begin{bmatrix} 1 & \gamma_{12} & \cdots & \gamma_{1p} \\ \gamma_{12} & 1 & \cdots & \gamma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{1p} & \gamma_{2p} & \cdots & 1 \end{bmatrix};$$

like the parameters  $\theta$  and  $\rho$ , the correlations  $\{\gamma_{ij}\}$  are system inputs. For now, the basic quantities  $Z_{jt}$  have unit variance, so correlation equals covariance.

### 3.3 Correlation Structure for all Within-Patient Data

Observations between endpoints at different timepoints are correlated. There are number of possibilities for defining this structure, the most convenient and common is the Kronecker product model used in multivariate longitudinal models (Gao et al, 2006). This model implies that the covariance matrix of the entire set of  $p(T + 1)$  endpoint and timepoint measurements for a given patient is

$$\text{Cov}(\mathbf{Z}'_1, \dots, \mathbf{Z}'_p) = \mathbf{\Gamma} \otimes \mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma} & \gamma_{12}\mathbf{\Sigma} & \cdots & \gamma_{1p}\mathbf{\Sigma} \\ \gamma_{12}\mathbf{\Sigma} & \mathbf{\Sigma} & \cdots & \gamma_{2p}\mathbf{\Sigma} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{1p}\mathbf{\Sigma} & \gamma_{2p}\mathbf{\Sigma} & \cdots & \mathbf{\Sigma} \end{bmatrix}, \quad (4)$$

where  $\mathbf{Z}'_j = (Z_{j0}, \dots, Z_{jT})$ , so that  $(\mathbf{Z}'_1, \dots, \mathbf{Z}'_p) = (Z_{10}, \dots, Z_{1T}, \dots, Z_{p0}, \dots, Z_{pT})$ .

As noted above following (3),  $(\mathbf{Z}'_1, \dots, \mathbf{Z}'_p)$  may be generated using the Cholesky decomposition: generate a  $p(T + 1)$ -vector  $\mathbf{W}$  of independent  $N(0, 1)$  variates, decompose  $\mathbf{B} = \mathbf{\Gamma} \otimes \mathbf{\Sigma}$  as  $\mathbf{B} = \mathbf{U}'\mathbf{U}$  and let  $(\mathbf{Z}'_1, \dots, \mathbf{Z}'_p) = \mathbf{W}'\mathbf{U}$ .

However, the special structure of  $\mathbf{B}$  can be exploited, giving a more efficient algorithm. First, generate  $p$  independent vectors  $\mathbf{V}'_i = (V_{i0}, \dots, V_{iT})$  having

covariance matrix from (3) using (1) and (2). Then factor  $\mathbf{\Gamma}$  as  $\mathbf{\Gamma} = \mathbf{U}'\mathbf{U}$ .

Taking

$$\begin{aligned}\mathbf{Z}_1 &= u_{11}\mathbf{V}_1 + \cdots + u_{p1}\mathbf{V}_p \\ &\vdots \\ \mathbf{Z}_p &= u_{1p}\mathbf{V}_1 + \cdots + u_{pp}\mathbf{V}_p\end{aligned}$$

gives basic variables with the desired covariance structure.

An interesting fact about covariance structure (4) is that the correlation between variables at any time point is identical to the correlation between difference scores; specifically;  $Corr(Z_{jt}, Z_{j't}) = Corr(Z_{jt} - Z_{j0}, Z_{j't} - Z_{j'0}) = \gamma_{j,j'}$ . This fact can be helpful when deciding values of the  $\gamma_{j,j'}$ , since it does not matter whether the raw variables or difference scores are considered.

## 4 Mean Structures

Our recommendation is to specify, as inputs, mean structures for the different endpoint\*timepoint\*treatment combinations. Such structures can be determined purely *a priori*, from earlier phase data, suggested by PK/PD models, or from studies on similar interventions. Specifically, for treatment  $g$  ( $g = 0, \dots, G$ ), endpoint  $j$ , the mean time-response profile is assigned to be  $\mu_{j1}^{(g)}, \dots, \mu_{jT}^{(g)}$ .

To simplify the burden of assigning  $(G + 1) \times p \times (T + 1)$  distinct values  $\mu_{jt}^{(g)}$ , one might assume linear, quadratic or other time-response functions; an alternative is to assign  $\mu_{jt}^{(g)}$  for a few specific  $t \in \{0, \dots, T\}$ , and linearly interpo-

late to specify the remaining  $\mu_{jt}^{(g)}$ . Since time-response profiles tend to be most curved at the beginning of the study, it is best to assign more values  $\mu_{jt}^{(g)}$  for small  $t$ ; fewer values are needed for large  $t$  where the patients have (typically) reached a steadier state.

Bayesian predictive data generation can be accommodated by generating the  $\mu_{jt}^{(g)}$  from probability distributions rather than assigning the values specifically. We do not pursue that angle here, but instead suggest sensitivity analysis using alternative configurations  $\mu_{jt}^{(g)}$  that are considered probable, *a priori*.

## 5 Distribution Filters

Clinical data can be non-normal: adverse event data are binary, self-reported and physician-reported patient conditions are often reported in five-point and seven-point Likert scales, laboratory data contain outliers, and primary endpoints are often censored time-to-event data (such as time to cure or time to death). In this section we show how to generate an ideal data set having desired distributional characteristics and correlation structures; the next section will consider the more realistic situation where there are patient dropouts and noncompliance effects.

### 5.1 Normal

Given that the basic random elements  $Z_{ijt}$  are normal, this is the simplest case. Standard deviations  $\sigma_{jt}^{(g)}$  must be specified in addition to the means, and the

data set contains elements

$$Y_{ijt}^{(g)} = \mu_{jt}^{(g)} + \sigma_{jt}^{(g)} Z_{ijt}. \quad (5)$$

### 5.1.1 Input Issues

Homoscedasticity assumptions simplify inputs; in some cases it may be reasonable to assume  $\sigma_{jt}^{(g)} = \sigma_{jt}$ ; or even more simply that  $\sigma_{jt}^{(g)} = \sigma_j$ .

## 5.2 Lognormal

For positive, positively skewed clinical data, the lognormal distribution often is realistic. Here we assume simply

$$\ln(Y_{ijt}^{(g)}) = \mu_{jt}^{(g)} + \sigma_{jt}^{(g)} Z_{ijt}.$$

### 5.2.1 Input Issues

The  $\mu_{jt}^{(g)}$  may be specified as means of the logged data. If it is more convenient to think in terms of the unlogged data, then the  $\mu_{jt}^{(g)}$  may be specified as  $\mu_{jt}^{(g)} = \ln\{\text{median}(Y_{ijt}^{(g)})\}$ .

Again homoscedasticity assumptions simplify inputs; in some cases it may be reasonable to assume  $\sigma_{jt}^{(g)} = \sigma_{jt}$ ; or even more simply that  $\sigma_{jt}^{(g)} = \sigma_j$ , assuming that it is convenient to specify these quantities in terms of the logged data. However, even assuming  $\sigma_{jt}^{(g)} = \sigma_j$  the raw data  $Y_{ijt}^{(g)}$  are heteroscedastic across time and group, depending on the  $\mu_{jt}^{(g)}$ . If it is more convenient to think in terms of unlogged data, then the  $\sigma_{jt}^{(g)}$  may be specified in terms of  $\mu_{jt}^{(g)}$  and the standard deviation  $\sigma_{jt}'^{(g)}$  of the unlogged  $Y_{ijt}^{(g)}$  by solving

$$\{\sigma_{jt}^{(g)}\}^2 = \exp\{2\mu_{jt}^{(g)} + 2\sigma_{jt}^{(g)}\} - \exp\{2\mu_{jt}^{(g)} + \sigma_{jt}^{(g)}\}$$

for  $\sigma_{jt}^{(g)}$ .

The correlation inputs  $\rho$  and  $\theta$  in the repeated measures covariance matrix  $\Sigma$  may be specified in terms of the logged data. If it is more convenient to consider unlogged data, one may use the identity

$$\text{Corr}(Y_{ijt}^{(g)}, Y_{ijt'}^{(g)}) = \frac{\exp\{\sigma_{jt}^{(g)} \sigma_{jt'}^{(g)} \times \text{Corr}(Z_{ijt}, Z_{ijt'})\} - 1}{\{\exp(\sigma_{jt}^{(g)}) - 1\}^{1/2} \{\exp(\sigma_{jt'}^{(g)}) - 1\}^{1/2}} \quad (6)$$

to help identify values of  $\rho$  and  $\theta$  that are reasonably consistent with the values of  $\text{Corr}(Z_{ijt}, Z_{ijt'})$  that are implied by (6). However, when attempting to identify input values for the endpoint correlation matrix  $\Gamma$ , one cannot use (6) in general, as some endpoints might be normal, others lognormal, others ordinal, etc. Thus, it is best to identify inputs for lognormal case in terms of the logged data, rather than to use (6). On the other hand, (6) can be helpful as a diagnostic check.

### 5.3 Mixture

While the lognormal distribution allows positive outliers, clinical data often contain outliers in both directions and can be negative; a common example of both is percentage change data. A general mixture random variable is easily generated as  $X = I(U \geq c)X_1 + I(U < c)X_2$ , where  $X_i \sim F_i$ ,  $U$  is a  $U(0, 1)$  random variable independent of  $X_i$ , and where  $c$  is the contamination fraction.

Suppose the values  $E(Y_{ijt}^{(g)}) = \mu_{jt}^{(g)}$  and  $\text{Var}(Y_{ijt}^{(g)}) = \{\sigma_{jt}^{(g)}\}^2$  are given as

inputs. To simplify inputs and to facilitate consistency across the various distribution filters, one might specify normal mixing distributions with common means. This is accomplished easily: consider (5) and make the following substitution:

$$Z_{ijt} \leftarrow I(U \geq c_j) \frac{Z_{ijt}}{(1 - c_j + c_j r_j^2)^{1/2}} + I(U < c_j) \frac{r_j Z_{ijt}}{(1 - c_j + c_j r_j^2)^{1/2}}, \quad (7)$$

where  $c_j$  is the contamination fraction for endpoint  $j$  and  $r_j$  is the ratio of contaminated to normal standard deviation. The resulting  $Z_{ijt}$  remain unit variance, so the desired means and variances of  $Y_{ijt}^{(g)}$  are achieved.

### 5.3.1 Input Issues

The mixing fraction  $c_j$  and the standard deviation ratio  $r_j$  might be estimated from historical data (e.g., using maximum likelihood), or simply assigned. In the latter case, the relationship between excess kurtosis  $\kappa_j$  and  $(c_j, r_j)$  is helpful:

$$\kappa_j = \frac{3c_j(1 - c_j)(r_j - 1)^2}{(1 - c_j + c_j r_j^2)^2}.$$

For example, one might suspect a certain contamination fraction  $c_j$  (e.g., .05), and know the approximate kurtosis from historical data (e.g., 20), in which case the standard deviation ratio  $r_j$  can be set to 5.48.

Interestingly, the correlations between the repeated measures  $Y_{ijt}^{(g)}$  obtained after applying (7), and then (5), within endpoint  $j$  are identical to the corresponding correlations between the basic quantities  $Z_{ijt}$ . Hence, for determining  $\rho$  and  $\theta$  to generate the basic normal quantities  $Z_{ijt}$ , one may use raw data,

despite its nonnormal characteristics. Correlations between mixture endpoints and other types of endpoints are attenuated, however. To use historical data to estimate or suggest correlations among the basic normal quantities  $Z_{ijt}$ , it is therefore advisable to transform such historical data to normal scores before estimation.

## 5.4 Binary

Binary data abound in clinical trials: adverse events are typically binary, and efficacy measures such as cure/no cure are binary as well. The binary outcomes may be generated from a multivariate probit model in terms of the basic variables  $Z_{ijt}$ :

$$Y_{ijt}^{(g)} = I(Z_{ijt} > t_{jt}^{(g)}),$$

where  $t_{jt}^{(g)} = \Phi^{-1}(1 - \mu_{jt}^{(g)})$ , and where  $\mu_{jt}^{(g)}$  is the desired probability of success.

### 5.4.1 Input Issues

The  $\mu_{jt}^{(g)}$  may be instantiated using PK/PD models, earlier phase studies, or studies on similar compounds. With binary data there is no need to specify standard deviations  $\sigma_{jt}^{(g)}$ .

When variables are binary, the correlations in (3) and (4) refer to tetrachoric correlations in the case of multiple binary variables, or to biserial correlations when some variables are binary and others are normal. Correlations involving the raw binary variables should not be used when specifying parameters in  $\Gamma$  and  $\Sigma$ ; such correlations are generally too small.

## 5.5 Ordinal

Ordinal data are common in clinical data sets; such data arise from patient-reported pain scales and quality-of-life (QOL) surveys, as well as from Physician's Global Assessments. Such data are typically recorded as 5-point or 7-point Likert scales. To simulate data from such a process, the multivariate probit model is extended from 0,1 to several categories, say  $1, \dots, k$ .

Suppose the mean structure  $E(Y_{ijt}^{(g)}) = m_{jt}^{(g)}$  is desired. Unlike the binary case, there are many probability distributions  $P(Y_{ijt}^{(g)} = c)$ , ( $c = 1, \dots, k$ ), for which  $E(Y_{ijt}^{(g)}) = \sum_c cP(Y_{ijt}^{(g)} = c) = m_{jt}^{(g)}$ . It is unwieldy to specify separate probability distributions  $P(Y_{ijt}^{(g)} = c)$  for all timepoints, treatment groups and endpoints, and the process can be simplified by specifying a single baseline distribution, then relating all distributions to the baseline distribution through an ordinal probit model.

To that end let the baseline distribution (assumed common for all treatment groups) be given by  $P(Y_{ij0} = c) = p_{jc}$ , with  $\sum_c p_{jc} = 1$ . In the baseline category, the data are generated in terms of the fundamental quantities as

$$Y_{ij0}^{(g)} = 1 + I(Z_{ij0} > t_{j1}) + \dots + I(Z_{ij0} > t_{j,k-1}),$$

where  $t_{jc} = \Phi^{-1}(p_{j1} + \dots + p_{jc})$ . For other timepoint\*group combinations, set

$$Y_{ijt}^{(g)} = 1 + I(Z_{ijt} > t_{j1} - \mu_{jt}^{(g)}) + \dots + I(Z_{ijt} > t_{j,k-1} - \mu_{jt}^{(g)}).$$

Here the  $\mu_{jt}^{(g)}$  are determined by solving  $E(Y_{ijt}^{(g)}) = m_{jt}^{(g)}$ , which implies that

they are the solution to

$$k - \sum_{c=1}^{k-1} \Phi(t_{jc} - \mu_{jt}^{(g)}) = m_{jt}^{(g)}. \quad (8)$$

Software to solve nonlinear equations is required for (8); nonlinear regression software such as PROC NLIN will work if more sophisticated tools are unavailable.

### 5.5.1 Input Issues

The  $m_{jt}^{(g)}$  are suggested, as before, from early phase studies, PK/PD models, or studies on similar compounds. The baseline distributions  $P(Y_{ij0} = c) = p_{jc}$  are specified similarly.

Correlations in (3) and (4) refer to polychoric correlations in the case of multiple ordinal variables, or to ordinal extensions of biserial correlation when some variables are ordinal and others are normal. Again, correlations involving the raw ordinal variables should not be used when specifying parameters in  $\Gamma$  and  $\Sigma$ ; such correlations are generally too small.

## 5.6 Survival

As seen above, the simulation model we present revolves around the basic patient quantities  $Z_{ijt}$ , which may be thought of as "latent health indicators." This paradigm generalizes easily to a certain kind of survival model known as the "first hitting time model" (Lee and Whitmore, 2004). As in the case of binary variables, thresholds  $t_{jt}^{(g)}$ ,  $t = 1, \dots, T$ , must be assigned. Survival times  $Y_{ij}^{(g)}$

and censoring indicators  $C_{ij}^{(g)}$  are then given as

$$Y_{ij}^{(g)} = \min\{t : Z_{ijt} > t_{jt}^{(g)}\} \text{ and } C_{ij}^{(g)} = 0, \text{ when such a } t \text{ exists} \quad (9)$$

$$Y_{ij}^{(g)} = t \text{ and } C_{ij}^{(g)} = 1 \text{ otherwise} \quad (10)$$

Such a model is realistic when  $Z_{ijt}$  can be thought of as "progression of disease" or "progression of cure." Note that an equivalent form of the model has  $Y_{ijt}^{(g)} = \min\{t : Z_{ijt} - t_{jt}^{(g)} > 0\}$  so that the quantities  $Z_{ijt} - t_{jt}^{(g)}$  may be thought of as group-specific "progression", with 0 denoting the threshold that determines a survival "event."

### 5.6.1 Input Issues

In cases where "progression"  $Z_{ijt} - t_{jt}^{(g)}$  is manifest, the inputs  $t_{jt}^{(g)}$  can be determined from historical data, PK/PD models, or studies on similar compounds. In such a case the standard deviation of the progression measurement must be "absorbed" into the  $t_{jt}^{(g)}$ . When progression is not manifest, it may be convenient to specify survival probabilities  $S_{jt}^{(g)} = P(Y_{ij}^{(g)} > t)$ , and then determine the  $t_{jt}^{(g)}$  as a function of the  $S_{jt}^{(g)}$  using (9) by solving  $P(\min\{t : Z_{ijt} > t_{jt}^{(g)}\} > t) = S_{jt}^{(g)}$ . Unfortunately, this equation seems to defy simple solution for general  $t_{jt}^{(g)}$  and the correlation structure (3), although the relationship has been established for certain special cases as noted by Aalen and Gjessing (2001). We suggest modeling using the  $t_{jt}^{(g)}$ , and checking that the implied survival functions  $S_{jt}^{(g)}$  (e.g., using simulation) are reasonable.

As far as correlation inputs go, the values in (3) can be suggested by historical

progression data when such is available. Otherwise, the inputs (3) are not much of a concern, as there is only one survival outcome  $Y_{ij}^{(g)}$ , not repeated measures indexed by  $t$ . However, the relationship between  $t_{jt}^{(g)}$  and  $S_{jt}^{(g)}$  depends on the values chosen in (3), so again, the implied values of  $S_{jt}^{(g)}$  should be examined.

Correlations between the latent "progression" and other endpoints (as given in (4)) can possibly be estimated from maximum likelihood analysis of data using the "first hitting time model," or simply assigned *a priori*.

## 6 Dropouts and Noncompliance

The mechanisms in section 5 create an "ideal data set." However, in the real world of clinical trials, data sets are rarely ideal: patients skip visits, they drop out of the study altogether, and they do not comply with their assigned regimen, by failing to take doses, or by mixing their medication with other medications and/or nutraceuticals. Thus the actual data  $D_{ijt}^{(g)}$  differ from the ideal data  $Y_{ijt}^{(g)}$  that are defined above.

### 6.1 Dropouts and Missing Data

The dropout and missing data mechanisms are complex, and generally cannot be assumed to be missing at random (MAR) or missing completely at random (MCAR). More realistically, the missing values should depend on patient experience in the trial. The following models simultaneously aim for realism and simplicity, using the framework of the previous sections.

First we make a distinction: a "dropout" refers to a patient's discontinuation

in the trial. "Missing data" refer to a patient's missing of a visit. In some cases (e.g., for  $t = T$ ) it is impossible to distinguish the two simply from the missing value in the clinical record; nevertheless, it makes sense to model the data generating mechanisms as distinct.

### 6.1.1 Missing Data

A simplifying assumption is that the missing values (not the dropouts) are MCAR, reflecting a sporadic tendency to miss visits that is consistent across all treatment groups. If  $U_{ijt}^{(g)} \sim U(0, 1)$  are generated independently of the  $Y_{ijt}^{(g)}$ , and if the missing value rate is  $\beta$ , then assign  $D_{ijt}^{(g)}$  to missing when  $U_{ijt}^{(g)} < \beta$ , else  $D_{ijt}^{(g)} = Y_{ijt}^{(g)}$ .

### 6.1.2 Dropouts

Patients drop out frequently. In some cases dropouts are sporadic, with MCAR mechanism; in other cases dropout may be related to lack of safety and/or efficacy of the patient's experience. There are several possible ways to model the dropout mechanism; some examples and further references are contained in O'Brien, Zhang and Bailey (2005). We have found it convenient to define a "misery index" in terms of the basic quantities  $Z_{ijt}^{(g)}$ , allowing group-specific dropout rates that are defined by quantiles of the distribution of the misery index. The misery index is a combination of efficacy and safety: if both efficacy and safety are very bad, then the patient drops out. On the other hand, if safety is somewhat bad but efficacy is very good, the patient might stay; conversely, if

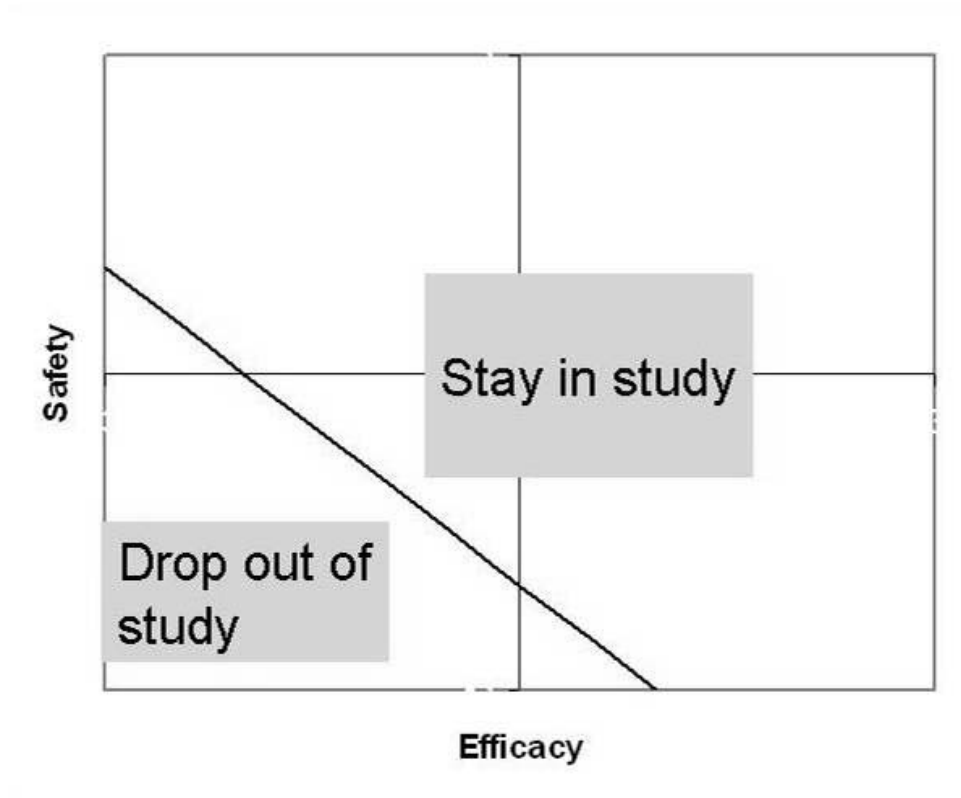


Figure 3: Dropout mechanism.

efficacy is somewhat bad but safety is good the patient might stay. This model is illustrated in Figure 3:

To be specific, define  $\{\text{Safety}\}$  to be the set of safety endpoints, and  $\{\text{Efficacy}\}$  to be the set of efficacy endpoints. Then, for  $j \in \{\text{Safety}\}$  let  $c_j = -1(1)$  if lower(higher) values of  $Z_{ijt}^{(g)}$  indicate greater safety; for  $j \in \{\text{Efficacy}\}$  let  $c_j = -1(1)$  if lower(higher) values of  $Z_{ijt}^{(g)}$  indicate greater efficacy. Define

$$S_{it}^{(g)} = \sum_{j \in \{\text{Safety}\}} c_j Z_{ijt}^{(g)}, \quad E_{it}^{(g)} = \sum_{j \in \{\text{Efficacy}\}} c_j Z_{ijt}^{(g)},$$

the combined safety and efficacy scores respectively. The dropout mechanism shown in Figure 3 is related to these quantities, but we assume further that the effects may be cumulative, and that the cumulative effect may be more local or global. Thus we define an exponential smooth parameter  $s \in [0, 1]$ , where  $s = 1$ , denotes a completely local effect,  $s = 0$  denotes a cumulative effect, and values intermediate denote intermediate cumulative effect of misery. Our "misery index" that allows differential weight  $w \in [0, 1]$  to safety and efficacy is defined as follows:

$$I_{it}^{(g)} = w \frac{S_{it}^{(g)}}{Var^{1/2}(S_{it}^{(g)})} + (1-w) \frac{E_{it}^{(g)}}{Var^{1/2}(E_{it}^{(g)})},$$

$$I_{it}^{(g)} \leftarrow I_{it}^{(g)} + (1-s) I_{i,t-1}^{(g)} + (1-s)^2 I_{i,t-2}^{(g)} + \dots,$$

to allow exponentially smoothed carryover effects of the misery index, and finally we have our index

$$I_{it}^{(g)} \leftarrow \frac{I_{it}^{(g)}}{Var^{1/2}(I_{it}^{(g)})},$$

a  $N(0, 1)$  quantity. Patient  $i$  is assigned to leave whenever  $I_{it}^{(g)} > \Phi^{-1}(1 - r_t^{(g)})$ , where  $r_t^{(g)}$  is the group-specific dropout rate.

## 6.2 Noncompliance

Realistically, patients will not adhere to their regimens. Our model assumes that, within each between-visit interval (indexed by  $t$ ), a patient's compliance  $C_{it}^{(g)}$  is a continuous measurement on the  $[0, 1]$  scale. If compliance is meant to refer to actual percentage of doses taken during the interval, then true compliance is actually discrete, taking values such as 14/14, 13/14, etc., and the

value  $C_{it}^{(g)}$  may be considered as a continuous approximation to this discrete value. On the other hand, "true compliance" might more actually reflect a variety of behaviors including proper timing of doses, and compliance with food and additional medication requirements. In such case the use of a continuous distribution for  $C_{it}^{(g)}$  has better justification.

It is sensible that there are subject-specific and carryover effects for the compliance data. Thus, the same CS + AR(1) model of (3) might therefore be used as above, by first generating CS + AR(1) normally distributed fundamental quantities  $Z_{ict}$  ( $i$  denotes patient,  $c$  denotes compliance endpoint,  $t$  denotes time interval), and then convert to proportions through the normal probability transform. While the  $Z_{ict}$  can be thought to be dependent on the endpoints, it simplifies inputs to make them independent. Dependence of the patient responses upon compliance can then be modeled directly in terms of effects of noncompliance on the patient-specific mean response, which is more consistent with PK/PD models (Holford and Peace, 1992; Lee et al, 2003).

The  $C_{it}^{(g)}$  are determined in terms of the  $Z_{ict}$  using group-specific target compliance rates. Assuming compliance is stationary over time, the median compliance at any time might be prespecified as  $C_{.5}^{(g)}$ , e.g.,  $C_{.5}^{(0)} = 0.95$  and  $C_{.5}^{(1)} = 0.90$  specifies median compliances of 95% and 90% in the control and dosed groups, respectively. An additional constraint in the form of another percentile is needed; let us suppose the tenth percentile is specified. Then the  $C_{it}^{(g)}$  are determined as

$$C_{it}^{(g)} = \Phi(a^{(g)} + b^{(g)}Z_{ict}),$$

where

$$a^{(g)} = \Phi^{-1}(C_{.5}^{(g)})$$

and

$$b^{(g)} = \frac{\Phi^{-1}(C_{.1}^{(g)}) - \Phi^{-1}(C_{.5}^{(g)})}{\Phi^{-1}(C_{.1}^{(g)})}.$$

### 6.2.1 Effect of Noncompliance on Treatment Response

Motivated by models used by Holford and Peace (1992), and by Lee et al (2003) respectively, we assume that noncompliance regresses the patient response toward natural history or placebo, depending on user preference. No matter which distribution structure is assumed, the parameters  $\mu_{jt}^{(g)}$  are closely tied to drug or placebo effects. Noncompliance is assumed to alter these values by "regressing" them towards baseline values  $\mu_{jt}^{(b)}$  as shown in the following replacement operation:

$$\mu_{jt}^{(g)} \leftarrow \mu_{jt}^{(b)} + CE_{it}^{(g)}(\mu_{jt}^{(g)} - \mu_{jt}^{(b)}),$$

where  $CE_{it}^{(g)}$  denotes "compliance effect," which like compliance, is between 0 and 1, and is explained further below.

In some cases it is sensible to take  $\mu_{jt}^{(b)} = \mu_{jt}^{(0)}$ ; i.e., noncompliance regresses effects towards the placebo response (Lee et al, 2003). In other cases it may be sensible to input the  $\mu_{jt}^{(b)}$  as "natural history" values, and assume that non-compliance regresses all treatment groups (and perhaps even the placebo group) toward the natural history (Holford and Peace, 1992).

The compliance effect  $CE_{it}^{(g)}$  is assumed to be related to actual compliance depending on the study – some studies model compliance effects as cumulative,

suggesting

$$CE_{it}^{(g)} = (C_{i1}^{(g)} + \dots + C_{it}^{(g)})/t.$$

In other cases, such as with antibacterials, the effect of compliance might be purely local:

$$CE_{it}^{(g)} = C_{it}^{(g)}.$$

In general, one might suppose both local and global effects via an exponential smooth specified as follows:

$$\begin{aligned} CE_{i1}^{(g)} &= C_{i1}^{(g)} \\ &\dots \\ CE_{it}^{(g)} &= C_{it}^{(g)} + (1-s)CE_{i,t-1}^{(g)} \end{aligned}$$

Converting to averages so that all compliance effects are between 0 and 1, we perform the following substitution:

$$\begin{aligned} CE_{it}^{(g)} &\leftarrow \frac{s\{CE_{it}^{(g)}\}}{1 - (1-s)^t}, \text{ for } s \neq 0 \\ CE_{it}^{(g)} &\leftarrow \frac{CE_{it}^{(g)}}{t}, \text{ for } s = 0 \end{aligned}$$

### 6.2.2 Correlation with Misery Index

The local compliance values logically might be related to the misery index. If the correlation desired between local compliance and cumulative misery index is  $\tau$ , then the local compliance may be defined via the following substitution:

$$C_{it}^{(g)} \leftarrow \tau^{1/2} I_{it}^{(g)} + (1-\tau)^{1/2} C_{it}^{(g)}.$$

## 7 An Example

Gatekeeping strategies allow both determination of clinical success and determination of a collection of secondary indications, all with strong control of the familywise error rate (or FWE; see Westfall and Krishen, 2001, and Wiens and Dmitrienko, 2005). The complex multivariate nature of these procedures makes mathematical determination of their operating characteristics (true FWE and various power functions) difficult. Adding issues of noncompliance, nonnormality (including survival and other types of endpoints), and informative dropouts, makes the problem completely intractable in finite samples.

We consider a hypothetical trial with two arms, a primary endpoint that normal with slight positive kurtosis, and a collection of three equally important secondary endpoints, two of which are measured on 5-point Likert scales, and one of which is a survival endpoint. We suppose that the (latent) endpoint correlations, subject correlation, and carryover correlation are all .5; as well as cumulative noncompliance that regresses the response toward the placebo. Compliance is assumed to be good, with median and 10th percentiles equal to 95% and 80% respectively in both groups. We assume there are five binary safety variables, and that the dropout mechanism is equally weighted by safety and efficacy, where the safety is determined by averaging all safety measures and efficacy is determined by averaging all efficacy measures. The dropout rates are low; 1% and 1.5% at each timepoint as determined by the misery index in the control and treatment groups, respectively and that the random missing value rate is 1%. Missing values will be imputed using last observation carried

forward (LOCF).

The study is considered for either 12 weeks or 8 weeks; part of the analysis will be to determine pros and cons of each. Mean response functions are assumed to increase rapidly, with 90% of the efficacy at 8 weeks and levelling off at 12 weeks for all efficacy measures. Only one of the safety measures is assumed to have a drug-related effect; all others are assumed null.

Interesting questions are (a) What is the best test for the primary endpoint -The two sample t-test on the raw scores? The two-sample t-test on the changes from baseline? The Kruskal-Wallis test on the raw scores? The Kruskal-Wallis test on the change scores? and (b) Among studies where the primary endpoint has been found significant by the best test found in the answer to (a), how many secondary endpoints will be found significant using the proportional hazards model for the survival endpoint and Kruskal-Wallis tests for the ordinal scores? The gatekeeping strategy chosen here uses Hochberg's procedure for all secondary endpoints. Hochberg's method is particularly attractive for its simplicity: because it requires only the  $p$ -values for the marginal tests, it can be used easily for endpoints with mixed data types, such as survival and metric. The sequential nature of the Hochberg procedure as well as the conditional nature of the gatekeeping structure make this problem virtually intractable mathematically.

Table 1 summarizes the results of various tests on the primary endpoint for a variety of designs.

Table1. Simulated power for primary endpoints.

| Design          | AOV  | ANCOV<br>(Mean) | ANCOV<br>(Med) | Diff<br>(Mean) | Diff<br>(Med) | KW   | KWdiff<br>(Mean) | KWdiff<br>(Med) |
|-----------------|------|-----------------|----------------|----------------|---------------|------|------------------|-----------------|
| 12 wks/ 30,30   | 0.41 | 0.55            | 0.54           | 0.48           | 0.47          | 0.58 | 0.67             | 0.65            |
| 12 wks/ 50,50   | 0.57 | 0.73            | 0.72           | 0.67           | 0.64          | 0.80 | 0.87             | 0.86            |
| 12 wks/ 100,100 | 0.83 | 0.94            | 0.93           | 0.90           | 0.89          | 0.97 | 0.99             | 0.99            |
| 8 wks/ 30,30    | 0.36 | 0.49            | 0.48           | 0.43           | 0.41          | 0.51 | 0.59             | 0.57            |
| 8 wks/ 50,50    | 0.51 | 0.67            | 0.66           | 0.59           | 0.58          | 0.73 | 0.82             | 0.80            |
| 8 wks/ 100,100  | 0.78 | 0.90            | 0.90           | 0.86           | 0.84          | 0.95 | 0.98             | 0.98            |

Clearly, the power of the test is very sensitive to the primary analysis.

Table 2 shows the results for the secondary endpoints, using Hochberg's method, among cases where the KW difference test (from the mean) is significant.

Table2. Simulated power for individual secondaries,

for any secondary, and for all secondaries.

| Design          | Sec 1 | Sec 2 | Sec 3 | Any  | All  |
|-----------------|-------|-------|-------|------|------|
| 12 wks/ 30,30   | 0.23  | 0.54  | 0.54  | 0.69 | 0.18 |
| 12 wks/ 50,50   | 0.49  | 0.78  | 0.78  | 0.89 | 0.42 |
| 12 wks/ 100,100 | 0.83  | 0.98  | 0.98  | 0.99 | 0.82 |
| 8 wks/ 30,30    | 0.04  | 0.40  | 0.40  | 0.54 | 0.03 |
| 8 wks/ 50,50    | 0.16  | 0.62  | 0.61  | 0.75 | 0.14 |
| 8 wks/ 100,100  | 0.41  | 0.91  | 0.91  | 0.97 | 0.39 |

Thus, the 8 weeks/50,50 design offers acceptable power (82%) for the primary endpoint for the KWdiff test. If it is acceptable to get at least one significance among the secondaries following a primary significance, then the 8 weeks/50,50 design remains possibly acceptable, with 75% conditional power. However, if difference among all secondaries is desired, then the 12 weeks/100,100 design is needed. It is worth noting that the Survival test (secondary endpoint 1) behaves

poorly for small  $n$ , this is a failure of asymptotics of the usual proportional hazards model test. The simulation tool suggests that an alternative test should be sought if smaller sample sizes are to be used.

There were 20,000 simulations per trial configuration to reduce Monte Carlo error; this was accomplished in reasonable time by using the grid computing environment of SAS (Bremer et al, 2004) with 40 machines; total compute time for the three designs was  $\sim 1$  hour using the grid but would have been  $\sim 35$  hours on a stand-alone machine.

## 8 Conclusion

We have presented details concerning a system for simulation of typical clinical trials that can help to answer fundamental, yet complex, questions that are often mathematically intractable. While the system offers a rich flexible structure, simplifying modeling assumptions are made that enhance its usability but simultaneously limit its generality. Our hope is that pharmaceutical statisticians find the system information presented here to be not only useful in its own right, but also a useful reference point for developing other models for use in improved systems. The system is currently the proprietary property of Vertex Pharmaceuticals, Inc.

## References

Aalen, O.O., Gjessing, H.K. (2001). Understanding the shape of the hazard rate: A process point of view. *Statistical Science* 16: 1–22.

Anderson, J.J., Bolognese, J.A., Felson, D.T. (2003). Comparison of rheumatoid arthritis clinical trial outcome measures. *Arthritis and Rheumatism* 48: 3031–3038.

Bremer, R., Perez, J., Smith, P., Westfall, P. (2004). Grid computing at Texas Tech University using SAS. *Proceedings of the 14th Annual South-Central SAS User's Group Regional Conference*: 64–72.

Frison, L., Pocock, S.J. (1992). Repeated measures in clinical trials: Analysis using mean summary statistics and its implications for design. *Statistics in Medicine* 11:1685–1704.

Gao, F., Thompson, P., Xiong, C., Miller, J.P. (2006). Analyzing multivariate longitudinal data Using SAS<sup>®</sup>. *Proceedings of the Thirty-first Annual SAS<sup>®</sup> Users Group International Conference*, Paper 187–31.

Holford, N.H.G., Peace, K.E. (1992). Methodologic aspects of a population pharmacodynamic model for cognitive effects in Alzheimer patients treated with Tacrine. *Proc. Natl. Acad. Sci.* 89, 11466–11470.

Kimko, H.C., Duffull, S.B. (2003). *Simulation for Designing Clinical Trials: A Pharmacokinetic-Pharmacodynamic Modeling Perspective*. New York: Marcel Dekker Inc.

Lee, H., Kimko, H.C., Rogge, M., Wang, D., Nestorov, I., Peck, C.C. (2003). Population pharmacokinetic and pharmacodynamic modeling of Etanercept using logistic regression analysis. *Clin. Pharmacol. Ther.* 73: 348–365.

Lee, M.-L. T., Whitmore, G.A. (2004). First hitting time models for life-time data. in *Handbook of Statistics 23, Advances in Survival Analysis*; N.

Balakrishnan, and C. Rao, eds., Elsevier: Amsterdam, 537–543.

O’Brien, P.C., Zhang, D., Bailey, K.R. (2005). Semi-parametric and non-parametric methods for clinical trials with incomplete data. *Statistics in Medicine* 24:341–358.

Peck, C.C., Rubin, D.B., Sheiner, L.B.(2003). Hypothesis: A single clinical trial plus causal evidence of effectiveness is sufficient for drug approval. *Clin. Pharm. Ther.* 73: 481–490.

Poland, B., Wadda, R.(2001). Combining drug–disease and economic modelling to inform drug development decisions. *Drug Discovery Today* 6: 1165–1170.

U.S. Department of Health and Human Services, Food and Drug Administration (1999). Guidance for Industry: Clinical Development Programs for Drugs, Devices, and Biological Products for the Treatment of Rheumatoid Arthritis (RA); <http://www.fda.gov/cder/guidance/index.htm>.

Westfall, P.H., Krishen, A. (2001). Optimally weighted, fixed sequence, and gatekeeping multiple testing procedures. *Journal of Statistical Planning and Inference* 99: 25–40.

Wiens, B. L., Dmitrienko, A. (2005). The fallback procedure for evaluating a single family of hypotheses. *Journal of Biopharmaceutical Statistics* 15: 929–942.