Read in and arrange the data.

```
clear;
rng('default');
cd('/Users/sbiswas/GitHub/pqe/data');

d =
 dataset('file', 'GSE87069_E10.5-13.5_SingleCells_rpkms.tab', 'ReadVarNames',
 true, 'ReadObsNames', true);
genes = get(d, 'ObsNames');
samples = get(d, 'VarNames')';

y = double(d); % log2(RPKM + 0.001)
yexp = 2.^y - 0.001; yexp(yexp < 1e-8) = 0; % RPKM. Convert numerical
 zeros to zeros.
mean_exp = mean(y,2);
std_exp = std(y,0,2);
GFP = y(1,:);
GFPexp = yexp(1,:);

% Read in sample metadata. This mainly contains embryonic stage
% information.
md = dataset('file', 'GSE87069_series_matrix.txt', 'ReadObsNames',
 false, 'ReadVarNames', true);
mdc = dataset2cell(md)';
md_samples = mdc(2:end,1);
E_stage = strrep(mdc(2:end,7), 'developmental stage: ', '');

% Order the metadata so it's congruent with the expression data.
[~,~,ib] = intersect(samples, md_samples, 'stable');
md_samples = md_samples(ib);
E_stage = E_stage(ib);

% The relevant variables going forward are:
% y -> log2(RPKM + 0.001)
% yexp -> RPKM
% GFP -> log2(RPKM + 0.001) of EGFP transgene
% GFPexp -> RPKM of EGFP transgene
% E_stage -> embryonic stage of each cell
% samples -> Cell IDs
% genes -> Gene IDs
```

Extract an informative gene-set.

```
% The authors omit that the expression data is log transformed RPKM.
 In
% examining this data many of the expression values are -9.9658. This
 is
% equal to log2(0.001). Thus we assume that expression values in this
% dataset are log2(RPKM + 0.001).

% Let's get an idea of the dropout rate
```

```
dr = sum(sum(yexp == 0))/numel(yexp);
fprintf('Dropout rate: %0.2f%%\n', 100*dr)

% I didn't think the use of the bulk samples was a good way to define
% interesting genes since bulk averages can mask important
 variability.
% Thus, let's extract a gene set directly from the single-cell data
 they've
% obtained.

% Let's first consider removing genes that are completely zero in at
 least
% 95% of the samples.
PCT_CUTOFF = 0.95;
mask = sum(yexp == 0,2) > floor(PCT_CUTOFF*size(y,2)); % genes to
 remove

% If however, among the remaining top 5% of abundant samples there is
% strong induction of expression, then maybe this gene has some signal
 to
% it. Therefore we should keep it.
check_further = find(mask);
for i = 1 : length(check_further)
    yrem = yexp(check_further(i),:);
    yrs = sort(yrem, 'descend');

    p10 = prctile(yrs(1:round((1-PCT_CUTOFF)*size(y,2))), 10);

    if p10 > 0.01
        mask(check_further(i)) = false;
    end

end

fprintf('Number of genes removed by expression filter: %0.0f\n',
 sum(mask));
y(mask,:) = [];
yexp(mask,:) = [];
genes(mask,:) = [];



% Now let's build a linear model for every gene. We will regress
 log(RPKM +
% 0.001) onto EGFP signal and three categorical variables, each of
 which
% are indicator variables of E11.5, E12.5, or E13.5. If any of the
 four
% regression coefficients are significnat, even before correcting for
% multiple testing, we will keep the gene. Effectively this model is
% testing to see if a gene has differential expression wrt EGFP+ vs
 EGFP-
% cells or if it is DE wrt at least one embryonic stage.
%
```

```
% Note that by virtue of treating embryonic stage as a categorical
% variable, this model is not making the assumption that expression is
% linear in time. Expression just needs to be different in that stage
 vs
% all other stages, which is a much more relaxed criterion and should
 be
% able to capture events in which a gene's expression goes up and then
 back
% down again.
X = [GFP; strcmpi(E_stage, 'E11.5')'; strcmpi(E_stage, 'E12.5')'; ...
    strcmpi(E_stage, 'E13.5')']'; % design matrix
Xs = standardize(X); % standardize design matrix
Ys = standardize(y'); % Standardize responses

disp('Design matrix has minor multi-colinearity');
disp(corr(Xs));


pvals = zeros(size(Ys,2),1);
for i = 1 : length(pvals)
    stats = regstats(Ys(:,i), Xs);
    pvals(i) = stats.fstat.pval;

    if mod(i,1000) == 0
        disp(i);
    end
end

tokeep = pvals < 0.01;


y(~tokeep,:) = [];
yexp(~tokeep,:) = [];
genes(~tokeep,:) = [];
pvals(~tokeep,:) = [];


% Export the filtered dataset.
% Do not include the EGFP transgene from the expression matrix as this
 will
% introduce artificial signal into the dataset during downstream
 modeling.
design = cell2dataset([num2cell(GFP'), E_stage], 'ObsNames', ...
    samples, 'VarNames', {'EGFP', 'EStage'});

dout = mat2dataset(y(2:end,:), 'VarNames', samples, 'ObsNames',
 genes(2:end));

export(dout, 'file', 'expression_filtered_and_DE_genes_expression_mat.txt');
export(design, 'file', 'expression_filtered_and_DE_genes_design_mat.txt');

Dropout rate: 78.97%
Number of genes removed by expression filter: 10919
Design matrix has minor multi-colinearity
```

```
 1.0000   -0.0862   -0.1702   -0.1536
-0.0862    1.0000   -0.3060   -0.2647
-0.1702   -0.3060    1.0000   -0.4139
-0.1536   -0.2647   -0.4139    1.0000

     1000

     2000

     3000

     4000

     5000

     6000

     7000

     8000

     9000

    10000

    11000

    12000

    13000

    14000
```

Published with MATLAB® R2015b