

A Confidence Interval Approach to Gene Chip Analysis

Jennifer L. Waller, Medical College of Georgia, Augusta, GA
Mark G. Anderson, Medical College of Georgia, Augusta, GA

ABSTRACT

Recently, gene chip technology and data generated from this technology have given statisticians a new realm in which to develop statistical methodology. In gene chip analysis, mRNA from a tissue sample for a disease of interest is isolated and placed on the chip. From expression data generated from the chip, we then determine what genes are highly expressed for the particular disease of interest. The challenge is to find approximately 100 genes from 12,000 that warrant further investigation. To further complicate matters, gene chips are very expensive (~\$1000 each) and sample sizes tend to be very small, with $n \leq 5$ chips typically used.

Using SAS 8.0 ® under Windows 2000 © and programmed mostly within a DATA step, a confidence interval approach was developed that examines various thresholds for gene expression. mRNA from 8 normal and 8 apolipoprotein AI (apo AI, a gene known to play a pivotal role in HDL metabolism) knocked out mice were isolated and 16 gene chips were analyzed. Expression ratios (red/green) for each gene were determined within a gene chip (mouse) and 99% confidence intervals calculated for the mean expression ratio across the 8 mice within a treatment group (normal and apo AI). The lower limit of each confidence interval is compared to a series of expression ratio thresholds and those genes whose lower limit is greater than the threshold are flagged in both groups. The total number of genes flagged for each threshold is calculated for each group and plotted using SAS/GRAPH ®. Additionally, the genes flagged are compared between groups to determine how the genes are differentially expressed.

INTRODUCTION

cDNA microarrays, or gene chips, is a technology that enables a researcher to examine gene expression levels for thousands of genes simultaneously. The goal with most microarray experiments is to identify a number of genes that are differentially expressed in a diseased sample versus a referent sample. Gene identification is not the end of the investigation, however, but it is just the tip of the iceberg. From the identification of a subset of genes, a microbiologist, cellular biologist, physiologist or other basic science researcher can then further explore the functions of the subset of genes rather than having to sift through thousands. Thus the goal is to narrow down the possibilities, so that the more intensive and time consuming proteomics work can begin.

While opening up doors for clinical research, this microarray technology has also introduced a host of analytic opportunities for statisticians. This paper describes a relatively simple confidence interval approach to determining specific genes that are over or under expressed when there are two groups with replicate gene chips. This paper will give a brief overview of microarrays, describe the microarray data used, describe the confidence interval approach, and finally give further analytic issues which are currently being investigated.

OVERVIEW OF CDNA MICROARRAYS

Wildsmith and Elcock (2001), Hedenfalk et al. (2001), and Hamadeh and Afshari (2000) give good descriptions of microarrays and microarray processing. There are essentially two types of arrays, microarrays and oligonucleotide arrays. Microarrays are created by depositing a large number of genes onto a glass slide. cDNA clones are spotted in an array arrangement onto a glass slide or nylon membrane using a robotic spotting printer. mRNA from a reference sample and from a disease sample are reverse transcribed with different

fluorescent dyes, green and red. Usually, green is used for the referent sample and red is used for the disease sample.

Following the reverse transcriptions with the dyes, the mRNA samples are hybridized to a cDNA microarray containing the robotically printed cDNA clones. The microarrays are then washed several times to remove unbound mRNA. Following the washes, the microarrays are scanned with a laser scanning microscope to obtain color images of the hybridization mRNA from the diseased and referent cells. Two images are then sent to a computer which generates the location and intensity of spots, one image for the green and one image for the red. These images are then overlaid. Genes which are over expressed in the diseased sample appear as red dots on the microarray image, those which have decreased expression appear as green dots, and those which are not differentially expressed in either the diseased sample or the referent sample appear as yellow dots. Because of the limitation of analyzing an image, each spot is assigned an expression value for red and an expression value for green. These two values are then used to create a red to green expression ratio and this ratio is used in analyses.

APOLIPOPROTEIN AI AND CONTROL GROUP MICROARRAY DATA

Callow et al. (2000) studied two lines of mice with very low HDL cholesterol levels compared to inbred control mice. Data for this paper consisted on one of the two studied mouse lines, the apolipoprotein AI (or apo AI) knock-out line. Of interest in this study was to determine genes with differential expression for low HDL. Tissue samples were taken from the livers of the mice.

The treatment group consisted of 8 mice with the apo AI gene knocked out. The control group consisted of 8 control C57B1/6 mice. Target cDNA was obtained from mRNA by reverse transcription and labeled using red fluorescent dye, Cy5 from each of the 16 mice. The referent sample cDNA was prepared by combining the cDNA from the 8 control mice and was labeled using green fluorescent dye, Cy3. Each microarray consisted of the cDNA for one mouse (i.e. the "diseased" sample representing the red dye) and the combined cDNA of the 8 control mice (i.e. the referent sample representing the green dye).

The cDNA was hybridized to a microarray that contained 5,548 cDNA probes, including 200 probes that were related to lipid metabolism. The microarrays were then imaged and red and green fluorescence intensities were generated for each cDNA probe.

THE IMPORTANCE OF REPLICATION

Due to the cost (approximately \$1,000) and the vastness of the data that is produced by a single microarray, many researchers choose to perform only a single replicate or to pool data from several diseased samples and several referent samples and use these pooled data on a single chip. While cost effective, this is not an ideal approach. Lee et al. (2000) examined the variability between replicated microarrays and showed that a single microarray experiment has substantial variability from a variety of sources. In their paper, three replicates were used and one replicate gave different results than the other two indicating that "replication does not ensure duplication of results, a fact that cannot be quantified when replication is not used." (Lee et al., 2000). Thus, when working with researchers on microarray data it is necessary to educate them about replication and how replication can help them get a handle on the different sources of variability.

CONFIDENCE INTERVAL APPROACH TO GENE IDENTIFICATION

The confidence interval approach taken to examine genes and determine which are differentially expressed in the diseased samples versus the control samples is a relatively simple one. The first step in construction of a confidence interval is to examine the skewness of the ratio statistic, R/G , calculated for each replicate within each gene. Transformations of the ratio statistic may be necessary before the confidence intervals are constructed. In this case, a family of transformations was examined and the transformation that produced the lowest mean skewness across genes was used. The family of transformations (Box and Tiao, 1992, pg. 530) used was

$$y^{(\lambda)} = \begin{cases} \frac{y^\lambda - 1}{\lambda} & (\lambda \neq 0) \\ \log(y) & (\lambda = 0) \end{cases} \quad y > 0$$

where $y=R/G$ and $-2 \leq \lambda \leq 2$. After examining the skewness of the transformed ratio statistic, it was determined that the $\log(R/G)$ was the least skewed. Because many researchers are interested in at least 2-fold increases, $\log_2(R_r/G_r)$, r =the replicate, was used. SAS code for examining the skewness of different transformations is included with comments indicating code for various steps.

For each gene, a $100(1-\alpha)\%$ confidence interval for the mean ratio of R/G using the number of replicates within a group was calculated by the following:

$$\bar{x}_g \pm t_{(r-1, 1-\alpha/2)} se(\bar{x}_g)$$

where g is the group, diseased or control, r is the number of replicates in each group, and $\bar{x}$ is calculated as the mean of $\log_2(R_r/G_r)$.

After calculating the confidence interval for each group, diseased or control, the lower and upper limits are compared to various expression ratio thresholds. If the lower limit is greater than the threshold, then the gene is considered to be over expressed for that particular group. If the upper limit is lower than the inverse of the threshold then the gene is considered to be under expressed for that particular group.

While knowing that a gene is differentially expressed within the diseased group or the control group of interest, it is also necessary to determine whether the gene was differentially expressed in both groups, only in the diseased group, only in the control group, or in neither group. This additional information is helpful in assessing which genes to concentrate further laboratory research on in the future. Newton et al. (2000) indicate that the ratio of the expression genes within a microarray vary greatly, i.e. some may have high expression for any sample and some may have very low expression. Thus knowing whether the gene is expressed in both, neither, or only in one group is necessary.

SAS code is given for the calculation of 99% confidence intervals and commonality of expression with comments indicating the steps taken.

RESULTS

The confidence interval method identified 7 genes which were under expressed in the diseased group of microarrays. Dudoit and Yang et al. (2000) used the same apo AI data in the investigation of their methodology. For the apo AI experiments they identified 8 genes which were differentially expressed in the apo AI knocked out mice as compared to the control mice. The confidence interval method identified 7 genes which were under expressed at the 2-fold level, six of which were the same genes identified by Dudoit and Yang. The different results between the CI approach and that taken by Dudoit and Yang is probably due to the different alpha levels used, here 0.01 and Dudoit and Yang

0.05.

The output of the number of genes identified at each threshold are given, the commonality of the genes expressed at each threshold in the apo AI mice and the control mice, and the plots of differentially (either over or under) expressed genes identified at the 2-fold level are shown following the SAS code. Output is shown only when genes were identified as over or under expressed.

FUTURE INVESTIGATIONS

Newton et al. (2001) indicate that when considering fold changes (e.g. 1.5-fold or 2-fold) in gene expression utilizing a ratio measure ignores the variation of the ratio across genes. In essence their argument is that a 1.5-fold change for one gene may be perfectly adequate to show that the specific gene is differentially expressed, but that a 3-fold change in another gene may be required to show differential expression. While the methodology he describes deals with a single microarray, he indicates that combining the information from replicates may aid in identifying the contribution of different sources of variation. Thus one possibility to investigate is the variability between genes in their expression and consider using different ratio threshold levels for each gene based on their individual distribution.

Second, Yang and Dudoit et al. (2000) and Chen et al. (1997) describe different normalization techniques for cDNA microarray data. These include within slide location normalizations (global, intensity dependent, and within print-tip-group), within slide scale normalizations, paired slide dye-swap normalizations, and multiple slide normalizations. Implementing these various normalizations within SAS before analysis is another area for further investigation.

Third, the number of confidence intervals which are generated is extremely large and the potential of misidentification of differentially expressed genes is large. Dudoit and Yang et al. (2000) investigated a two-sample t-statistic for determining differential expression, performed permutation tests on this test statistic, and calculated adjusted p-values using a variety of different methods. Implementing Dudoit and Yang et al.'s methodology in SAS is currently underway.

CONCLUSION

The relatively simple confidence interval method presented here provides similar results to others who have utilized the data from the apo AI experiments conducted by Callow et al. (2000). Improvements in normalization and implementation of permutation tests within SAS will provide additional tools for the analysis of microarray data.

Analysis of microarray data is of great interest and new methodologies and utilization of and improvements in old methodologies emerge every day. Additionally, many new software programs are currently on the market which aid a researcher in analyzing microarray data using both old and new methodologies. However, researchers must be aware of the pitfalls of just pointing and clicking without understanding their data and the methods they have used for analysis in these new programs. Rather than investing in new software, which can be quite costly, investing the time to learn the new package and verifying that the software is performing an analysis as it should, implementing current and new methodologies and investigating the capabilities in SAS is desirable. SAS is a powerful tool, we just need to educate the people we work with as to how powerful it is!

REFERENCES

Box, GEP, Tiao, GC. (1992) Bayesian Inference in Statistical Analysis. New York, NY: John Wiley and Sons.

Callow, MJ, Dudoit, S, Gong, EL, Speed TP, and Rubin EM. Microarray expression profiling identifies genes iwht altered expression in hdl deficient mice. *Genome Research* 10(12):2022-9, 2000.

Chen, Y, Dougherty, ER, Bittner, ML. Ratio-based decision and the quantitative analysis of cDNA microarray images. *Journal of Biomedical Optics* 2:364-374, 1997.

Dudoit, S, Yang, YH, Callow, MJ, Speed, TP. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. (Technical report #578) Submitted, *Journal of the American Statistical Association*, 2000.

Hamadeh, H, and Afshari, CA. Gene chips and functional genomics. *American Scientist* 88:508-515.

Hedenfalk, I, Duggan, D, Chen, Y, Radmacher, M, Bittner, M, Simon, R, Meltzer, P, Busterson, B, Esteller, M, Kallioniemi, OP, Wilfond, B, Borg, A, Trent J. Gene expression profiles in hereditary breast cancer. *The New England Journal of Medicine* 344(8):539-548, 2001.

Lee, MLT, Kuo, FC, Whitmore, GA, Sklar, J. Importance of replication in microarray gene expression studies: Statistical methods and evidence from repetitive cDNA hybridizations. *Proceedings of the National Academy of Science*, 97(18):9834-9839, 2000.

Newton, MA, Kendzierski, CM, Richmond, CS, Blattner, FR, Tsui, KW. On differential variability of expression ratios: Improving statistical inference about gene expression changes from microarray data. *Journal of Computational Biology* 8(1): 37-52, 2001.

Wildsmith, SE, Elcock, FJ. Microarrays under the microscope. *Molecular Pathology*, 54(1):8-16, 2001.

Yang, YH, Dudoit, S, Luu, P, Speed, T. Normalization for cDNA microarray data. Technical Report #589, submitted, 2001.

CONTACT INFORMATION

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

Jennifer L. Waller
Medical College of Georgia
Office of Biostatistics and Bioinformatics (AE-3031)
Augusta, GA 30912-4900
Work Phone: (706) 721-3785
Fax: (706) 721-6294
Email: jennifer@stat.mcg.edu

SAS, SAS/GRAPH and all other SAS products or services are registered trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration, © indicates copyright.

Other brand and product names are registered trademarks or trademarks of their respective companies.

TRANSFORMATION SAS PROGRAM

```
*****
** This program examines different      **
** transformations of R/G to determine **
** which has the lowest mean skewness  **
** across genes.                       **
*****
data apoal;
  set in.apoal;
```

```
*****
** Creates 8 observations per gene. Creates**
** ratio of R/G for each gene for each   **
** group, disease (t) and control (c).    **
*****;
array cg {8} clg c2g c3g c4g c5g c6g c7g c8g;
array cr {8} clr c2r c3r c4r c5r c6r c7r c8r;
array tg {8} klg k2g k3g k4g k5g k6g k7g k8g;
array tr {8} klr k2r k3r k4r k5r k6r k7r k8r;
do i=1 to 8;
  gene=genenum;
  cgexp=cg{i};
  crexp=cr{i};
  tgexp=tg{i};
  trexp=tr{i};
  crg=cr{i}/cg{i};
  trg=tr{i}/tg{i};
  output;
end;
keep gene cgexp crexp tgexp trexp crg trg;
*****
** Create transformations with lambda **
** in the interval -2 to 2, by 0.5.    **
*****;
data new;
  set apoal;
  array cln {4} cl_n05 cl_n1 cl_n15 cl_n2;
  array cl {4} cl_05 cl_1 cl_15 cl_2;
  array tln {4} tl_n05 tl_n1 tl_n15 tl_n2;
  array tl {4} tl_05 tl_1 tl_15 tl_2;
  do i=1 to 4;
    cln{i}=((crg)**(-(i/2))-1)/(-(i/2);
    cl{i}=((crg)**((i/2))-1)/(i/2);
    tln{i}=((trg)**(-(i/2))-1)/(-(i/2);
    tl{i}=((trg)**((i/2))-1)/(i/2);
  end;
  cl_0=log(crg);
  tl_0=log(trg);

proc sort data=new; by gene;
*****
** Calcualte skewness measure for the ratio**
** and transformations of R/G in each group**
** for each gene.                          **
*****;
proc means data=new skew noprint;
  var trg tl_n2 tl_n15 tl_n1 tl_n05 tl_0 tl_05
      tl_1 tl_15 tl_2 crg cl_n2 cl_n15 cl_n1
      cl_n05 cl_0 cl_05 cl_1 cl_15 cl_2;
  output out=vs skew=strg stl_n2 stl_n15
      stl_n1 stl_n05 stl_0 stl_05 stl_1
      stl_15 stl_2 scrg scl_n2 scl_n15
      scl_n1 scl_n05 scl_0 scl_05 scl_1
      scl_15 scl_2;
  by gene;
*****
** Calculate the mean skewness of R/G      **
** transformations across genes to determine **
** which gives a mean skewness nearest zero. **
*****;
proc means data=vs n mean std median;
  var strg stl_n2 stl_n15 stl_n1 stl_n05
      stl_0 stl_05 stl_1 stl_15 stl_2;
  title 'Diseased Group Mean Skewness Across
      Genes';

proc means data=vs n mean std median;
  var scrg scl_n2 scl_n15 scl_n1 scl_n05
      scl_0 scl_05 scl_1 scl_15 scl_2;
  title 'Control Group Mean Skewness Across
      Genes';
run;
```

CONFIDENCE INTERVAL SAS PROGRAM

```
*****
** This program calculates a CI for each gene **
** within each group, disease (t) or control **
** (c ). The lower and upper limits are **
** compared to different thresholds to **
** determine the number of differentially **
** expressed genes in each group. A **
** commonality variable is created to show **
** whether the gene expressed in both, only **
** one, or neither the disease or control **
** group. Finally graphs are produced which **
** show the lower limit, mean, and upper limit **
** of the log2(R/G) expression for each group. **
*****;
proc format;
  value oexp 1='Over Expressed'
            0='Not Expressed';
  value uexp 1='Under Expressed'
            0='Not Expressed';
  value common 1='Both Exp'
              2='Cntl Exp'
              3='Trt Exp'
              4='Neither Exp';

data apoa1;
*****
** Create ratios of log2(r/g) for each **
** sample. Also create **
** log2(r/g)_trt/log2(r/g)_ctl. **
*****;
  array cg c1g c2g c3g c4g c5g c6g c7g c8g;
  array cr c1r c2r c3r c4r c5r c6r c7r c8r;
  array tg k1g k2g k3g k4g k5g k6g k7g k8g;
  array tr k1r k2r k3r k4r k5r k6r k7r k8r;
  array cratio cratiol-cratio8;
  array tratio tratiol-tratio8;
  do over cg;
    cratio=log2(cr/cg);
    tratio=log2(tr/tg);
  end;
*****
** Calculate the mean, standard error **
** and effective sample size across **
** samples within the groups. **
*****;
  /* Normal - Control Group */
  m_cratio=mean(of cratiol-cratio8);
  se_cratio=stderr(of cratiol-cratio8);
  n_cratio=n(of cratiol-cratio8);
  /* Disease - Treatment Group */
  m_tratio=mean(of tratiol-tratio8);
  se_tratio=stderr(of tratiol-tratio8);
  n_tratio=n(of tratiol-tratio8);
*****
** Set alpha, degrees of freedom, and **
** reliability coefficient for each **
** group. **
*****;
  alpha=0.995;
  cdf=n(of cratiol-cratio8)-1;
  tdf=n(of tratiol-tratio8)-1;
  ctcoeff=tinv(alpha,cdf);
  ttcoeff=tinv(alpha,tdf);
*****
** Calculate upper and lower confidence **
** intervals for each group. **
*****;
  cucl=m_cratio+ctcoeff*se_cratio;
  clcl=m_cratio-ctcoeff*se_cratio;
  tucl=m_tratio+ttcoeff*se_tratio;
  tlcl=m_tratio-ttcoeff*se_tratio;
```

```
*****
** Determine if lower confidence limit **
** is greater than a threshold value. **
** Determine if upper confidence limit **
** is less than a threshold value. **
** Determine the commonality of **
** expression. **
*****;
  /* R greater than G */
  array cthresh {9} cthresh1 cthresh1_25
                  cthresh1_5 cthresh1_75
                  cthresh2 cthresh2_25
                  cthresh2_5 cthresh2_75
                  cthresh3;
  array tthresh {9} tthresh1 tthresh1_25
                  tthresh1_5 tthresh1_75
                  tthresh2 tthresh2_25
                  tthresh2_5 tthresh2_75
                  tthresh3;
  array commona {9} common1 common1_25
                  common1_5 common1_75
                  common2 common2_25
                  common2_5 common2_75
                  common3;

  do k=1 to 9;
    cthresh{k}=0;
    tthresh{k}=0;
    if clcl>((k+1)-((3*k)/4)) then cthresh{k}=1;
    if tlcl>((k+1)-((3*k)/4)) then tthresh{k}=1;
    if cthresh{k}=tthresh{k}=1 then
      commona{k}=1;
    else if cthresh{k}=1 and tthresh{k}=0 then
      commona{k}=2;
    else if cthresh{k}=0 and tthresh{k}=1 then
      commona{k}=3;
    else if cthresh{k}=tthresh{k}=0 then
      commona{k}=4;
  end;
  /* G greater than R */
  array cthresha {9} cthreshn1 cthreshn1_25
                    cthreshn1_5 cthreshn1_75
                    cthreshn2 cthreshn2_25
                    cthreshn2_5 cthreshn2_75
                    cthreshn3;
  array tthresha {9} tthreshn1 tthreshn1_25
                    tthreshn1_5 tthreshn1_75
                    tthreshn2 tthreshn2_25
                    tthreshn2_5 tthreshn2_75
                    tthreshn3;
  array commonaa {9} commonn1 commonn1_25
                    commonn1_5 commonn1_75
                    commonn2 commonn2_25
                    commonn2_5 commonn2_75
                    commonn3;

  do j=1 to 9;
    cthresha{j}=0;
    tthresha{j}=0;
    if cucl<-((j+1)-((3*j)/4)) then
      cthresha{j}=1;
    if tucl<-((j+1)-((3*j)/4)) then
      tthresha{j}=1;
    if cthresha{j}=tthresha{j}=1 then
      commonaa{j}=1;
    else if cthresha{j}=1 and tthresha{j}=0
      then commonaa{j}=2;
    else if cthresha{j}=0 and tthresha{j}=1
      then commonaa{j}=3;
    else if cthresha{j}=tthresha{j}=0 then
      commonaa{j}=4;
  end;
```

```

*****
** Format threshold variables and common **
** gene variables. **
*****;
format cthresh1 cthresh1_25 cthresh1_5
cthresh1_75 cthresh2 cthresh2_25
cthresh2_5 cthresh2_75 cthresh3
tthresh1 tthresh1_25 tthresh1_5
tthresh1_75 tthresh2 tthresh2_25
tthresh2_5 tthresh2_75 tthresh3 oexp.
cthreshn1 cthreshn1_25 cthreshn1_5
cthreshn1_75 cthreshn2 cthreshn2_25
cthreshn2_5 cthreshn2_75 cthreshn3
tthreshn1 tthreshn1_25 tthreshn1_5
tthreshn1_75 tthreshn2 tthreshn2_25
tthreshn2_5 tthreshn2_75 tthreshn3
uexp.
common1 common1_25 common1_5 common1_75
common2 common2_25 common2_5 common2_75
common3 commonn1 commonn1_25 commonn1_5
commonn1_75 commonn2 commonn2_25
commonn2_5 commonn2_75 commonn3
common.;
*****
** Get frequency distribution of threshold **
** variables for control and disease groups. **
*****;
proc freq data=apoal;
tables cthresh1 cthresh1_25 cthresh1_5
cthresh1_75 cthresh2 cthresh2_25
cthresh2_5 cthresh2_75 cthresh3;
title 'Number of Genes with Lower CL';
title2 'Above the Threshold for Control Group';

proc freq data=apoal;
tables cthreshn1 cthreshn1_25
cthreshn1_5 cthreshn1_75
cthreshn2 cthreshn2_25
cthreshn2_5 cthreshn2_75
cthreshn3;
title 'Number of Genes with Upper CL Below';
title2 '1/Threshold for Control Group';

proc freq data=apoal;
tables tthresh1 tthresh1_25 tthresh1_5
tthresh1_75 tthresh2 tthresh2_25
tthresh2_5 tthresh2_75 tthresh3;
title 'Number of Genes with Lower CL';
title2 'Above the Threshold for Disease Group';

proc freq data=apoal;
tables tthreshn1 tthreshn1_25
tthreshn1_5 tthreshn1_75
tthreshn2 tthreshn2_25
tthreshn2_5 tthreshn2_75
tthreshn3;
title 'Number of Genes with Upper CL Below';
title2 '1/Threshold for Disease Group';

proc freq data=apoal;
tables common1 common1_25 common1_5
common1_75 common2 common2_25
common2_5 common2_75 common3
commonn1 commonn1_25 commonn1_5
commonn1_75 commonn2 commonn2_25
commonn2_5 commonn2_75 commonn3;
title 'Number of Common Expressions Control vs
Disease';

```

```

*****
** Plot genes with lower limit greater **
** than 2-fold expression ratio or upper**
** limit less than 1/2-fold expression. **
*****;
goptions reset=(axis, legend, pattern,
symbol, title, footnote)
norotate hpos=0 vpos=0 htext=
ftext= ctext= target= gaccess=
gsfmode= ;
goptions device=WIN ctext=blue graphrc
interpol=join;

symbol1 c=DEFAULT i=none ci=black v=dot;
symbol2 c=DEFAULT i=none ci=black v=x;
symbol3 c=DEFAULT i=none ci=black v=dot;

axis1 order=(-0 to 6 by 0.50) color=blue
width=2.0 offset=(1 cm)
label=('Log 2(LCL) Threshold');
axis2 color=blue width=2.0 offset=(1 cm)
label=('Gene Number');
axis3 order=(-6 to 0 by 0.50) color=blue
width=2.0 offset=(1 cm)
label=('Log 2(UCL) Threshold');

proc gplot data=apoal;
plot genenum*clcl
genenum*m_cratio
genenum*cucl / overlay haxis=axis1
vaxis=axis2 frame;
where cthresh1=1;
title 'Control Expressions with LCL > 1
- 2-fold';

proc gplot data=apoal;
plot genenum*tlcl
genenum*m_tratio
genenum*tucl / overlay haxis=axis1
vaxis=axis2 frame;
where tthresh1=1;
title 'Disease Expressions with LCL > 1
- 2 fold';

proc gplot data=apoal;
plot genenum*clcl
genenum*m_cratio
genenum*cucl / overlay haxis=axis3
vaxis=axis2 frame;
where cthreshn1=1;
title 'Control Expressions with UCL < -1
- 1/2-fold';

proc gplot data=apoal;
plot genenum*tlcl
genenum*m_tratio
genenum*tucl / overlay haxis=axis3
vaxis=axis2 frame;
where tthreshn1=1;
title 'Treatment Expressions with UCL < -1
- 1/2-fold';

run;
quit;
run;

```

Number of Genes with Upper CL Below
1/Threshold for Disease Group

The FREQ Procedure

tthreshn1	Frequency	Percent
Not Expressed	6377	99.89
Under Expressed	7	0.11

tthreshn1_25	Frequency	Percent
Not Expressed	6379	99.92
Under Expressed	5	0.08

tthreshn1_5	Frequency	Percent
Not Expressed	6381	99.95
Under Expressed	3	0.05

tthreshn1_75	Frequency	Percent
Not Expressed	6382	99.97
Under Expressed	2	0.03

tthreshn2	Frequency	Percent
Not Expressed	6382	99.97
Under Expressed	2	0.03

tthreshn2_25	Frequency	Percent
Not Expressed	6383	99.98
Under Expressed	1	0.02

tthreshn2_5	Frequency	Percent
Not Expressed	6383	99.98
Under Expressed	1	0.02

tthreshn2_75	Frequency	Percent
Not Expressed	6383	99.98
Under Expressed	1	0.02

tthreshn3	Frequency	Percent
Not Expressed	6383	99.98
Under Expressed	1	0.02

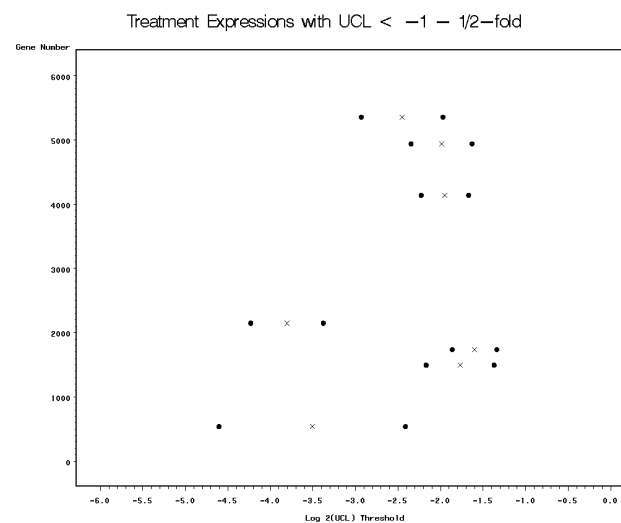
commonn2	Frequency	Percent
Trt Exp	2	0.03
Neither Exp	6382	99.97

commonn2_25	Frequency	Percent
Trt Exp	1	0.02
Neither Exp	6383	99.98

commonn2_5	Frequency	Percent
Trt Exp	1	0.02
Neither Exp	6383	99.98

commonn2_75	Frequency	Percent
Trt Exp	1	0.02
Neither Exp	6383	99.98

commonn3	Frequency	Percent
Trt Exp	1	0.02
Neither Exp	6383	99.98



Number of Common Expressions Control vs Disease

commonn1	Frequency	Percent
Trt Exp	7	0.11
Neither Exp	6377	99.89

commonn1_25	Frequency	Percent
Trt Exp	5	0.08
Neither Exp	6379	99.92

commonn1_5	Frequency	Percent
Trt Exp	3	0.05
Neither Exp	6381	99.95

commonn1_75	Frequency	Percent
Trt Exp	2	0.03
Neither Exp	6382	99.97