

INTRODUCTION

When analysing pharmacokinetic (PK) data it is necessary to combine and handle various items of data (assay date/times, dosing date/times, concentration results, demographic data etc.).

Typically, such data has to go through a series of SAS® programming steps e.g. validation, merge, reporting of potential problems before it can be handled by the pharmacokineticist.

Another factor is that the decisions the pharmacokineticist makes about data – e.g. which results are anomalous – require databasing prior to the production of tables, figures, and listings.

To improve turnaround times and enable more autonomy - a hybrid SAS environment was created that enabled direct use of data by the pharmacokineticist, including databasing of decisions, all without the need for SAS programming intervention

Approach

Our approach was to analyse in terms of :

- ❖ what data was being handled
- ❖ what decisions and instructions were made by the pharmacokineticist that affected data preparation
- ❖ the tools available,
- ❖ the range and variability of studies and data

This paper concentrates on how we made our changes, the outcome, and the lessons learned.

SCOPING THE SYSTEM

We scoped the system by looking at the data used, pharmacokineticist actions and the tools available.

The data

Typically the data required were :

- ❖ assay dates and times
- ❖ dosing dates and times
- ❖ demography
- ❖ randomisation
- ❖ body weight (for some studies)
- ❖ sample results

Sample results were usually available only some time after the rest of the data. The nature of PK assays meant there were frequently delays and/or complex queries. It was not uncommon therefore to have several transmissions of such data.

From this data a SAS dataset would be created containing sample profiles that the pharmacokineticist would use for input into their non-linear regression analysis software.

Pharmacokineticist decisions

Every study is different and we realised there was no exhaustive list of what might be asked for. Those we anticipated were :

- ❖ identification of anomalous results
- ❖ profile start and end points
- ❖ profiles that were “intermediate” – that is would start with some drug already inside the body
- ❖ treatment of values below quantification level

Also the pharmacokineticist needed alerting of situations such as:

- ❖ positive results prior to dosing
- ❖ embedded values below level of quantification
- ❖ “late” positive values – i.e. after two or more values below quantification

Tools available

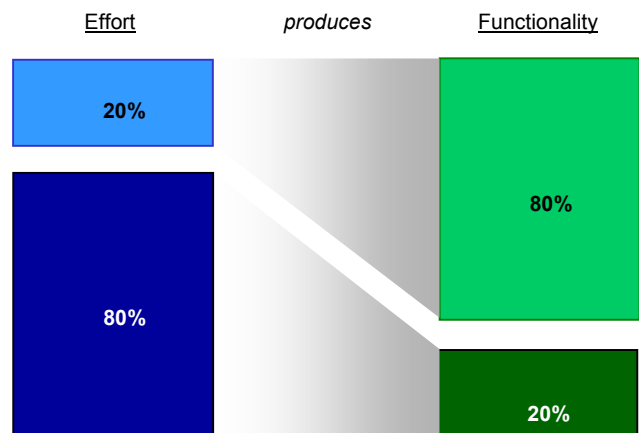
The tools available to us were principally SAS and Microsoft Office. There was some use of WinNonlin and knowledge of S and SPSS. We preferred for not introducing new software because of costs (i.e. initial and ongoing costs of support and training).

In SAS we had made extensive use of SAS MACRO for process improvements and some use of SAS/AF (mainly for utilities).

Range and variability of studies

Given that we are a contract research unit undertaking studies for many tens of diverse clients we had an instinctive view that we could not cover every single study we might encounter in future.

The Pareto principle suggests that in software development the cost – results structure is as follows :



We wanted to encompass all our studies and thus offer all our clients the same high-quality service but we were conscious that we could stray into the problem of over-elaboration in our solution. The likely cause of such variability in the system was perceived to be varying data structures.

SYSTEM DESIGN

What tools to use

The software tools decided upon were :

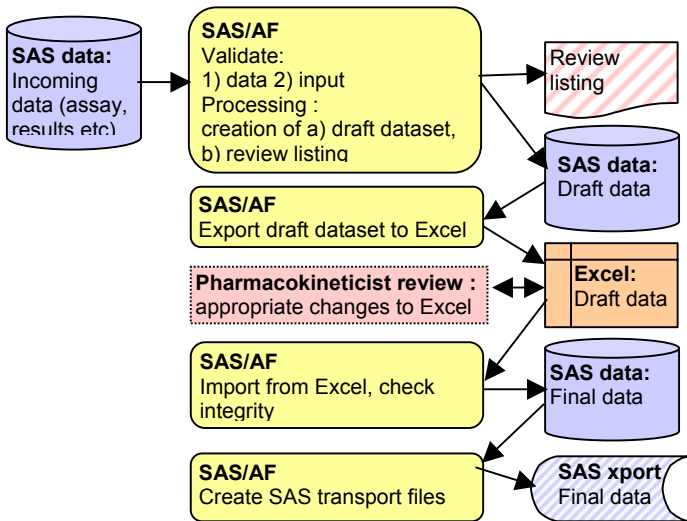
- ❖ the basic techniques of merging data were obviously to be done in SAS – we had the skills and a lot of code
- ❖ the flagging of data by the pharmacokineticist was to be done using Excel with data being transferred from SAS to Excel and back. This has some problems but Excel was a familiar tool.
- ❖ moving SAS data to Excel and back we would have to build in some integrity checking for the data – PROC COMPARE has useful facilities for this
- ❖ to handle the variability in studies we would use a SAS/AF interface – allowing the pharmacokineticist to use SAS code flexibly but without needing to write code
- ❖ the whole system was to have a menu system using SAS/AF.

Flexibility “Custom code”

To cope with varying data structures a method of using “custom code” was devised. The code behind the AF screens would use macros with code such as :

```
%macro name(macro parms,custom=No);
data randxxx;
  set {random dataset};
run;
%if %upcase(&custom)= YES %then %do;
  %include {custom code};
%end;
data rand;
  set randxxx;
run;
```

Flowchart



DATA CHECKING

On screen checks

As a general principle all parameters entered on screen are checked for validity. The user of the system – the pharmacokineticist – has some knowledge of the data but is not expected to perform data checks themselves.

Also there is a facility to “tie-up” the assays taken and the results received. Given both the later time at which PK results can be supplied and the possibility of data being re-supplied (often in time pressure situations) this was seen as improving reliability and autonomy for the pharmacokineticist.

The screen shot below shows the screen areas checked by the system :

AF code can use the same SAS facilities as MACRO and the DATA step. Particularly useful for such checking are the functions of EXIST, OPEN, VARxxx, ATTRN, ATTRC e.g. :

```
if not exist(teDosdsn.text) then
  {error code for no dose ds}
else do;
  tableid=open(teDosdsn.text,'i');
  chknum=varnum(tableid,teDosVar.text);
  if (chknum=0) then
    {error code for no such dose var}
  else do;
    cond=teDosVar.text||'='||teDval.text;
    rc=where(tableid,cond);
    if rc ^= 0 then
      {error code for incompatible value}
    else if attrn(tableid,'nlobsf')=0 then
      {error code for no instance of var=val}
  end;
  rc=close(tableid);
end;
```

REPORTS

Several reports are produced including

- ❖ validation results
- ❖ items for review
- ❖ pharmacokineticist decisions
- ❖ any custom code use

Example of review listing

Items identified for review in study ABC							
ANALYTE	VOL	PROF	DAY	NTIME	COMMENT	CONC	ACTUAL TIME
XYZ-1	3	1	1	0	Day 1 pre-dose not blq	4.26	-0.33
XYZ-1	4	2	7	3	Sample date missing	6.63	.
XYZ-1	5	1	1	0.5	Time deviation	17.79	1.2

IMPLEMENTATION

Because of the nature of this system is had to undergo validation in order to prove its fitness for use. Our approach was :

- ❖ run the system in parallel with existing methods for three months, testing at least one of each study type (single dose, parallel, cross-over etc.) making a formal comparison of results
 - ❖ use previous complex studies for stress-testing
- From this we have made a testbed for future system validations.

SAS TECHNIQUES

Macro

The number of profiles that pharmacokineticist can request is flexible. To cope with this we wrote a macro with a parameter that expected the profiles as a repeating block of “<treatment order> <start day> <start time> <end day> <end time> <intermediate interval flag> / ” and then used macro code of the form :

```
%let i = 1;
%let profile = %qscan(&profiles,&i,%str(,));
%do %while (&profile ^= );
  %let proford = %qscan(&profile,1,%str(/));
  %let rest = %qscan(&profile,2,%str(/));
  %let profsday = %qscan(&rest,1,%str( ));
  %let profstim = %qscan(&rest,2,%str( ));
  {etc}
  ...
  %let i = %eval (&i + 1);
  %let profile = %qscan(&profiles,&i,%str(,));
%end;
```

not particularly elegant, but effective for our purposes.

PROC COMPARE

We used PROC COMPARE on re-importing data from Excel to:

- ❖ to ensure that the structure of the data had not changed (rows and columns added, changed, or deleted)
- ❖ to identify what changes had been made by the pharmacokineticist.

PROC COMPARE has a useful summary return code facility, e.g. :

```
proc compare base=.. compare=.. out=.. outdif
outnoequal nosummary;
  by ... ; var <variables changable in Excel>;
run;
%let comprc=&sysinfo; /* review compare results */
%let comprc2 = %sysfunc(putn(&comprc,binary16.));
%if 0 ^= %substr(&comprc3,4,1) %then
  <Changes in values>
%if 0 ^= %substr(&comprc3,10,1) %then
  <Base data set has obs not in comparison>
%if 0 ^= %substr(&comprc3,15,1) %then
  <Data set types differ>
%if 0 ^= %substr(&comprc3,16,1) %then
  <Data set labels differ> etc. etc.
```

REVIEW

When we started out we had doubts whether we could meet all our objectives. However we have done this : the system works well, it has made big savings on effort and time, and has enabled the pharmacokineticists to work with much more autonomy.

What worked well for us :

- ❖ the overall design is robust and has been working well. It was achieved without huge cost or a long build time.
- ❖ AF with MACRO is great for achieving flexibility – both for validation and data processing.

What we might do differently :

- ❖ maybe not use Excel – as the transfer format - we will look at the facilities offered SAS v9 (e.g. XML)
- ❖ a better designed interface could be devised – at times the current interface is a bit clunky.

In the future :

- ❖ there will be more use of Oracle Clinical and CDISC style datasets. Potentially, these present some synergies and more variability – we don’t anticipate a change to the system design.

CONCLUSION

It is possible in SAS to build successful systems for the clinical trials environment. Key to the success in this project was keeping the system flexible - something which AF and MACRO are major tools for.

Another aspect of the system was making different software tools work together, this “mix and match” approach worked well and we will use this approach again. It isn’t necessary or desirable to stay totally within SAS.