A SAS® Macro Tool for Selecting Differentially Expressed Genes from Microarray Data

Huanying Qin, Laia Alsina, Hui Xu, Elisa L. Priest
Baylor Health Care System, Dallas, TX

ABSTRACT

DNA Microarrays measure the expression of thousands of genes simultaneously. Commercial software such as JMP®/Genomics and GeneSpring® use T-tests, ANOVA, or mixed models for statistical analysis to identify differentially expressed genes. These methods are valid for larger sample sizes. We work with an immunology laboratory that often needs to analyze data from experiments with less than 10 samples. The researchers developed an Excel-based process to select differentially expressed genes from microarray experiments with a small sample size. This process required complex, manual manipulation of data and could take weeks to complete. We created a SAS MACRO to automate the whole process. The program reads microarray data from and provides a summary report in Excel. The researchers can easily modify the parameters and repeat the analysis. The program made it possible to reduce data processing time from weeks to minutes with no mistakes related to manual manipulation. In addition, it provides more output information for further analysis. This paper describes the tool and uses real data to demonstrate that it is valid and efficient.

INTRODUCTION

Microarray has become a common tool for identifying differentially expressed genes under different experimental conditions. Unlike traditional statistical set ups, microarray data are summarized in a matrix format where the genes (outcomes) are in rows and different subjects or samples are in columns. In addition, the number of subjects is much smaller than the number of genes. For example, the Illumina human array V4 contains over 47,000 probe sets. After applying filtering criteria for probes with a present call in at least one sample using a cutoff detection p-value of 0.01, it is common to use around 20,000 probes for input into an analysis. Software such as JMP/Genomics and Gene Spring are often used to handle this large data. These programs are used primarily when there is a reasonable sample size because they use statistical methods based on differences in gene expression value across groups (such as cases and controls) to select differentially expressed genes. However, in the setting of in vitro experiments, it is common to have less than 10 samples per study group. The standard software programs are limited in their ability to handle this type of data because they treat subjects as a group, not individually. In addition, GeneSpring selects differentially expressed genes based on fold change and does not take difference in intensity into account, and fold change can be magnified in low expressed probes.

Because of the limitations of the standard software to analyze microarray data from in vitro experiments with a small sample size, the researchers used experience and clinical knowledge to develop an algorithm that selects differentially expressed genes based on both difference in expression intensity and fold change. This algorithm was first implemented as an Excel -based tool. However, using the Excel tool required complex and tedious manual work and the analysis of each experimental condition took researchers many days to complete. Usually, the researchers needed to analyze data for multiple experimental conditions and because of the complexity of the microarray gene expression data, the full analysis could easily take weeks to finish. The SAS macro tool described in this paper is able to run an analysis for multiple experimental conditions automatically within minutes and only requires the user to provide values for 5 macro parameters.

WORK FLOW AND PROGRAM SETUP

We will demonstrate the use of the SAS tool to analyze a microarray experiment with 2 samples (baseline and stimulation) for each of the subjects. The analysis work flow of a microarray experiment involves three key steps:

1) The researcher defines the macro parameters and prepares the data following a standard formatting and naming convention.

2) The macro parameters in the tool are updated by the statistician and the SAS program is executed.

3) The researcher is informed of the results in the output folder.

MACRO PARAMETERS

Infold= the folder name where the researcher stores the input microarray data, using the format: Project_experiment_researcher name_date_difference_fold change

Difference= the cutoff value for difference between the two samples (ie baseline and stimulation) for each subject and equal to the value for difference in the infold name

Fold= the cutoff value for fold change between baseline and stimulated samples for each subject and equal to the value for fold change in the infold name

N= the number of subjects

Thresh= the threshold used to determine whether a gene is truly up or down regulated genes. To consider a gene up or down regulated, it needs to be up/down regulated in more than one subject. This threshold varies from experiment to experiment, but our laboratory established the use of two thirds of subjects tested for most experiments. This means that a probe will be considered differentially expressed upon a stimulation (named truly up/down) only if it is so in two thirds of the subjects tested.

As shown in Fig 1, first the researcher determines the values for the 5 macro parameters listed above. Next, they prepare the data following a standard naming and formatting convention which documents the parameters used in the algorithm in each analysis. We have dedicated a folder on a shared drive to store all input and output data files.

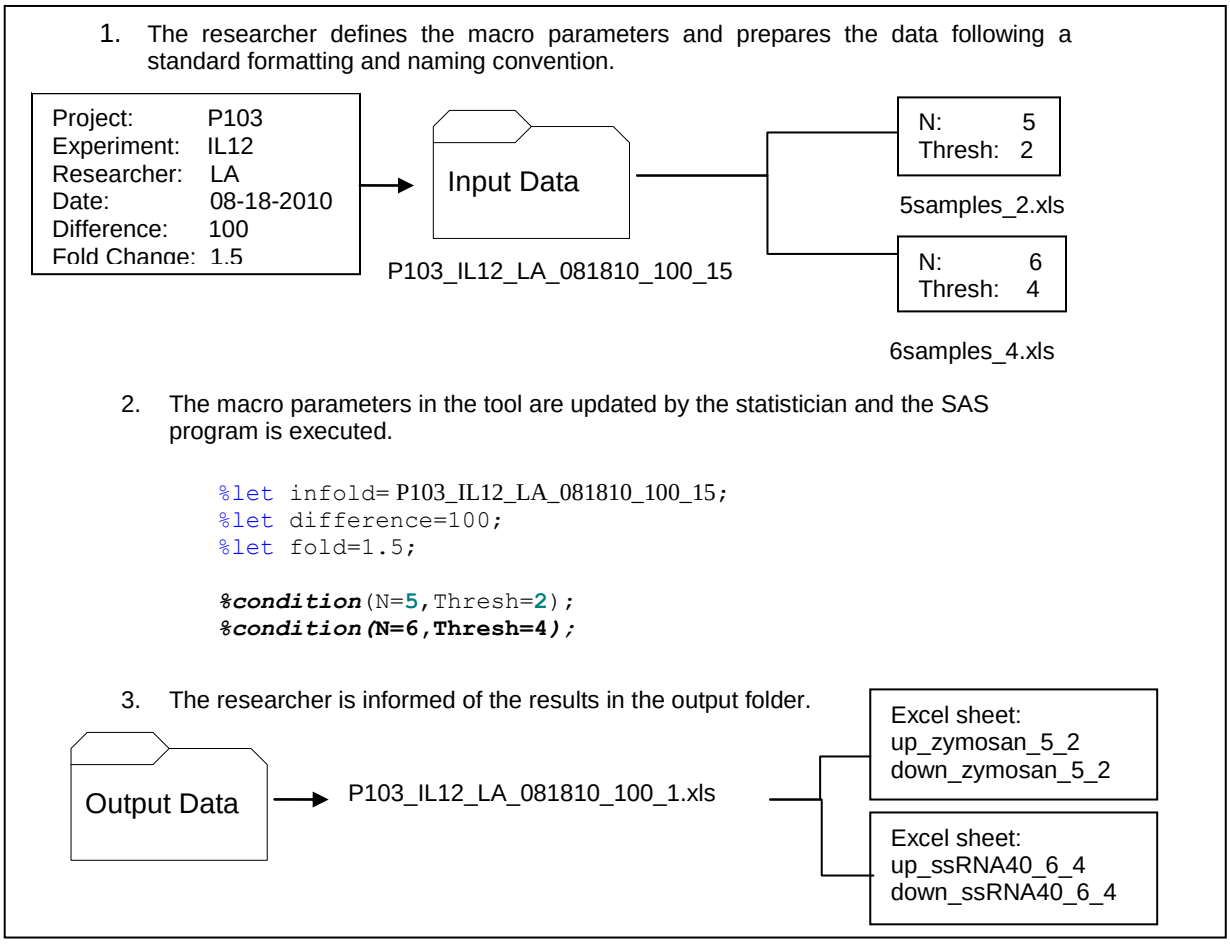
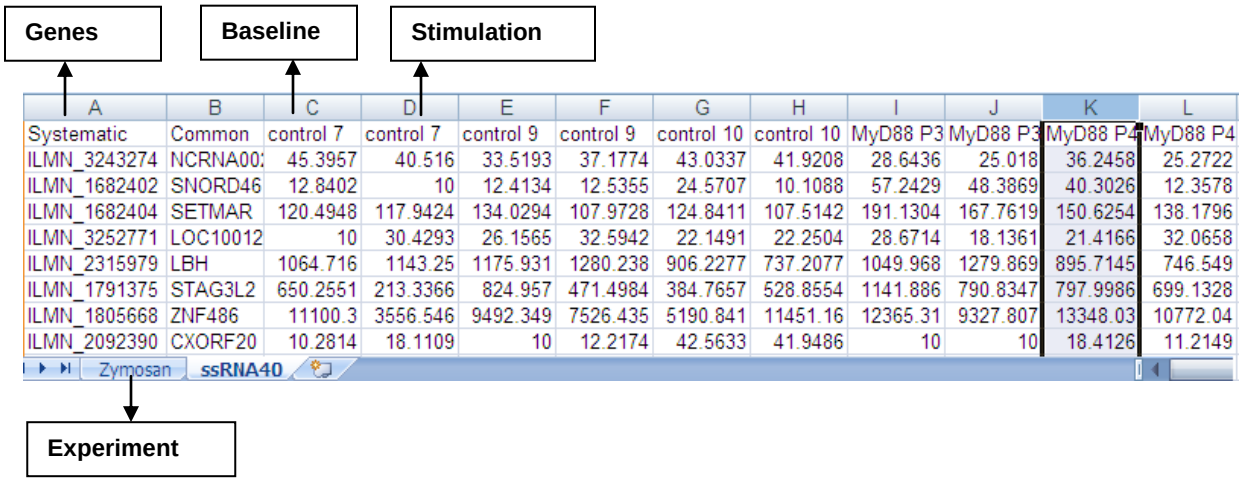


Figure 1. Workflow and program setup

Whenever a researcher has input Excel datasets ready, he or she stores them in a newly created subfolder. The researcher uses the values for difference and fold change to name the subfolder. The subfolder name contains information for project, researcher name, date, and cutoff values for difference and fold change. Common values used for fold change are 1.5 and 2.0, but because the infold name will be used as a part of output file name, the '.' is removed and 15 is used to denote 1.5.

Next, the researcher uses the values from the threshold and the number of subjects to name the input Excel files in the format Nsamples_Thresh. Each file can have multiple sheets and each sheet is named after the experimental condition. However, each of the sheets within one Excel file needs to have the same number of subjects.

Finally, the researcher confirms that each Excel worksheet is setup correctly. The data on the worksheet is arranged with the following rules (Fig 2): 1) sheet name reflects the experimental condition, 2) in each sheet, data for the subjects is arranged in columns, 3) baseline and stimulation samples are side by side for each subject, with baseline first, 4) columns for each subject have the same name.



	A	B	C	D	E	F	G	H	I	J	K	L
	Systematic	Common	control 7	control 7	control 9	control 9	control 10	control 10	MyD88 P3	MyD88 P3	MyD88 P4	MyD88 P4
	ILMN_3243274	NCRNA00	45.3957	40.516	33.5193	37.1774	43.0337	41.9208	28.6436	25.018	36.2458	25.2722
	ILMN_1682402	SNORD46	12.8402	10	12.4134	12.5355	24.5707	10.1088	57.2429	48.3869	40.3026	12.3578
	ILMN_1682404	SETMAR	120.4948	117.9424	134.0294	107.9728	124.8411	107.5142	191.1304	167.7619	150.6254	138.1796
	ILMN_3252771	LOC10012	10	30.4293	26.1565	32.5942	22.1491	22.2504	28.6714	18.1361	21.4166	32.0658
	ILMN_2315979	LBH	1064.716	1143.25	1175.931	1280.238	906.2277	737.2077	1049.968	1279.869	895.7145	746.549
	ILMN_1791375	STAG3L2	650.2551	213.3366	824.957	471.4984	384.7657	528.8554	1141.886	790.8347	797.9986	699.1328
	ILMN_1805668	ZNF486	11100.3	3556.546	9492.349	7526.435	5190.841	11451.16	12365.31	9327.807	13348.03	10772.04
	ILMN_2092390	CXORF20	10.2814	18.1109	10	12.2174	42.5633	41.9486	10	10	18.4126	11.2149

Zymosan | ssRNA40

Figure 2. Input dataset

SAS PROGRAM OVERVIEW

Once the researchers have the folders and Excel files setup, they contact the statistician. The statistician uses all the information in the folder names and Excel files to get the values for the macro variables needed to run the analysis tool.

```
%let infold= P103_IL12_LA_081810_100_15;
%let difference=100;
%let fold=1.5;

%macro condition(N, Thresh);
libname tool "I:\Huanying\tool\data\&infold\&N.samples_&Thresh..xls";

***Process data and apply selection algorithm here***;

PROC EXPORT DATA= WORK.up_fold
            OUTFILE= "I:\Huanying\tool\output\&infold..xls"
            DBMS=EXCEL REPLACE;
     SHEET="up_&&dn&k.._&N._&Thresh"; RUN;

PROC EXPORT DATA= WORK.down_fold
            OUTFILE= "I:\Huanying\tool\output\&infold..xls"
            DBMS=EXCEL REPLACE;
     SHEET="down_&&dn&k.._&N._&Thresh";
RUN;
%end;
%mend;

%condition(N=5,Thresh=2)
%condition(N=6,Thresh=4)
```

TECHNICAL DETAILS

The SAS Macro code includes 3 important steps. These steps can be repeated for each Excel file in the input data folder.

1. Use sheets in Excel file to create two macro lists of variables
2. %do loop to carry out all calculations
3. Separate true up and true down probes and output them to excel file.

Step 1: Use sheets in Excel file to create two macro lists of variables

```
libname datain"I:\Huanying\tool\data\&infold\&N.samples_&Thresh..xls";
proc contents data=datain._all_ ;
ods output members=work.member;
run;
```

The ODS OUTPUT statement creates a dataset work.member that contains all sheet names for the input Excel file. Next, the program uses PROC SQL to count the number of experiments and create two macro variable lists. Macro variables &DA1--&DA&K provide the input for the do loop, and macro variables &DN--&DN&K are modified values that will be used in the output file names.

```
proc sql ;
select count(distinct name) into :k
from member;
quit;
%LET K=&K;

proc sql;
select distinct name into :da1-:da&k
from member;
quit;

data member1;
set member;
substr(name, index(name, '$'), 1)='';
run;

proc sql;
select distinct name into :dn1-:dn&k
from member1;
quit;
```

Step 2: %do loop to carry out all calculations

For each experiment, compare the data from the same subject for the two samples (the baseline sample and a stimulation sample). The difference and fold change between the baseline and stimulation samples will be calculated using several arrays, but before that, all baseline and stimulation sample names need to be separated.

Code 1: create 2 variable lists for arrays.

To create the variable lists, first drop the variables systematic and common. Next, we use the column position to determine whether a sample is a baseline or stimulation. The odd values are the baseline and the even values are the stimulation.

```
proc contents data=datain."&&da&k"n(drop=systematic common ) varnum  ;
ods output position=name;
run;

data odd even;
set name;
if num/2>intz(num/2) then output odd;
else output even;
run;

PROC SQL ;
```

```
SELECT variable INTO :base SEPARATED BY " "
FROM Odd;
SELECT variable INTO :stim SEPARATED BY " "
FROM even;
Quit;
```

Code 2: a data step to calculate difference and fold change for each subject.

Next, using the macro variable lists created above, create arrays of values. After the arrays are created, perform the calculations for the differences and fold change for each gene for each subject.If you need to include source code:

```
array base(&N) &base;
array stim(&N) &stim;
array diff(&N) diff1-diff&N;
array fold(&N) fold1-fold&N;
array up(&N) up1-up&N;
array down(&N) down1-down&N;

do j=1 to &N;
diff(j)=abs(stim(j)-base(j));
if stim(j)>=base(j) then fold(j)=stim(j)/base(j);
else fold(j)=-base(j)/stim(j);
```

Compare each calculated difference to the &difference. and &fold. cutoff values specified by the researcher. This will determine whether each gene is up or down regulated for each subject.

```
if diff(j)>=&difference and fold(j)>=&fold then up(j)=1;
else up(j)=0;
if diff(j)>=&difference and fold(j)<=-&fold then down(j)=1;
else down(j)=0;
```

Next, use the &Thresh cutoff specified by the researcher to determine whether genes are considered ‘truly’ up or down regulated for the subjects.

```
sum_up=sum(of up1-up&N);
sum_down=sum(of down1-down&N);
if sum_up>=&Thresh then true_up=1; else true_up=0;
if sum_down>=&Thresh then true_down=1; else true_down=0;
```

Finally, summarize the data across all of the genes. Count the up regulated genes as well as those considered ‘truly’ upregulated based on the &Thresh value. Then use a nested macro to add up the fold change across probes among the true up and upregulated genes. Similar code is repeated to summarize the down regulated and ‘truly’ downregulated genes.

```
proc means data=signature sum noprint;
var true_up up1-up&n_base;
where true_up=1;
output out=sum_up sum=/autoname;
run;

%macro mean(N);
%do i=1 %to &N;
proc means data=signature sum;
var fold&i;
where true_up=1 and up&i=1;
output out=sum_fold&i sum=/autoname;
run;
%end;
%mend;
```

Step 3: Separate the ‘truly’ up and down regulated genes and output them to Excel.

The last step in the SAS tool separates the ‘truly’ up and down regulated genes and outputs them to Excel. This is the gene list of interest for the researcher (contains usually hundreds of probes). The upregulated and downregulated

genes are output to separate Excel worksheets. Each experimental condition from the input Excel file will have two output worksheets.

Another highlight of this tool is that it provides summary information along with the differentially expressed gene list. First, it outputs the fold change value for each probe and each subject. Also, in order to quantify and compare the degree of transcriptional change observed between subjects for a particular experimental condition, the tool provides researchers with two different metrics based on the differentially expressed gene list: 1) the sum of genes that are differentially expressed in each subject 2) the sum of all fold changes greater than a certain threshold (up or down regulation). This last metric is named Cumulative Score and has already been described (Pankla R, Buddhisa S, Berry M, et al. Genomic transcriptional profiling identifies a candidate blood biomarker signature for the diagnosis of septicemic melioidosis. Genome Biol. 2009; 10:R127).

```
data up_fold(drop=true_up);
merge signature(keep=systematic common true_up fold1-fold&N where=(true_up=1))
sum_up sum_fold1-sum_fold&N;
drop _type_ _freq_;
run;

PROC EXPORT DATA= WORK.up_fold
            OUTFILE= "I:\Huanying\tool\output\&infold..xls"
            DBMS=EXCEL REPLACE;
     SHEET="up_&&dn&k.._&N._&Thresh";
RUN;
```

Similar code is used to output the downregulated genes.

Finally, all outputs from the same input folder are contained in one single Excel file named after the input folder. This format allows the researcher to compare multiple experiments easily. Each sheet name contains information for up/down, experimental condition, number of subject and threshold value in the format: UP_EXPERIMENT_&N._&Thresh. This way, the researchers can easily find the output file and understand what each sheet is.

Each output sheet contains several levels of data (Fig 3). First, fold changes for each subject for each gene are listed. Next, the total number of up or down regulated genes for each subject, and the total fold change across genes for each subject are given. Finally, the total number of true up and true down regulated genes for all subjects is given.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
Systematic	Common	fold1	fold2	fold3	fold4	fold5	true_up_Sum	up1_Sum	up2_Sum	up3_Sum	up4_Sum	up5_Sum	fold1_Sum	fold2_Sum	fold3_Sum	fold4_Sum	fold5_Sum
ILMN_169	RALGDS	1.243103	1.626375	1.703258	1.510406	1.254188	120	107	102	77	10	2	638.3053	451.9702	253.7087	17.05172	3.420654
ILMN_171	GADD45B	3.031967	2.935594	-1.33413	-1.38797	-1.24485											
ILMN_171	SLC25A24	2.981478	3.674712	2.569509	1.600667	-1.13814											
ILMN_172	TNF	10.95338	7.941783	4.392648	1.072195	1.110519											
ILMN_178	GPR84	6.431412	3.230008	2.097955	1.104472	-1.10171											
ILMN_165	IL1A	65.12101	33.2153	14.82087	1.575842	-2.03183											
ILMN_169	LOC64281	1.664819	-1.02497	1.54187	1.049777	-1.2124											
ILMN_166	IL8	4.403394	3.481886	1.890956	1.282659	-1.22559											

up_ssRNA40_5_2 | down_ssRNA40_5_2 | up_Zymosan_6_3 | down_Zymosan_6_3

Figure 3. Output dataset

CONCLUSION

In conclusion, the SAS macro tool can greatly save time by reducing data processing time in Excel from weeks to minutes. The researchers can also easily analyze the same datasets repeatedly by changing any of the macro parameters on difference, fold change or threshold in order to achieve the best output. In addition, mistakes related to manual data manipulation in Excel are eliminated since the whole analysis is automated. The Excel-based process was originally designed to only obtain the differentially expressed gene list and Cumulative Score, but the SAS macro tool also provides extra outputs including fold change and number of differentially expressed genes for each subject.

ACKNOWLEDGMENTS

Thank you to Dr. Derek Blankenship and Dr. Damien Chaussabel for the support on this project, and to Timothy Zumwalt for validating the tool with your data.

CONTACT INFORMATION

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

Name: Huanying Qin
Enterprise: Baylor Institute of Immunology Research
Address: 3434 Live Oak St.
City, State ZIP: Dallas, Texas 75204
Work Phone: 214-820-9064
Fax: 214-265-3628
E-mail: huanyinq@baylorhealth.edu
Web: NA