Melanophore RNA-seq analysis post mapping/normalization

Table of Contents

Lauren Saunders and Meredith Bache-Wiig March 14, 2016

load the dataset

```
load melRNAseq.mat;
```

assign groups for ablated (-TH) and unablated (+TH) (our 2 conditions)

```
ablated=melFPKM(1:6,:);
unablated=melFPKM(7:12,:);

% means with zeros
mA = mean(ablated);
mUA = mean(unablated);

zerMeanA = mA;
zerMeanUA = mUA;

plusMeanA = zerMeanA + 1;
plusMeanUA = zerMeanUA + 1;

% estimate pseudo-reference with geometric mean row by row
% melFPKMalt1=melFPKM';
% pseudoRefSample = geomean(melFPKMalt1,2);
% nz = pseudoRefSample > 0;
% ratios = bsxfun(@rdivide,melFPKMalt1(nz,:),pseudoRefSample(nz));
% sizeFactors = median(ratios,1);
```

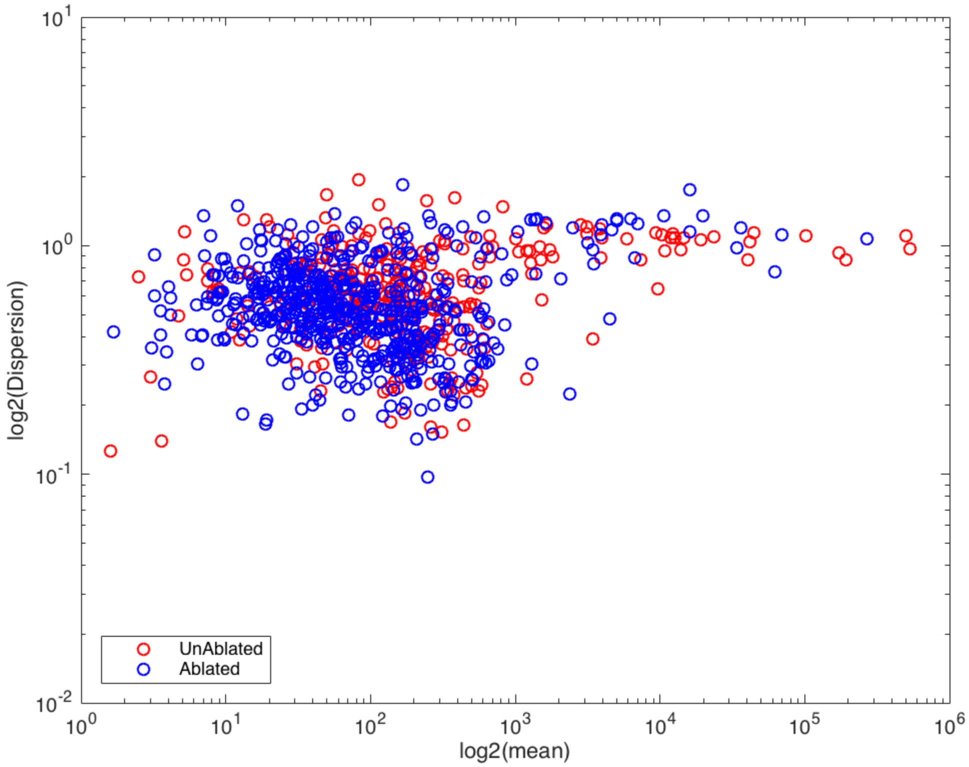
Removing FPKM values below 1

```
melFPKMalt=melFPKM;
for i=1:38125;
```

```
    for j=1:12;
        if melFPKMalt(j,i)<1;
            melFPKMalt(j,i)=NaN;
        end;
    end;
end;
```

To plot the mean FPKM values between the two groups

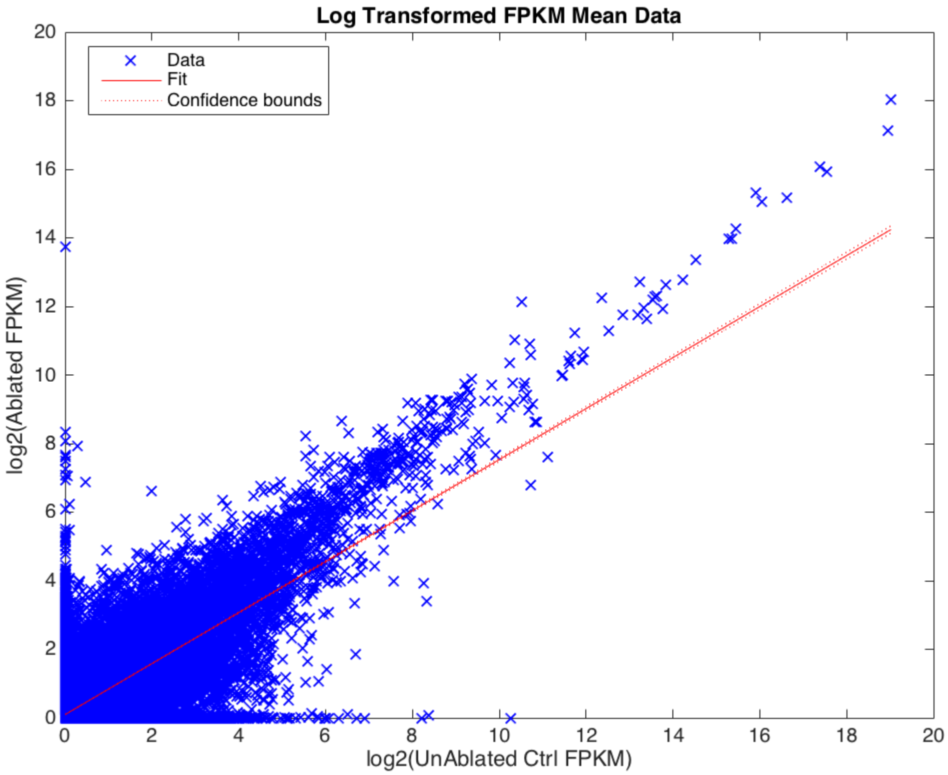
```
mF4=melFPKM';
mF2 = melFPKMalt'; % it didn't work with the non-transposed version

% find mean (with transpose)
meanUAnew = mean(mF2(:, 7:12),2);
meanAnew = mean(mF2(:, 1:6),2);

meanUA = mean(mF4(:, 7:12),2);
meanA = mean(mF4(:, 1:6),2);

% convert zeros to NaN
% meanAnew(meanA ==0) = NaN;
% meanUAnew(meanUA ==0) = NaN;

% find dispersion (with transpose)
dispUA = std(mF2(:,7:12),0,2) ./ meanUAnew;
dispA = std(mF2(:,1:6),0,2) ./ meanAnew;

% plot on a log-log scale (with transpose)
figure;
loglog(meanUAnew, dispUA, 'or');
hold on;
loglog(meanAnew, dispA, 'ob');
xlabel('log2(mean)');
ylabel('log2(Dispersion)');
legend('UnAblated', 'Ablated', 'Location', 'southwest');

% convert zeros to NaN
meanA(meanA ==0) = NaN;
meanUA(meanUA ==0) = NaN;

% linear regression model
mdl=fitlm(meanUA,meanA);
```

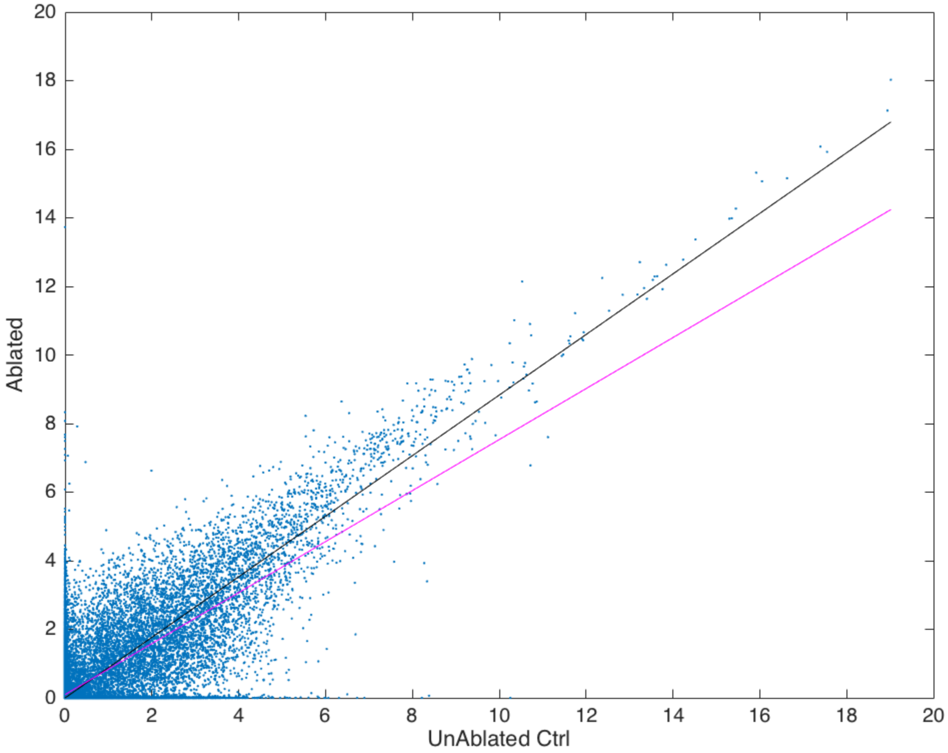


try this plot (FPKM with pseudocount)

```
[p,S] = polyfit(log2(plusMeanUA), log2(plusMeanA), 1);
[y, delta] = polyval(p,log2(plusMeanUA),S);

% b1 = log2(plusMeanUA)/log2(plusMeanA);
% yCalc1 = b1 * log2(plusMeanUA);

x = log2(plusMeanUA);
y2 = log2(plusMeanA);

mdl = fitlm(x, y2, 'Linear');
lmCI = coefCI(mdl);

figure;
plot(mdl);
hold on;
% plot(x,yCalc1, 'g-');
xlabel('log2(UnAblated Ctrl FPKM)');
ylabel('log2(Ablated FPKM)');
title('Log Transformed FPKM Mean Data');

% scatter plot of means
figure;
plot(x, y2, '.');
```

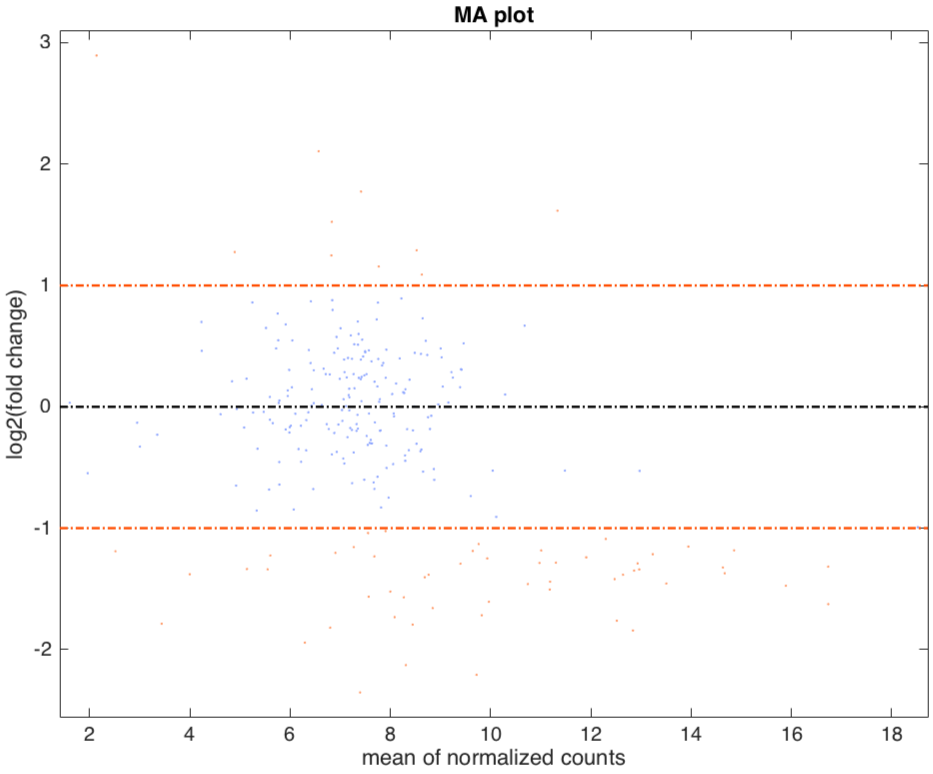
```
hold on;
plot(x,yCalc1, 'k-');
hold on;
plot(x, y, 'm-');
%hold on;
%plot(x, y-delta, 'g', x, y+delta, 'c');
xlabel('UnAblated Ctrl');
ylabel('Ablated');
```





Fold Change

```
% compute the mean and the log2FoldChange
meanBase = (meanUAnew + meanAnew) / 2;
foldChange = meanAnew ./ meanUAnew;
log2FC = log2(foldChange);

% plot mean vs. fold change (MA plot)
mairplot(meanAnew, meanUAnew,'Type','MA','Plotonly',true);
set(get(gca,'Xlabel'),'String','mean of normalized counts')
set(get(gca,'Ylabel'),'String','log2(fold change)')
```



Published with MATLAB® R2015b