

The Ultimate Update to ADaM Specification Template that Simplifies ADaM Programming and Creation of Define XML

March 6, 2023

Xiang Wang, Yuanyuan Liu, Ryan Krause, Bristol-Myers Squibb
Daniel Huang, BeiGene

Introduction

Define.xml is required when submitting data packages to regulatory agencies like FDA, PMDA to inform the regulators which datasets, variables, controlled terms and other metadata were used. **With the help of Pinnacle 21 Enterprise (P21E), define.xml can be created from P21E compatible specifications (P21E define spec).** There are two ways of creating P21E define spec:

- Create P21E define spec from company standard ADaM spec: easy for creating and reviewing spec, while require complicated SAS macros and manual copy/paste process to convert the ADaM spec into P21E define spec.
- Use [P21 Excel Template for Define.xml | \(pinnacle21.net\)](https://pinnacle21.net) from the very beginning: easy for downstream Define.xml creation but not user-friendly for ADaM programming because programmers need to spend tremendous time and effort to fill out the complicated specification *with many columns and complexity/dependencies across columns/tabs for P21E define spec creation.*

Maior tabs in P21E define spec

Datasets	Variables	ValueLevel	WhereClauses	Codelists	Dictionaries	Methods	Comments
----------	-----------	------------	--------------	-----------	--------------	---------	----------

Introduction - continued

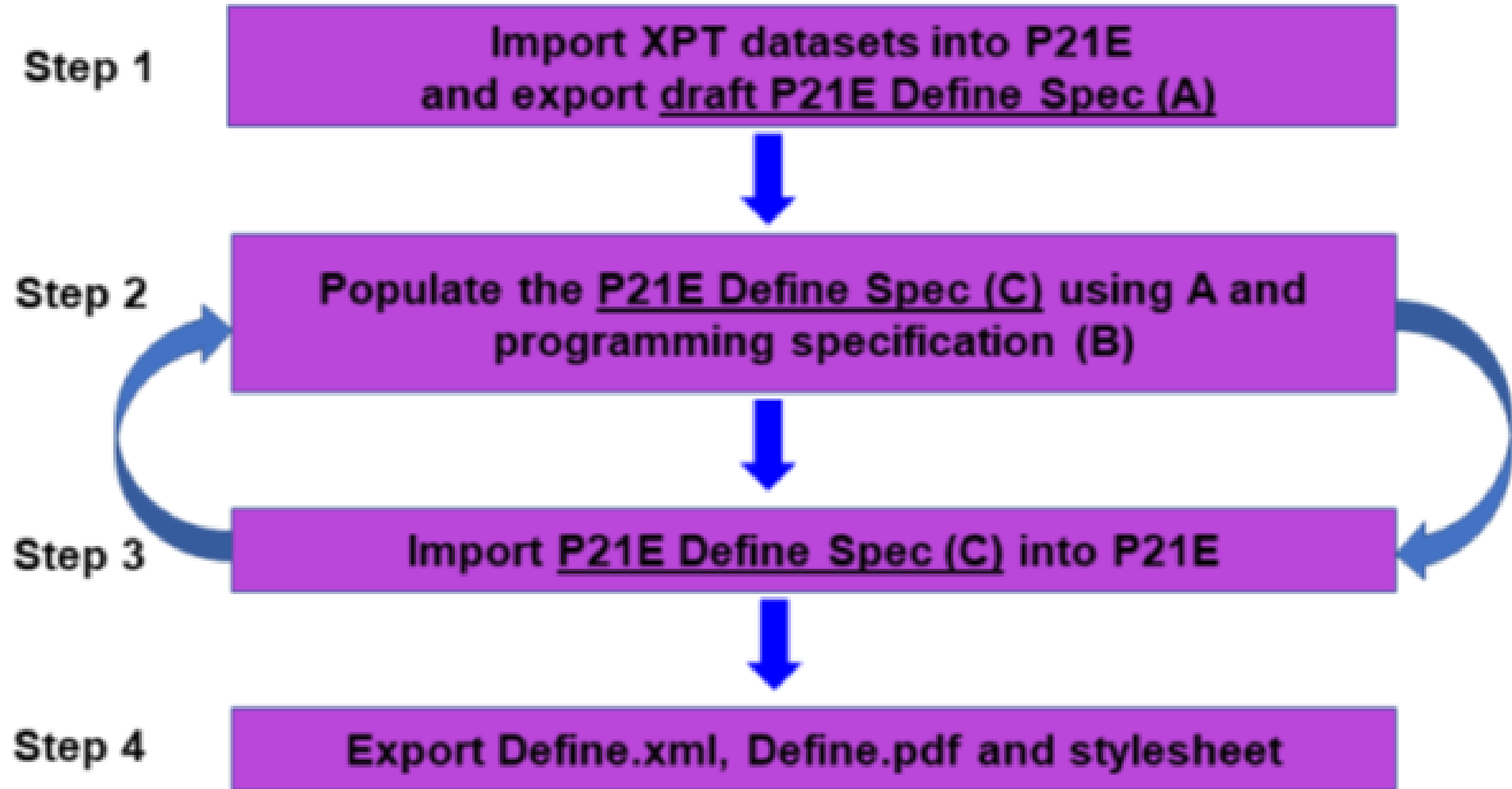
- Inspired by the fact that P21E can automatically populate the four time-consuming tabs (**Variables**, **Codelists**, **ValueLevel** and **WhereClauses**) in define specification with quality after importing ADaM transport (.XPT) datasets
- This presentation is a major update to our previous PharmaSUG paper (Xiang Wang, Daniel Huang. 2021. Design ADaM Specification Template that Simplifies ADaM Programming and Creation of Define XML in CDISC Era. PharmaSUG, SS-016)
- The updated ADaM specification template in this paper will greatly **simplify** the effort for statistical programmers to create **ADaM datasets** and **Define.xml** with simpler and more user-friendly features

Old vs New ADaM specs

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
Order	Dataset	Variable	Label	Data Type	Length	Significant Digits	Format	Mandatory	Codelist	Origin	Pages	Method	Predecessor	Role	Comment	Source/Derivation	Comment Description
▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼

A	B	C	D	E	F
Variable	Label	Data Type	Length	Origin	Source/Derivation
▼	▼	▼	▼	▼	▼

RECAP: FOUR STEPS OF CREATING DEFINE.XML UNDER P21E



NEW ADAM SPECIFICATION TEMPLATE (Spec B)

The newly designed ADaM specification template has the following tabs: **Study**, **Datasets**, **Documents**, **Dictionaries**, and **individual dataset tab** (ADSL, ADAE, ADBASE etc) shown below

Study	Datasets	Documents	Dictionaries	ADSL	ADBASE	ADAE	ADCM	ADMH	ADLB
-------	----------	-----------	--------------	------	--------	------	------	------	------

The first four tabs (**Study**, **Datasets**, **Documents**, and **Dictionaries**) are quite straightforward, and the information can be entered according to the instruction from P21E user guide.

Study tab: example is shown right.

A	B
Attribute	Value
StudyName	XXX-XXXX
StudyDescription	XXXXXXXXXXXXXXXXXX
ProtocolName	XXX-XXXX
StandardName	ADaM-IG
StandardVersion	1.1
Language	en

NEW ADAM SPECIFICATION TEMPLATE (Spec B)

Datasets tab: example is shown below.

A	B	C	D	E	F	G	H	I
Dataset	Description	Class	Structure	Purpose	Key Variables	Repeating	Reference Data	Comment
ADSL	Subject-Level Analysis Dataset	SUBJECT LEVEL ANALYSIS DATASET	One record per study per subject	Analysis	STUDYID, USUBJID	No	No	
ADBASE	Baseline Analysis Data	ADAM OTHER	One record per study per subject	Analysis	STUDYID, USUBJID	No	No	
ADAE	Adverse Events Analysis Data	OCCURRENCE DATA STRUCTURE	One record per study per subject per preferred term per verbatim term per analysis start date per analysis end date	Analysis	STUDYID, USUBJID, AESPID, AEDECOD, AETERM, ASTDT, AENDT	Yes	No	
ADCM	Concomitant Medications Analysis Data	OCCURRENCE DATA STRUCTURE	One record per study per subject per category per indication per reported medication name per standardized medication name per start date per end date	Analysis	STUDYID, USUBJID, CMCAT, CMINDC, CMTRT, CMDECOD, CMSTDTC, CMENDTC, CMSPID	Yes	No	
ADLB	Lab Analysis Data	BASIC DATA STRUCTURE	One record per study per subject per analysis parameter per analysis timepoint	Analysis	STUDYID, USUBJID, PARAMCD, ADT, VISITNUM, ASEQ, LBSTRESU, AVAL	Yes	No	
ADMH	Medical History Analysis Data	OCCURRENCE DATA STRUCTURE	One record per study per subject per verbatim term per start date per end date per sponsor-defined identifier	Analysis	STUDYID, USUBJID, MHTERM, MHSTDTC, MHENDTC, MHSPID	Yes	No	

Documents tab: example is shown below.

A	B	C
ID	Title	Href
complexAlgorithm	Complex Algorithm	complexalgorithm.pdf
ADRG	Analysis Data Reviewer's Guide	adrg.pdf

Dictionaries tab: example is shown below.

A	B	C	D	E
ID	Name	Data Type	Dictionary	Version
AEDICT	Adverse Event Dictionary	text	MedDRA	8
MedDRA	Medical Dictionary for Regulatory Activities	text	MedDRA	24

NEW ADAM SPECIFICATION TEMPLATE (Spec B)

- **ADSL** tab
 - 1) Only **six** columns are required (add **Common Variable** (column **G**) for ADSL)
 - 2) Simplify **Data Type** (column **C**) into **Char** and **Num**
 - 3) Variable **Length** (column **D**) can be 200 for all **Char** variables and 8 for **Num** variables (length can be trimmed during each dataset creation)
 - 4) Write simple SAS macro to read each dataset tab to get **variables** and **attributes**

A	B	C	D	E	F	G
Variable	Label	Data Type	Length	Origin	Source/Derivation	Common Variable
STUDYID	Study Identifier	Char	200	Predecessor	DM.STUDYID	Y
USUBJID	Unique Subject Identifier	Char	200	Predecessor	DM.USUBJID	Y
SUBJID	Subject Identifier for the Study	Char	200	Predecessor	DM.SUBJID; SUBJID is required in ADSL, but permissible in other ADaM datasets	
SITEID	Study Site Identifier	Char	200	Predecessor	DM.SITEID	Y
INVTNAM	Investigator Name	Char	200	Predecessor	DM.INVTNAM	
SITEGR1	Pooled Site Group 1	Char	200	Derived	Description of grouping/pooling clinical sites for analysis purposes	
SITEGR1N	Pooled Site Group 1 (N)	Num	8	Assigned	The numeric one-to-one mapping between SITEGR1 and SITEGR1N	
COUNTRY	Country	Char	200	Predecessor	DM.COUNTRY	
COUNGR1	Country Group 1	Char	200	Derived	Description of grouping/pooling countries for analysis purposes	
COUNGR1N	Country Group 1 (N)	Num	8	Assigned	The numeric one-to-one mapping between COUNGR1 and COUNGR1N	
REGION1	Geographic Region 1	Char	200	Derived	Description of grouping/pooling geographical regions for analysis purposes such as 'Asia', 'Europe', 'North America', 'Rest of World'	
REGION1N	Geographic Region 1 (N)	Num	8	Assigned	The numeric one-to-one mapping between REGION1 and REGION1N	
BRTHDTC	Date/Time of Birth	Char	200	Predecessor	DM.BRTHDTC	
BRTHDT	Date of Birth	Num	8	Derived	Numeric version of BRTHDTC; Note: if BRTHDTC is partial or missing, set BRTHDT to missing without	
AGE	Age	Num	8	Predecessor	DM.AGE	
AGEU	Age Units	Char	200	Predecessor	DM.AGEU	
AAGE	Analysis Age	Num	8	Derived	Age used for analysis that may be derived differently than DM.AGE. AAGE is	

NEW ADAM SPECIFICATION TEMPLATE (Spec B)

Other ADaM tabs such as ADAE, ADCM, ADMH

A	B	C	D	E	F
Variable	Label	Data Type	Length	Origin	Source/Derivation
AESEQ	Sequence Number	Num	8	Predecessor	AE.AESEQ
AETRTEM	Treatment Emergent Flag	Char	1	Predecessor	SUPPAE.QVAL where SUPPAE.QNAM='AETRTEM'
TRTEMFL	Treatment Emergent Analysis Flag	Char	1	Derived	TEAEs are defined as AEs that begin on/after the first XXX or non-conforming product infusion through XX days after the infusion end date (do not include AEs after other anti-cancer treatment). An AE with completely missing onset and stop dates, or with the onset date missing and a stop date later than the XXX infusion date, will be considered to be treatment-emergent. For non-missing ADSL.TRTSDT, set TRTEMFL='Y' if (1) both ADAE.ASTDT and ADAE.AENDT are missing ; OR (2) ADSL.ASTDT is missing and ADAE.AENDT is on or later than XXX infusion date; OR (3) ADSL.TRTSDT <= ADAE.ASTDT <= min(ADSL.TRTEDT + XX, ADSL.SACTDT);
TREMxxFL	Treatment Emergent Period xx Flag	Char	1	Derived	This variable is required if there are multiple periods where treatment emergence is a key feature of the analysis for each period. If TREMxxFL is included, TRTEMFL is defined as the overall treatment-emergent flag.
TRTEMwFL	Treatment Emergent Analysis w Flag	Char	1	Derived	This variable is used if there are other analysis needs (e.g., different cut-offs) where treatment emergence is a key feature of the analysis. If TREMwFL is included, TRTEMFL is defined as the overall treatment-emergent flag.

How to Populate P21E Define Spec (C) using draft P21E Define Spec (A) and programming specification (B)

- P21E Define Spec (C) below has the same structure as draft P21E Define Spec (A)

Study	Datasets	Variables	ValueLevel	WhereClauses	Codelists	Dictionaries	Comments	Documents	Methods
-------	----------	-----------	------------	--------------	-----------	--------------	----------	-----------	---------

- Programming specification (B) should follow the same structure as the **NEW ADAM SPECIFICATION TEMPLATE**
- **Task:** populate correct information from **A** and **B** into **C**
- ✓ The 4 tabs (Study, Datasets, Dictionaries, Documents) can be directly copied from **B**
- ✓ P21E can populate these three time-consuming tabs (**Codelists**, **ValueLevel** and **WhereClauses**) in **A** well - directly copy from **A** (may need minor modifications);
- ✓ Use simple **SAS code** to automate the population of tabs (**Variables**, **Methods**, **Comments**) from **A** and **B**

How to populate tabs ‘Variables’, ‘Methods’, ‘Comments’

Variables tab: P21E populates these first columns quite well 

Order	copy this column from A into C	
Dataset Name	copy this column from A into C	
Variable	copy this column from A into C	
Label	copy this column from A into C	
Data Type	copy this column from A into C	
Length	copy this column from A into C	
Significant Digits	copy this column from A into C	
Format	copy this column from A into C	
Mandatory	copy this column from A into C	
Codelist	copy this column from A into C	

Use SAS code to automatically populate these columns from B:
Origin, Method, Predecessor, Comment

Use SAS code to automatically populate these two time-consuming tabs from B:
Methods, Comments

SAS code to populate tabs 'Variables', 'Methods', 'Comments'

```
***** Read the draft P21E Define Spec (A)
    exported from P21 immediately after importing xpt datasets *****/
proc import datafile= "&path.\A.xlsx"
    out=exported_variables DBMS=xlsx replace;
    sheet="Variables";
    getnames=YES;

run;

/*create a dataset domain which contains all domain names*/
proc sql;
    create table domain as select distinct dataset from exported_variables;
quit;

/** Read programming spec (B) - the same structure as NEW ADAM SPECIFICATION TEMPLATE ***/
/** create one sas dataset for each domain ***/
%macro data_extract(dataset=);
proc import datafile= "&path.\B.xlsx"
    out=adam_variables_&dataset DBMS=xlsx replace;
    sheet="&dataset";
    getnames=YES;

run;

data adam_variables_&dataset;
    set adam_variables_&dataset;
    dataset = "&dataset";
run;
%mend;

data _null;
    set domain;
    call execute('%data_extract(dataset='||strip(dataset)||')');
run;
```

SAS code to populate tabs 'Variables', 'Methods', 'Comments'

```
/****** FINAL P21E Define Spec (C) for the tab 'Variables' get columns (Origin, Method,
Predecessor, Comment) and get two tabs (Methods, Comments) *****/
/******/
data adam_variables_(keep=dataset variable origin Source_Derivation common_variable id);
  length id $20;
  set adam_variables_;
  if missing(variable) then delete;
  id=strip(dataset) ||'.'|| strip(variable);
run;
proc sort; by dataset variable; run;

/****** get common (core) variables from programming spec (B) *****/
data commonVar;
  set adam_variables_(where=(common_variable='Y'));
  keep id dataset variable;
run;
proc sql noprint;
  select '''||strip(variable)||''' into :commVar separated by " " from commonVar;
quit;

data exported_variables_(drop=order_);
  length id $20;
  set exported_variables(rename=(order=order_) keep=order dataset variable);
  order=input(order_, best12.);
  id=strip(dataset) ||'.'|| strip(variable);
run;
proc sort; by dataset variable; run;
```

SAS code to populate tabs 'Variables', 'Methods', 'Comments'

```
data variables_;  
  merge exported_variables_(in=a)  adam_variables_(in=b);  
  by dataset variable;  
  if a;  
  
  if dataset='ADSL' then do;  
    if origin='Predecessor' then Predecessor=Source_Derivation;  
    if origin='Assigned' then Comment=strip(dataset) ||'.'|| strip(variable);  
    if origin='Derived' then Method=strip(dataset) ||'.'|| strip(variable);  
  end;  
  
  if dataset ^= 'ADSL' then do;  
    if variable in (&commVar.) then do; origin='Predecessor'; Predecessor='ADSL' ||'.'|| strip(variable); end;  
    else do;  
      if origin='Predecessor' then Predecessor=Source_Derivation;  
      if origin='Assigned' then Comment=strip(dataset) ||'.'|| strip(variable);  
      if origin='Derived' then Method=strip(dataset) ||'.'|| strip(variable);  
    end;  
  end;  
run;  
proc sort; by dataset order; run;
```

SAS code to populate tabs 'Variables', 'Methods', 'Comments'

```

/*****
/*****      Methods tab      *****/
/*****/

data methods;
    length name type $50 description $2000;
    set variables_ (where=(origin='Derived'));
    name='Algorithm to derive ' || strip(id);
    type='Computation';
    description=Source_Derivation;
    keep id name type dataset variable description;

run;

proc sort; by dataset variable; run;

/*****
/*****      Comments tab      *****/
/*****/

data comments;
    length description $2000;
    set variables_ (where=(origin='Assigned'));
    description=Source_Derivation;
    keep id dataset variable description;

run;

proc sort; by dataset variable; run;
```

SAS code to populate tabs 'Variables', 'Methods', 'Comments'

```

/*****
/* output to an excel file having information of three tabs (Variables, Comments, Methods)
   to be copied into FINAL P21E Define Spec (C) *****/
*****/
proc export data=variables_
            outfile="&path.\Define_variables_methods_comments.xlsx"
            dbms=xlsx replace;
            sheet="Variables";

run;

proc export data=methods
            outfile="&path.\Define_variables_methods_comments.xlsx"
            dbms=xlsx replace;
            sheet="Methods";

run;

proc export data=comments
            outfile="&path.\Define_variables_methods_comments.xlsx"
            dbms=xlsx replace;
            sheet="Comments";

run;
```

Demo

Recorded Steps

This file contains all the steps and information that was recorded to help you describe the recorded steps to others. Before sharing this file, you should verify the following:

- The steps below accurately describe the recording.
- There is no information below or on any screenshots that you do not want others to see.

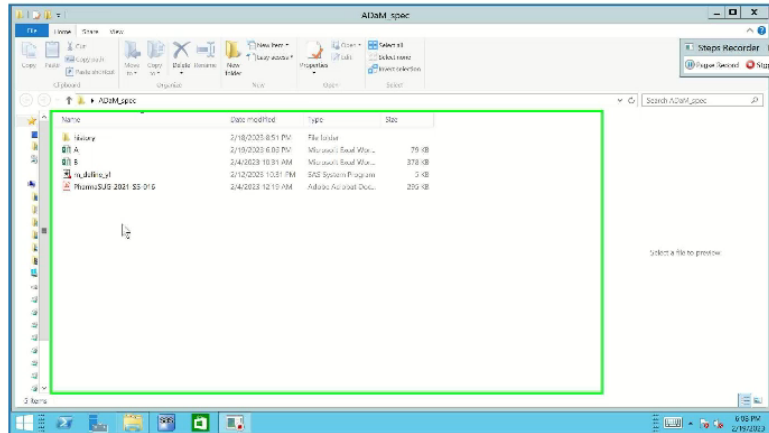
Passwords or any other text you typed were not recorded, except for function and shortcut keys that you used.

You can do the following:

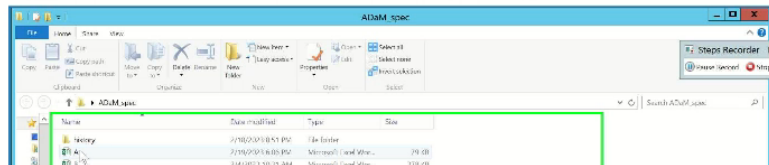
- [Review the recorded steps](#)
- [Review the recorded steps as a slide show](#)
- [Review the additional details](#)

Steps

Step 1: (2/19/2023 6:08:02 PM) User left click on "Items View (list)" in "ADaM_spec"



Step 2: (2/19/2023 6:08:04 PM) User left double click on "Name (edit)" in "ADaM_spec"



Conclusions

- The updated ADaM Specification template supports for both ADaM programming and Define.xml creation
- It is **easy to use and maintain** with minimal specification entry and short programming time
- It can be well adopted and utilized by both small and big pharmaceutical/biotech companies which seek **simplicity** and **efficiency** without complicated and difficult implementation
- Simple SAS code to automate the population of final define spec

Thank you

Xiang Wang

Independent Consultant and Founder
White Crane LLC

Email: xwang929@gmail.com

Ryan Krause

Director, Statistical Programming
Bristol-Myers Squibb

Email: Ryan.Krause@bms.com

Yuanyuan Liu

Associate Director, Statistical Programming
Bristol-Myers Squibb

Email: Yuanyuan.Liu@bms.com

Daniel Huang

Director, Statistical Programming
BeiGene

Email: Daniel.Huang@beigene.com