/*
These SAS macros were written locally and are maintained by Mayo Clinic staff. They contain the SAS source code, a brief description of the macro's function and an example of the macro call.

Author:

Douglas W. Mahoney
mahoney@mayo.edu

Background:

This macro normalizes data from multiple iTRAQ experiments using a two stage linear model. The first stage consists of modeling the itraq run and channel effects and the second stage consists of modeling the protein and peptide. The linear model to be fit is given by

abundance=itraqrun+channel+itraqrun*channel+peptide(protein)+protein.

Since the main focus is to obtain normalized abundances and not estimates of the individual effects, a cell means approach is used and the model becomes

abundance=itraqrun*channel + peptide(protein).

The easy way to view this is as a two anova with rows corresponding to each combination of itraqrun and channel and columns corresponding to each combination
of protein and peptide. The estimates of row effects are given by the cell means of

abundance-estimated(peptide(protein)

and the estimates of column effects are given by the cell means of

abundance-estimated(itraqrun*channel).

After iterating back and forth, the final normalized data is given by

abundance-estimated(itraqrun*channel)-estimated(peptide(protein)).

Parameters:

```
data=Name of the dataset that contains the data to be normalized

iter=Number of iterations. Typically 3 to 4 is enough but use smallset=1 and
more iterations to determine convergence.

abundance=log(raw abundance) ... the scale is up to the user. This is typically
the peak area or peak height of the peptide within a protein.

itraqrun=Variable in the dataset that indicates which run the experiment is from

channel=Variable in the dataset that indicates the channel

peptide=Variable in the dataset that identifies a peptide

protein=Variable in the dataset that identifies a protein

out=Name of the dataset where results are to be stored

smallset= Indicator macro variable whether you want a small set of
information returned (i.e.,=1 returns the input variables and the
normalized value) whereas 0 would return each step of the
iteration. The last option would be helpful in determining
the total number of iteractions to run.

*/


%macro itraqnorm(data=,iter=10,itraqrun=,channel=,
    peptide=,abundance=,protein=,out=normalized,
    smallset=1);

    %do i=1 %to &iter;

        %if &i=1 %then %do;

            /*Grab the overall mean*/

            proc means noprint data=&data;
                var &abundance;
                output out=omean mean=overall;
            run;

            data omean;
                set omean;
                call symput("omean",overall);
            run;

            /*Overall mean is the first estimate of yhat*/
            /*s0 is a iteration variable for peptide(protein)*/
            /*r0 is a iteration variable for itraqrun*channel*/

            data step&i;
                set &data;
                s0=0;
                r0=0;
                yhat0=&omean;
                resid1=&abundance;
```

```
    run;
    %end;

%else %do;

    data step&i;
        set step&i;
    run;
    %end;

proc means data=step&i noprint;
    class &itraqrun &channel;
    var resid&i;
    output out=steping mean=r&i;
run;

/*Cell means for itraqrun and channel*/

data steping;
    set steping;
    if &itraqrun=.|&channel=. then delete;
run;

proc sort data=steping;
    by &itraqrun &channel;
run;

proc sort data=step&i;
    by &itraqrun &channel;
run;

%let g=%eval(&i-1);

/*Merge in the cell means for itraqrun and channel*/
/*sresid is the residuals of itraqrun and channel and
    used to estimate peptide(protein) */

data step_s;
    merge step&i steping;
    by &itraqrun &channel;
    yhat&i=r&i+s&g;
    sresid=&abundance-r&i;
run;

proc means data=step_s noprint;
    class &peptide &protein;
    var sresid;
    output out=steping mean=s&i;
run;

/*Cell means for peptide(protein) */

data steping;
    set steping;
    if &peptide^=" " and &protein^=" ";
run;
```

```
        proc sort data=steping;
            by &peptide &protein;
        run;

        proc sort data=step_s;
            by &peptide &protein;
        run;

        %let k=%eval(&i+1);

        /*Merge in the cell means for peptide(protein) */
        /*resid&k becomes the residual of abundance -
estimated(peptide(protein)) */
        /*in the next iteration for itraqrun and channel estimates*/
        /*abs&i is the difference between the last iteration and current
iteration
            estimates of yhat */
        /*normalized is the normalized abundance estimate*/

        data step&k;
            merge step_s steping;
            by &peptide &protein;
            yhat&i=r&i+s&i;
            resid&k=&abundance-s&i;
            abs&i=abs(yhat&i-yhat&g);
            normalized=&abundance-yhat&i;
        run;

        /*Mopping up disk space*/

        proc datasets;
            delete step&g;
        run;

        %end;

    data &out;
        set step&i;

        %if &smallset=1 %then %do;
        keep &itraqrun &channel &peptide &protein &abundance normalized;
        %end;

    run;

    proc datasets;
        delete steping step_s step&i omean;
    run;

    %mend;
```