Paper RI-10

Data Merging and Exploration to Identify Associations Between Environmental Factors and Disease Outbreaks

Neeta Shenvi, Department of Biostatistics and Bioinformatics, Rollins School of Public Health, Emory University, Atlanta, GA

Xin Zhang, Department of Biostatistics and Bioinformatics, Rollins School of Public Health, Emory University, Atlanta, GA

Azhar Nizam, Department of Biostatistics and Bioinformatics, Rollins School of Public Health, Emory University, Atlanta, GA

ABSTRACT

This paper describes data merging and visualization techniques for epidemiological and environmental surveillance data. The ultimate goal is to learn about associations between specific environmental factors and disease outbreaks.

Results included in this paper were produced using SAS 9.3 on a Windows XP platform, using Base SAS, SAS/STAT software, and SAS/GRAPH. SAS 9.2 or later is required for ODS graphics extensions.

INTRODUCTION

Cholera epidemics have been reported in over 75 countries in South East Asia, Africa and South America in past three decades [1]. The global burden of cholera is substantial. In 2005, 131,943 cases and 2,272 deaths were reported to the WHO, and recently major, sustained epidemics have been reported in West Africa [2]. Cholera is endemic in Bangladesh, and the rates of cholera here are amongst the highest in the world. Epidemiological and ecological surveillance for cholera has been under way since 1997 in rural Bangladesh as part of the 'Epidemiology and Ecology of *Vibrio cholerae*' study, funded by the National Institutes of Health [3]. The main objective of this study includes elucidating the influence of specific environmental factors on outbreaks of cholera, and developing a model for predicting cholera outbreaks. Such a model would be useful for investigating the impact of potential interventions to mitigate cholera epidemics.

Between March 2004 and September 2007, regular environmental sampling was conducted in two areas of rural Southern Bangladesh, Mathbaria and Bakerganj, in order to determine the physical, chemical, and biological parameters of the natural bodies of water used by rural residents as sources of water for drinking and other household purposes.

In concurrent clinical surveillance in these two areas, physicians examined patients presenting with watery diarrhea at a central hospital or clinic. Complete clinical assessments of patients and microbiological investigations of rectal swabs taken from patients were performed to establish the causes of diarrhea.

Environmental sampling and clinical surveillance for cholera were conducted at 15-day intervals in Mathbaria, and monthly in Bakerganj. Clinical surveillance was conducted for three consecutive days each month.

In such studies, environmental and clinical surveys often occur on different timelines. As such, data merging for the purpose of correlating the two data series can be difficult and subjective. Furthermore, scientists are often interested in exploring chronological lags between the series, making merging more complicated. Visualization of the data series by means of overlaid scatterplots and other multi-dimensional graphics, and exploratory quantification of possible lags and correlations is an important first step in building predictive models.

In this paper, we illustrate five data merging and visualization techniques that enabled us to identify potential associations between cases of the disease and environmental variables, with a variety of possible lags. Clinical and environmental data collected in rural Bangladesh is used here. Only partial data is shown.

We use graph template language (GTL) to create the graphs. The code in this paper was tested using SAS® 9.2 and SAS® 9.3 software on the Windows XP platform.

ILLUSTRATION

1: OPTIMAL MAPPING OF CLINICAL AND ENVIRONMENTAL DATA

In such studies, the environmental and clinical surveys often occur on different timelines and lagged correlations between incident cases and environmental variables are of interest. Graphs can play an important role in providing a quick view of the data series timelines, and the general trend in lag between the two data series. A simple illustration is presented in Figure 1.

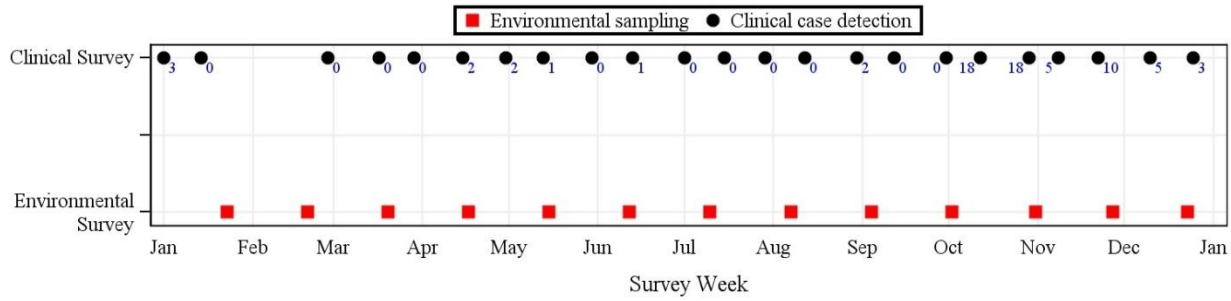

Figure 1. Illustrating Clinical and Environmental Data Series Timelines in Bakerganj

Here, the closed circles denote the timeline for the clinical data sampling; the observed numbers of cases is shown beside the circle in blue. The rectangle denotes the timeline for the environmental sample collection. In Figure 1, it is apparent that clinical environmental sampling did not always coincide, and that several possible lags between environmental and clinical observations may need to be explored.

In general, this graph is very effective for understanding the lag between the occurrence of clinical cases and environmental sampling. Below, we describe SAS DATA and PROC steps used to create this plot.

Data Preparation:

Two sample datasets are presented. The “environ” dataset has e_date (date of environmental sampling) and the “cases” dataset has two variables: c_date (date of observed clinical case) and cases (observed number of cases) for clinical survey.

```
DATA environ;
input e_date ddmmyy8. ;
datalines;
23/1/06
20/2/06
20/3/06
17/4/06
15/5/06
12/6/06
10/7/06
7/8/06
4/9/06
2/10/06
31/10/06
27/11/06
23/12/06;
RUN;
```

```
DATA cases;
Input c_date ddmmyy8. cases;
datalines;
1/1/06          3
14/1/06         0
27/2/06         0
17/3/06         0
29/3/06         0
15/4/06         2
30/4/06         2
13/5/06         1
30/5/06         0
13/6/06         1
1/7/06          0
15/7/06         0
29/7/06         0
12/8/06         0
30/8/06         2
12/9/06         0
30/9/06         0
12/10/06       18
29/10/06       18
8/11/06         5
22/11/06       10
10/12/06        5
25/12/06        3;RUN;
```

Next we create the union of the two datasets. The resulting dataset, “both”, has three variables: e_date, c_date, and cases.

```
PROC SQL;
CREATE TABLE both AS
SELECT e_date, "" AS c_date, "" AS cases FROM environ
union
SELECT "" , c_date , cases FROM cases;
QUIT;
```

Based on e_date and c_date, we create week variables for the two dates (e_week,c_week), and two y-axis coordinate indicators (c_occurred,e_occurred) for two data series.

```
DATA both; set both;
e_week = intck('WEEKV','01JAN04'd, e_date);
c_week = intck('WEEKV','01JAN04'd, c_date);

if c_date ~= . then c_occurred=0.52;
if e_date ~=. then e_occurred=0.5;
format e_date c_date date7.;
RUN;
```

Template Code:

We use GTL to visualize the timeline of these two data series. We use two scatterplot statements within layout overlay. The first scatterplot renders environmental data series. The second scatterplot renders clinical case series. Here is the program snippet to create GTL template.

```
PROC TEMPLATE;
…
layout overlay /
xaxisopts=(Label="Survey Week" griddisplay=on display=(label tickvalues)
linearopts=(integer=true tickvaluesequence=(start=100 end=160 increment=4)))

yaxisopts=(griddisplay=on   label=" " linearopts=(integer=false
tickvaluesequence=(start=0.5 end=0.53 increment=0.01)));

scatterplot x=e_week y=e_occurred/Markerattrs=( size=3pt color=red
symbol=squareFilled) LEGENDLABEL="Environmental sampling" NAME="envir";
scatterplot x=c_week y=c_occurred/Markerattrs=( size=3pt symbol=circleFilled )
datalabel=cases datalabelattrs=(color=blue) LEGENDLABEL="Clinical case detection"
NAME="Cases";

discretelegend "envir" "Cases"/location=outside halign=center valign=top
titleborder=true borderattrs=(thickness=2);

endlayout;

…

RUN;
```

Now that we have the data and the template, we can generate the time series graph using the SGRENDER procedure as shown below.

```
PROC FORMAT;
value y 0.5="Environmental\n Survey" 0.51= " " 0.52="Clinical Survey";
RUN;

PROC SGRENDER data=both template=oneplot;
format c_occurred y.;
RUN;
```

2: MERGING CLINICAL AND ENVIRONMENTAL DATASETS WITH LAGS

Examination of Figure 1 indicated that a variety of lags between the data series could be considered. An enhancement to this graph which indicates possible pairings of observations from the two series can further assist in the investigation of lags and correlations. In Figure 2 we show the clinical dataset merged with a lag of -5 to 14 days with environmental dataset. The connecting straight and "V" lines indicate merged observations. The numbers in purple show the lag time (in days) between environmental date and clinical case date. The numbers in blue are the number of cases occurred.

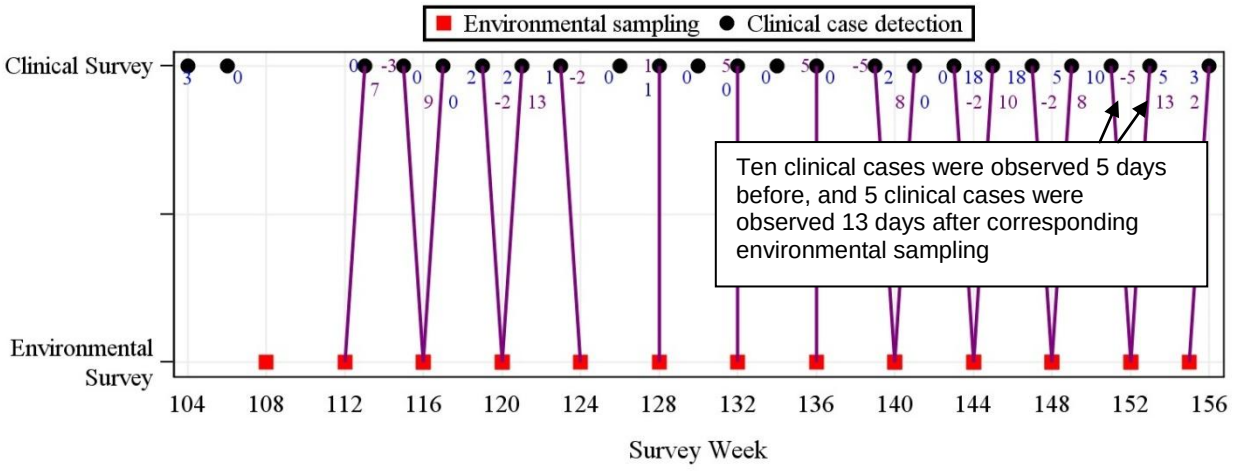

Figure 2. Merging clinical dataset merged with a lag of -5 to 14 days with environmental dataset

DATA PREPARATION:

We create "environ" and "cases" dataset as described above. For each environmental date, we create two lag dates (prev_e_date, next_e_date) for a lag of -5 to 14 days.

```
DATA environ;
set environ;
prev_e_date = e_date - 5;
next_e_date=e_date + 14;
format prev_e_date next_e_date e_date date7.;
RUN;
```

Next we merge two datasets to create "both" as below.

```
PROC SQL;
CREATE TABLE both AS
SELECT a.*, c_date, cases ,"Matched Lag" AS comments, (c_date - e_date) AS lag
FROM environ AS a, cases AS b
WHERE c_date between prev_e_date AND  next_e_date;
QUIT;
```

We generated the merged table as show in Table 1.

e_date	prev_e_date	next_e_date	c_date	cases	comments	lag
20-Feb-06	15-Feb-06	6-Mar-06	27-Feb-06	0	Matched Lag	7
20-Mar-06	15-Mar-06	3-Apr-06	17-Mar-06	0	Matched Lag	-3
20-Mar-06	15-Mar-06	3-Apr-06	29-Mar-06	0	Matched Lag	9
17-Apr-06	12-Apr-06	1-May-06	15-Apr-06	2	Matched Lag	-2
17-Apr-06	12-Apr-06	1-May-06	30-Apr-06	2	Matched Lag	13
15-May-06	10-May-06	29-May-06	13-May-06	1	Matched Lag	-2
12-Jun-06	7-Jun-06	26-Jun-06	13-Jun-06	1	Matched Lag	1
10-Jul-06	5-Jul-06	24-Jul-06	15-Jul-06	0	Matched Lag	5
7-Aug-06	2-Aug-06	21-Aug-06	12-Aug-06	0	Matched Lag	5
4-Sep-06	30-Aug-06	18-Sep-06	30-Aug-06	2	Matched Lag	-5
4-Sep-06	30-Aug-06	18-Sep-06	12-Sep-06	0	Matched Lag	8
2-Oct-06	27-Sep-06	16-Oct-06	30-Sep-06	0	Matched Lag	-2
2-Oct-06	27-Sep-06	16-Oct-06	12-Oct-06	18	Matched Lag	10
31-Oct-06	26-Oct-06	14-Nov-06	29-Oct-06	18	Matched Lag	-2
31-Oct-06	26-Oct-06	14-Nov-06	8-Nov-06	5	Matched Lag	8
27-Nov-06	22-Nov-06	11-Dec-06	22-Nov-06	10	Matched Lag	-5
27-Nov-06	22-Nov-06	11-Dec-06	10-Dec-06	5	Matched Lag	13
23-Dec-06	18-Dec-06	6-Jan-07	25-Dec-06	3	Matched Lag	2

Table 1. ”Both” dataset: Merged for a lag of -5 to 14 days

Five observations in “cases” dataset that did not have any match in “environ” dataset and 1 observation in “environ” dataset did not have a match with “cases” dataset. Following code appends such observations in “both” dataset. The resulting “both” dataset will have following additional rows.

```
PROC SQL;
CREATE TABLE cases_matched AS
SELECT a.*,b.c_date AS c_date_both
FROM  cases AS a LEFT JOIN both AS b
ON a.c_date =b.c_date;
QUIT;

PROC SQL;
INSERT INTO BOTH( c_date, cases,comments)
SELECT c_date, cases, "No Environ Match"
FROM  cases_matched
WHERE c_date_both is null;
QUIT;
```

The resulting “both” dataset will have following additional rows. Partial data is shown here.

e_date	prev_e_date	next_e_date	c_date	cases	comments	lag
23-Jan-06	.	.	.	.	No case data	.
.	.	.	1-Jan-06	3	No Environ data	.
.	.	.	14-Jan-06	0	No Environ data	.
.	.	.	30-May-06	0	No Environ data	.

TEMPLATE CODE

We use a vectorplot statement with the X1, Y1 columns as their origin and X2, Y2 columns for their ends to connect the lines as below.

```
PROC TEMPLATE;
….
vectorplot x=c_week y=c_occurred  xorigin=e_week yorigin=e_occurred /
arrowheads=false datalabel=lag datalabelattrs=(color=purple)
lineattrs=(pattern=solid thickness=2px color=purple) shaftprotected=true;
….
```

Finally, we use sgrender procedure as described above to produce the graph.

3: CORRELATION PLOTS FOR CLINICAL CASES AND ENVIRONMENTA PARAMETERS

After exploring and identifying appropriate lags, and merging the clinical and environmental data, visualizations of the number of cholera cases and various environmental parameters is an important step in predictive model building. Figure 3 below graphically illustrates association of clinical cases with environmental parameters. Here we use different environmental parameters such as: water pH, water depth, air temp, water temp.

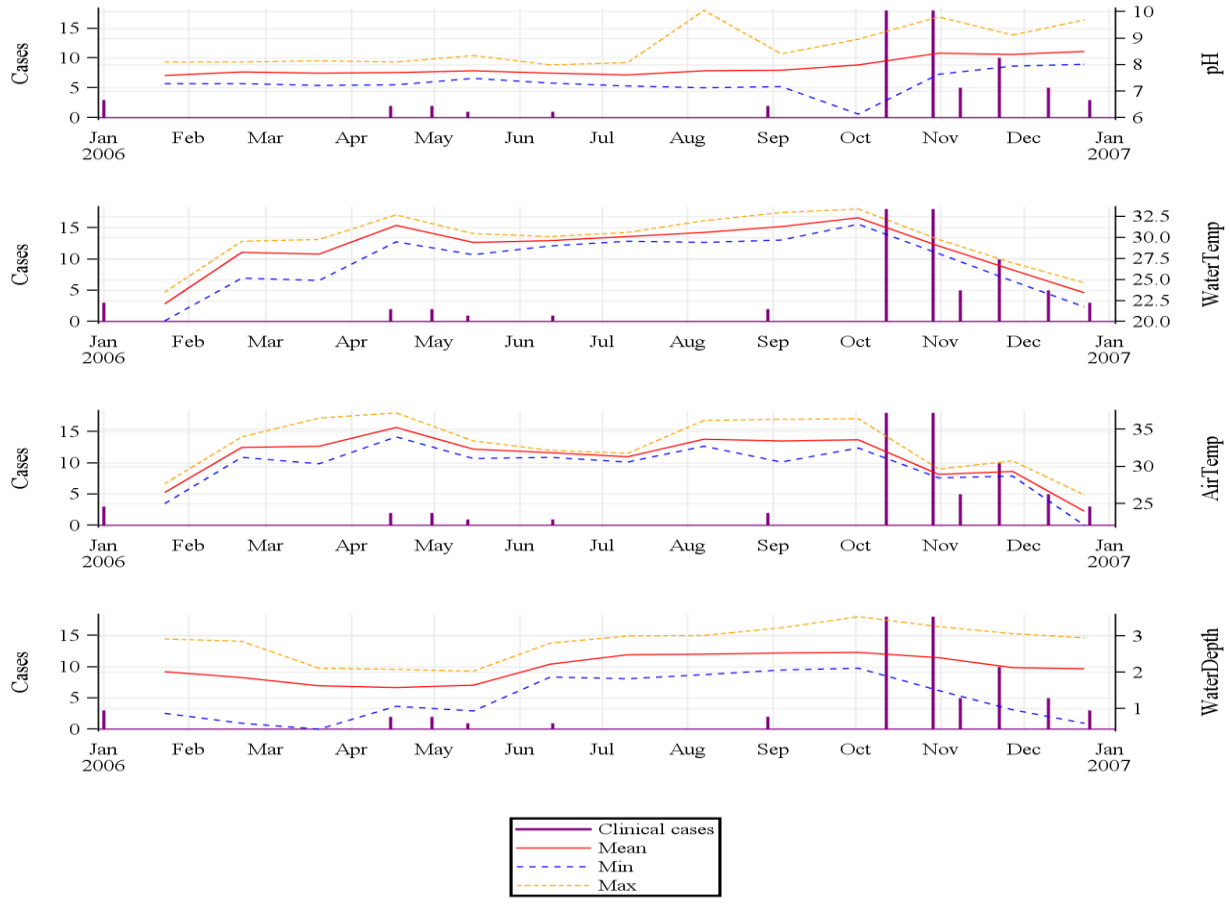

Figure 3. Environmental Parameters and Cholera Cases Over Time

DATA PREPARATION:

During each environmental survey, environmental samples from 8 water bodies are collected. The partial data table is below. The "site" variable represents water bodies. Environmental predictors are pH, water temperature, water depth etc.

```
DATA PRED;
INPUT Site    E_DATE DDMMYY8.      Ph     WaterTemp     WaterDepth    AirTemp
       DOT    TDS    Conductivity Salinity;
DATALINES;
1       23-1-06       8.06   22.5   2.7    25     8.38   88.3   184.7    0
2       23-1-06       7.62   21.8   1.6    25.8   6.28   126.3  263       0.1
3       23-1-06       7.31   20.9   2.92   26.5   6.97   62.2   130.6    0
4       23-1-06       8.11   23.5   2.09   26.5   8.22   81.9   171.5     0
...;
RUN;
```

We take mean, max, min of each parameter for each survey date and merge with "both" dataset. The code is below.

```
PROC SQL;
CREATE TABLE group_pred  AS
SELECT e_date, min(ph) AS ph_min, avg(ph) AS ph_mean, max(ph) AS ph_max,
       min(airtemp) AS airtemp_min, avg(airtemp) AS airtemp_mean, max(airtemp) AS
airtemp_max,…
FROM pred   GROUP BY e_date;
QUIT;

PROC SQL;
CREATE TABLE both2 AS
SELECT a.*, b.*   FROM a.both AS a LEFT JOIN group_pred AS b
ON a.e_date=b.e1_date;
QUIT;
```

TEMPLATE CODE:

We define two global variables num_rows, y2_array. The "num_rows" specifies number of panels. "y2_array" specifies the array of predictor variables. The macro "do_plot_template" defines template using global variables. You can use the lattice layout with one cell. The do loop creates series of overlay layout provided by the user input in macro variable num_rows. In this example, four layouts are rendered dynamically (num_rows=4). Each cell overlay layout contains one needleplot statement to draw number of cases and 3 series plot. Each series plot renders mean, max and min series line for the predictor variable against cases during survey sampling. We use Y-axis for cases and Y2-axis for the predictor variables.

Next, we construct legend that directly references four legend items, cases, min, max and mean. The template code snippet is given below.

```
%let num_rows=4;
%let y2_array=pH WaterTemp AirTemp WaterDepth;
%macro do_plot_template();
PROC TEMPLATE;
define statgraph predplot;
…
layout lattice /  rowgutter=10px columns=1 rows=%eval(&num_rows.+1)
rowdatarange=union;

%do i=1 %to &num_rows.;
%let y2_label=%scan(&y2_array,&i);
layout overlay /……;

needleplot x=c_date y=cases/ LEGENDLABEL="Clinical cases" NAME="clin"
       lineattrs=(color=purple thickness=2px pattern=solid);
seriesplot x=e_date y=&y2_label._mean/yaxis=y2 LEGENDLABEL="Mean" NAME="Mean"
       LINEATTRS= (COLOR=red PATTERN=1 );
seriesplot x=e_date y=&y2_label._min/yaxis=y2 LEGENDLABEL="Min" NAME="Min"
       LINEATTRS= (COLOR=blue PATTERN=2 ) ;
seriesplot x=e_date y=&y2_label._max/yaxis=y2 LEGENDLABEL="Max" NAME="Max"
       LINEATTRS= (COLOR=orange PATTERN=3 ) ;
endlayout;
%end;

layout overlay /;
discretelegend  "clin" "Mean" "Min" "Max"/location=outside
halign=center valign=top
titleborder=true
borderattrs=(thickness=2);
endlayout;
/*****************/
endlayout;*end lattice;
endgraph;
end;
run;
%mend;
```

We are finished with the data and template. All that is left is to generate the graph by submitting “do_plot_template” macro call and the SGRENDER procedure below:

```
%do_plot_template;
PROC SGRENDER data=both2 template=predplot;run;
```

4: CORRELATION COEFFICIENT (R) FOR CLINICAL CASES AND ENVIRONMENTAL PARAMETERS

To accompany Figure 3, we compute Pearson and Spearman correlation coefficients for clinical cases with predictor variables. Environmental predictors with the strongest correlations with cholera cases are identified in the resulting traffic light plot (Figure 4).

The partial output is given below. We used two cutoff conditions to highlight correlation statistics. For predictors (pH, AirTemp, WaterTemp), we look for positive correlation of at least 0.2 to ascertain any association with cholera cases (i.e. rp >0.2 and rs> 0.2). We then use traffic light method to highlight those sites where correlation of predictors with cases is at least 0.2.

For the predictor “Water depth”, we look for a negative association of at least -0.1 to ascertain association with cholera cases. (i.e. rp >-0.1 and rs> -0.1). We then use traffic light method to highlight those sites where correlation of “water temp” with cases is at least -0.1.

	Site											
	1		2		3		4		5		6	
Predictor	Pearson r	Spearman r	Pearson r	Spearman r	Pearson r	Spearman r	Pearson r	Spearman r	Pearson r	Spearman r	Pearson r	Spearman r
Ph	0.45	0.52	0.43	0.56	0.37	0.39	0.52	0.55	0.52	0.53	0.63	0.59
WaterDepth	-.03	-.13	-.10	-.17	-.02	-.13	-.01	-.37	0.12	-.11	-.10	-.15
AirTemp	-.15	-.33	-.29	-.45	-.25	-.47	-.23	-.45	-.23	-.40	-.33	-.41
WaterTemp	0.25	-.12	0.15	-.01	0.45	0.36	0.42	0.39	0.03	-.12	0.24	-.01

Figure 4. Traffic Light to view desired correlations

DATA PREPARATION

We create a dataset “pred_corr” by merging previously created two datasets, “pred” and “both”.

```
proc sql;
create table pred_corr as
select a.*, b.ph,watertemp,waterdepth,dot,AirTemp,site
from a.both as a left join pred as b
on a.e_date=b.date
order by site;
quit;
```

COMPUTE CORRELATION COEFFICIENTS:

The following code snippet generates spearman and pearson correlation statistics between cases and environmental predictors. We ods output pearson and speraman correlation tables, then merge these two tables to create “corr_all” table. The “corr_all” table has one row for each predictor variable and its corresponding r_p and r_s for each of water site.

```
%let y2_array=pH WaterTemp AirTemp WaterDepth;
ods output PearsonCorr=corrp;
ods output SpearmanCorr=corrs;
PROC CORR data=pred_corr pearson spearman;
 by site;
 var  cases  &y2_array.;
RUN;
/** merge two r sets ********/
PROC SQL;
CREATE TABLE corr_all AS
SELECT   a.variable, a.site, a.pcases AS spearman_p, a.cases AS
spearman_r, b.pcases AS pearson_p, b.cases AS pearson_r format=5.2
FROM corrs AS a, corrp AS b
WHERE a.variable=b.variable AND a.site=b.site;
quit;


DATA corr_all; set corr_all;
spearman_r=round(spearman_r,0.01);
pearson_r=round(pearson_r,0.01);
RUN;
```

TRAFFIC LIGHT TO HIGHLIGHT OUTPUT

We used PROC REPORT and traffic lighting tricks in previously published papers ‘Beyond the Basics: Advanced PROC REPORT Tips and Tricks’ [4] and ‘Traffic Lighting: The Next Generation’ [5]. The code snippet is given below.

```
PROC REPORT data=corr_all nofs split='_' ;
where variable not in ('cases');
column variable site, (pearson_r spearman_r) dummy;
define variable/group order=data 'Predictor'
                    style(column)=[font_size=7pt just=left ];
define site/across order=data center 'Site'
                    style(column)=[font_size=9pt  just=center ];
define pearson_r/display 'Pearson_r' format=4.2
                    style(column)=[font_size=9pt just=left ] ;
define spearman_r/display 'Spearman_r' format=4.2
                    style(column)=[font_size=9pt just=left ];
define dummy/ noprint;

%let startcol = 2; /* position of the first computed var under the across */
%let sites = 6; /* Number of sites across */
%let varsunder = 2; /* Number of variables under the across */

%macro create;
compute spearman_r;
%do i=2 %to %eval(&sites*&varsunder) %by &varsunder;
/* if r is > 0.2 for all predictors other than watertemp */
if(abs(_c&i._) GT 0.2 and abs(_c%eval(&i.+1)_) GT 0.2) and variable ne "WaterDepth"
then do;
      call define ("_c&i._" , "style", "style=[background = pink" ) ;
       call define ("_c%eval(&i.+1)_" , "style", "style=[background = pink" );
end;
else if (_c&i._ LE -0.10 and _c%eval(&i.+1)_ LE -0.10) and variable eq "WaterDepth"
then do;
      call define ("_c&i._" , "style", "style=[background = pink" ) ;
      call define ("_c%eval(&i.+1)_" , "style", "style=[background = pink" ) ;
end;
%end;
endcomp;

%mend create;
%create;
run;
```

5: VISUALIZATION OF POTENTIAL ASSOCIATIONS BETWEEN ENVIRONMENTAL INDICATORS OF *V. CHOLERAE* AND CLINICAL CASES

The visualizations in Figures 1, 2, 3 and 4 are useful when examining associations between quantitative environmental variables and cholera cases. However, environmental studies often involve multiple categorical indicators of the presence or absence of disease-causing agents in ecological samples. To visualize the associations between such indicators and the numbers of cholera cases observed, multi-dimensional plots for categorical predictors are needed.

The Figure 5 shows detection of *Vibrio cholerae* in two types of environmental samples, water and zooplanktons. The *Vibrio cholera* detection was done by four methods: Fluorescent antibody (DFA), PCR, culture, and total cultureable *V.cholerae* count (TCVCC). The lower panel shows clinical cases over time. The middle and top panels represent *Vibrio cholerae* detection from zooplankton and water samples respectively. The Y-axis for the top 2 panels corresponds to the 8 sites. The “closed” and “open” symbols represent positive and negative *Vibrio cholerae* detection respectively.

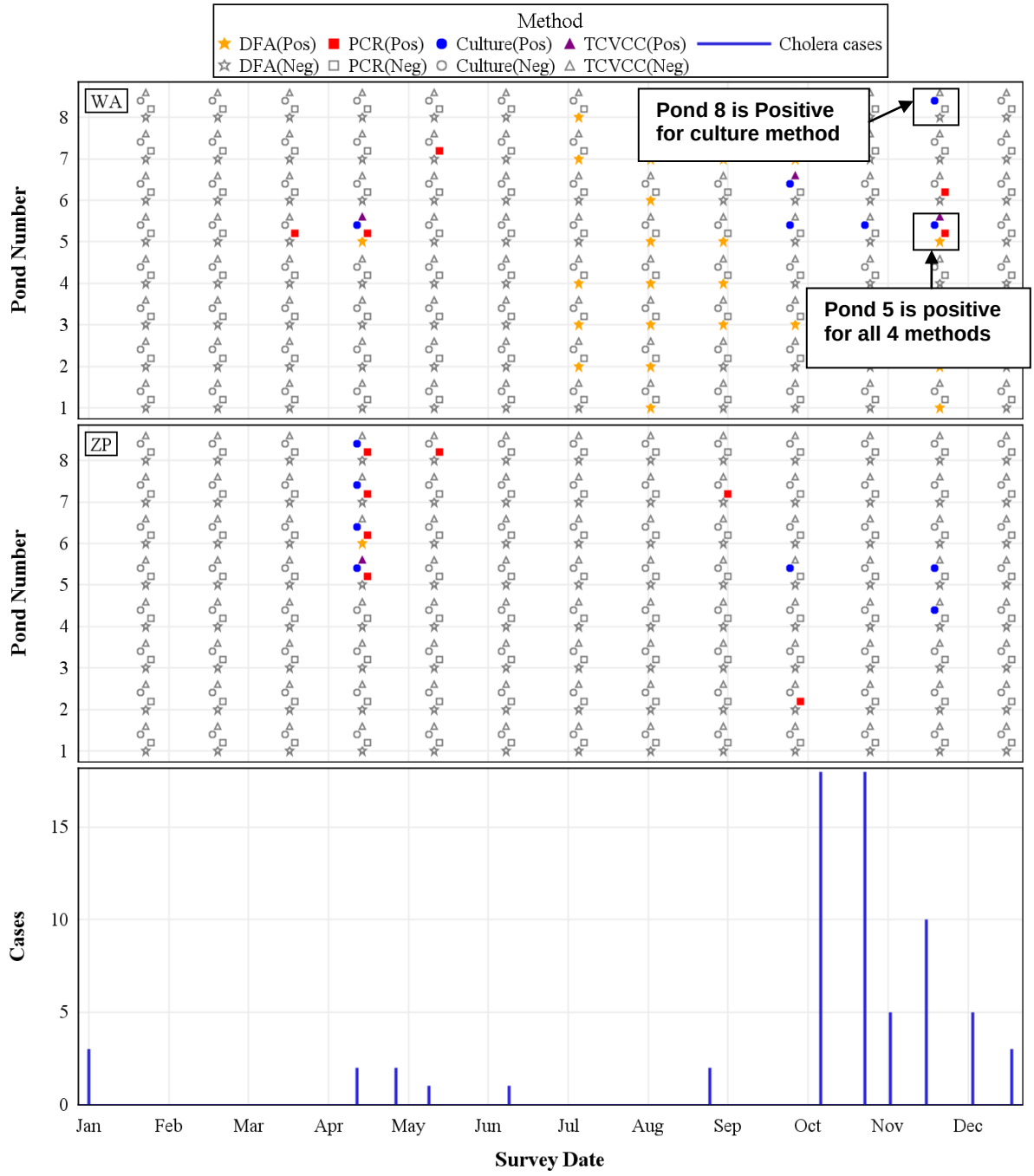

Figure 5. Detection of Vibrio cholerae in two types of environmental samples

DATA PREPARATION:

For each environmental survey, two types of samples (water and zooplankton) from 8 pond sites are used to detect Vibrio cholera. Each sample was tested by four methods (DFA, PCR, Culture, and TCVCC). The partial data table is below. The code for positive and negative detection is 1 and 0.

```
data method;
input e_date ddmmyy8. site    sampletype DFA PCR Culture TCVCC;
Datalines;
…
2-10-06  1      WA      0       0       0       0
2-10-06  1      ZP      0       0       0       0
2-10-06  2      WA      0       0       0       0
2-10-06  2      ZP      0       1       0       0
2-10-06  3      WA      1       0       0       0
2-10-06  3      ZP      0       0       0       0
2-10-06  4      WA      0       0       0       0
2-10-06  4      ZP      0       0       0       0
2-10-06  5      WA      0       0       1       0
2-10-06  5      ZP      0       0       1       0
2-10-06  6      WA      0       0       1       1
2-10-06  6      ZP      0       0       0       0
2-10-06  7      WA      1       0       0       0
2-10-06  7      ZP      0       0       0       0
2-10-06  8      WA      0       0       0       0
2-10-06  8      ZP      0       0       0       0
……..
;
run;
```

Now we create “Positive” and “Negative” variable for each method as follows.

```
DATA method; set method;
DFA_Pos=ifn(dfa=1, site,. ); DFA_Neg=ifn(dfa=0,site,.);
….;
RUN;
```

Next, we transform data to create four variables for each method, two for each sample type (e.g. DFA_POS_WA, DFA_NEG_WA, DFA_POS_ZP, DFA_NEG_ZP)

```
%macro data_transform(in= ,out=);
DATA &out (rename=( DFA_POS=DFA_POS_&out.  DFA_Neg=DFA_Neg_&out. ….  )
       DROP=DFA PCR CULTURE TCVCC SAMPLETYPE);
       set &in; where sampletype in ("&out.");
RUN;
%mend;

%data_transform(in=method ,out=WA);
%data_transform(in=method ,out=ZP);

DATA method; MERGE WA ZP; by e_date site;RUN;
```

The partial “Method” data table is below.

e_date	site	DFA_POS_WA	DFA_Neg_WA	DFA_POS_ZP	DFA_Neg_ZP	PCR…	CULTURE...	TCVCC...
2/10/2006	1	.	1	.	1			
2/10/2006	2	.	2	.	2			
2/10/2006	3	3	.	.	3			
2/10/2006	4	.	4	.	4			
2/10/2006	5	.	5	.	5			
2/10/2006	6	.	6	.	6			

Now, we merge “method” dataset with “cases” dataset.

```
DATA plotdata; MERGE method cases; RUN;
```

DECLARE GLOBAL VARIABLES FOR PLOT:

We declare some global variables that are referenced dynamically in GTL template code. The variable "methods_vector" stores 4 methods used to detect *Vibrio cholerae* (DFA, PCR, Culture, TCVCC). Variable "sampletype_vector" stores 2 sample types used (WA, ZP).

We declare variables "colorarray" and "symbolarray" to distinguish 4 methods by distinct color and symbol.

The variable "x_jitter" takes 4 constants to jitter x-coordinate. These global variables are given below.

```
%let Methods_vector=DFA PCR Culture TCVCC;  *** 4 methods;
%let SampleType_vector=WA ZP;   *** 2 samples;
%let colorarray=darkorange red blue purple; *** 4 colors for 4 methods;
%let symbolarray=star square circle triangle; *** 4 symbols for 4 methods;
%let y_jitter=0;  *** initialize constant to jitter y-axis coordinate;
%let x_jitter=0 2 -2 0; ***this jitters symbols for 4 methods on x-axis;
```

TEMPLATE CODE:

To implement this in GTL, you can use the lattice layout with one cell arranged in three rows. The top two rows show data corresponding to two sample types (WA, ZP) and the lowermost row shows cholera cases. The top two rows contain an overlay layout with two scatter plots to show positive or negative results. The lowermost row contains an overlay layout with needle plot to show cholera cases over time.

We use two nested do loops to correctly render top two rows. The outer do loop generates top two rows with an overlay layout corresponding to two sample types (WA ZP):

```
%do i=1 %to %SYSFUNC(COUNTW(&SampleType_vector));
```

The inner do loop then renders scatter plots corresponding to 4 methods.

```
%do j=1 %to %SYSFUNC(COUNTW(&Methods_vector));
```

Here is the program snippet to build the template:

```
%macro do_plot_template();
%let rownum=%eval(%SYSFUNC(COUNTW(&SampleType_vector))+1);

PROC TEMPLATE;
define statgraph methodplot;
begingraph …;
layout lattice / columns=1 rows=&rownum rowdatarange=union
columndatarange=unionall;
%do i=1 %to %SYSFUNC(COUNTW(&SampleType_vector)); ***puts 2 panels for 2 samples;
%let samplename = %scan(&SampleType_vector,&i);
       layout overlay/;
         %do j=1 %to %SYSFUNC(COUNTW(&Methods_vector));
              ***puts 4 symbols for each site;
              %let Methodname = %scan(&Methods_vector,&j);
              /* << -- SCATTER PLOTS FOR POSITIVES or NEGATIVE;>>   ---*/
          %end; *** close inner do;

              entry halign=left "&samplename" /valign=top border=true;
       endlayout;
%end;  *-- close outer do ;
/*-- << PLOT cholera cases >> --*/
/*-- << Specify Panel Y-Axis >> --*/
/*-- << Specify Common X-Axis >> --*/
/*-- << Specify Legends and Markers >> --*/

endlayout; *--Close Lattice layout ;
endgraph;  *-- Close begingraph ;
end; *-- Close define ;

%mend;
```

Next, we illustrate scatter plot components. We have two scatter plot statements to show positive and negative result.

We jitter x and y coordinates in scatter plot statement using eval function as follows:

```
X=eval(e_date + %scan(&x_jitter, &j, " "))
Y=eval(&Methodname._POS_&samplename. + &y_jitter )
```

Next, we dynamically specify marker attributes (color, symbol) and legend attributes (name, label) as follows.

```
Markerattrs=(size=3pt color= %scan(&colorarray, &j)
         symbol=%scan(&symbolarray, &j)Filled  )
         LEGENDLABEL="&Methodname.(Pos)"
         NAME="&Methodname.Pos";
```

At the end of two scatter plot statements, we increment y_jitter value to separate 4 symbols from one another.

```
%let y_jitter=%SYSEVALF(&y_jitter + 0.2)
```

Here is the program snippet:

```
/* << -- PLOT POSITIVES or NEGATIVE;>>   ---*/
scatterPlot X=eval(e_date+%scan(&x_jitter, &j, " "))
            Y=eval(&Methodname._POS_&samplename. + &y_jitter )
                  / Markerattrs=(size=3pt color= %scan(&colorarray, &j)
                    symbol=%scan(&symbolarray, &j)Filled  )
                    LEGENDLABEL="&Methodname.(Pos)" NAME="&Methodname.Pos";
     *-- PLOT NEGATIVES ;
scatterPlot X=eval(intnx('day',e_date,%scan(&x_jitter, &j, " ")))
            Y=eval(&Methodname._NEG_&samplename. + &y_jitter)
                 / Markerattrs=(size=3pt color=grey symbol=%scan(&symbolarray,
&j))
                   LEGENDLABEL="&Methodname.(Neg)" NAME="&Methodname.Neg";
    /*-- Increment y_jitter --*/
%let y_jitter=%SYSEVALF(&y_jitter +0.2);
```

Now we show other pieces of GTL code to build the template.

We show cholera cases in the lowermost panel using layout overlay. We use needle plot statement with cases on Y-axis and c_date (case date) on X-axis. The program snippet is below:

```
/* PLOT cholera cases*/
layout overlay/ ;
needleplot x=c_date y=Cases /  lineattrs=graphdata1(thickness=2px pattern=solid)
                               NAME="Cholera" legendlabel="Cholera cases";
endlayout;
```

Next, we specify Y-axis for 3 panels within “rowaxes” block. In this block, we have two “rowaxis” statements, first for top two panels and second for lowermost panel. The program snippet is below:

```
/**** Specify Panel Y-Axis ***********/
rowaxes;
       %do i=1 %to %SYSFUNC(COUNTW(&SampleType_vector)); *** two top panels;
              %let samplename = %scan(&SampleType_vector,&i);
              rowaxis / griddisplay=on display=(label tickvalues)
              labelattrs=(weight=bold) label="Pond Number"
              linearopts=( integer=true TICKVALUELIST=(1 2 3 4 5 6 7 8));
       %end;

rowaxis / griddisplay=on display=(label tickvalues)   *** Lowest panel;
          linearopts=(integer=true)  labelattrs=(weight=bold) ;
endrowaxes;
```

Next, we specify common X-axis for all 3 panels within “columnaxes” block. The program snippet is below:

```
/**** Specify Common X-Axis ***********/
columnaxes;
columnaxis / griddisplay=on display=(label tickvalues) labelattrs=(weight=bold)
             linearopts=(tickvalueSequence=(start="1Dec2005"d increment=31
             end="31Dec2006"d) tickvalueformat=Monname3.) label="Survey Date";
endcolumnaxes;
```

Next, we specify legends and markers. The program snippet is below:

```
/**** Specify Legend and Markers ***********/
sidebar / align=top;
layout overlay / pad=(bottom=2px);
       DiscreteLegend
       %do j=1 %to %SYSFUNC(COUNTW(&Methods_vector));
        %let Methodname = %scan(&Methods_vector,&j);
            "&Methodname.Pos" "&Methodname.Neg"
       %end
    "Cholera"/ORDER=COLUMNMAJOR down=2 valign=top
              halign=center DISPLAYCLIPPED=TRUE
     title="Method" ;
endlayout;
endsidebar;
```

All the pieces are now in place. All that is left is to generate the graph by submitting “do_plot_template” macro call and the SGRENDER procedure below:

```
%do_plot_template;

PROC SGRENDER data=plotdate template= methodplot;
RUN;
```

CONCLUSION

This paper demonstrates some helpful ways to visualize potential associations between environmental and clinical surveillance data effectively. Such exploratory tools can be invaluable in understanding the nature of multiple data series, and in building models relating the series. As seen in this paper, creating graphs using SAS® procedures such as PROC SGPLOT and GTL, together with the available plotting options, keeps the needed programming from becoming more complicated than it needs to be. The ultimate goal is to visually communicate the data at hand that scientists can use to interpret and draw conclusions.

REFERENCES

1. World Health Organization. 2000. W.H.O. report on global surveillance of epidemic prone infectious diseases, communicable diseases and surveillance response. W.H.O./CDS/CSR/ISR/2000. World Health Organization, Geneva, Switzerland.
2. World Health Organization. Cholera, 2005.
3. Epidemiology and Ecology of V. Cholerae in Bangladesh NIH Grant Citation. http://projectreporter.nih.gov/project_info_details.cfm?aid=8090286&icde=13374901.
4. Beyond the Basics: Advanced PROC REPORT Tips and Tricks. Allison McMahill SAS Global Forum 2011, Paper 246-2011
5. Traffic Lighting: The Next Generation, Julie VanBuskirk and Jennifer S. Harper SAS Global Forum 2012, Paper 282-2012

ACKNOWLEDGMENTS

We would like to thank principle investigators Drs. Brad Sack and David Sack, and all investigators involved in the 'Epidemiology and Ecology *of V. cholera* in Bangladesh' study.

CONTACT INFORMATION

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

Neeta V. Shenvi
Department of Biostatistics and Bioinformatics
Rollins School of Public Health
Emory University
Atlanta, GA 30322
E-mail: nshenvi@emory.edu