PharmaSUG 2019 - Paper PO-110

Raw Data Sets Tracker: Time and Project Management Based on the Volume of Available Clinical Data Using SAS® Software.

Girish Kankipati, Seattle Genetics, Seattle WA

ABSTRACT

Time management and availability of clinical data play an important role in the successful execution of a project. In order to plan programming activities and resources, it is very important to understand the availability of clinical data. The volume of raw data depends on a few aspects including the enrollment speed and the type of clinical trial (Phase I, II, or III). Sometimes, enrollment rates can be slow and can cause data unavailability issues, hindering the programming activities, as programming could be challenging with limited data. To address this issue, it is important to establish a robust procedure to track raw data for the successful completion of project within a given timeline.

This paper will discuss how to track raw data availability by creating a raw data set tracker using a SAS® program. This dynamic SAS® program will be demonstrated to create this tracker. The raw data set tracker proposed is to identify the number of data sets that are programmable on a weekly basis. It gives summary statistics on number of subjects and total number of records present in each raw data set during a particular week, displayed as pie and bar charts. Thus, the tracker application will help the programmer plan the activities efficiently (for example, Week 1: DM AE EX; Week 2: DS MH).

INTRODUCTION

For successful completion of a study, timeline management plays an important role in developing the TLFs and data sets. The study timeline is dependent on availability of clinical data. It's always challenging to program SDTM and ADaM data sets with limited data. However, programmers don't want to delay programming activities due to limited data. Data availability depends on the active enrollment of study subjects and timely data entry at clinical sites, and it varies from study to study. In a phase I clinical trial, enrollment rates are slower when compared to a phase III trial. Hence, to manage the issue of uncertainty in predicting enrollment rates this issue this paper introduces a standard procedure to track availability of raw data using SAS. This paper also discusses tracking the availability of raw data from one data cut to another. This will help the study lead programmer to plan programming activities accordingly.

Before moving to further details, it's best to understand some background about data cuts.

WHAT IS A DATA CUT?

A data cut is a predefined subset of the clinical study database. Typically, it includes all available data at a pre-specified point in time, or a subset of data collected up to that point in time.

WHY USE A DATA CUT?

Interim analyses are prepared to support the objectives of a study and enable decisions for a project. A data cut provides a static set of data on which to perform an interim analysis. It allows for traceability and reproducibility of results while the study is ongoing and data continue to be collected. With a predefined data cutoff, relevant data can be "cleaned."

FIXED-CUTOFF DATA CUT

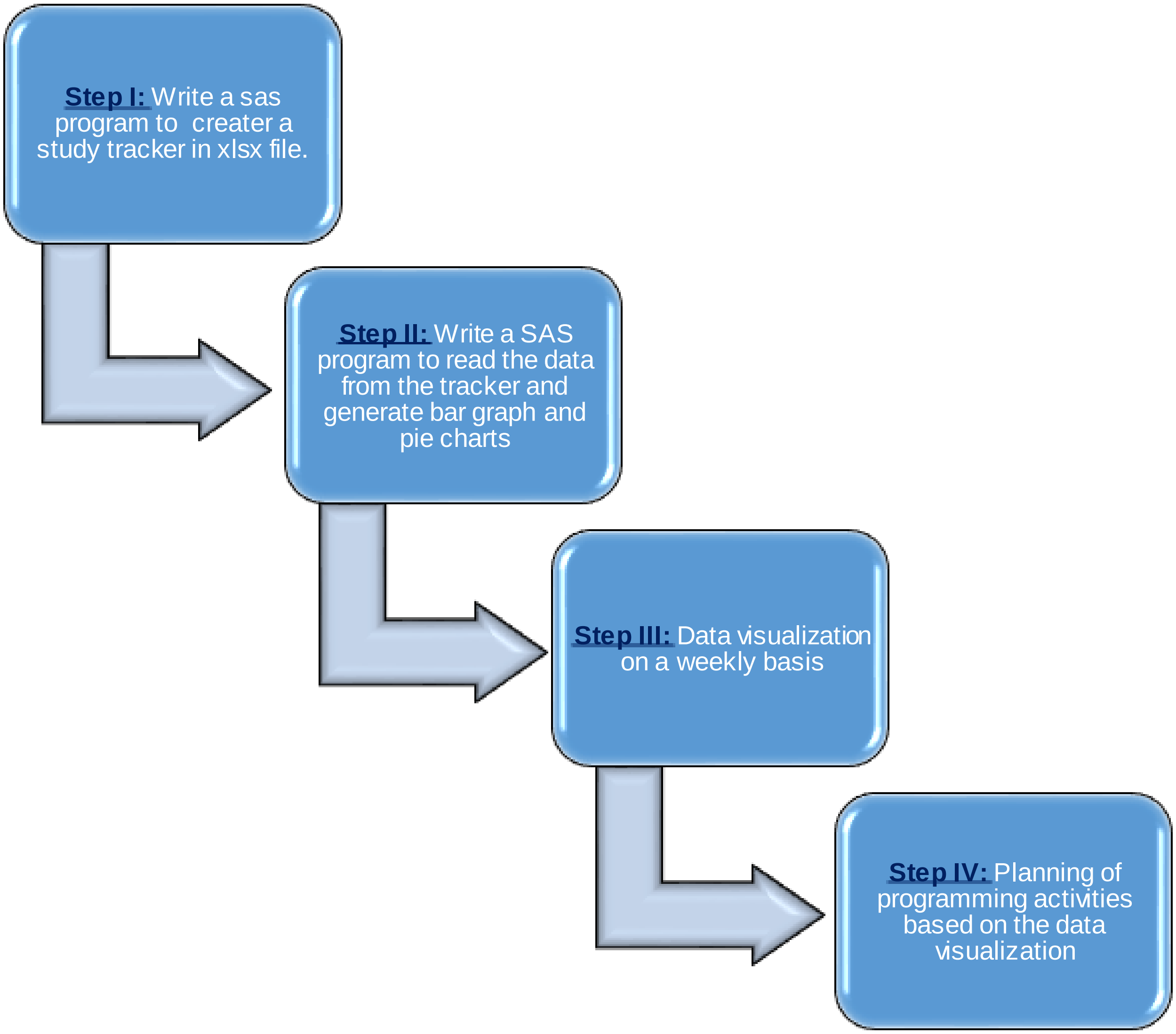
All the data collected on or before a study milestone.

VARIABLE-CUTOFF DATA CUT

All data collected as of a subject-specific milestone.

PROCESS OVERVIEW

See Figure 1 for an overview of the steps involved in creating the study tracker and planning of study related activities.



Step I: Write a sas program to creater a study tracker in xlsx file.
Step II: Write a SAS program to read the data from the tracker and generate bar graph and pie charts
Step III: Data visualization on a weekly basis
Step IV: Planning of programming activities based on the data visualization

Figure 1. Step-By-Step Process to Track the Volume of Clinical Data

STEP I: CREATE A PROGRAM TRACKER

The following code can be used to perform both fixed and variable-cut off data cuts. The data cuts are mostly dependent on the study milestones. In Phase I and Phase II studies, the data cuts often happen on a weekly basis to evaluate safety.

The following dynamic SAS program pulls the raw data set names from the vcolumn SASHELP view and the library 'R' and creates a raw data set tracker.

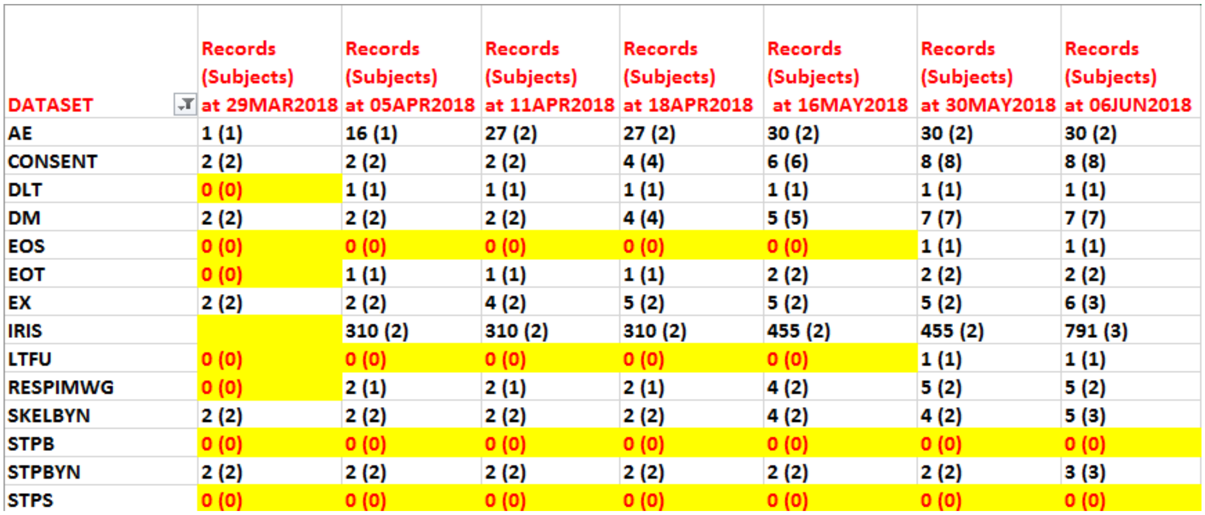
```
*-------------------------------------------------------------------*
|Read raw data information from SAS help library column view:
|Subject id will be present with different names in different raw data
|sets so get unique data sets from memnames
*------------------------------------------------------------------*;
data dsets;
      set sashelp.vcolumn;
      if libname = 'R' and name in ( 'SUBJID','USUBJID','SUBJECT');
      if name = 'USUBJID' then
            patient = 'USUBJID';
      else if name = 'SUBJECT' then
            patient = 'SUBJECT';
      if name = 'SUBJID' then
            patient = 'SUBJID';
      keep memname patient;
run;

proc sort data = dsets nodupkey;
      by memname;
run;

*------------------------------------------------------------------*
|Assign sequence number to each raw data set
*------------------------------------------------------------------*;
data dsets;
      set dsets;
      by memname;
      dsseq  =  n ;
      dataset = memname;
      keep dataset dsseq patient;
run;

%macro cnt (subject = );
      *------------------------------------------------------------*
      |Get the maximum total number of SAS data sets
      *------------------------------------------------------------*;
      proc sql;
            select count(dsseq) into :dsmax from dsets;
            select put(datepart(crdate),date9.) into :creatdt from
sashelp.vtable where libname = 'R' and memname = 'CONSENT';
      quit;

      %put DSMAX -----:::: &dsmax.;
      %put CREATDT -----:::: &creatdt.;

      *------------------------------------------------------------*
      |a) Create a loop for 1 to total number of data sets
      |b) Get each data set name based on the sequence number
      |c) Get unique patients number from each data set
```

```
     *-------------------------------------------------------------*;
     %do i = 1 %to &dsmax.;

          proc sql;
          select dataset into : dset from dsets where dsseq =  &i.;
          select patient into : patient from dsets where dsseq = &i.;
          quit;

          %put DSET ----::: &dset.;
          %put Unique Subject identifier variable  ----::: &patient.;

          *-----------------------------------------------------*
          |a) Count the unique patient from each raw data set and
          | Total number of records from each data set
          |b) Concatenate number of patients and number of records
          | for each raw data set
          *----------------------------------------------------*;
          proc sql;
               create table cntds&i. as select unique "&dset." as
dataset ,count(unique &patient.) as subcnt ,count(*) as reccnt
,strip(put(calculated reccnt,best.))||' ('||strip(put(calculated
subcnt,best.))||')' as cnts_&creatdt. label = "Number of Records (no.
of Subjects) at &creatdt."
               from r.&dset.
               ;
          quit;

          %if &i. eq 1 %then
               %do;
               data cntds;
               set cntds1;
               run;
               %end;
          %else %if &i. gt 1 %then
               %do;
               *------------------------------------------*
               |Combine all the raw data sets that were created
               | in the for loop
               *------------------------------------------*;
               proc append base = cntds data = cntds&i.;
               run;
          %end;
     %end;
%mend;

%cnt;

*-------------------------------------------------------------------*
|Read in most recent Data Tracker dataset
*------------------------------------------------------------------*;
data lds;
     set dtracker;
```

```
run;


proc sort data = lds;
      by dataset;
run;

proc sort data = cntds;
      by dataset;
run;

*---------------------------------------------------------------------*
|Combine with the previous week raw data set
*--------------------------------------------------------------------*;
data out.dtracker;
      merge lds cntds;
      by dataset;
run;
*---------------------------------------------------------------------*
|Export the Sas data set into XLSX format
*--------------------------------------------------------------------*;
proc export data = out.DataTracker (drop = subcnt reccnt)
      outfile = 'P:\Projects\abc\data\raw\DataTracker.xlsx'
      dbms = xlsx
      label replace;

run;
```

DESCRIPTION OF DATA TRACKER

The tracker consists of total number of records and total number of subjects in the raw data each week. As shown in the example in Figure 2 as the weeks pass by, the volume of clinical data increases and the data availability can be seen in the tracker. The tracker shows number of subjects and records available; if the dataset is empty it is highlighted in yellow. Based on the dataset metrics, we can plan safety and efficacy analyses. For example, STPB (soft tissue plasma baseline) and STPS (soft tissue plasma post baseline) are the tumor-related raw data sets and do not have any records. Whereas, for AE, CONSENT, SKELBYN, and DM there is an increase in the number of observations and subjects each week. This sheet is updated every week such that an additional column is added for the weekly metrics.



DATASET	Records (Subjects) at 29MAR2018	Records (Subjects) at 05APR2018	Records (Subjects) at 11APR2018	Records (Subjects) at 18APR2018	Records (Subjects) at 16MAY2018	Records (Subjects) at 30MAY2018	Records (Subjects) at 06JUN2018
AE	1 (1)	16 (1)	27 (2)	27 (2)	30 (2)	30 (2)	30 (2)
CONSENT	2 (2)	2 (2)	2 (2)	4 (4)	6 (6)	8 (8)	8 (8)
DLT	0 (0)	1 (1)	1 (1)	1 (1)	1 (1)	1 (1)	1 (1)
DM	2 (2)	2 (2)	2 (2)	4 (4)	5 (5)	7 (7)	7 (7)
EOS	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	1 (1)	1 (1)
EOT	0 (0)	1 (1)	1 (1)	1 (1)	2 (2)	2 (2)	2 (2)
EX	2 (2)	2 (2)	4 (2)	5 (2)	5 (2)	5 (2)	6 (3)
IRIS		310 (2)	310 (2)	310 (2)	455 (2)	455 (2)	791 (3)
LTFU	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	1 (1)	1 (1)
RESPIMWG	0 (0)	2 (1)	2 (1)	2 (1)	4 (2)	5 (2)	5 (2)
SKELBYN	2 (2)	2 (2)	2 (2)	2 (2)	4 (2)	4 (2)	5 (3)
STPB	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
STPBYN	2 (2)	2 (2)	2 (2)	2 (2)	2 (2)	2 (2)	3 (3)
STPS	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)

Figure 2. Raw Data Set Tracker Based on the Step-1 SAS Program

STEP II: READ DATA

In step II, generate a SAS program to read the data from the tracker to display it graphically. SAS column views will be used in the code to pull the number of weeks. The following SAS program creates a bar chart and a pie chart:

```
*--------------------------------------------------------------------*
|Pull the Excel sheet and convert into SAS data set
*-------------------------------------------------------------------*;
proc import out=sample
      datafile="'p:\projects\abc\data\raw\datatracker.xlsx"
      dbms=xlsx replace;
      getnames=yes;
run;
******Get the number of observations in each raw data set(Convert all
records columns into week1 week2 week3 format during the
transformation of excel to SAS data set format)*****;
data sample1;
      set sample;
      array terms week:;
      do over terms;
       if terms ne '' then
       terms=scan (terms,1,'(');
      end;
run;

data vcolumn1;
      set sashelp.vcolumn;
      if memname ='SAMPLE' and name ^='DATASET';
run;
*--------------------------------------------------------------------*
|Calculate the maximum number of weeks
*-------------------------------------------------------------------*;
proc sql;
```

```
      select distinct (name) into :n1 from VCOLUMN1;
      select count (name) into :n2 from VCOLUMN1;
quit;

****Calculate the number of observations per week by using below
macro******;
%macro count ();
      %do i=1 %to &n2;
      data sample2(rename=(week1_&i.=week&i.));
            set sample1;
            week1 &i. = input(week&i.,best.);
            keep week1_&i.;
      run;
      proc sql  noprint;
            create table wk&i. as select sum (week&i.) as weeknum,
"Week&i." as week  from sample2;
            quit;
      %end;
%mend;

%count;

data all;
      set wk:;
run;
*******Create axis definitions******;
axis1  offset=(5)  major=none label=(h=10pt 'Weekly data extract' );
axis2    minor=none label=(a=90 h=10pt 'Number of observations');

********Create symbol definitions*****;
symbol1 c=vibg i=needle v=none w=35 pointlabel =(height =8pt);
symbol2 c=depk i=join   v=dot h=1.5;

**********Generate bar chart***************;
proc gplot data=all;
      plot weeknum*week weeknum*week / overlay haxis=axis1 vaxis=axis2;
run;
quit;
********Set the graphics environment*******;
goptions border cback=white htitle=12pt;

**********Generate pie chart***************;
proc gchart data=all1;
      pie week / sumvar=weeknum
            value=outside
            percent=none
            other=0
            slice=outside
            noheading;

run;
quit;
```

STEP III: DATA VISIUALIZATION BASED ON TRACKER

Using proc gchart, number of observations per week is shown in the example bar chart in Figure 3. The total number of observations is displayed at the top of each bar. This helps to understand the pattern of increase in availability of clinical trial data over time.

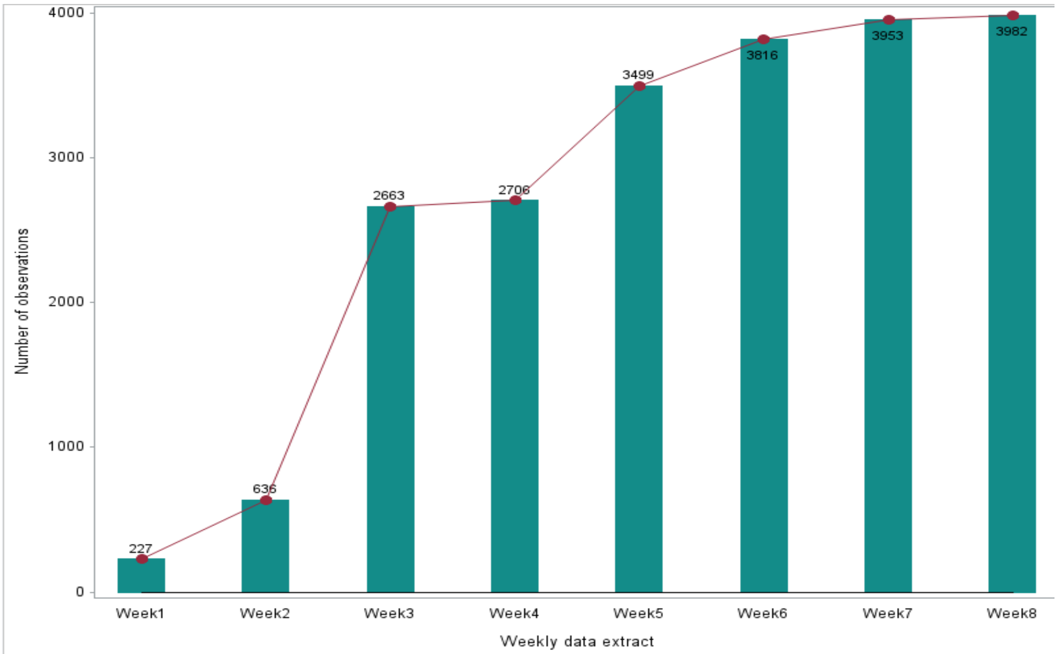


Figure 3. Bar Chart Representing the Weekly Availability of Clinical Data

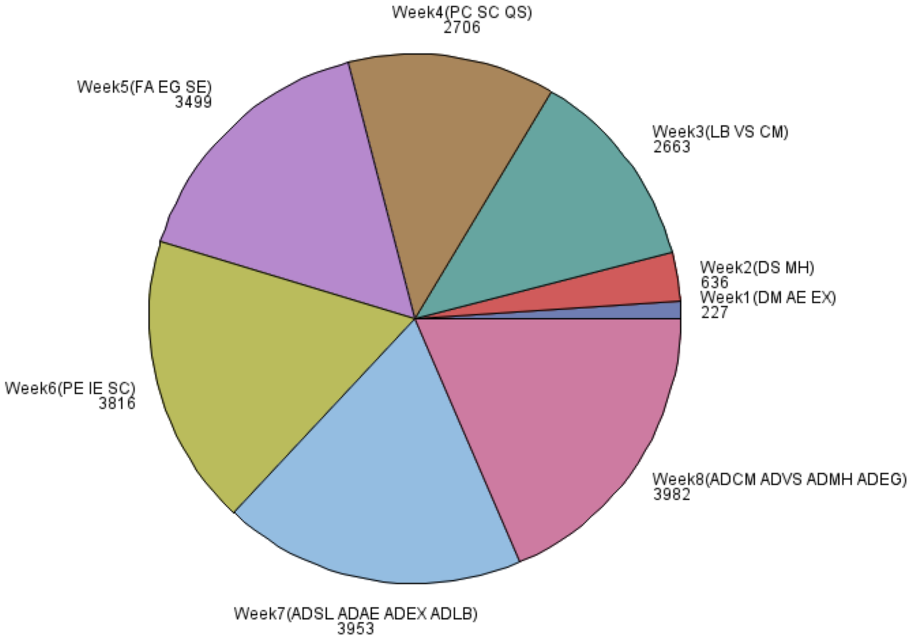


Figure 4. Pie Chart Representing the Total Number of Records Available Each Week

The example pie diagram in Figure 4shows the number of observations and data sets that can be programmed each week. Based on the pie diagram and bar graph a programmer can plan programming activities appropriately for each weekly data extract. Prioritization of tasks plays a very important role. In a Phase I and Phase II studies, available clinical data depends on active subject enrollment. Sometimes, efficacy data will be available after 3-4 weeks from the study start date. Study teams often request programmers to create safety and efficacy analysis outputs for DMCs (Data Monitoring Committees) and SMCs (Safety Monitoring Committees). In these scenarios, the lead programmer can effectively utilize the tracker and check the data availability for that particular analysis and plan the programming activities accordingly.

STEP IV: PLANNING PROGRAMMING ACTIVITIES

Based on the raw data tracker and bar chart results programming can be done as below:

Week 1: DM AE EX

Week 2: DS MH

Week 3: LB VS CM

Week 4: PC SC QS

Week 5: FA SE

Week 6: PE QS IE SC

Week 7: ADSL ADAE ADEX ADLB

Week 8: ADCM ADVS ADMH

Week 9: EG ADEG

This whole process can be repeated on a weekly basis. Every Monday morning, tracker graphs and sheet will be updated with the data sets need to focus on the coming week.

CONCLUSION

In order to meet key program objectives, timelines play a critical role in the successful delivery of statistical output supporting CSRs, publications, and safety deliverables, including numerous interim analyses. This raw data set tracker program helps a SAS programmer to plan activities efficiently.

REFERENCES

5 BASIC PHASES OF PROJECT MANAGEMENT. (n.d.). Retrieved from Project Insight: https://www.projectinsight.net/project-management-basics/basic-project-management-phases

Kelso, T. J. (2017). *NESUG 17.* Retrieved from lexjansen.com: https://www.lexjansen.com/nesug/nesug04/as/as06.pdf

Mei Dey, A. C. (2018). *pharmasug.org.* Retrieved from Pharmasug: https://www.pharmasug.org/proceedings/2018/DS/PharmaSUG-2018-DS19.pdf

Navarro, M. T. (2015, DEC 02). *Standardizing Data Cuts.* Retrieved from phusewiki: https://www.phusewiki.org/docs/2015_California_SDE/Standardizing%20Data%20Cuts_PhUSE%20SDE_REVISED-MNavarro.pdf

ACKNOWLEDGEMENTS

I would like to acknowledge my manager Avani Kaja for her valuable contributions to this paper. I would like to sincerely thank Sr Programmer John Shaik and Statistical Programming Director Wei Qian for encouragement to write this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Girish Kankipati
Seattle Genetics, Inc.
21823 - 30th Drive S.E.
Bothell, WA 98021
425-527-2140
gkankipati@seagen.com