MWSUG 2017 - Paper PH04

AIR Binder 2.0: A Dynamic Visualization, Data Analysis and Reporting SAS® Application for Preclinical and Clinical ADME Assays, Pharmacokinetics, Metabolite Profiling and Identification

Hao Sun, Covance Inc., Madison, Wisconsin
Kristen Cardinal, Covance Inc., Madison, Wisconsin
Richard Voorman, Covance Inc., Madison, Wisconsin

ABSTRACT

Although regulatory agencies request pharmacometric datasets be submitted as SAS® transport files for review, current clinical and preclinical ADME data analysis is handled mostly by non-SAS® programs. Previously, we reported a SAS®-based application, AIR Binder, for automatic analysis and reporting of a specific cytochrome P450 (CYP) inhibition assay, a key preclinical drug metabolism assay for the prediction of drug-drug interactions (PharmaSUG 2017). Significantly improved productivity and efficiency resulted in the expansion of the application to a wide range of preclinical and clinical ADME assays for dynamic visualization, data analysis and reporting. Considering the complexity of data structures and presentation styles across these assays, SAS® macros were designed and written to be more generalized and object-oriented. Key features include: various styles of ODS panel plots implemented to visualize profiles of metabolites for cross-species comparison and toxicology species selection; enhanced pharmacokinetic parameter analysis and display for metabolites; comprehensive statistical analysis of plasma protein binding data with PROC GLM; customized non-linear fitting for CYP inhibition and induction assays with kinetic parameter calculation and display using PROC NLIN. With the current infrastructure it is convenient to expand the program with the integration of new drug metabolism assay types for data analysis and visualization. Overall, AIR Binder 2.0 dynamically visualized data to efficiently convey information for quick decision making, which enhanced communications within study teams, between CRO and clients, and significantly shortened reporting turnaround time of drug metabolism projects for drug discovery and development.

INTRODUCTION

Preclinical and clinical ADME assays are important components to support drug discovery and development. Data from these ADME assays are critical to understand pharmacokinetic and metabolic fate of drug molecules, providing essential supporting material for IND and NDA applications with regulatory agencies. Cross-species metabolite profiling and identification using microsomes or hepatocytes as well as plasma protein binding are two primary ADME assays required for IND filing of a new chemical entity (NCE), both of which could help in selecting animal species for drug safety testing. Other in vitro ADME assays eventually required for clinical drug development and NDA filing include CYP inhibition and induction assays for drug-drug interactions (DDI), as well as CYP reaction phenotyping, which is important to understand DDI and also serves as a tool to identify the NCE's potential impact on polymorphic populations. In addition, definitive metabolite profiling and identification are required for mass balance studies in animals and human together with the determination of pharmacokinetic parameters for the NCE and its key metabolites.

Today, these preclinical and clinical ADME assays are often standardized in most pharmaceutical companies and CROs; however, data tables and figures across these assays including key messages to be delivered and presented are diverse in format. In the past, data analysis, table making, figure plotting, and curve fitting in our organization heavily relied on Microsoft Excel, together with a few specialized scientific software packages such as SigmaPlot or Prism for non-linear curve fitting due to their integrated kinetic equations. However, the process was labor-intensive, time-consuming and inefficient, and required data transfer between unrelated programs. Considering the complexity of data structures and presentation styles across these ADME assays, a better, more efficient, flexible process was needed. We sought a flexible, efficient and low maintenance platform to handle various assay reporting. SAS® macros were thus selected, designed and written in a generalized and object-oriented format to guide us to the right direction to achieve that goal.

We further developed AIR Binder (**A**utomatic **I**n Vitro/In Vivo ADME **R**eport **Binder**), a SAS®-based application (Sun 2017), for dynamic visualization, data analysis and reporting of preclinical and clinical ADME assay data. Some key features include: (1) ODS panel plots were implemented to visualize profiles of metabolites for cross-species comparison and toxicology species selection, pharmacokinetics, and other ADME assays; (2) PROC GLM was applied for comprehensive statistical analysis for plasma protein binding data; (3) PROC REG and NLIN were used for various linear and nonlinear curve fitting in CYP inhibition and induction assays (IC_{50} and EC_{50} determination using a 4-parameter logistic model), metabolic stability ($t_{1/2}$ determination), and pharmacokinetics (non-compartment and compartment models). AIR Binder dynamically visualized these ADME assay data to efficiently convey information for quick decision making within study teams, which also significantly enhanced communications between CRO and clients, and greatly shortened reporting turnaround time for drug metabolism projects in discovery and development. With the current infrastructure of AIR Binder, it is convenient to expand the program with the integration of new ADME assay types for data analysis and visualization. Overall, AIR Binder is a standardized and customized program for efficiently reporting preclinical and clinical ADME data, significantly reducing turnaround time, eliminating unnecessary errors due to data transfer, minimizing labor cost, and enhancing productivity.

In this paper, we describe the design of SAS® macros in AIR Binder to be more object-oriented, such that code was shared efficiently across all ADME assays. Then, for each individual ADME assay, the specific requirements and challenges in data analysis and presentation are discussed, which include CYP inhibition and induction, plasma protein binding, CYP phenotyping, metabolic stability and intrinsic clearance, metabolite profiling and identification, and pharmacokinetics. For each part, key assay information, data structure, scientific calculation criteria, data visualization and the role of AIR Binder in enhancing data analysis and visualization are discussed.

OBJECT-ORIENTED MACROS: CODE SHARE AND REUSE IN AIR BINDER

Object-oriented programming (OOP) is a software design philosophy. The emphasis of OOP is on the data and operations rather than concentrating on the order of programming steps. Java, Python, and C++ are the most popular OOP languages today. Significant reusability of code can be gained by means of object-oriented programming concepts including “encapsulation”, “polymorphism”, and “inheritance”. Real object-oriented SAS® applications can be created with SAS® Component Language (SCL), which is not covered in this paper.

Instead, AIR Binder was developed in Base SAS® (version 9.2), which was designed with streamlined SAS® macros. Although SAS® macro language is not inherently object-oriented, structurally it has some characteristics of object-oriented languages. Thus, more accurately, AIR Binder was designed with object-oriented concepts to organize SAS® macros, so they could be extensively shared across ADME assay types. These SAS® macro functions were reusable to analyze, visualize and report all preclinical and clinical ADME assay data.

By definition, encapsulation indicates dividing the code into a public interface and a private implementation of the interface. Main deliverables of these ADME assays are tables and figures, which construct the top class of AIR Binder. For example, in the generalized “report table” class, several basic components were included, resembling a generic recipe that generate the ADME assay tables but at the same time did not contain specific individual assay data (Figure 1). The report table class may contain various subclass and individual methods. Common subclasses include those that read in raw tables such as CSV files, round numbers, calculate basic statistics such as mean and standard deviation, transpose various numbers of replicates, display certain significant figures, and finally draw tables with desired features and styles. Accordingly, AIR Binder was written this way to generalize high-level macros that were used across individual ADME assays (“objects” in OOP). For example, to create a table for the CYP inhibition assay, a series of class/subclass macros were lined up, together with unique methods for this specific assay type. Each macro function written in AIR Binder was a self-sustainable "object" with local variable defined and macro parameters generalized. The same concept was used across ADME assays.

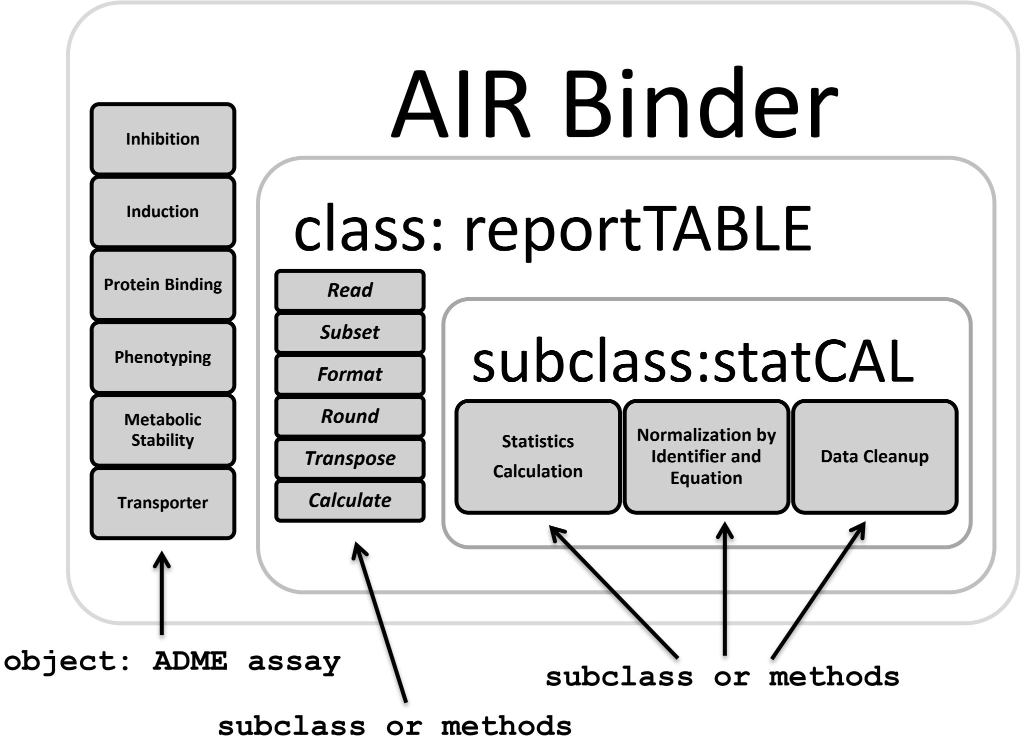


Figure 1 An example of AIR Binder code organization

SAS® macros to possess the polymorphism feature of OOP should have the ability to overload generic SAS® functions so that they have appropriate behavior based on the context at runtime. The following function STAT in AIR Binder is a generalized macro to calculate arithmetic means and standard deviation of a specified variable Y from an input file INFILE. The calculated mean (MEAN_Y) and standard deviation (STD_Y) using PROC MEANS are merged with INFILE to create the output file OUTFILE. Two other parameters that are passed in STAT are ID and X. The ID parameter is the identifier variable or a list of variables such as different enzymes, hepatocyte donors, species, patient groups, concentrations, time points, or combinations of two or more of these variables. The X parameter is a separate identifier variable that can be further used for normalization purposes, a common requirement to present preclinical and clinical ADME assay data, which are a series of concentration or time points in general.

```
%macro stat(infile=infile,outfile=outfile,id=in,x=in,y=in,mean_y=out,std_y=out);

        proc sort data=&infile; by &id &x; run;

        data &outfile._stat (keep=&id &x &y); set &infile; run;

        proc means data=&outfile._stat; by &id &x;
                output out=&outfile._staty
                mean(&y)=&mean_y
                std(&y)=&std_y;
        run;

        data &outfile; merge &infile &outfile._staty(drop=_type_ _freq_); by &id &x; run;
%mend;
```

Similarly, for the macro function NORM, the parameter EQUATION describes an assay-dependent calculation at runtime and the parameter ZERO defines the X variable value for normalization, which can be a zero concentration in the CYP inhibition and induction assays, or a CYP name with the highest formation of metabolites in the CYP phenotyping assay. Thus, this single macro is a generic interface that has multiple implementations decided at runtime, the concept of polymorphism in OOP.

```
%macro norm(infile=in,outfile=out,id=in,x=in,y=in,mean_y=in,std_y=in,
        equation=in,zero=in,pct=out);

        proc sort data=&infile; by &id; run;

        data &outfile._norm; set &infile (keep=&id &x &mean_y); by &id; run;

        data &outfile._zero (drop=&x &mean_y);
                set &outfile._norm;
                where &x="&zero";
                ctrl=&mean_y;
        run;

        proc sort data=&outfile._zero nodup; by &id; run;

        data &outfile(drop=ctrl);
                merge &infile &outfile._zero;
                by &id;
                &pct=&equation;
        run;
%mend;
```

To leverage polymorphism in other instances, some SAS® macro functions in AIR Binder were nested, defined and executed at runtime. One example was the code setup for figure plotting schemes: a single fitting curve (for all assays required), two fitting curve overlay (such as ±NDAPH results for the IC$_{50}$ shift plot in CYP inhibition assays), and three fitting curve overlay (such as three hepatocyte donors normally used for CYP induction assays). The following code is a part of a 4-parmater logistic fitting curve plot. The PREP4PL macro can be called once or multiple times to generate datasets based on the kinetic parameters determined with PROC NLIN, then to generate datasets for fitting curves, and also to create a sub dataset for the inset display in figures. The PLOTFITTING1/2/3 macro functions are called subsequently to draw corresponding figures using the SGPLOT procedure:

```
/* part of a 4-paramter logistic fitting curve plot function */
%if &assayname=fitting and &overlay=single %then %do;
        %prep4pl(…all parameters…)
        %plotfitting1(…all parameters…)
%end;

%else %if &assayname=fitting and &overlay=double %then %do;
        %prep4pl(…all parameters…)
        %prep4pl(…all parameters…)
        %plotfitting2(…all parameters…)
%end;

%else %if &assayname=fitting and &overlay=triple %then %do;
        %prep4pl(…all parameters…)
        %prep4pl(…all parameters…)
        %prep4pl(…all parameters…)
        %plotfitting3(…all parameters…)
%end;
```

To enhance code reusability, conditional definition of SAS® macros was implemented. The following is a function called PLUM in AIR Binder that was used in all ADME assays to insert a "±" sign between arithmetic mean and standard deviation values. It was also used to insert a "×/÷" sign between geometric mean and standard deviation values for special situations such as CYP induction assays when using an mRNA expression endpoint. The TYPE parameter defined runtime output. In addition, macro variables of these mathematic signs were defined as local variables using the %LET statement including PLUSMINUS, MULT and DIVI, to enhance functionality and reusability.

```
%macro plum(infile=in,outfile=out,type=in,mean_y=in,std_y=in,y_stat=out);

  %let plusminus=(*ESC*){unicode 00b1};
  %let mult=(*ESC*){unicode 00d7};
  %let divi=(*ESC*){unicode 00f7};
```

```
        %if &type=ari %then %do;
                data &outfile(drop=&mean_y &std_y);
                        set &infile;
                        if &mean_y NE . and &std_y NE . then do;
                         &y_stat =
                                right(&mean_y) || " " || "&plusminus" || " " || left(&std_y);
                        end;
                        else if &mean_y NE . and &std_y EQ . then do;
                                &y_stat = " " || right(&mean_y) || " ";
                        end;
                        else do; &y_stat = "NA"; end;
                run;
        %end;

        %if &type=geo %then %do;
                data &outfile(drop=&mean_y &std_y);
                        set &infile;
                        if &mean_y NE . and &std_y NE . then do;
                         &y_stat =
                                right(&mean_y) || " " || "&mult.|&divi" || " " || left(&std_y);
                        end;
                        else if &mean_y NE . and &std_y EQ . then do;
                                &y_stat = " " || right(&mean_y) || " ";
                        end;
                        else do; &y_stat = "NA"; end;
                run;
        %end;
    %mend;
```

Inheritance indicates the ability to create a subclass that contains specializations of the parent class, which is the mechanism an object acquires some or all properties of another object. The %INCLUDE statement is an efficient tool for code reuse. For the examples above, macro functions STAT and NORM can be called alternatively for multiple times to form a new function ANNE below. The new function first calculates the mean and standard deviation of the variable Y, and then the variable Y is normalized by a “ZERO” value and a defined “EQUATION” parameter. Following that, mean and standard deviation for the normalized Y (or the new variable PCT) is calculated. Thus, the new macro ANNE processes full functions of STAT and NORM but the combination creates a new function. The variation of different combination patterns of STAT and NORM creates different applications.

```
        %include "c:\sas\siri\stat.sas";
        %include "c:\sas\siri\norm.sas";

        %macro anne(infile=in,outfile=in,id=in,x=in,y=in,mean_y=out,
                std_y=out,pct=out,mean_pct=out,std_pct=out,equation=in,zero=in);

                %stat(infile=&infile,outfile=&outfile._ystat,
                        id=&id,x=&x,y=&y,mean_y=&mean_y,std_y=&std_y)

                %norm(infile=&outfile._ystat,outfile=&outfile._norm,
                        id=&id,x=&x,y=&y,mean_y=&mean_y,std_y=&std_y,
                        equation=&equation,zero=&zero,pct=&pct)

                %stat(infile=&outfile._norm,outfile=&outfile,
                        id=&id,x=&x,y=&pct,mean_y=&mean_pct,std_y=&std_pct)
        %mend;
```

Therefore, with the designed organization of code in AIR Binder, which was more object-oriented by applying the concepts of encapsulation, polymorphism, and inheritance, the reusability of code was significantly enhanced. The code rewriting time was dramatically reduced across preclinical and clinical ADME assay reporting. Also, minimal time was needed to develop an application for a new assay type integrated to AIR Binder. Specific examples for these ADME assays including data handling, workflow, and typical output are summarized in detail below.

CYP INHIBITION AND INDUCTION

CYP inhibition (such as reversible inhibition and mechanism-based inactivation) and induction are essential contributors for clinical drug-drug interactions. Key kinetic parameters can be derived from preclinical in vitro CYP inhibition and induction assays and then applied to predict clinical drug-drug interactions. The combined effects of inhibition and induction were integrated into a mechanistic static model (Einolf 2014), the so-called “net effect” model with the equation below:

$$AUCR = \frac{1}{f_m \times \Pi_{liver} + (1 - f_m)} \times \frac{1}{\Pi_{gut} \times \left(1 - f_g\right) + f_g}$$

where AUCR is the ratio of the pharmacokinetic AUC in the presence and absence of a potential inhibitor/inducer (“perpetrator”), f_m is the fraction of the “victim” drug metabolized by a specific enzyme (such as CYP3A4), f_g is the fraction of the “victim” drug escaping from gut extraction, and Π_{liver} and Π_{gut} are the product of the following terms in liver and gut, respectively:

$$\Pi_{liver\ or\ gut} = \frac{1}{1 + \frac{[I]}{K_i}} \times \frac{k_{degradation}}{k_{degradation} + \frac{[I] \times k_{inact}}{[I] + K_I}} \times (1 + \frac{d \times E_{max} \times [I]}{[I] + EC_{50}})$$

where [I] is the concentration of the inhibitor/inducer (such as its C_{max}), $k_{degradation}$ is the degradation rate of a specific enzyme (a constant, for example, 0.0003 min^{-1} and 0.0005 min^{-1} for CYP3A4 in liver and gut, respectively), K_i is the inhibition constant for reversible inhibition, k_{inact} is the maximum inactivation rate, K_I is the concentration of inactivator at half k_{inact}, E_{max} is the maximum induction ratio, EC_{50} is the half-maximal effective concentration for induction, and d is an estimated value derived from linear weighted least-square regression. The methods to determine these kinetic parameters are described as follows.

First, IC_{50} assay for direct inhibition is a simplified form for a representative K_i determination. Previously we described the application of a 4-parameter logistic model to calculate IC_{50} values (Sun 2017):

$$f(x) = \Phi_0 + \frac{\Phi_1 - \Phi_0}{1 + \left(\frac{x}{\Phi_2}\right)^{-\Phi_3}}$$

where Φ_0 is the lower asymptote, $\Phi_1$is the upper asymptote, Φ_2 is the inflection point or IC_{50}, and Φ_3 is the Hill slope. A typical graphic output for direct inhibition IC_{50} curve fitting in AIR Binder is shown in Figure 2.

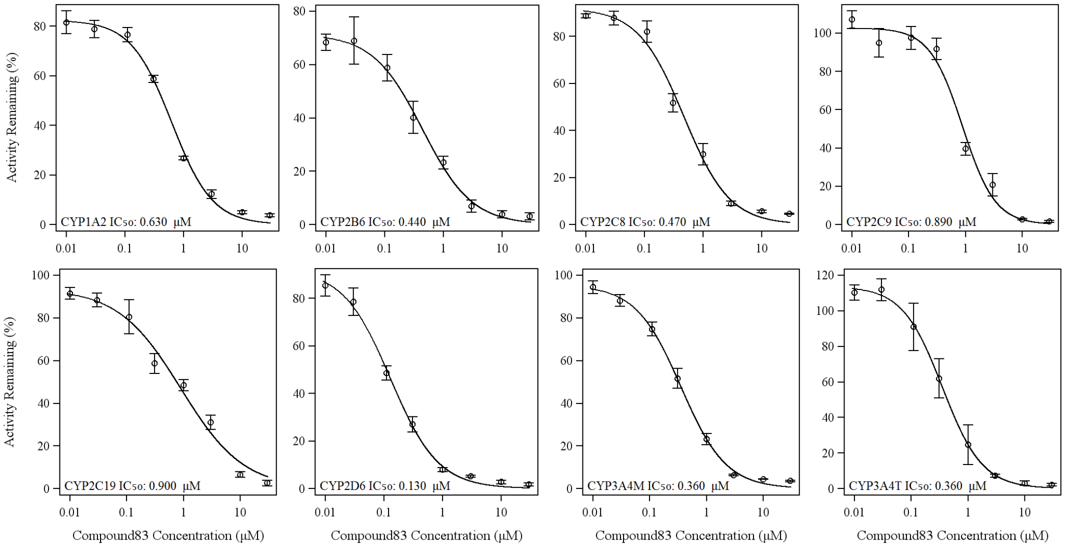


Figure 2 IC_{50} curve fitting and plot for direct inhibition assay

Four variations of IC_{50} fitting were also described previously (Sun 2017) that include (1) a relative IC_{50} (&fittype=RELATIVE); (2) an absolute IC_{50} (&fittype=ABSOLUTE); (3) a lower asymptote constrained IC_{50} (&fittype=LOWERCSN); and (4) an upper asymptote constrained IC_{50} (&fittype=UPPERCSN). They were integrated into a flexible single SAS® function by conditionally selecting 4 macro variables: “&MINCOL”, “&MAXCOL”, “&IC50COL”, and “&HILLCOL”, as defined by various %LET macro statements as follows:

```
...part of a 4-paramter logistic model preparation function...

     %if &fittype=relative %then %do;
             %let maxcal=max;
             %let mincal=min;
             %let maxcol=col1;
             %let mincol=col2;
             %let ic50col=col3;
             %let hillcol=col4;
             %let dropcol=col1 col2 col3 col4 i;
     %end;

     %if &fittype=absolute %then %do;
             %let maxcal=100;
             %let mincal=0;
             %let maxcol=100;
             %let mincol=0;
             %let ic50col=col1;
             %let hillcol=col2;
             %let dropcol=col1 col2 i;
     %end;

     %if &fittype=lowercsn %then %do;
             %let maxcal=max;
             %let mincal=0;
             %let maxcol=col1;
             %let mincol=0;
             %let ic50col=col2;
             %let hillcol=col3;
             %let dropcol=col1 col2 col3 i;
     %end;

     %if &fittype=uppercsn %then %do;
             %let maxcal=100;
             %let mincal=min;
             %let maxcol=100;
             %let mincol=col1;
             %let ic50col=col2;
             %let hillcol=col3;
             %let dropcol=col1 col2 col3 i;
     %end;
```

The values of these variables are assigned from the PARAMETERESTIMATES table (COL1-4) generated by the ODS OUTPUT statement in PROC NLIN. Additionally, the macro variables &MAXCAL and &MINCAL control the MODEL statement in PROC NLIN, and &DROPCOL adjusts the variable list to draw a curve fitting line. In practice, a 3-parameter logistic model by constraining the lower asymptote to zero (LOWERCSN) was commonly used for IC_{50} determination:

$$f(x) = \frac{\Phi_1}{1 + \left(\frac{x}{\Phi_2}\right)^{-\Phi_3}}$$

where Φ_1 is the upper asymptote, Φ_2 is IC_{50}, and Φ_3 is the Hill slope. IC_{50} is in general a good indicator of K_i for many compounds in drug development especially when they are not tightly bound inhibitors. However, the K_i value is an intrinsic parameter that is independent of substrate, enzyme and other experimental conditions. In practice, K_i assay is a combination of several IC_{50} assays at different substrate concentrations. Its data structure and processing for table and figure generation in AIR Binder are exactly the same as that of an individual IC_{50} assay.

K_i values for several typical inhibition mechanisms including competitive, non-competitive, uncompetitive, and mixed inhibition are determined by fitting the following equations using PROC NLIN:

$$V_{competitive} = \frac{V_{max} \times [S]}{K_m\left(1 + \frac{[I]}{K_i}\right) + [S]}$$

$$V_{non-competitive} = \frac{V_{max} \times [S]}{K_m\left(1 + \frac{[I]}{K_i}\right) + [S](1 + \frac{[I]}{K_i})}$$

$$V_{uncompetitive} = \frac{V_{max} \times [S]}{K_m + [S](1 + \frac{[I]}{K_i})}$$

$$V_{mixed} = \frac{V_{max} \times [S]}{K_m\left(1 + \frac{[I]}{K_i}\right) + [S](1 + \frac{[I]}{\propto \times K_i})}$$

where V is the rate of reaction as measured by a specific metabolite formation, V_{max} is the maximal rate of reaction, K_m is the affinity constant, [S] is the substrate concentration, [I] is the inhibitor concentration, K_i is the inhibition constant, and α is the interaction parameter that determines the degree to which the inhibitor binding changes the substrate binding affinity to the enzyme.

The relationship between IC_{50} and K_i for a competitive inhibition is expressed as follows:

$$K_{i,competitive} = \frac{IC_{50}}{1 + \frac{[S]}{K_m}}$$

The direct inhibition assay is conducted at the substrate concentration of K_m and thus K_i is estimated as a half of IC_{50}. However, for tightly bound inhibitors, the relationship between IC_{50} and K_i for competitive inhibition is expressed as follows:

$$K_{i,competitive} = \frac{IC_{50} - \frac{[E_0]}{2}}{1 + \frac{[S]}{K_m}}$$

where $[E_0]$ is the initial enzyme concentration. Thus, for a tight binder, K_i is also determined by the initial enzyme concentration, in addition to its IC_{50} value.

Second, the IC_{50} shift assay (Figure 3) in the presence and absence of a catalytic factor (±NADPH) serves as an indicator for further investigation of mechanism-based inactivation. The kinetic parameters K_I and k_{inact} of a specific enzyme are derived from a two-step process. First, a linear regression analysis of a time-course inhibition (using PROC REG) is used to calculate the initial rate constant (k_{obs}) at various concentrations of a compound. Then, k_{obs} values are used to fit a non-linear function with PROC:

$$k_{obs} = k_{obs,[I]=0} + \frac{k_{inact} \times [I]}{K_I + [I]}$$

where k_{obs} is the first order rate constant for a mechanism-based inactivation as estimated from the slope of a time-course inhibition (natural logarithm of residual activity), [I] is the compound concentration, k_{inact} is the maximal rate of enzyme inactivation, and K_I is the compound concentration resulting in the sum of k_{obs} at [I]=0 and 50% of the maximum enzyme inactivation.

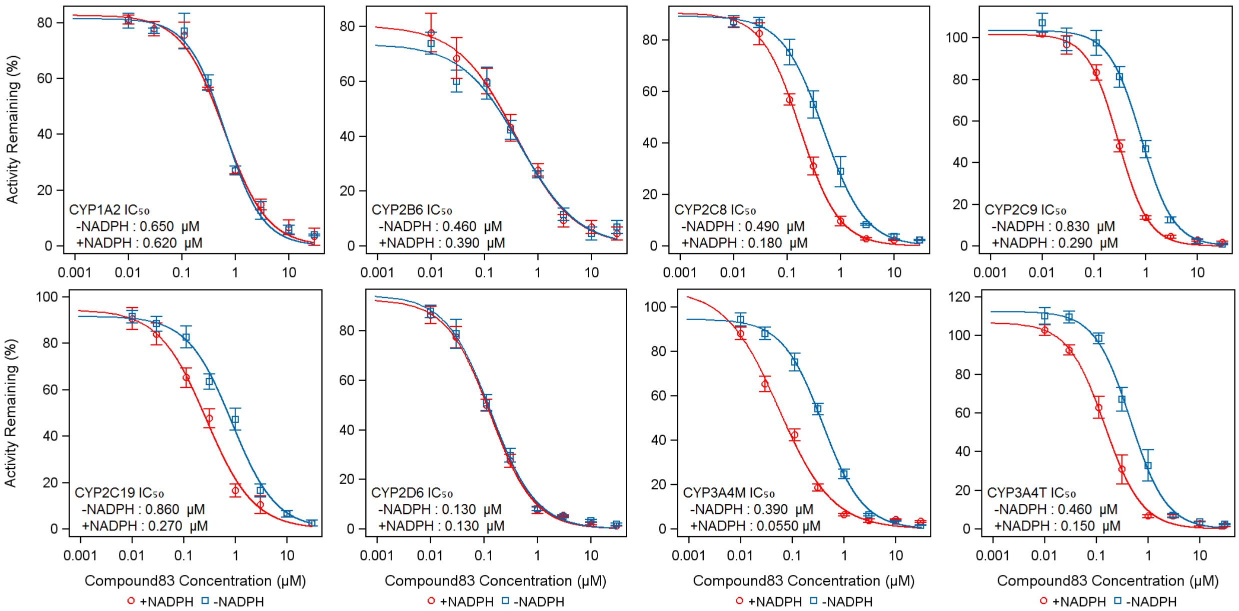


Figure 3 Display of IC_{50} shift for metabolism-dependent inhibition

Third, for a CYP induction assay two separate sets of data are generally collected: a fold change of enzyme activity and mRNA expression levels. Similarly, a four-parameter logistic model by constraining the lower asymptote is often used to determine EC_{50} and E_{max}:

$$Fold\ Induction = \frac{E_{max}}{1 + (\frac{[I]}{EC_{50}})^{-Hill}}$$

The fold induction for the enzyme activity at a specific concentration of a compound is commonly normalized to the solvent control directly. But the fold induction for mRNA expression level requires additional interpretation and is calculated with the following equations:

$$\Delta C_T = C_{T,CYP} - C_{T,reference}$$

$$\Delta\Delta C_T = \Delta C_{T,treated} - \Delta C_{T,vehicle}$$

$$Fold\ Induction = 2^{-\Delta\Delta C_T}$$

where C_T (threshold cycle) is the cycle number in real time PCR at which the fluorescence generated within a reaction crosses the threshold of detection, ΔC_T is the difference in threshold cycle between the specific CYP mRNA studied and an endogenous control such as GAPDH mRNA (glyceraldehyde-3-phosphate dehydrogenase, one of the most commonly used housekeeping gene), $\Delta\Delta C_T$ is the difference in ΔC_T between potential inducer-treated sample and vehicle control.

The 4-parameter logistic model preparation function for CYP inhibition curve overlay and CYP induction donor curve overlay are two objects from a generic logistic model preparation class in AIR Binder. A typical graphic output for induction mRNA expression EC_{50} curve fitting is presented in Figure 4. Compared to CYP inhibition IC_{50} curves, the mean and standard deviation of the fold induction used for mRNA expression are geometric instead of arithmetic, for the consideration that the fold induction of mRNA expression follows an exponential form. The GEOMEAN function was integrated to calculate the geometric mean of fold induction for any non-missing values of replicates. The EXP and LOG functions were used to calculate the geometric standard deviation.

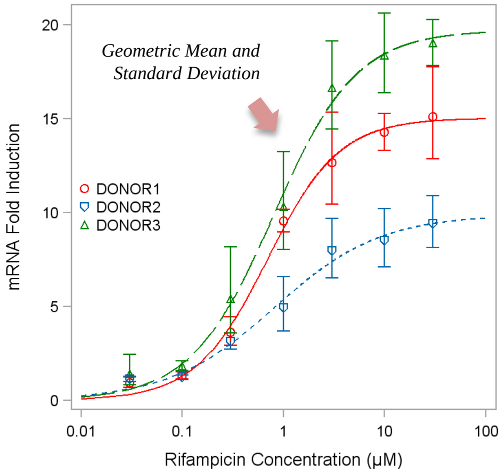


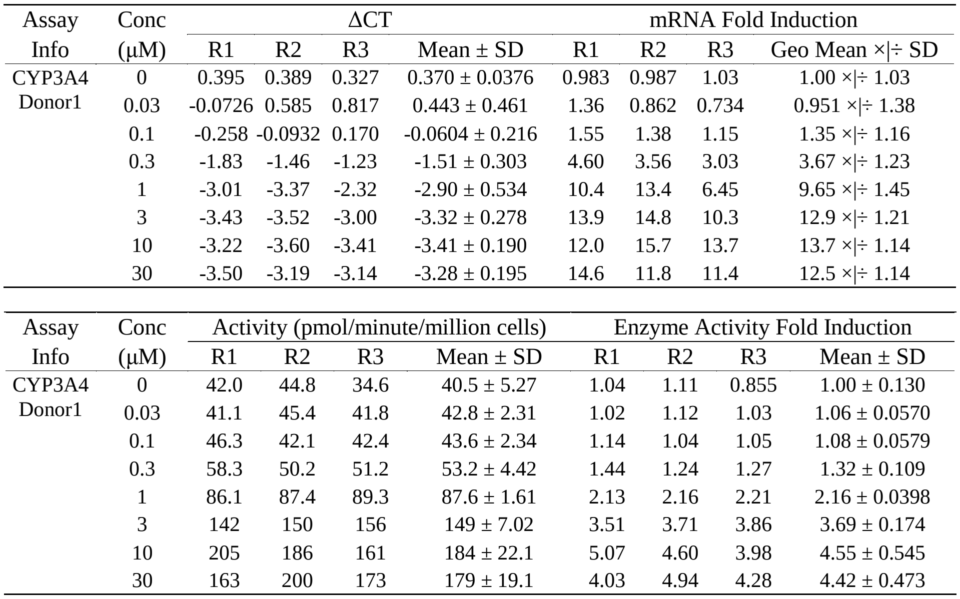
Figure 4 EC_{50} curve overlay for mRNA fold induction in multiple donors

In detail, the geometric mean and standard deviation of fold induction for mRNA expression are calculated with the following equations:

$$\mu_g = \left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}}$$

$$\sigma_g = \exp(\sqrt{\frac{\sum_{i=1}^{n}(ln\frac{x_i}{\mu_g})^2}{n}})$$

where μ_g is the geometric mean, x_i is the fold induction for mRNA expression, and σ_g is the geometric standard deviation. A comparison table output for geometric and arithmetic mean and standard deviation for the mRNA and enzyme activity fold induction in an induction assay are shown in Figure 5.



Assay	Conc	ΔCT				mRNA Fold Induction			
Info	(µM)	R1	R2	R3	Mean ± SD	R1	R2	R3	Geo Mean ×|÷ SD
CYP3A4 Donor1	0	0.395	0.389	0.327	0.370 ± 0.0376	0.983	0.987	1.03	1.00 ×|÷ 1.03
	0.03	-0.0726	0.585	0.817	0.443 ± 0.461	1.36	0.862	0.734	0.951 ×|÷ 1.38
	0.1	-0.258	-0.0932	0.170	-0.0604 ± 0.216	1.55	1.38	1.15	1.35 ×|÷ 1.16
	0.3	-1.83	-1.46	-1.23	-1.51 ± 0.303	4.60	3.56	3.03	3.67 ×|÷ 1.23
	1	-3.01	-3.37	-2.32	-2.90 ± 0.534	10.4	13.4	6.45	9.65 ×|÷ 1.45
	3	-3.43	-3.52	-3.00	-3.32 ± 0.278	13.9	14.8	10.3	12.9 ×|÷ 1.21
	10	-3.22	-3.60	-3.41	-3.41 ± 0.190	12.0	15.7	13.7	13.7 ×|÷ 1.14
	30	-3.50	-3.19	-3.14	-3.28 ± 0.195	14.6	11.8	11.4	12.5 ×|÷ 1.14

Assay	Conc	Activity (pmol/minute/million cells)				Enzyme Activity Fold Induction			
Info	(µM)	R1	R2	R3	Mean ± SD	R1	R2	R3	Mean ± SD
CYP3A4 Donor1	0	42.0	44.8	34.6	40.5 ± 5.27	1.04	1.11	0.855	1.00 ± 0.130
	0.03	41.1	45.4	41.8	42.8 ± 2.31	1.02	1.12	1.03	1.06 ± 0.0570
	0.1	46.3	42.1	42.4	43.6 ± 2.34	1.14	1.04	1.05	1.08 ± 0.0579
	0.3	58.3	50.2	51.2	53.2 ± 4.42	1.44	1.24	1.27	1.32 ± 0.109
	1	86.1	87.4	89.3	87.6 ± 1.61	2.13	2.16	2.21	2.16 ± 0.0398
	3	142	150	156	149 ± 7.02	3.51	3.71	3.86	3.69 ± 0.174
	10	205	186	161	184 ± 22.1	5.07	4.60	3.98	4.55 ± 0.545
	30	163	200	173	179 ± 19.1	4.03	4.94	4.28	4.42 ± 0.473

Figure 5 Geometric and arithmetic mean and standard deviation display for fold induction

PLASMA PROTEIN BINDING

Plasma protein binding is a routine preclinical and clinical ADME assay designed to measure the percentage of drug molecules bound to plasma proteins using the equilibrium dialysis method. Free or unbound fraction (f_u) of drug molecules is determined and compared across species for the selection of optimal preclinical toxicology species, and/or compared across patient groups in clinical studies for optimal pharmacokinetic analysis. Sometimes, free fraction may also be measured in a time course postdose to determine concentration dependency, to elucidate unexpected pharmacokinetic profiles. For most drug molecules, the unbound drug concentration in vivo determines efficacy and toxicity, which may not always be related to free fraction values derived from the plasma protein binding assay in vitro (Liu 2014). An explanation is that free fraction impacts both bioavailability and clearance, as presented below with the total and free drug exposure in plasma by a well-established mathematical liver model called "well-stirred model":

$$AUC_{total} = \frac{\mathrm{F} \times \mathrm{dose}}{\mathrm{CL}_{total}} = \frac{\frac{\mathrm{Q}}{Q + f_u \times CL_{in}} \times \mathrm{dose}}{\frac{\mathrm{Q} \times f_u \times CL_{in}}{Q + f_u \times CL_{in}}} = \frac{\mathrm{dose}}{f_u \times CL_{in}}$$

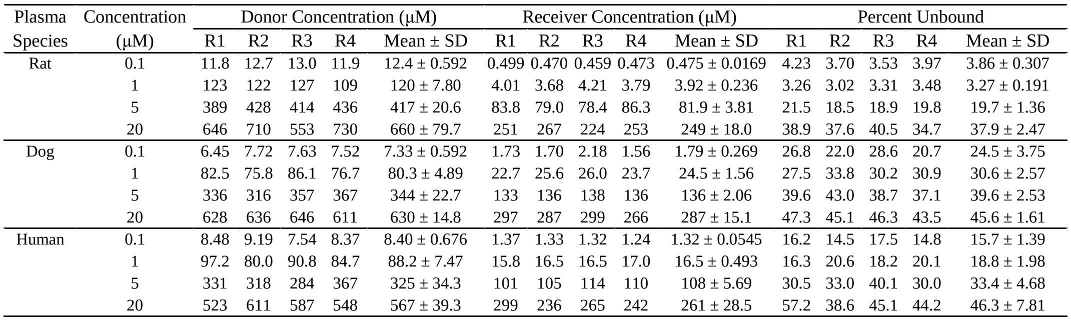
$$AUC_{unbond} = f_u \times AUC_{total} = \frac{\mathrm{dose}}{CL_{in}}$$

where $\mathrm{AUC}_{\mathrm{total}}$ is the total plasma concentration of a drug molecule or the overall drug exposure, $\mathrm{AUC}_{\mathrm{unbound}}$ is the overall unbound drug exposure, F is bioavailability, $\mathrm{CL}_{\mathrm{total}}$ is total clearance for the elimination of a drug molecule via metabolism and/or drug transport, f_u is free fraction, and Q is hepatic blood flow.

The unique data structure of the plasma protein binding assay is the paired concentration columns in raw tables: both donor- and receiver-side concentrations, which are separated by a dialysis cellulose membrane in an equilibrium dialysis device. Before generation of statistics, exactly paired data points should be used first to calculate the free fraction with the following equation:

$$Percent\ Unbound\ (f_u) = 100 \times (1 - \frac{C_{donor} - C_{receiver}}{C_{donor}})$$

where $\mathrm{C}_{\mathrm{donor}}$ is the concentration of a drug molecule in the donor side of the membrane at equilibrium in an equilibrium dialysis experiment, $\mathrm{C}_{\mathrm{receiver}}$ is the concentration of the drug molecule in the receiver side of the membrane (or dialysate). The standard table output in AIR Binder for plasma protein binding assay includes plasma species, concentration of a drug molecule, replicates (usually 4-6) with calculated mean and standard deviation of the donor-side concentration, receiver-side concentration, and free fraction values (Figure 6).



Plasma	Concentration	Donor Concentration (µM)					Receiver Concentration (µM)					Percent Unbound				
Species	(µM)	R1	R2	R3	R4	Mean ± SD	R1	R2	R3	R4	Mean ± SD	R1	R2	R3	R4	Mean ± SD
Rat	0.1	11.8	12.7	13.0	11.9	12.4 ± 0.592	0.499	0.470	0.459	0.473	0.475 ± 0.0169	4.23	3.70	3.53	3.97	3.86 ± 0.307
	1	123	122	127	109	120 ± 7.80	4.01	3.68	4.21	3.79	3.92 ± 0.236	3.26	3.02	3.31	3.48	3.27 ± 0.191
	5	389	428	414	436	417 ± 20.6	83.8	79.0	78.4	86.3	81.9 ± 3.81	21.5	18.5	18.9	19.8	19.7 ± 1.36
	20	646	710	553	730	660 ± 79.7	251	267	224	253	249 ± 18.0	38.9	37.6	40.5	34.7	37.9 ± 2.47
Dog	0.1	6.45	7.72	7.63	7.52	7.33 ± 0.592	1.73	1.70	2.18	1.56	1.79 ± 0.269	26.8	22.0	28.6	20.7	24.5 ± 3.75
	1	82.5	75.8	86.1	76.7	80.3 ± 4.89	22.7	25.6	26.0	23.7	24.5 ± 1.56	27.5	33.8	30.2	30.9	30.6 ± 2.57
	5	336	316	357	367	344 ± 22.7	133	136	138	136	136 ± 2.06	39.6	43.0	38.7	37.1	39.6 ± 2.53
	20	628	636	646	611	630 ± 14.8	297	287	299	266	287 ± 15.1	47.3	45.1	46.3	43.5	45.6 ± 1.61
Human	0.1	8.48	9.19	7.54	8.37	8.40 ± 0.676	1.37	1.33	1.32	1.24	1.32 ± 0.0545	16.2	14.5	17.5	14.8	15.7 ± 1.39
	1	97.2	80.0	90.8	84.7	88.2 ± 7.47	15.8	16.5	16.5	17.0	16.5 ± 0.493	16.3	20.6	18.2	20.1	18.8 ± 1.98
	5	331	318	284	367	325 ± 34.3	101	105	114	110	108 ± 5.69	30.5	33.0	40.1	30.0	33.4 ± 4.68
	20	523	611	587	548	567 ± 39.3	299	236	265	242	261 ± 28.5	57.2	38.6	45.1	44.2	46.3 ± 7.81

Figure 6 Paired data display for free fraction calculation in a plasma protein binding assay

The main workflow in AIR Binder to generate the previous free fraction table (Figure 6) is a batch of functions (called “FROG”, a new macro inherited from six upper-level macro functions):

```
%macro frog(infile=infile,outfile=outfile,keeplist=in,
     id=in,x=in,y=in,mean_y=out,std_y=out,y_stat=out,sigfig=3);

     %tres(infile=&infile,outfile=&outfile._tres,keeplist=&keeplist);

     %stat(infile=&outfile._tres,outfile=&outfile._stat,
             id=&id,x=&x,y=&y,mean_y=&mean_y,std_y=&std_y)

     %flip(infile=&outfile._stat,outfile=&outfile._flip,
             id=&id,x=&x,y=&y,mean_y=&mean_y,std_y=&std_y)

     %snip(infile=&outfile._flip,outfile=&outfile._round,
             roundvarlist=&mean_y &std_y,sigfig=&sigfig)

     %deco(infile=&outfile._round,outfile=&outfile._sig,
             sfvarlist=&y.1 &y.2 &y.3 &y.4 &mean_y &std_y,sigfig=&sigfig)

     %plum(infile=&outfile._sig,outfile=&outfile,type=ari,
             mean_y=sf_&mean_y,std_y=sf_&std_y,y_stat=&y_stat)
%mend;
```

FROG was called three times to create separate tables for the donor-side concentration, receiver-side concentration, and free fraction values, and then these tables were merged into a final table. The process started with the selection of three columns of data: species, concentration, and a measured/calculated endpoint (donor-side concentration, receiver-side concentration, or free fraction) using TRES, followed by calculation of statistics (STAT), transpose of quadruplicates (FLIP), rounding to 3 significant figures (SNIP), display of 3 significant figures (DECO), and joining mean and standard deviation with a “±” sign (PLUM). All these functions were written in generalized forms.

In addition, the percent recovery is calculated with the following equation:

$$Percent\ Recovery = 100 \times (\frac{C_{donor} \times V_{donor} + C_{receiver} \times V_{receiver}}{C_{spike} \times V_{donor}})$$

where C_{donor} and V_{donor} are the concentration of a drug molecule and volume in the donor side of the membrane at equilibrium, respectively; $C_{receiver}$ and $V_{receiver}$ are the concentration of a drug molecule and volume in the receiver side of the membrane (or dialysate), respectively; C_{spike} is the concentration of a drug molecule spiked in plasma prior to loading the dialysis device. The only data processing difference in AIR Binder to create a recovery table was an additional step to calculate the mean of C_{spike} before executing FROG (considering that C_{spike} is not paired with C_{donor} and $C_{receiver}$, but determined separately).

In addition to data display, a key data analysis and visualization part for the plasma protein binding assay is to determine species and/or concentration dependency. Analysis of variance (ANOVA) was used in AIR Binder to determine whether there is a difference between the means of groups such as different concentration or patient groups. If only one independent variable is to be analyzed for variance, for example, to determine human plasma free fraction in vitro, a key question is the dependency of the independent variable (concentration). Similarly, we may want to understand if there is a species difference at the same concentration of a drug molecule, if there is a difference of plasma protein binding between patient groups, or if there is a difference of free fraction of a drug molecule at several time points post-dose. These comparisons are normally conducted with a one-way ANOVA using the ANOVA procedure with a null hypothesis:

$$H_0: \mu_1 = \mu_2 = \cdots = \mu_k$$

where μ_1, μ_2, …, μ_k are the means of each group to be compared. Thus, any individual free fraction can be written using the following 1-factor model:

$$f_{u,ik} = \mu + \lambda_k + \varepsilon_{ik}$$

where µ is the grand mean of all free fraction values, λ_k is the group effect (such as species or concentration dependency) on the free fraction value for the *k*th group, and ε_{ik} is the error term (also called residual or within-group term). The F-test is performed to test the significance with the following equation:

$$F = \frac{MS_{betwen}}{MS_{within}} = \frac{\frac{SS_{between}}{K-1}}{\frac{SS_{total} - SS_{between}}{N-K}} = \frac{\frac{\sum n_i m_i^2 - \mu^2 N}{K-1}}{\frac{\left(\sum x^2 - \frac{(\sum x)^2}{N}\right) - (\sum n_i m_i^2 - \mu^2 N)}{N-K}}$$

where MS is mean squares or mean square deviation for the between- and within-group variance to be compared, SS is sum of squares, N is the total number of subjects/samples, K is the number of groups to be compared, µ is the grand mean of all free fraction values, n_i is the number of subjects/samples within the group i, and m_i is the mean of free fraction values within the group i.

If significant difference is found from PROC ANOVA, the null hypothesis will be rejected. A multiple comparison test, also called post-hoc will be used to find group difference. There are a variety of post hoc tests available in SAS® such as Tukey's studentized range test (TUKEY), Duncan's multiple-range test (DUNCAN), the Student-Newman-Keuls' multiple comparison test (SNK), and Scheffe's multiple-comparison procedure (SCHEFFE). In AIR Binder, a Tukey-Kramer method was the default post hoc test to compare group difference of free fraction values for plasma protein binding assays. Specifically, Tukey's Honestly Significant Difference (HSD) is calculated as:

$$HSD_\alpha = q_\alpha \sqrt{\frac{MS_{within}}{N}}$$

where α is the Type I error rate, q_α is a value from the studentized range statistic table based on α, MS_{within} is the mean squares of the within group, for within-group degree of freedom (N-K), and number of groups (K), and N is the number of subjects/samples in each group. Then a pairwise difference in mean is compared to HSD at the α level, which will be considered significant if $\Delta\text{mean}_{\text{pairwise}}$ > HSD.

If variance of two independent variables is to be compared, for example, the difference of free fraction values across both species and concentration levels, a two-way AVOVA will be performed. In AIR Binder, PROC ANOVA was used for balanced design (equal numbers of subjects or replicates per cell) and the GLM procedure was used for unbalanced design (unequal numbers of subjects or replicates per cell).

Considering the nature of plasma protein binding assays, most of times unequal numbers of subjects or replicates were derived, and thus PROC GLM was used more routinely in AIR Binder. When both the species- and concentration-dependency are tested for mean free fraction values, the model is written as follows:

$$f_{u,ijk} = \mu + \tau_j + \lambda_k + (\tau\lambda)_{jk} + \varepsilon_{ijk}$$

where µ is the grand mean of all free fraction values, τ_j is the group effect on the free fraction for the jth category of the row variable (such as species), λ_k is the group effect on free fraction for the kth category of the column variable (such as concentration), $(\tau\lambda)_{jk}$ is the interaction effect for the combination of the jth row category and kth column category, and ε_{ijk} is the error term.

Compared with PROC ANOVA, LSMEANS was used to produce least squares in PROC GLM, which is the adjusted means for plasma free fraction of a drug molecule. In general, we also assume that a type III sum of squares (the SS3 option for the MODEL statement in PROC GLM) is suitable for the assay, which gives sum of squares for the effect of each variable (such as species or concentration) evaluated after all other factors have been accounted for. The equations to calculate the F values for the factor A (such as species), factor B (such as concentration), and their interaction effects (AxB) are presented as follows:

$$F_A = \frac{MS_A}{MS_{within}} = \frac{\frac{SS_A}{(K-1)}}{\frac{SS_{total} - SS_{between}}{N - JK}}$$

$$F_B = \frac{MS_B}{MS_{within}} = \frac{\frac{SS_B}{(J-1)}}{\frac{SS_{total} - SS_{between}}{N - JK}}$$

$$F_{A\times B} = \frac{\frac{SS_{A\times B}}{(J-1)(K-1)}}{\frac{SS_{total} - SS_{between}}{N - JK}} = \frac{\frac{SS_{between} - SS_A - SS_B}{(J-1)(K-1)}}{\frac{SS_{total} - SS_{between}}{N - JK}}$$

where MS is mean squares, SS is sum of squares, N is the total number of subjects/samples, J and K are row and column variables to compare, for both between- and within-group variance. Similarly, if significant differences are found from a two-way ANOVA, a post-hoc test is performed to find group difference (Figures 7 and 8).

Species/Concentration (µM)	Dog/0.1	Dog/1	Dog/5	Dog/20	Human/0.1	Human /1	Human /5	Human /20	Rat/0.1	Rat /1	Rat /5	Rat /20
Dog/0.1		0.2969	<.0001	<.0001	0.0211	0.3668	0.0195	<.0001	<.0001	<.0001	0.6157	<.0001
Dog/1	0.2969		0.0171	<.0001	<.0001	0.0005	0.9830	<.0001	<.0001	<.0001	0.0017	0.1015
Dog/5	<.0001	0.0171		0.3150	<.0001	<.0001	0.2722	0.1783	<.0001	<.0001	<.0001	0.9998
Dog/20	<.0001	<.0001	0.3150		<.0001	<.0001	0.0003	1.0000	<.0001	<.0001	<.0001	0.0724
Human/0.1	0.0211	<.0001	<.0001	<.0001		0.9699	<.0001	<.0001	0.0005	0.0002	0.8480	<.0001
Human /1	0.3668	0.0005	<.0001	<.0001	0.9699		<.0001	<.0001	<.0001	<.0001	1.0000	<.0001
Human /5	0.0195	0.9830	0.2722	0.0003	<.0001	<.0001		0.0001	<.0001	<.0001	<.0001	0.7168
Human /20	<.0001	<.0001	0.1783	1.0000	<.0001	<.0001	0.0001		<.0001	<.0001	<.0001	0.0341
Rat/0.1	<.0001	<.0001	<.0001	<.0001	0.0005	<.0001	<.0001	<.0001		1.0000	<.0001	<.0001
Rat /1	<.0001	<.0001	<.0001	<.0001	0.0002	<.0001	<.0001	<.0001	1.0000		<.0001	<.0001
Rat /5	0.6157	0.0017	<.0001	<.0001	0.8480	1.0000	<.0001	<.0001	<.0001	<.0001		<.0001
Rat /20	<.0001	0.1015	0.9998	0.0724	<.0001	<.0001	0.7168	0.0341	<.0001	<.0001	<.0001	

Figure 7 ANOVA post hoc analysis with p values displayed

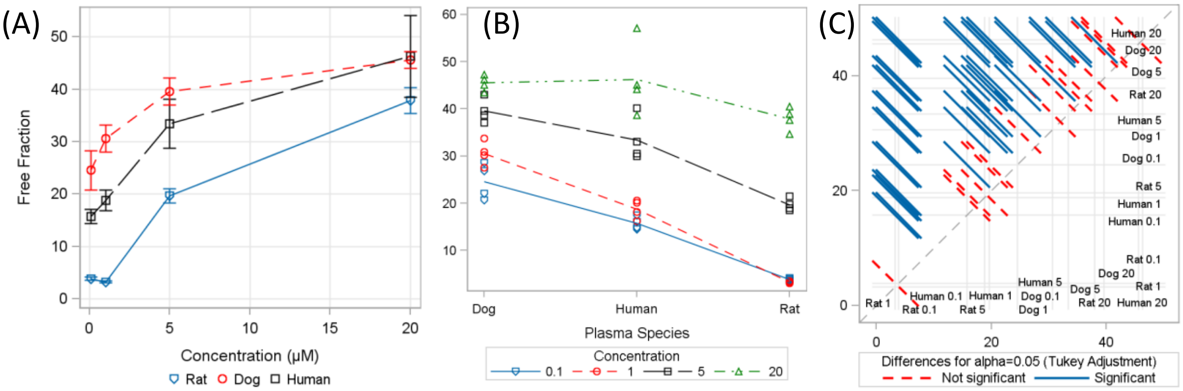


Figure 8 Plasma protein binding plots: (A) dose response; (B) interactions; and (C) pairwise comparison

CYP REACTION PHENOTYPING

CYP reaction phenotyping is a critical in vitro ADME assay for the assessment of potential drug-drug interactions. Fraction contribution to drug metabolism by a specific CYP is required to be determined to calculate the ratio of pharmacokinetic AUC in the presence and absence of the compound, as discussed previously in the “CYP Inhibition and Induction” section. On the other hand, due to the polymorphism of certain cytochrome P450s such as CYP2D6 and several CYP2Cs, the phenotyping assay is also key to the assessment of human pharmacokinetic variations across populations. Two separate assays are commonly used in parallel to quantitate the contribution of individual CYP enzymes: (1) using specific CYP chemical inhibitors with human liver microsomes, and (2) using recombinant cytochrome P450s (rCYP). For each assay, both parent compound depletion and metabolite formation are often monitored.

Due to the complexity of data structure especially special summary table and figure presentations for CYP reaction phenotyping assays, a standardized workflow was designed in AIR Binder (Figure 9). Four sub-processes were integrated into four macros: QPTRR, QPTRC, QPTCR, and QPTCC (Q for queue, PT for phenotyping, the fourth digit R/C for rCYP/chemical inhibitor, and the fifth digit R/C for raw /calculated data). For each sub-process, majority of integrated macros are shared functions, such a READ (read in raw tables), LOOK (format), TYPE (conditional variable selection and re-assignment), R2ID1V (table report for 2 identifier and 1 variable, used for raw data presentation), and REST/PNEE (tables/figure generation for summarized data).

Grouped macros were also created for special data handling, such as KPTTBL, which was used for the presentation of raw data of both rCYP and chemical inhibitor assays that included a series of data handling and calculation processes. On the other hand, for summarized tables and figures, the calculation for the parent depletion and metabolite formation processes was different (e.g. a distinct normalization approach), and thus used different SAS® macros or macro orders/organizations.

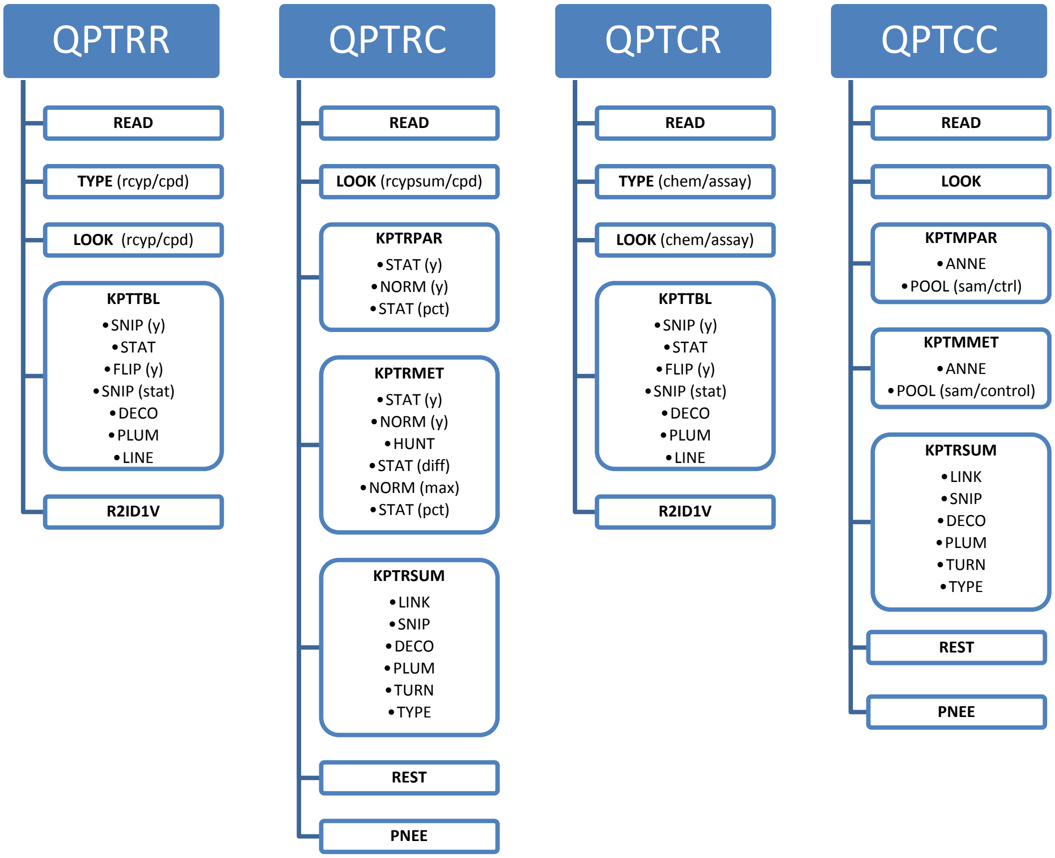


Figure 9 Workflow for CYP reaction phenotyping data handling, analysis, and reporting

Specifically, for the chemical inhibitor assay using human liver microsomes, the percent of metabolite present after an *x*-minute incubation period is calculated relative to that of the plus NADPH control incubation:

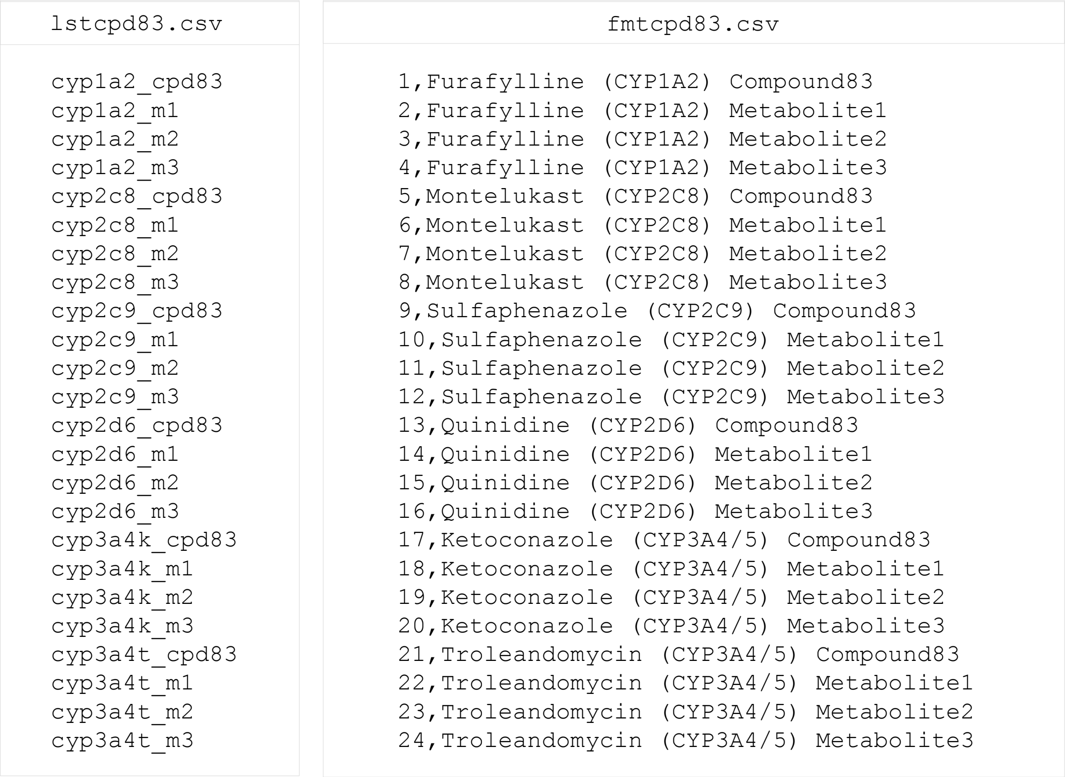
$$\%\ Metabolite\ Present\ _{chemical\ inhibition} = \frac{Metabolite\ at\ t_x}{+NADPH\ Control\ \text{at}\ \text{t}_x} \times\ 100$$

However, for the rCYP assay, the percent of metabolite present after an *x*-minute incubation period is calculated relative to the highest amount observed in any rCYP incubations that need to be determined by ranking a list of data values. Also, additional lookup tables for relative abundance of each rCYP in human liver microsomes need to be integrated beforehand:

$$rCYP\ Metabolite\ Present\ =\ (Metabolite\ at\ t_x - Metabolite\ at\ t_0) \times\ Relative\ Abundance$$

$$\%\ Relative\ Metabolite\ Present\ _{rCYP}\ =\ \frac{rCYP\ Metabolite\ Present}{Highest\ rCYP\ Metabolite\ Present} \times 100$$

From study to study, different CYP enzymes or even non-CYP enzymes are required to be evaluated. At the same time, different numbers (names) of metabolites are to be presented. Considering the challenges of formatting and ordering the identifier variables, an object-oriented design was used to maintain efficient and reusable code. All individual assay-specific information was saved in external files. The advantage of using external files was to minimize code maintenance. For CYP reaction phenotyping, the order of samples was saved in one CSV file ("lstcpd83.csv" in Figure 10), which was read in by a macro TYPE in AIR Binder and was assigned with a numeric order. A paired CSV file ("fmtcpd83.csv" in Figure 10) includes two variables: one numeric column for the sample order, and the other variable for detailed assay information. Then PROC FORMAT (integrated in the macro LOOK in AIR Binder) formatted them for both table and figure presentation.



lstcpd83.csv

```
cyp1a2_cpd83
cyp1a2_m1
cyp1a2_m2
cyp1a2_m3
cyp2c8_cpd83
cyp2c8_m1
cyp2c8_m2
cyp2c8_m3
cyp2c9_cpd83
cyp2c9_m1
cyp2c9_m2
cyp2c9_m3
cyp2d6_cpd83
cyp2d6_m1
cyp2d6_m2
cyp2d6_m3
cyp3a4k_cpd83
cyp3a4k_m1
cyp3a4k_m2
cyp3a4k_m3
cyp3a4t_cpd83
cyp3a4t_m1
cyp3a4t_m2
cyp3a4t_m3
```

fmtcpd83.csv

```
 1,Furafylline (CYP1A2) Compound83
 2,Furafylline (CYP1A2) Metabolite1
 3,Furafylline (CYP1A2) Metabolite2
 4,Furafylline (CYP1A2) Metabolite3
 5,Montelukast (CYP2C8) Compound83
 6,Montelukast (CYP2C8) Metabolite1
 7,Montelukast (CYP2C8) Metabolite2
 8,Montelukast (CYP2C8) Metabolite3
 9,Sulfaphenazole (CYP2C9) Compound83
10,Sulfaphenazole (CYP2C9) Metabolite1
11,Sulfaphenazole (CYP2C9) Metabolite2
12,Sulfaphenazole (CYP2C9) Metabolite3
13,Quinidine (CYP2D6) Compound83
14,Quinidine (CYP2D6) Metabolite1
15,Quinidine (CYP2D6) Metabolite2
16,Quinidine (CYP2D6) Metabolite3
17,Ketoconazole (CYP3A4/5) Compound83
18,Ketoconazole (CYP3A4/5) Metabolite1
19,Ketoconazole (CYP3A4/5) Metabolite2
20,Ketoconazole (CYP3A4/5) Metabolite3
21,Troleandomycin (CYP3A4/5) Compound83
22,Troleandomycin (CYP3A4/5) Metabolite1
23,Troleandomycin (CYP3A4/5) Metabolite2
24,Troleandomycin (CYP3A4/5) Metabolite3
```

Figure 10 External files used in AIR Binder to format identifier variables

In detail, the function LOOK controls format (when the LABEL variable is a character) and informat (when the LABEL variable is a numeric) in AIR Binder. The START variable for format and informat can be either character or numeric, and thus four types of format/informat can be chosen in LOOK including "C" for a character format, "N" for a numeric format, "J" for a character informat, and "I" for a numeric informat. For the previous CYP reaction phenotyping example ("fmtcpd83.csv" in Figure 10), the type was "N" (a numeric format). The format from raw external CSV file was created with the CNTLIN option:

```
%macro look(infile=infile,fmtname=in,type=in);

   data _fmt;
           infile &infile dsd dlm=",";
           retain fmtname "&fmtname" type "&type";
           input start label :$50.;
   run;

   proc format cntlin=_fmt; run;
%mend;
```

In addition, the macro TYPE was used to grab the identifier variable for each CYP reaction phenotyping sample from an external CSV file ("lstcpd83.csv" in Figure 10), and to create a new identifier variable by numeric order. First, the total number of unique identifiers was determined using the CALL SYMPUT routine and %EVAL function with a global macro viable DIMTYPE to store the information. Second, the CALL SYMPUT routine was used to extract each identifier line by line and store them in a &&VAR&I macro variable. Third, the IF/THEN statement was used to look up the identifier by comparing it to the &&VAR&I macro variable, and then to assign a new numeric value in order. TYPE was used in other assays as well to order the identifier variable, and was especially efficient for a lengthy/complicated identifier column.

```
%macro type(lstfile=in,infile=in,id=in,newid=in,outfile=in);

   data _dimtype; infile &lstfile dsd dlm=","; input _id :$30.; run;

   data _null_; set _dimtype end=eof;
           line+1;
           if eof then call symput("dimtypec",line);
   run;

   %let dimtype=%eval(&dimtypec);

   %do t=1 %to &dimtype;
           data &outfile._typeraw&t.;
                  infile &lstfile dsd dlm=",";
                  input _id :$30.; if _n_ =&t. then output; run;

           data _null_; set &outfile._typeraw&t.;
                  call symput('sam'||left(&t),strip(_id)); run;
   %end;

   %let j=0;
   %let outprefix=temp;

   data &outfile._&outprefix.0; set &infile; run;

   %do i=1 %to &dimtype;
           %put air binder typewriter: &&sam&i;

           data &outfile._&outprefix&i;
                  set &outfile._&outprefix&j;
                  if &id = "&&sam&i" then &newid=&i; run;
           %let j=%eval(&j+1);
   %end;

   data &outfile(drop=&id); set &outfile._&outprefix&dimtype; run;

   proc sort data=&outfile; by &newid; run;

%mend;
```

For the application of recombinant CYPs in CYP reaction phenotyping, extrapolation of the data from individual CYP enzymes to human liver microsomes was often required. Although a relative contribution method based on the highest contribution level and CYP protein expression abundance in liver is informative, sometimes, an intersystem extrapolation factor (ISEF) can be determined to facilitate the interpretation of data (Chen 2011):

$$ISEF_{\ V_{max}} = \frac{V_{max,HLM}}{V_{max,rCYP} \times CYP\ Abundance}$$

$$CL_{int} = \frac{V_{max}}{K_m}$$

$$ISEF_{\ CL} = \frac{\frac{V_{max,HLM}}{K_{m,HLM}}}{\frac{V_{max,rCYP}}{K_{m,rCYP}} \times CYP\ Abundance}$$

where ISEF can be calculated using either V_{max}, the maximum rate of reaction, or the intrinsic clearance value (CL_{int}). For the graphic presentation of CYP reaction phenotyping assays, the VBAR statement in PROC SGPANEL is an option. The NEEDLE statement is another option that takes advantage of using bar charts and hidden scatter plots to show error bars freely (Figure 11):

```
proc sgpanel data=cypphenotyping noautolegend;
  panelby metabolite/spacing=5 layout=panel columns=2 onepanel novarname sparse;
  scatter x=cypid y=percent /yerrorupper=upper markerattrs=(size=0 pct);
  needle x=cypid  y=percent /lineattrs=(thickness=18.83 color=vligb);
  format cypid cypidfmt.;
  colaxis label=" " type=discrete;
  rowaxis label="Relative Percentage";
run;
```

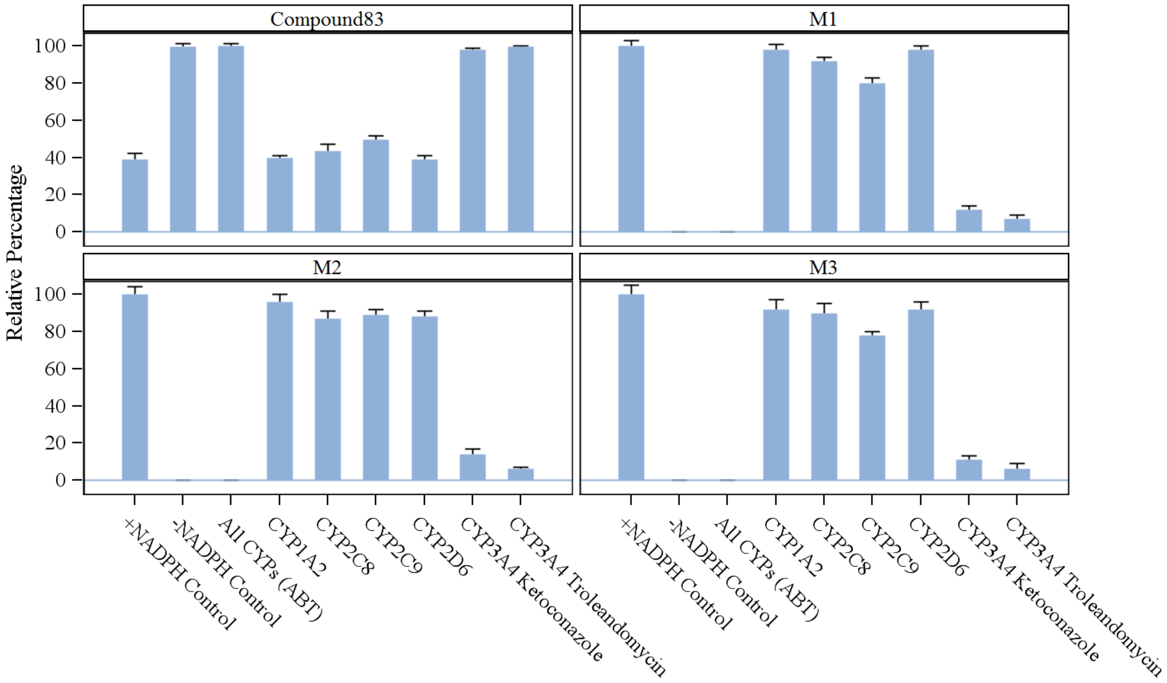


Figure 11 CYP reaction phenotyping graphics using PROC SGPANEL

METABOLIC STABILITY AND INTRINSIC CLEARANCE

Metabolic stability assays using human liver microsomes, S9 fractions (post-mitochondrial supernatant fraction), cytosols, and hepatocytes are used to determine intrinsic clearance of a compound, and thus to understand pharmacokinetic data. As described previously in the “Plasma Protein Binding” section, the AUC$_{unbound}$ is a function of dose and intrinsic clearance:

$$AUC_{unbond} = \frac{\text{dose}}{CL_{in}}$$

where AUC$_{unbound}$ is the overall unbound drug exposure, CL$_{int}$ is the intrinsic clearance that is determined in vitro via two methods in general: (1) “metabolite formation” by determining K$_m$ and V$_{max}$, as described previously in the “CYP Reaction Phenotyping” section; and (2) “parent compound depletion”, also called the “T$_{1/2}$ method” (Obach 1999). The T$_{1/2}$ method was more generally used in both drug discovery and development studies, which measures the first-order rate constant for the depletion of parent compound at a low concentration (such as 1 µM):

$$f(t) = 100 \times e^{-k \times t}$$

where f(t) is the percent remaining of a compound (as normalized by a 0-minute incubation control), *k* is the first-order rate constant. The half-life (t$_{1/2}$) and intrinsic clearance are calculated with the following equations:

$$t_{1/2} = \frac{0.693}{k}$$

$$CL_{int,m} = \frac{0.693}{t_{1/2} \times C_m}$$

where CL$_{int,m}$ is the intrinsic clearance as determined by a parent depletion method (mL/min/mg) using microsomes, C$_m$ is the concentration of microsomal protein in the incubation (mg/mL), and *k* is the first-order rate constant. Two options are available for data fitting: linear or non-linear regression. If the linear regression is used (PROC SGPLOT with the REG statement), the percent parent compound remaining was transformed to logarithm first (Figure 12). In that case, the linear fitting becomes:

$$f(t)_{natural\ logarithm} = 4.61 - k \times t$$

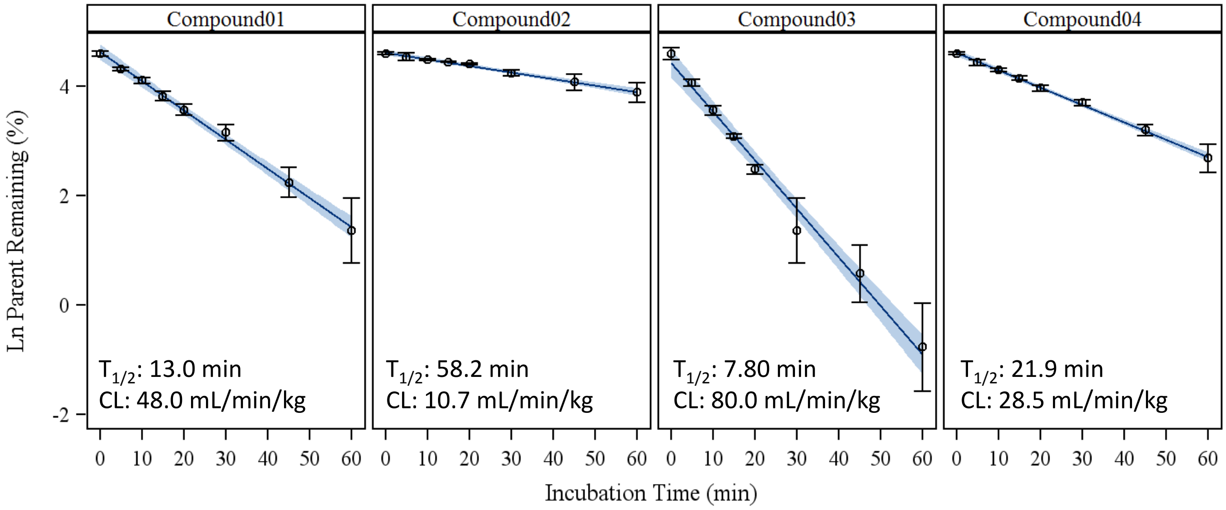


Figure 12 Microsomal intrinsic clearance determined by linear regression (SGPANEL/REG)

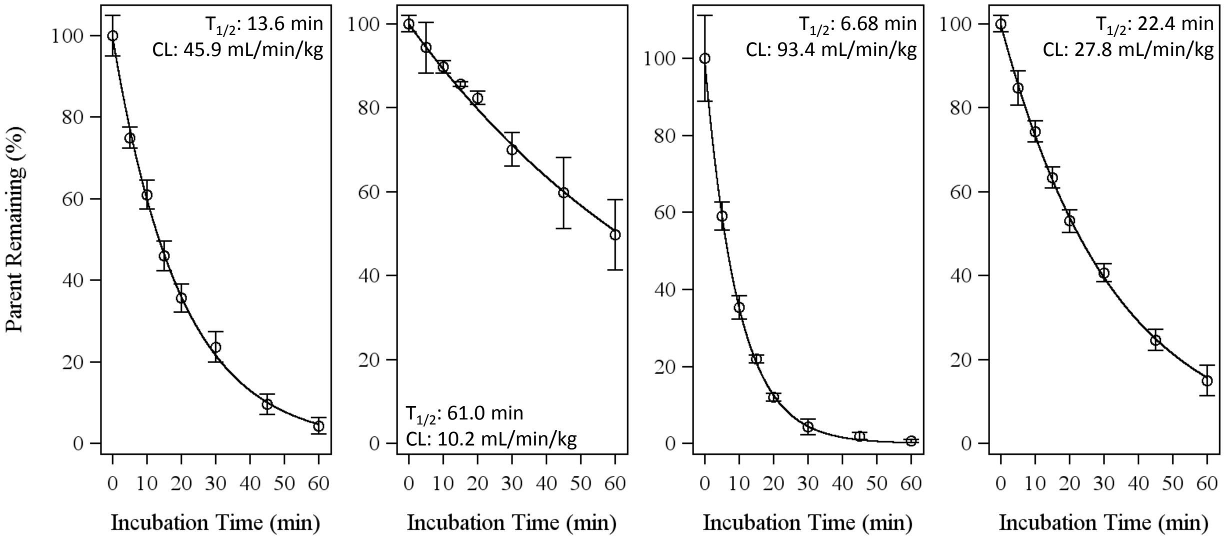


Figure 13 Microsomal intrinsic clearance determined by nonlinear regression (SGPLOT/NLIN)

Similarly, the intrinsic clearance can be determined by direct fitting the exponential equation using PROC NLIN with SGPLOT (Figure 13). The $t_{1/2}$ values for these compounds as determined by two methods were close; however, the data presentation especially the standard deviation rendered dramatic different look due to the scale of y-axis. The default REG procedure in SAS® for linear regression provides extensive information on the fitting performance, which is useful in some situation such as the evaluation of lack of fitting without additional programming (compared with the non-linear fitting using PROC NLIN).

The intrinsic clearance is scaled using the following equation as presented within the insets of Figures 12 and 13:

$$CL_{int,m,scaled} = \frac{0.693}{t_{1/2} \times C_m} \times \rho_m$$

where $CL_{int,m,scaled}$ is the scaled intrinsic clearance by body weight as determined by parent compound depletion in a time course (mL/min/kg), C_m is concentration of microsomal protein in the incubation (mg/mL), and ρ_m is the ratio of milligram microsomes per kilogram of body weight (a constant, for example, approximately 900 for human liver microsomes).

In addition, the intrinsic clearance for hepatocytes is determined similarly:

$$CL_{int,h} = \frac{0.693}{t_{1/2} \times C_h}$$

$$CL_{int,h,scaled} = \frac{0.693}{t_{1/2} \times C_h} \times \rho_h \times 0.00006$$

where $CL_{int,h}$ is the intrinsic clearance as determined by a parent depletion method in hepatocytes (µL/min/million cells of hepatocyte), C_h is concentration of hepatocytes (million cells/µL), ρ_h is the ratio of million hepatocytes per kilogram of body weight (a constant, for example, approximately 2880 for human hepatocytes), and $CL_{int,h,\ scaled}$ is often expressed as L/h/kg, and thus the equation is multiplied by 0.00006.

METABOLITE PROFILING, IDENTIFICATION, AND PHARMACOKINETICS

According to FDA's "*Guidance for Industry: Safety Testing of Drug Metabolites*", metabolites should be profiled and compared between human and other species used in nonclinical safety assessment. Cross-species metabolite profiling and metabolite identification using hepatocytes is a common preclinical assay that is often required for IND application. The purpose of the assay is to avoid late-stage discovery of potentially problematic drug metabolites (i.e, unique human metabolites or metabolites with disproportionately greater human exposure, as defined in the aforementioned guidance document).

Developing useful visualization tools for the metabolite profiling and comparison is important to efficiently convey information for quicker decision making. SAS® ODS provides powerful solutions to dynamically visualize these data to enhance communications within study teams, between CRO and clients. Stacked bar chart (Figure 14) using PROC SGPLOT (the VBAR statement with the GROUP option) was a solution to compare abundance of metabolites visually, which might be more efficient when the numbers of metabolites are limited.

When many metabolites were involved for comparison especially with a large range of abundance, paneled bar charts using PROC SGPANEL (the VBAR statement with the option LAYOUT=ROWLATTICE) provided better visualization effects (Figure 15). In addition, paneled plots using PROC SGPANEL (both SCATTER and SERIES statements with the GROUP option) were useful to demonstrate time-course formation of metabolites in vivo (Figure 16).

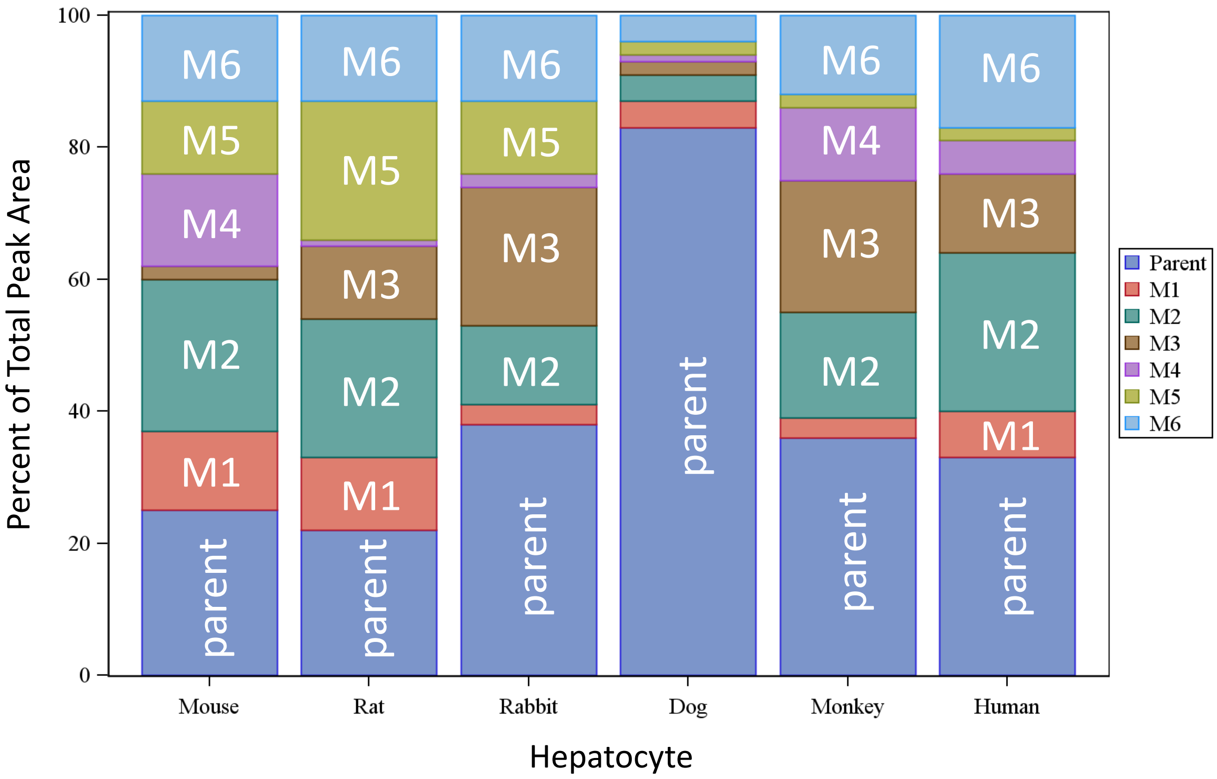


Figure 14 Stacked bar chart for cross-species metabolite profiles in hepatocytes

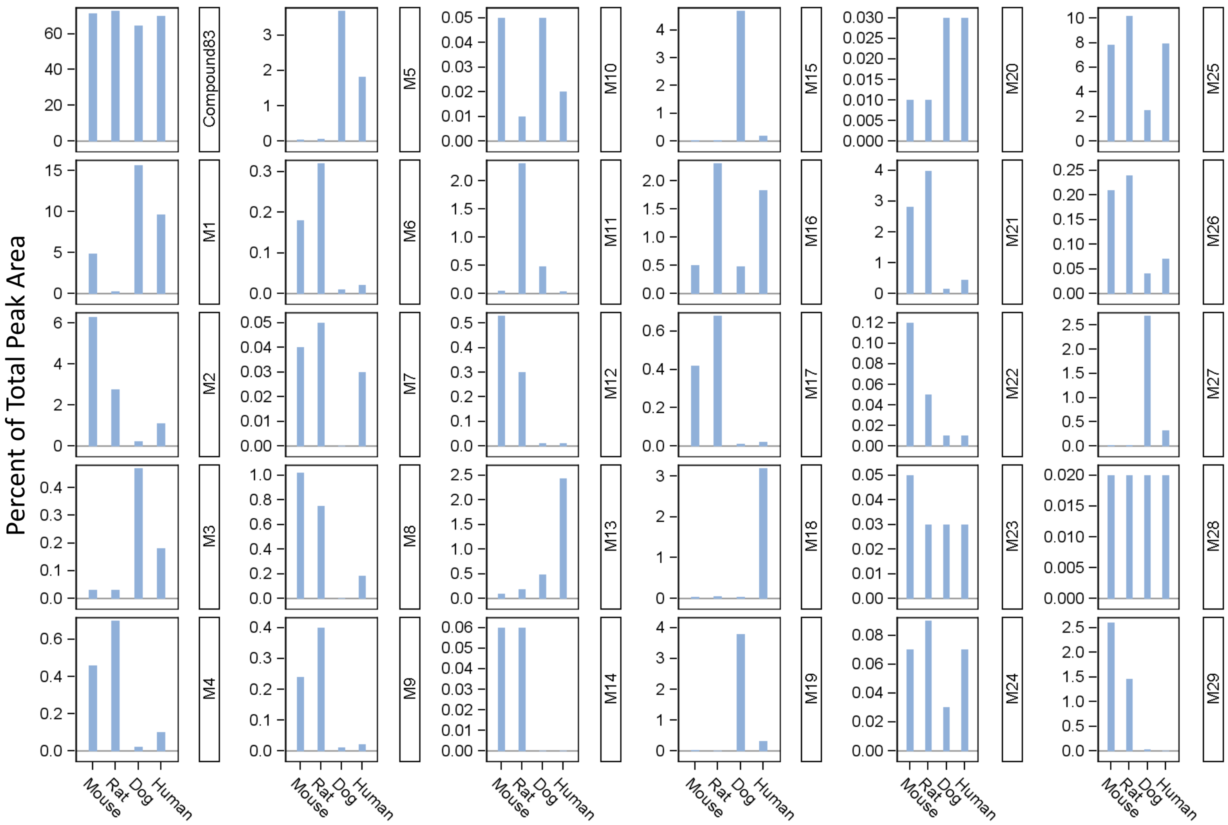


Figure 15 Paneled bar chart for cross-species metabolite profiles in hepatocytes

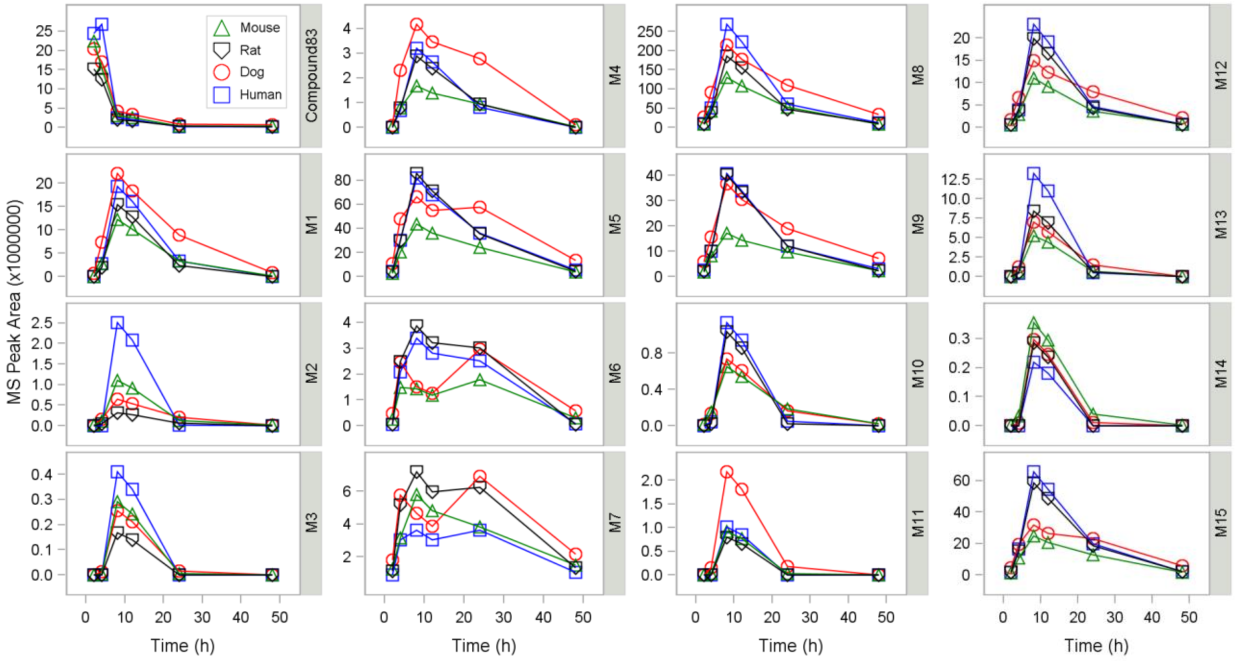


Figure 16 Time-course paneled plot for cross-species metabolite profiles in vivo

In addition, paneled box and whisker plots (Figure 17) implemented in AIR Binder provided informative graphics to compare metabolite formation profiles between patient groups, which demonstrated the distribution of results with outliers, a statistic advantage using SAS® as a platform.

For metabolites, like parent compound, one important parameter required to be calculated for their pharmacokinetics is the area under the concentration-time curve (AUC). The linear trapezoidal rule is the primary method for the estimation:

$$AUC_{total,i.v.} = \int_0^\infty C \times dt = C_0 \times \int_0^\infty e^{-k\times t} \times dt = \frac{C_0}{k}$$

where $AUC_{total,i.v.}$ is the total AUC when the drug is given as an intravenous bolus, C_0 is the zero-time concentration, and *k* is the elimination rate constant. For the elimination phase of an oral dose, the AUC can be calculated with the trapezoidal rule as follows:

$$C_{t_{i+1}} = C_{t_i} \times e^{-k_i \times \Delta t_i}$$

$$AUC_i = \int_0^\infty C_{t_{i+1}} \times dt - \int_0^\infty C_{t_i} \times dt = \frac{C_{t_{i+1}} - C_{t_i}}{k_i} = \frac{(C_{t_{i+1}} - C_{t_i}) \times \Delta t_i}{\ln(\frac{C_{t_i}}{C_{t_{i+1}}})}$$

where AUC_i is the AUC during the time interval from Δt_i to Δt_{i+1}, and k_i is the elimination rate constant. When the decline is exponential, a log trapezoidal method is more accurate and thus was used accordingly. For example, across patients the equal-volume time-point pooled samples were analyzed and followed by AUC calculation in AIR Binder. The paneled plot was again efficient to display these results, together with curve overlay (Figures 18 and 19).

Additionally, another usage of AIR Binder was reconstructing chromatograms between different LC/MS systems for metabolite profiling. Figure 20 is an example of reconstructing the time-slice profiles of metabolites from accelerator mass spectrometry (AMS), a technology used for micro tracer metabolite profiling studies.

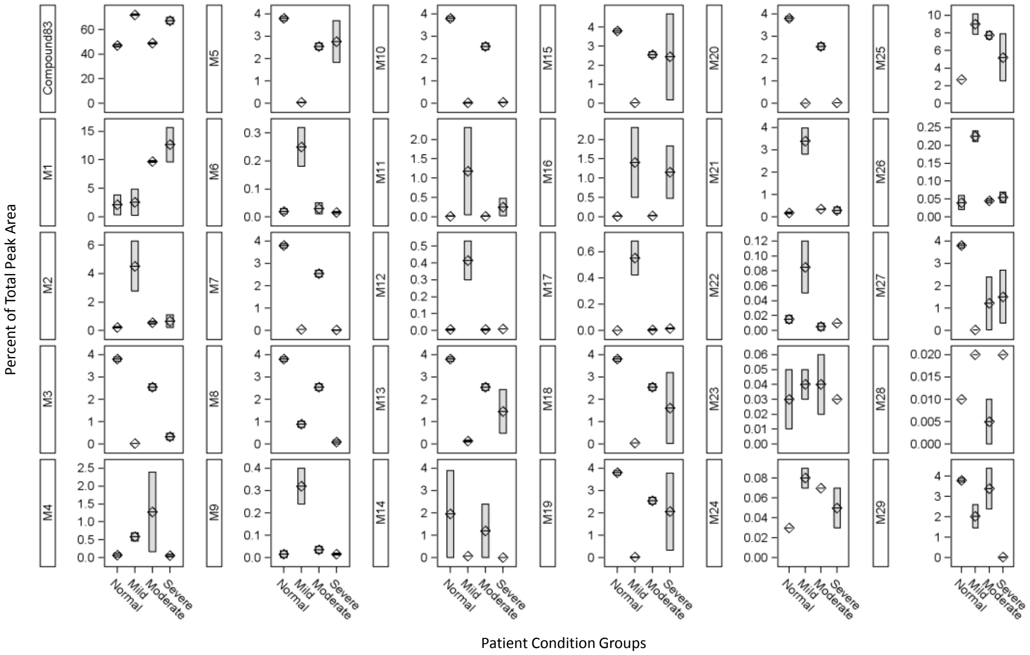


Figure 17 Paneled box and whisker plot to compare metabolite profiles between patient groups

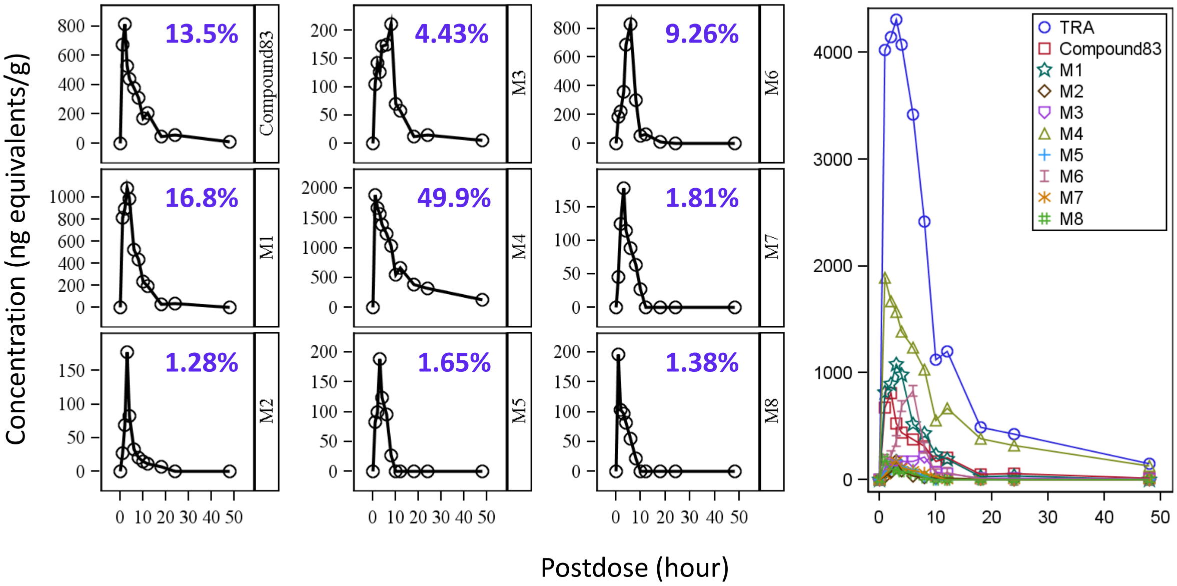


Figure 18 Pharmacokinetics of metabolites with AUC normalized by total radioactivity

Metabolite Designation	Postdose (Hours)											$AUC_{0\text{-}48h}$	
	1	2	3	4	6	8	10	12	18	24	48	hour×ng*(eq)*/g	%AUC
M1	**816**	899	1080	983	521	432	233	193	28.1	33.8	3.88	**8140**	**16.8**
M2	27.6	68.9	178	83.0	33.2	20.3	14.5	11.8	6.91	0	0	**623**	**1.28**
M3	105	143	126	172	175	211	70.1	58.2	12.9	15.5	5.38	**2150**	**4.43**
M4	1890	1670	1570	1390	1240	1030	554	667	388	322	126	**24200**	**49.9**
M5	82.4	99.3	189	124	95.0	26.6	0	0	0	0	0	**800**	**1.65**
M6	185	223	361	689	830	300	54.0	65.0	9.13	0	0	**4490**	**9.26**
M7	45.0	125	178	115	88.8	63.0	26.8	0	0	0	0	**878**	**1.81**
M8	196	104	96.5	82.0	54.6	21.3	0	0	0	0	0	**671**	**1.38**
Compound83	676	814	528	438	378	314	171	206	48.6	58.5	12.0	**6540**	**13.5**

Figure 19 Time-course metabolite formation table with AUC normalized by total radioactivity

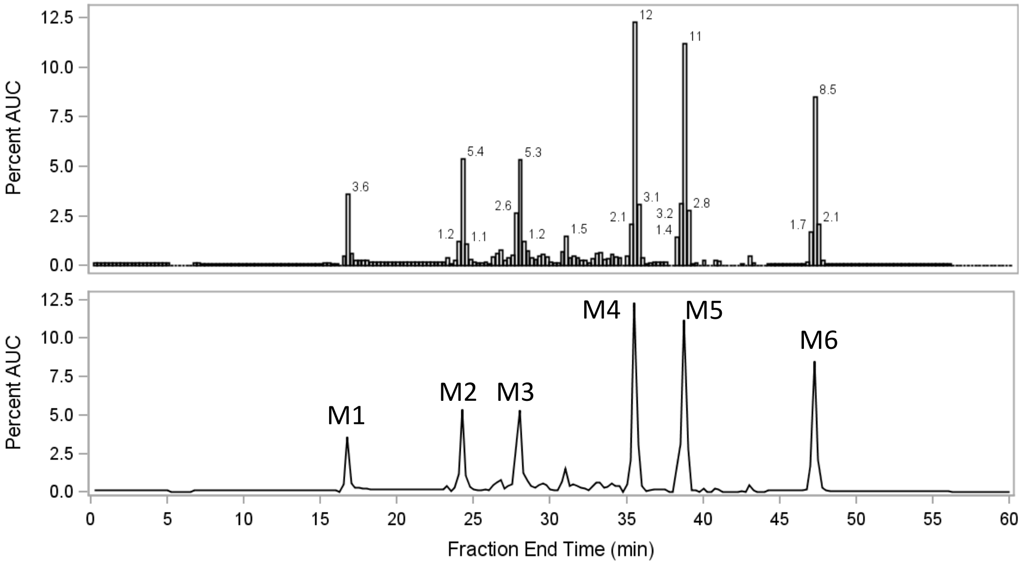


Figure 20 Chromatogram reconstruction using metabolite profiles from time slices in AMS

CONCLUSION

SAS® is a powerful and flexible platform for the development of customized applications on data analysis, visualization, and reporting of preclinical and clinical pharmacokinetic and drug metabolism assay data. A SAS®-based application called AIR Binder with expanded capacity to analyze and visualize diverse datasets for in vitro and in vivo ADME assays, was developed. Various customized ODS panel plots were extensively implemented to visualize profiles of metabolites for cross-species comparison, pharmacokinetics, and all other ADME assays, which turned out to be a powerful tool to convey information within study teams and also between CRO and clients, for enhanced communication and quick decision making. Customized linear and nonlinear fitting using PROC REG and NLIN for CYP inhibition and induction assays (IC_{50} and EC_{50} determination with a 4-parameter logistic model), metabolic stability ($t_{1/2}$ determination), and pharmacokinetics (non-compartment and compartment models) was another important feature for AIR Binder to enhance the quality of presentation and communication. In addition, with powerful statistical features from SAS®, AIR Binder handles all kinds of statistics in a streamlined fashion. For example, PROC GLM was applied for comprehensive statistical analysis on plasma protein binding data. Overall, AIR Binder dynamically and efficiently visualized ADME assay data, shortened reporting turnaround time significantly for drug metabolism projects in drug discovery and development. With the current designed infrastructure of AIR Binder using object-oriented concepts to organize SAS® macros for code share and reuse between assays, it is convenient to expand the program with the integration of new drug metabolism assay types for data analysis and visualization.

ACKNOWLEDGMENTS

The authors would like to thank Don McKenzie for his support on development and application of AIR Binder. The authors are also grateful to feedback from our clients and friends across pharmaceutical and biotech companies.

REFERENCES

Carpenter, A., 2004. *Carpenter's Complete Guide to the SAS Macro Language*. 2nd edition. SAS Publishing.

Chen, Y., Liu, L., Nguyen, K. and Fretland, A.J. 2011. “Utility of intersystem extrapolation factors in early reaction phenotyping and the quantitative extrapolation of human liver microsomal intrinsic clearance using recombinant cytochromes P450.” *Drug Metabolism and Disposition*, 39: 373-82.

Einolf, H.J., Chen, L., Fahmi, O.A., Gibson, C.R., Obach, R.S., Shebley, M., Silva, J., Sinz, M.W., Unadkat, J.D., Zhang, L., and Zhao, P. 2014. "Evaluation of various static and dynamic modeling methods to predict clinical CYP3A induction using in vitro CYP3A4 mRNA induction data." *Clinical Pharmacology and Therapeutics*, 95:179-88.

Liu, X., Wright, M. and Hop C.E.C.A. 2014, "Rational use of plasma protein and tissue binding data in drug design." *Journal of Medicinal Chemistry*, 57:8238-48.

Obach, R.S. 1999. “Prediction of human clearance of twenty-nine drugs from hepatic microsomal intrinsic clearance data: An examination of in vitro half-life approach and nonspecific binding to microsomes.” *Drug Metabolism and Disposition*, 27: 1350-9.

Ortiz de Montellano, P.R. 2004. *Cytochrome P450: Structure, Mechanism, and Biochemistry*. 3rd edition. Springer.

Sun, H., Cardinal, K., Kirby, C., and Voorman, R. 2017. “AIR Binder: an automatic reporting and data analysis SAS® application for cytochrome P450 inhibition assay to investigate DDI.” *PharmaSUG* Paper PO09.

ABBREVIATIONS

ADME, Absorption, Distribution, Metabolism and Excretion; AMS, Accelerator Mass Spectrometry; ANOVA, Analysis of Variance; AUC, Area Under the Concentration-Time Curve; CRO, Contract Research Organization; CYP, Cytochrome P450; DDI, Drug-Drug Interaction; HLM, Human Liver Microsomes; IND, Investigational New Drug; ISEF, Intersystem Extrapolation Factor; LC/MS, Liquid Chromatography/Mass Spectrometry; NCE, New Chemical Entity; NDA, New Drug Application; ODS, Output Delivery System; OOP, Object-Oriented Programming; QC, Quality Control; rCYP, Recombinant Cytochrome P450s; SOP, Standard Operating Procedure

CONTACT INFORMATION

Hao Sun, PhD
Drug Metabolism and Biotransformation
Covance, Inc.
3301 Kinsman Blvd, Madison, Wisconsin 53704
Email: hao.sun@covance.com
Phone: 608.241.7294

Kristen Cardinal, BS
Drug Metabolism and Biotransformation
Covance, Inc.
3301 Kinsman Blvd, Madison, Wisconsin 53704
Email: kristen.cardinal@covance.com
Phone: 608.230.1333