Multivariate Data and Singular Value Decomposition

Dataset: Hi-C data obtained from human colon cells Comparing Healthy vs cells with Chr 7 trisomy

Author: Koki Sasagawa

Date: 2/11/19

Detailed explanation about Hi-C and biological concepts added to provide additional context

Useful context on HI-C data references from the following paper: "Three dimensional folding of chromosomes compartmentalizes the genome and can bring distant functional elements such as promoters and enhancers into close spatial proximity" - Berkum et al. 2010

How the data is generated: 1. Cells are fixed in vivo (take place in live cells) with chemicals such as formaldehyde (DNA-protein cross linkage) 2. Lyse (cut) the DNA using a restriction enzyme. Cross linked loci remain linked (Where the spatial location of chromosomes were touching) 3. Biotin incorporated as 5' overhangs (when double stranded DNA is cut, one strand usually ends up being longer, in this case the 5' end) are filled in (biotin is a biomolecular marker used to tag certain sequences and extract later with streptavidin which has a natural affinity to biotin) 4. Blunt-end ligation 5. Purify and shear DNA, pulldown biotin 6. Sequence paired-ends 7. Chromatin interactions are visualized as a heat map, where x and y axis represent loci in genomic order, and each pixel representing the number of observed interactions between. Each pixel represents interactions between two locus. Intensity corresponds to the number of reads (0-200 reads). The stronger the signal, the interaction occurred in a large fraction of the population. A weaker signal indicates the interaction occurred in a much smaller population.

Contents

setup

```
clear
close all
```

Load data

1. hcecHic (N x N matrix containing HCEC Hi-C sample, obs / exp normalized) 2. hcec7Hic (N x N matric containing HCEC+7 Hi-C sample, obs / exp normalized) 3. chrSart (23 x 1 vector containing starting locations of chromosomes in Hi-C matrices (e.g., chr 4 start at index 669)

```
load('hcec7hic1Mb.mat')
```

1. Propose a method that can show the inequality between the two matrices using SVD

We can compare the singularvalues of the two matrices which is obtained from computing SVD. Plot the singularvalues for the two matrices on a scatter plot.

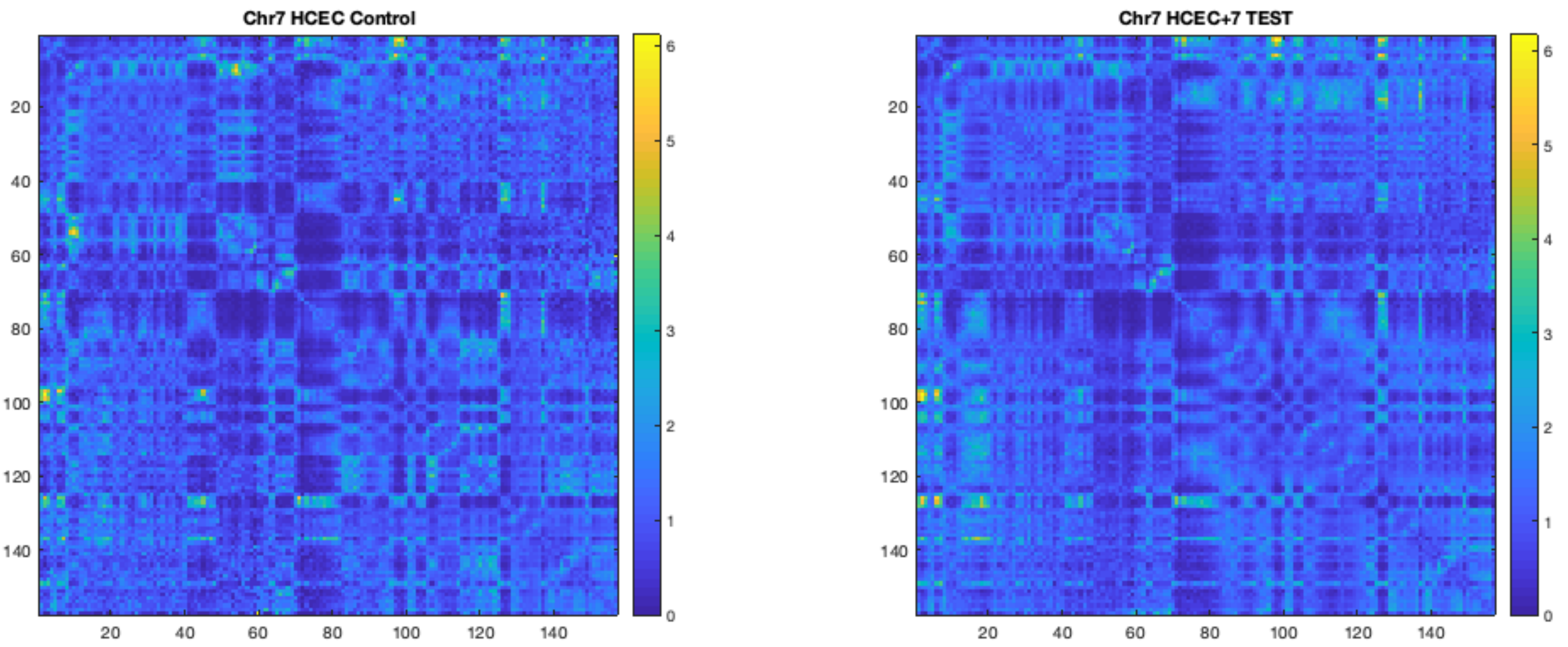
Calculate the rank 1 approximation of matrices and compare the control with test.

Compute the first principal component of each matrix, then subtract the two PC1 vectors and see where they are most different.

2. Compute the difference between chromosome 7 contact maps in the two data sets

```
% First, plot the Hi-C data of chromosome 7
fig(1) = figure(1);
set(fig(1), 'unit', 'normalized', 'Position', [.15 .15 .7 .4])

for k = 1:2
    ax(k) = subplot(1,2,k);
end

% select chr7 from normal cell data
chrSelect = 7; % select chr 7
chrSelectIdx = chrStart(chrSelect): chrStart(chrSelect+1)-1; % get the corresponding matrix indexes
hcecHicChr7 = hcecHic(chrSelectIdx, chrSelectIdx);

subplot(ax(1));
imagesc(hcecHicChr7)
title('Chr7 HCEC Control')
colorbar
```

```
axis('square')

% select chr7 from test cell data
hcec7HicChr7 = hcec7Hic(chrSelectIdx, chrSelectIdx);

subplot(ax(2));
imagesc(hcec7HicChr7)
title('Chr7 HCEC+7 TEST')
colorbar
axis('square')
```



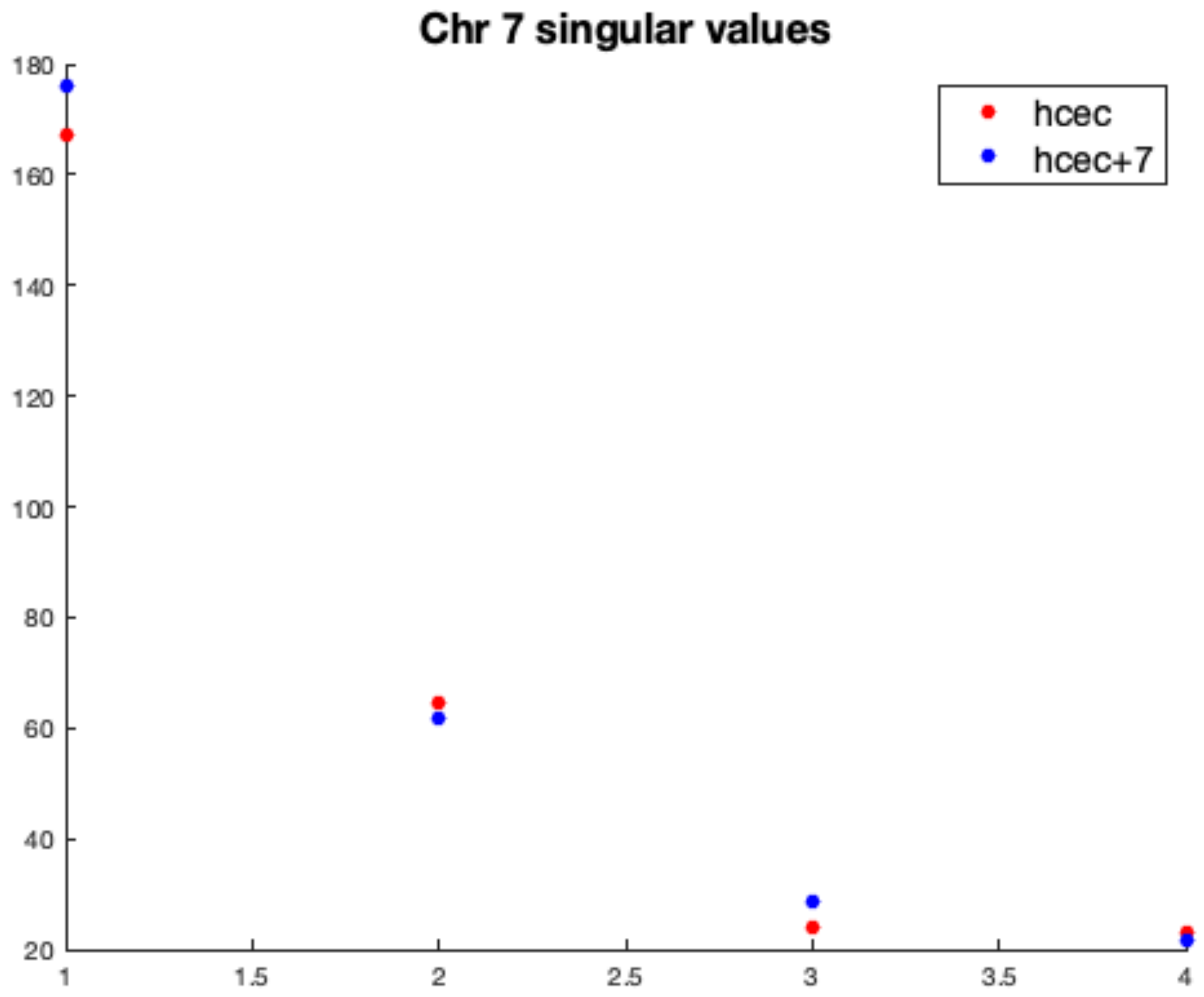
Compute SVD and plot singular values

The '0' argument produces a economy-size decomposition by m-by-n matrix A 'economy size' decomposition removes extra rows or columns for zeros from the diagonal matrix of singular values, S, along with columns in either U or V that multiply those zeros in the expression A = U * S * V

```
% singular values for control
[U1,S1,V1] = svd(hcecHicChr7,0);
sigma1 = diag(S1);

% singular values for test
[U2,S2,V2] = svd(hcec7HicChr7,0);
sigma2 = diag(S2);

% scatterplot of top 4 largest singular values
fig(2) = figure(2);
n = 1:4

p1 = scatter(n, sigma1(n), 'r', 'filled')
hold on % overlay second scatter plot on same figure
p2 = scatter(n, sigma2(n), 'b', 'filled')
hold off

legend([p1, p2], 'hcec','hcec+7', 'FontSize', 14)
title('Chr 7 singular values', 'FontSize', 16)
```

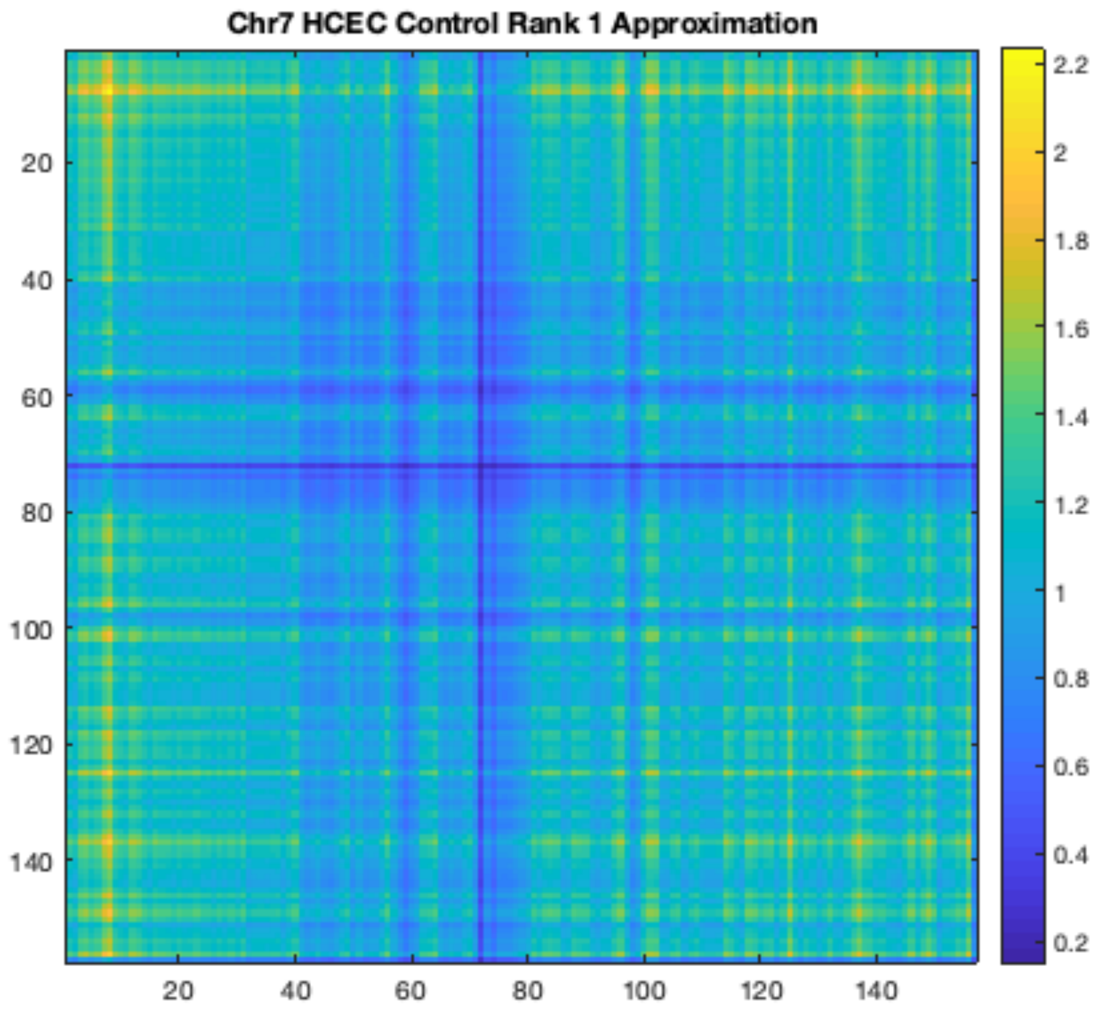
```
n =

     1     2     3     4


p1 = 

  Scatter with properties:

              Marker: 'o'
     MarkerEdgeColor: 'none'
     MarkerFaceColor: 'flat'
            SizeData: 36
           LineWidth: 0.5000
               XData: [1 2 3 4]
               YData: [167.3410 64.4796 23.8934 23.1735]
               ZData: [1×0 double]
               CData: [1 0 0]

  Use GET to show all properties


p2 = 

  Scatter with properties:
```

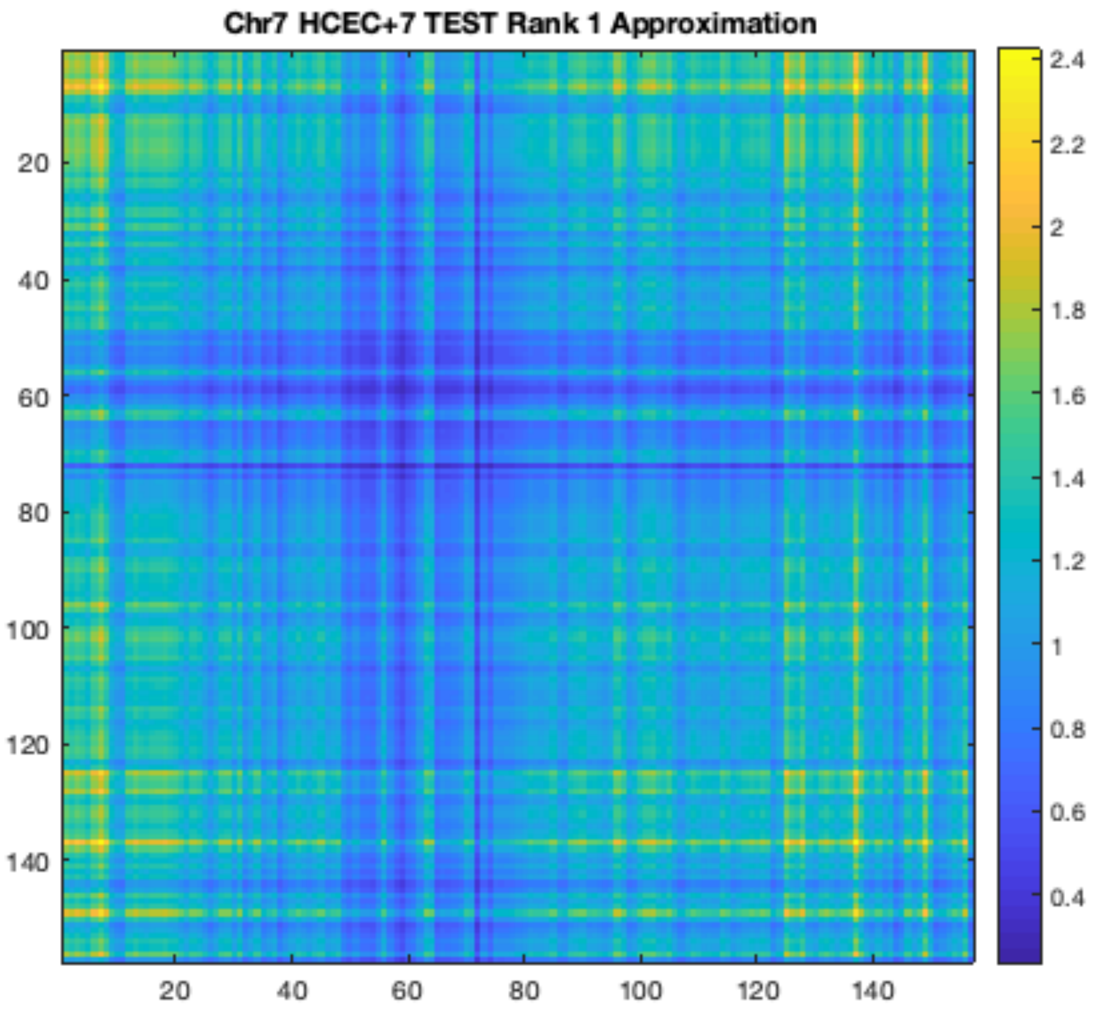
```
              Marker: 'o'
     MarkerEdgeColor: 'none'
     MarkerFaceColor: 'flat'
            SizeData: 36
           LineWidth: 0.5000
               XData: [1 2 3 4]
               YData: [175.8402 61.7937 28.8231 21.7912]
               ZData: [1×0 double]
               CData: [0 0 1]

  Use GET to show all properties
```



Compare Rank1 approximation

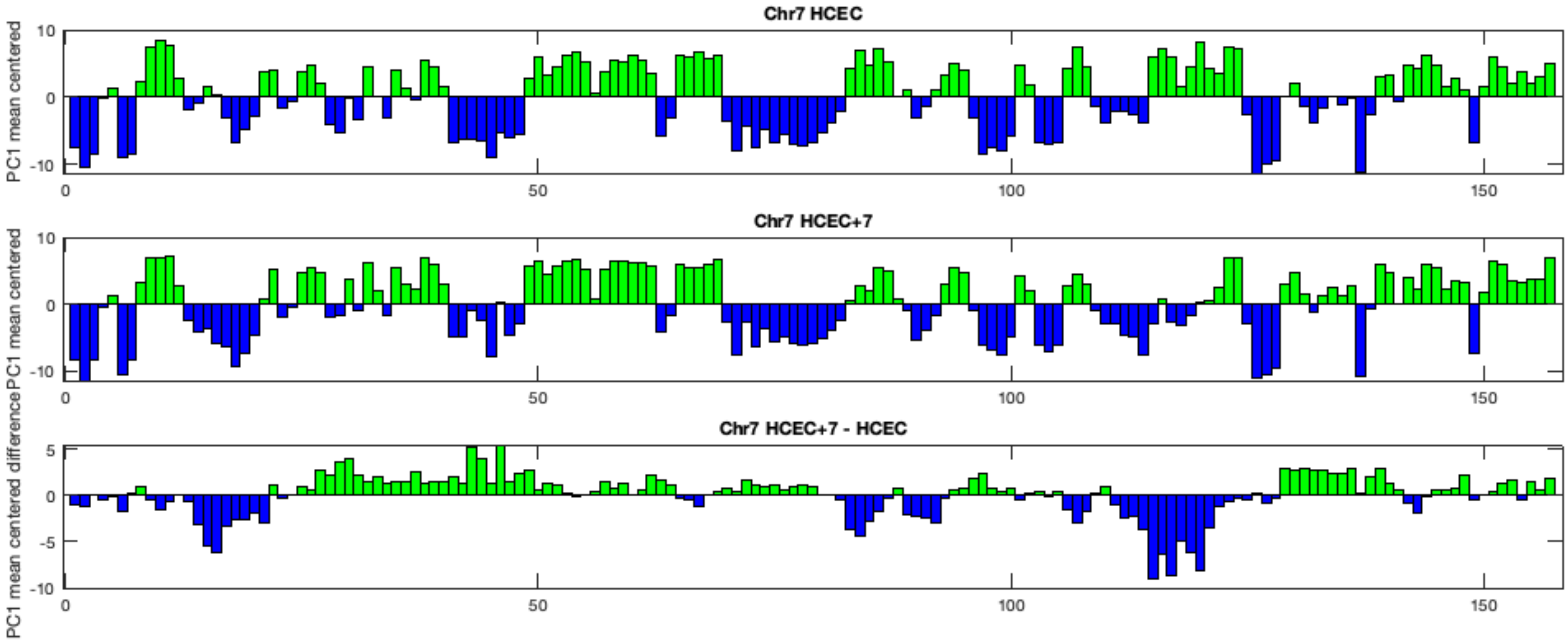
```
fig(2) = figure(2);
set(fig(2), 'unit', 'normalized', 'Position', [.15 .15 .7 .4])

for k = 1:2
    ax(k) = subplot(1,2,k);
end

% Control
subplot(ax(1));
A1 = sigma1(1)*U1(:,1)*V1(:,1)';
imagesc(A1)
title('Chr7 HCEC Control Rank 1 Approximation')
colorbar
axis('square')

% Test
subplot(ax(2));
A2 = sigma2(1)*U2(:,1)*V2(:,1)';
imagesc(A2)
title('Chr7 HCEC+7 TEST Rank 1 Approximation')
colorbar
axis('square')
```





Compare difference between first principle component

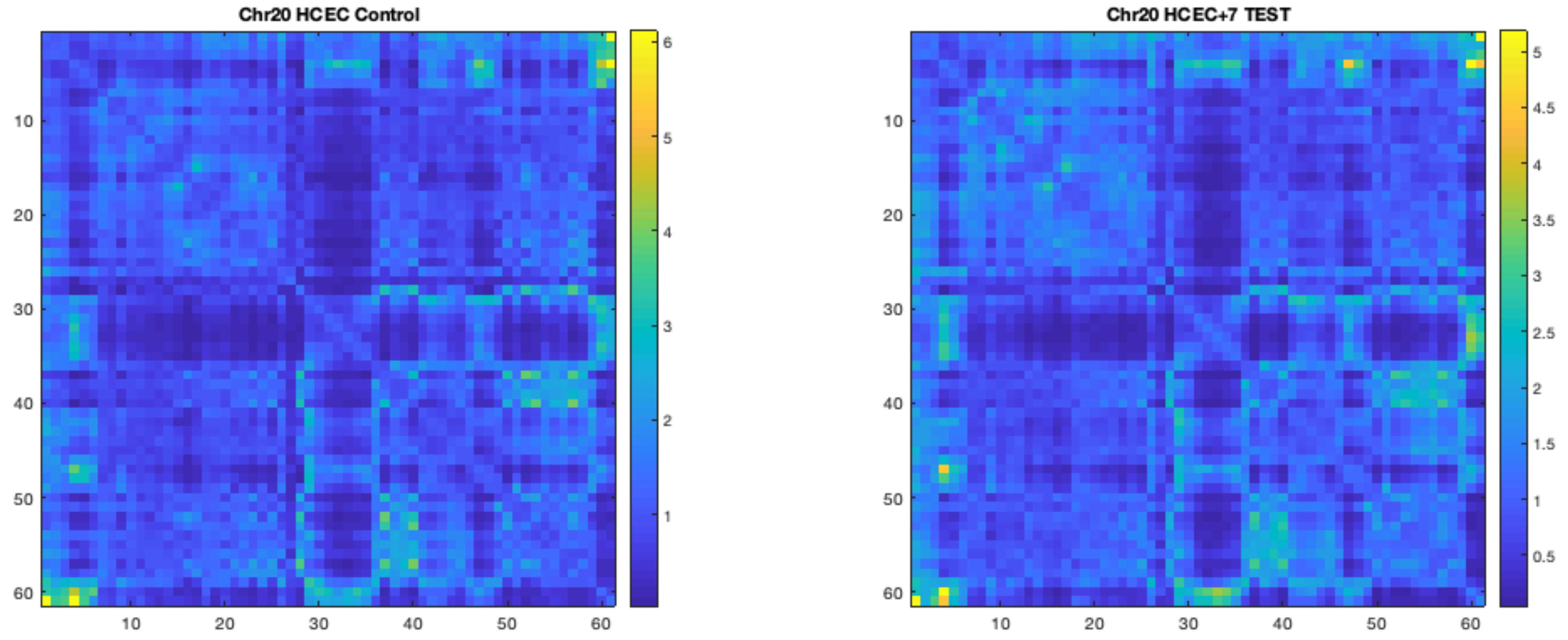
```
% PC1 of control
% mean center (on rows in this case since we want low dimensional)
center = hcecHicChr7 - mean(hcecHicChr7,2);
[U,S,V] = svd(center','econ');
svdScores = U * S;
PC1_control = svdScores(:,1);
x_values = 1:length(PC1_control);
pos_index_control = PC1_control >= 0;
neg_index_control = PC1_control < 0;

% PC1 of test
center = hcec7HicChr7 - mean(hcec7HicChr7,2);
[U,S,V] = svd(center','econ');
svdScores = U * S;
PC1_test = svdScores(:,1);
pos_index_test = PC1_test >= 0;
neg_index_test = PC1_test < 0;

% difference in PC1
PC1_difference = PC1_test - PC1_control;
pos_index_difference = PC1_difference >= 0;
neg_index_difference = PC1_difference < 0;

fig(3) = figure(3);
set(fig(3), 'unit', 'normalized', 'Position', [.15 .15 .7 .4])

for k = 1:3
    ax(k) = subplot(3,1,k);
end

% make barplot
subplot(ax(1));
bar(x_values(pos_index_control), PC1_control(pos_index_control), 'g', 'BarWidth', 1)
hold on
bar(x_values(neg_index_control), PC1_control(neg_index_control), 'b', 'BarWidth', 1)
ylabel('PC1 mean centered')
title('Chr7 HCEC')

subplot(ax(2));
bar(x_values(pos_index_test), PC1_test(pos_index_test), 'g', 'BarWidth', 1)
hold on
bar(x_values(neg_index_test), PC1_test(neg_index_test), 'b', 'BarWidth', 1)
ylabel('PC1 mean centered')
title('Chr7 HCEC+7')

subplot(ax(3));
bar(x_values(pos_index_difference), PC1_difference(pos_index_difference), 'g', 'BarWidth', 1)
hold on
bar(x_values(neg_index_difference), PC1_difference(neg_index_difference), 'b', 'BarWidth', 1)
ylabel('PC1 mean centered difference')
title('Chr7 HCEC+7 - HCEC')
```



3. Same computation for two other chromosomes of your choice

Chromosome of interest: 20 Knockdown of TP53 and expression of K-RasV12 in +7 HCECs resulted in the emergence of trisomy 20, another nonrandom aneuploidy observed in 85% of Colorectal Cancer - https://www.sciencedirect.com/science/article/pii/S1476558611800163

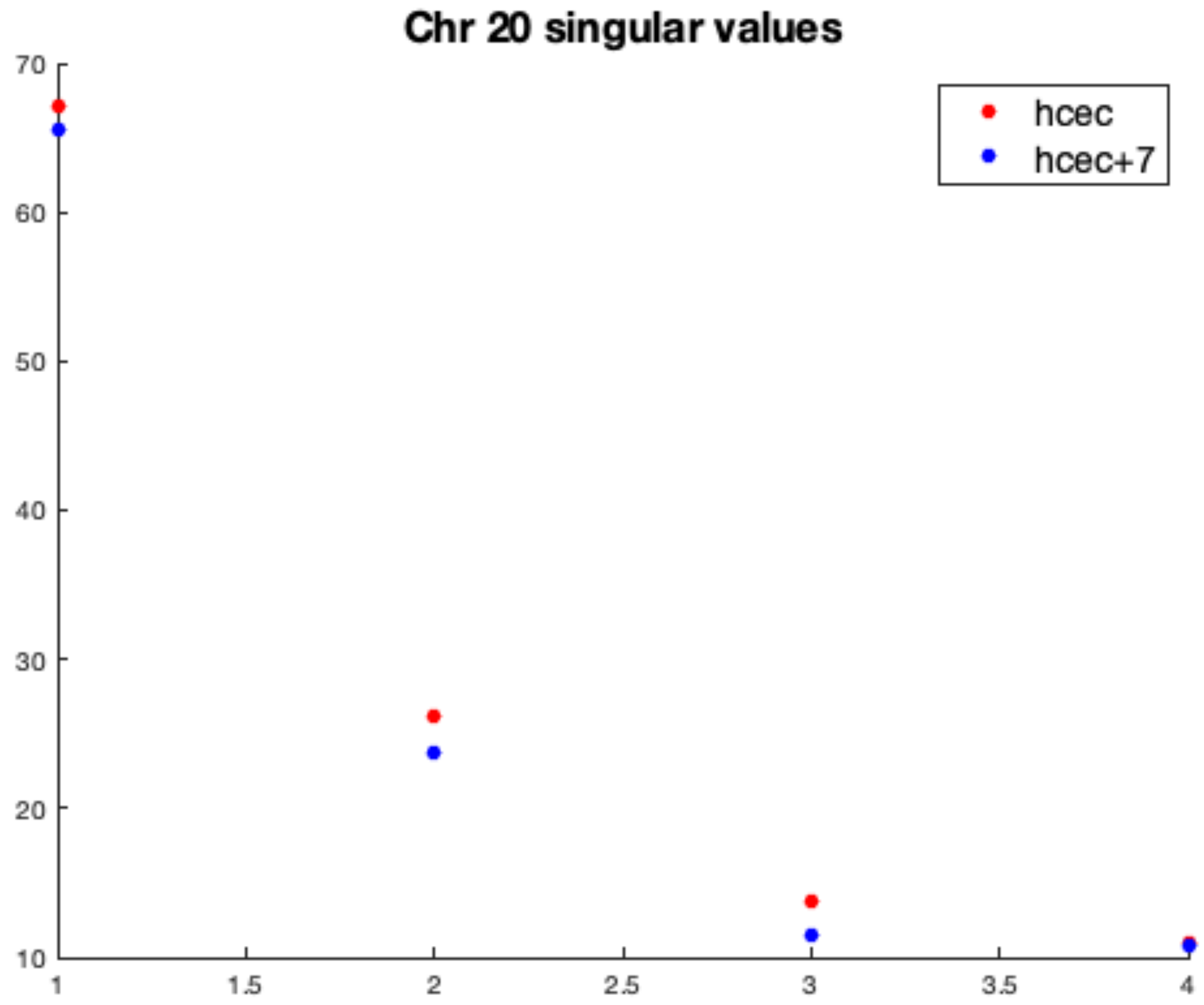
```
% TP53 - provide instructions to create tumor protein p53 which acts as
% a tumor suppressor - regulate cell division.
% K-RasV12 - Group of protein callled K-Ras part of pathway known as the
% RAS/MAPK pathway - proteins that carry signals from outside of the cell
% to the cell nucleus. Turned on by GTP and off by GDP.

fig(4) = figure(4);
set(fig(4), 'unit', 'normalized', 'Position', [.15 .15 .7 .4])

for k = 1:2
    ax(k) = subplot(1,2,k);
end

% Normal cell data
chrSelect = 20;
chrSelectIdx = chrStart(chrSelect): chrStart(chrSelect+1)-1; % get the corresponding matrix indexes
hcecHicChr20 = hcecHic(chrSelectIdx, chrSelectIdx);

% Test cell data
hcec7HicChr20 = hcec7Hic(chrSelectIdx, chrSelectIdx);

subplot(ax(1));
imagesc(hcecHicChr20)
title('Chr20 HCEC Control')
colorbar
axis('square')

subplot(ax(2));
imagesc(hcec7HicChr20)
title('Chr20 HCEC+7 TEST')
colorbar
axis('square')
```



Compute SVD and plot singular values

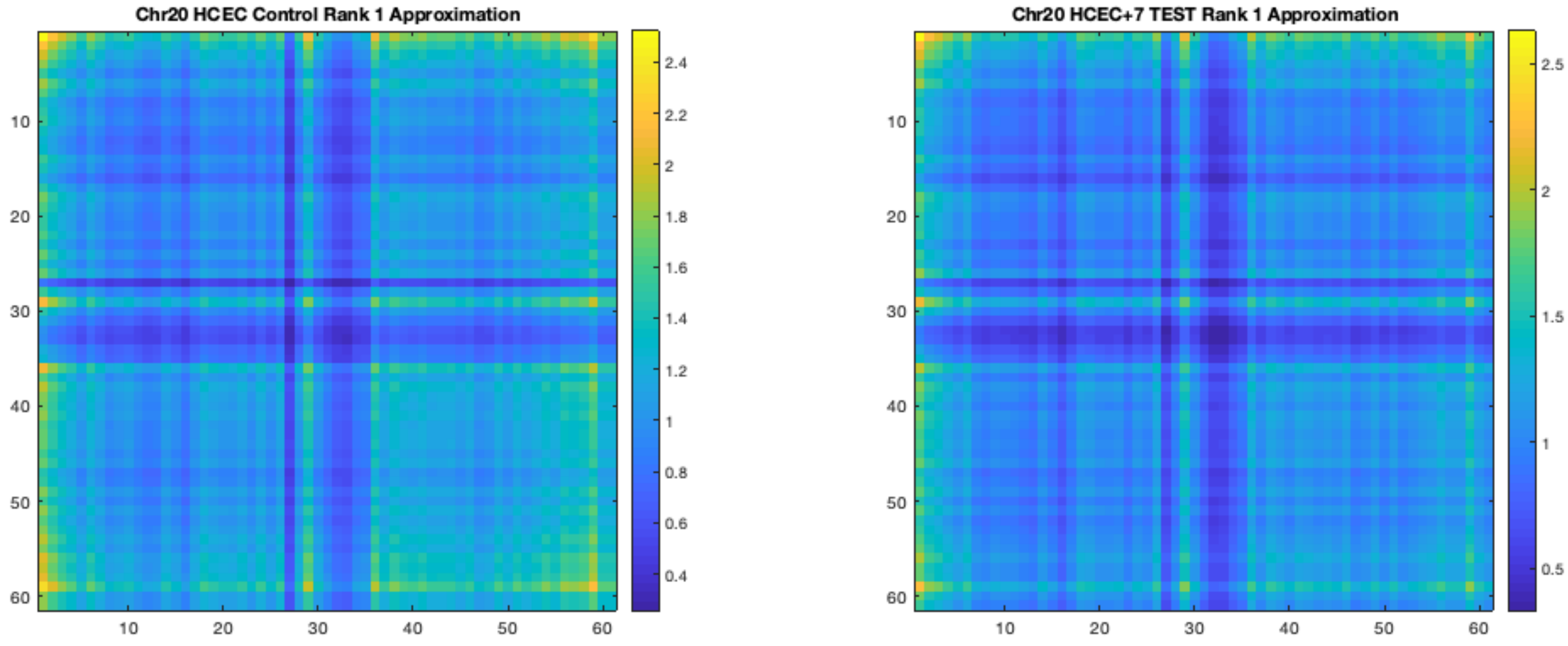
```
% singular values for control
[U1,S1,V1] = svd(hcecHicChr20,0);
sigma1 = diag(S1);

% singular values for test
[U2,S2,V2] = svd(hcec7HicChr20,0);
sigma2 = diag(S2);

% scatterplot of top 4 largest singular values
fig(5) = figure(5);
n = 1:4

p1 = scatter(n, sigma1(n), 'r', 'filled')
hold on % overlay second scatter plot on same figure
p2 = scatter(n, sigma2(n), 'b', 'filled')
hold off

legend([p1, p2], 'hcec','hcec+7', 'FontSize', 14)
title('Chr 20 singular values', 'FontSize', 16)
```

```
n =

     1     2     3     4


p1 =

  Scatter with properties:

             Marker: 'o'
    MarkerEdgeColor: 'none'
    MarkerFaceColor: 'flat'
           SizeData: 36
          LineWidth: 0.5000
              XData: [1 2 3 4]
              YData: [67.0326 26.1539 13.7053 10.8894]
              ZData: [1×0 double]
              CData: [1 0 0]

  Use GET to show all properties


p2 =

  Scatter with properties:

             Marker: 'o'
    MarkerEdgeColor: 'none'
    MarkerFaceColor: 'flat'
           SizeData: 36
          LineWidth: 0.5000
              XData: [1 2 3 4]
              YData: [65.4634 23.6572 11.5294 10.8037]
              ZData: [1×0 double]
              CData: [0 0 1]

  Use GET to show all properties
```



Compare Rank1 approximation

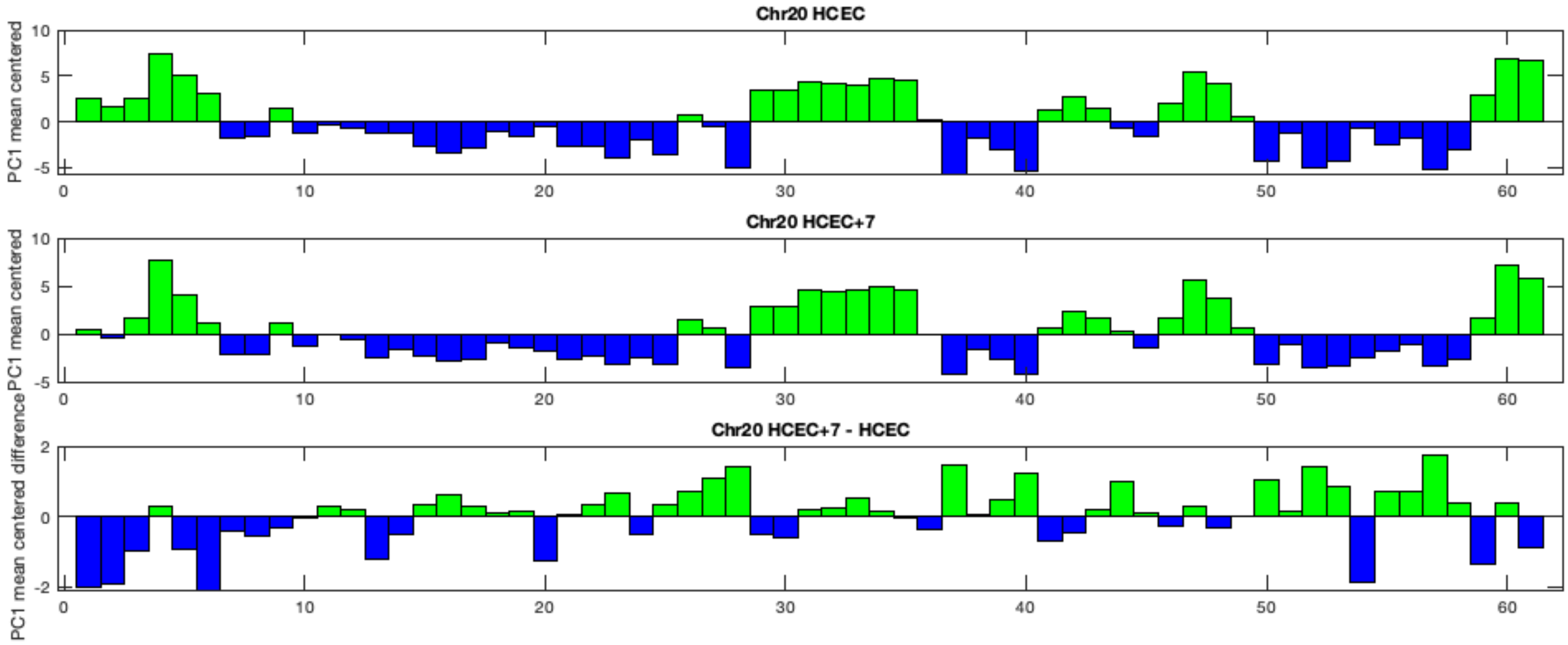
```
fig(6) = figure(6);
set(fig(6), 'unit', 'normalized', 'Position', [.15 .15 .7 .4])

for k = 1:2
    ax(k) = subplot(1,2,k);
end

% Control
subplot(ax(1));
A1 = sigma1(1)*U1(:,1)*V1(:,1)';
imagesc(A1)
title('Chr20 HCEC Control Rank 1 Approximation')
colorbar
axis('square')

% Test
subplot(ax(2));
A2 = sigma2(1)*U2(:,1)*V2(:,1)';
imagesc(A2)
title('Chr20 HCEC+7 TEST Rank 1 Approximation')
colorbar
axis('square')
```



Compare difference between first principle component

```
% PC1 of control
% mean center (on rows in this case since we want low dimensional)
center = hcecHicChr20 - mean(hcecHicChr20,2);
[U,S,V] = svd(center','econ');
svdScores = U * S;
PC1_control = svdScores(:,1);
x_values = 1:length(PC1_control);
```

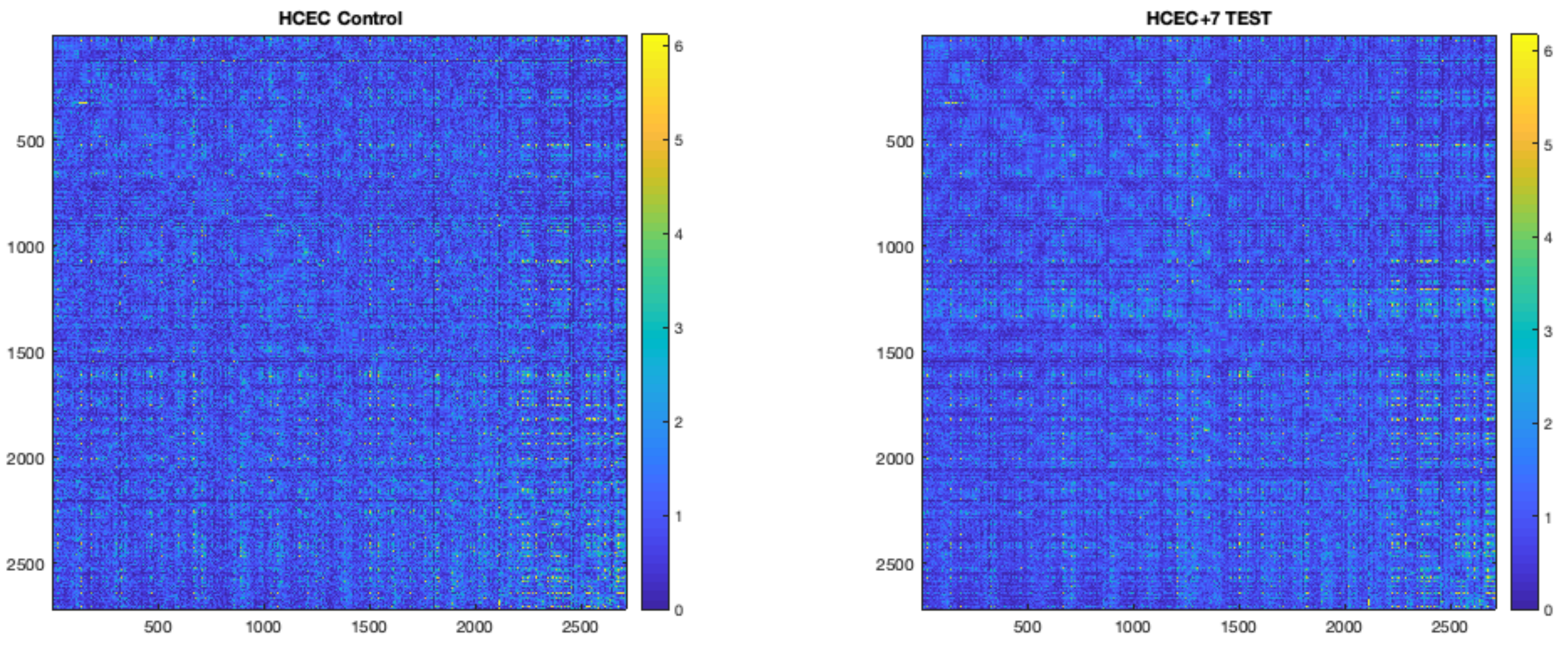
```
pos_index_control = PC1_control >= 0;
neg_index_control = PC1_control < 0;

% PC1 of test
center = hcec7HicChr20 - mean(hcec7HicChr20,2);
[U,S,V] = svd(center','econ');
svdScores = U * S;
PC1_test = svdScores(:,1);
pos_index_test = PC1_test >= 0;
neg_index_test = PC1_test < 0;

% difference in PC1
PC1_difference = PC1_test - PC1_control;
pos_index_difference = PC1_difference >= 0;
neg_index_difference = PC1_difference < 0;

fig(7) = figure(7);
set(fig(7), 'unit', 'normalized', 'Position', [.15 .15 .7 .4])

for k = 1:3
    ax(k) = subplot(3,1,k);
end

% make barplot
subplot(ax(1));
bar(x_values(pos_index_control), PC1_control(pos_index_control), 'g', 'BarWidth', 1)
hold on
bar(x_values(neg_index_control), PC1_control(neg_index_control), 'b', 'BarWidth', 1)
ylabel('PC1 mean centered')
title('Chr20 HCEC')

subplot(ax(2));
bar(x_values(pos_index_test), PC1_test(pos_index_test), 'g', 'BarWidth', 1)
hold on
bar(x_values(neg_index_test), PC1_test(neg_index_test), 'b', 'BarWidth', 1)
ylabel('PC1 mean centered')
title('Chr20 HCEC+7')

subplot(ax(3));
bar(x_values(pos_index_difference), PC1_difference(pos_index_difference), 'g', 'BarWidth', 1)
hold on
bar(x_values(neg_index_difference), PC1_difference(neg_index_difference), 'b', 'BarWidth', 1)
ylabel('PC1 mean centered difference')
title('Chr20 HCEC+7 - HCEC')
```



4. Compute the difference between the two data sets for the entire genome.

```
fig(8) = figure(8);
set(fig(8), 'unit', 'normalized', 'Position', [.15 .15 .7 .4])
for k = 1:2
    ax(k) = subplot(1,2,k);
end

subplot(ax(1));
imagesc(hcecHic)
title('HCEC Control')
colorbar
axis('square')

subplot(ax(2));
imagesc(hcec7Hic)
title('HCEC+7 TEST')
```

```
colorbar
axis('square')
```

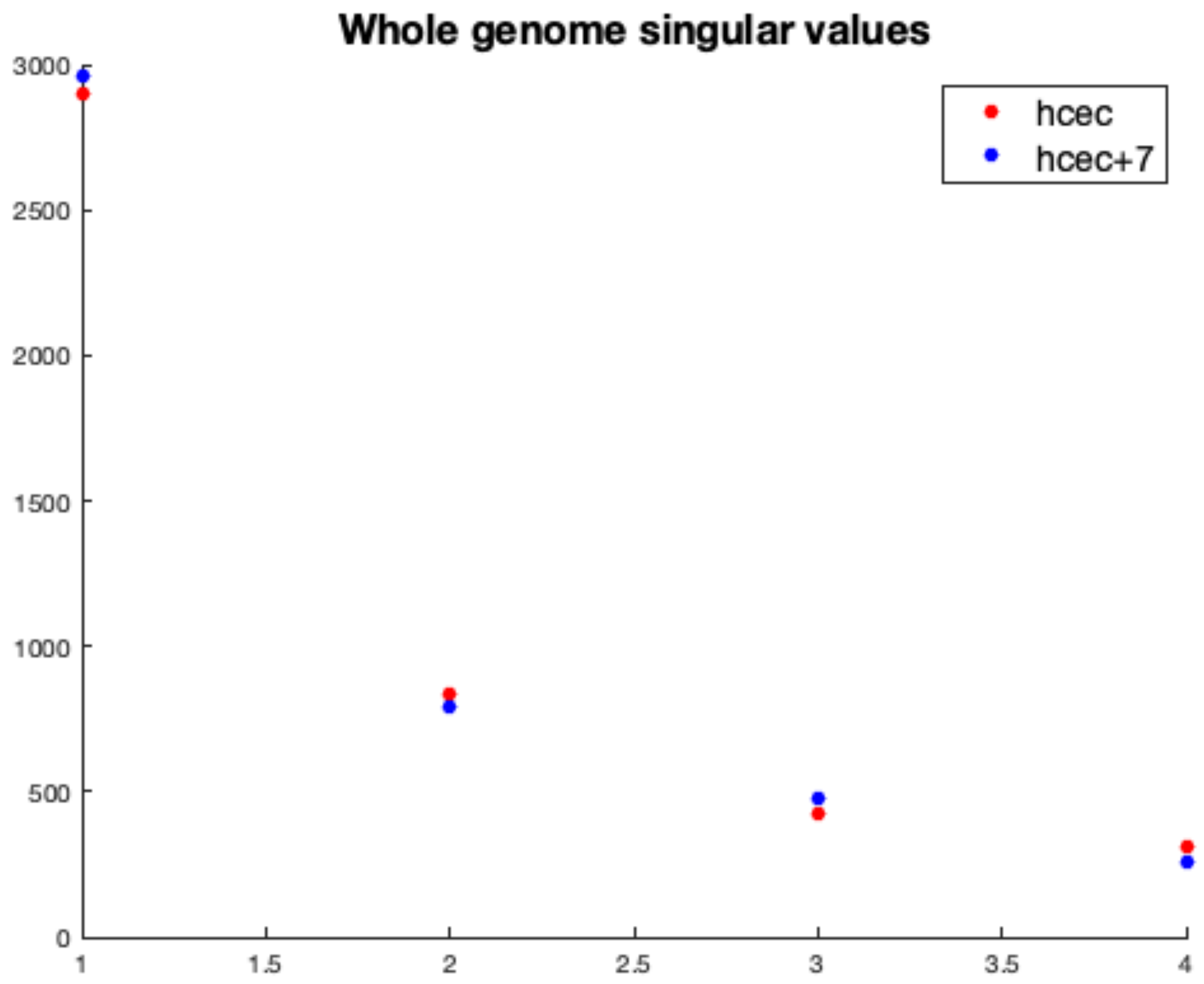


Compute SVD and plot singular values

```
% singular values for control
[U1,S1,V1] = svd(hcecHic,0);
sigma1 = diag(S1);

% singular values for test
[U2,S2,V2] = svd(hcec7Hic,0);
sigma2 = diag(S2);

% scatterplot of top 4 largest singular values
fig(9) = figure(9);
n = 1:4

p1 = scatter(n, sigma1(n), 'r', 'filled')
hold on % overlay second scatter plot on same figure
p2 = scatter(n, sigma2(n), 'b', 'filled')
hold off

legend([p1, p2], 'hcec','hcec+7', 'FontSize', 14)
title('Whole genome singular values', 'FontSize', 16)
```

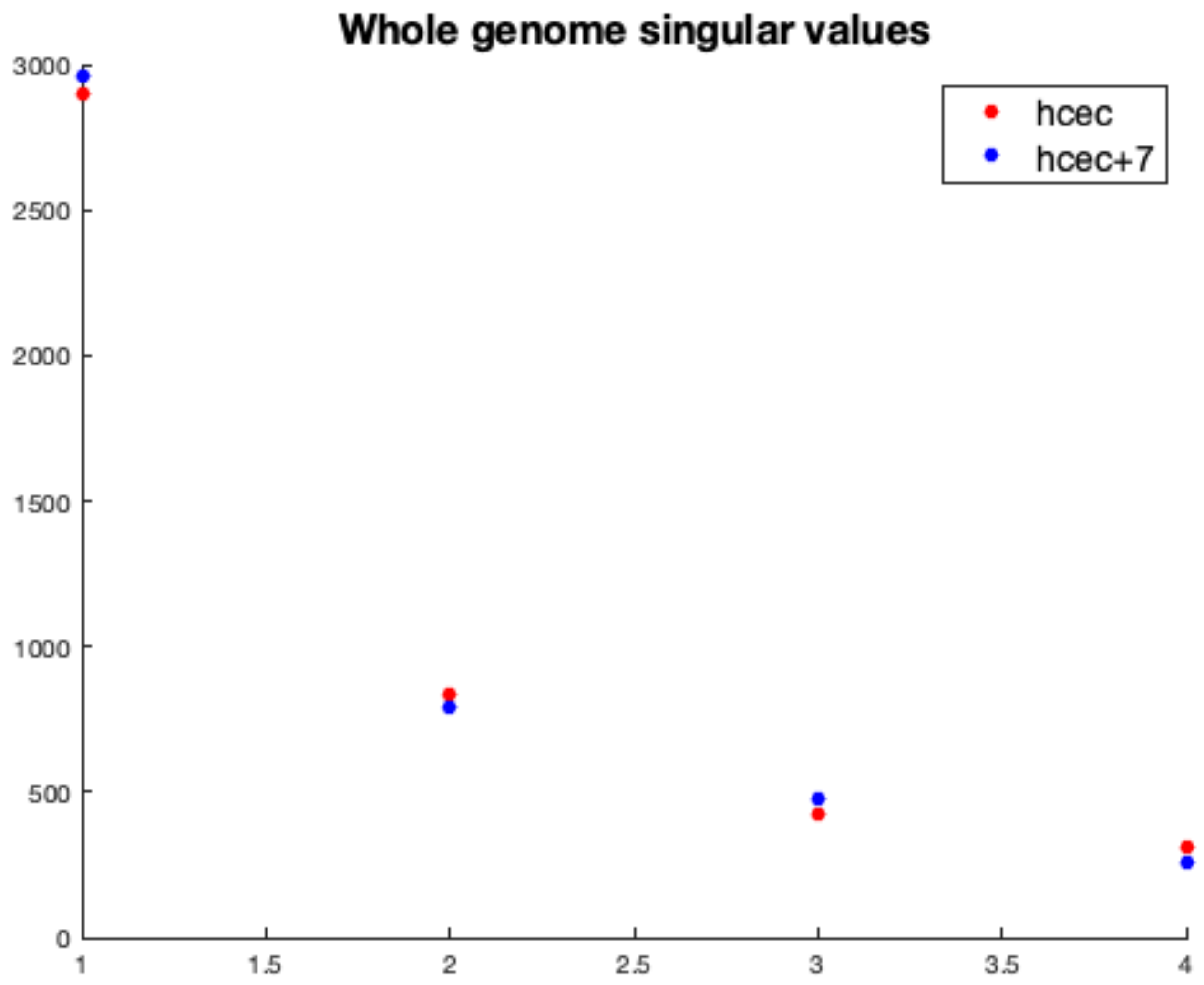
```
n =

     1     2     3     4


p1 = 

  Scatter with properties:

             Marker: 'o'
    MarkerEdgeColor: 'none'
    MarkerFaceColor: 'flat'
           SizeData: 36
          LineWidth: 0.5000
              XData: [1 2 3 4]
              YData: [2.9014e+03 833.4139 422.7666 311.3927]
              ZData: [1×0 double]
              CData: [1 0 0]

  Use GET to show all properties


p2 = 

  Scatter with properties:

             Marker: 'o'
    MarkerEdgeColor: 'none'
    MarkerFaceColor: 'flat'
           SizeData: 36
          LineWidth: 0.5000
              XData: [1 2 3 4]
              YData: [2.9646e+03 788.2809 477.3763 259.7240]
              ZData: [1×0 double]
              CData: [0 0 1]
```

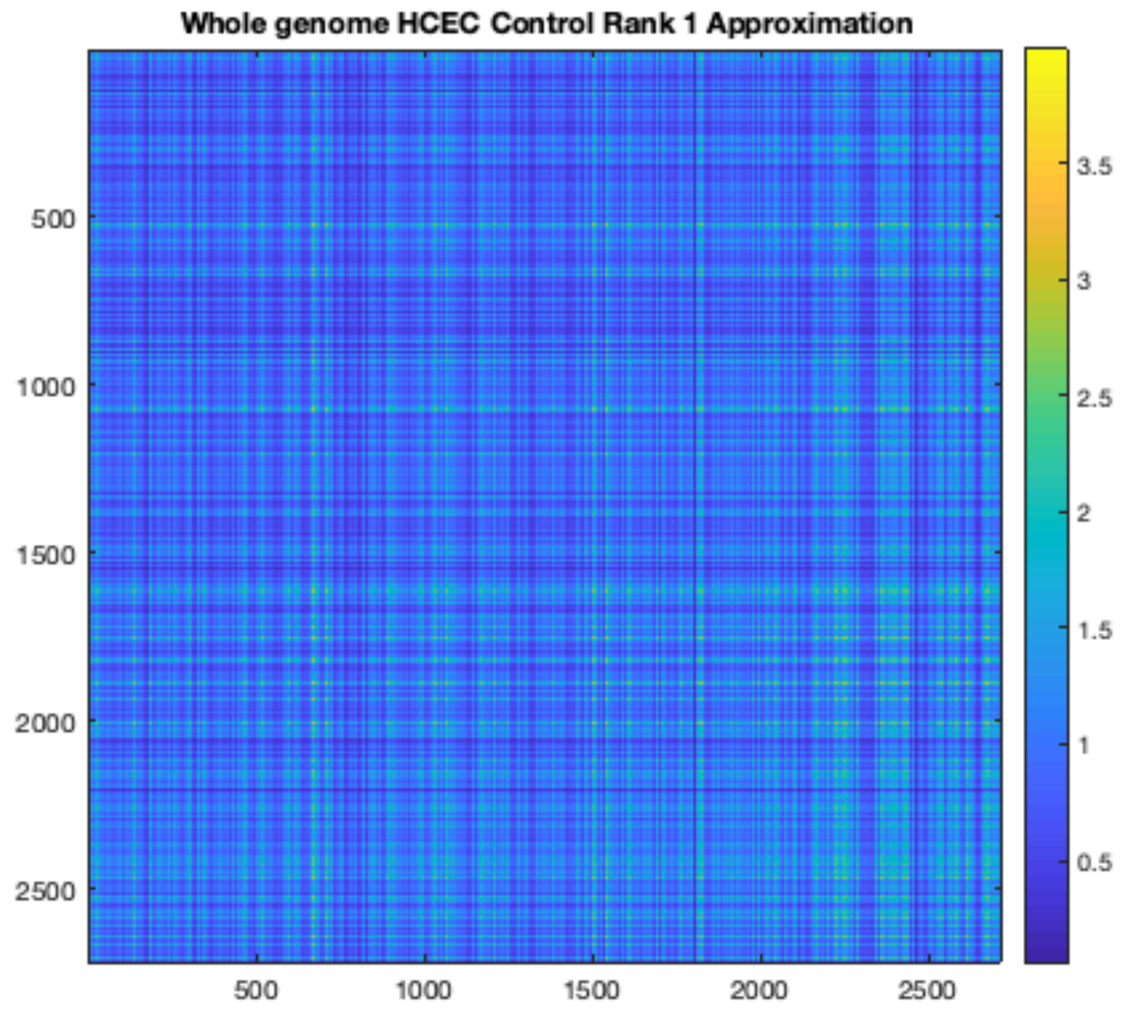
```
Use GET to show all properties
```

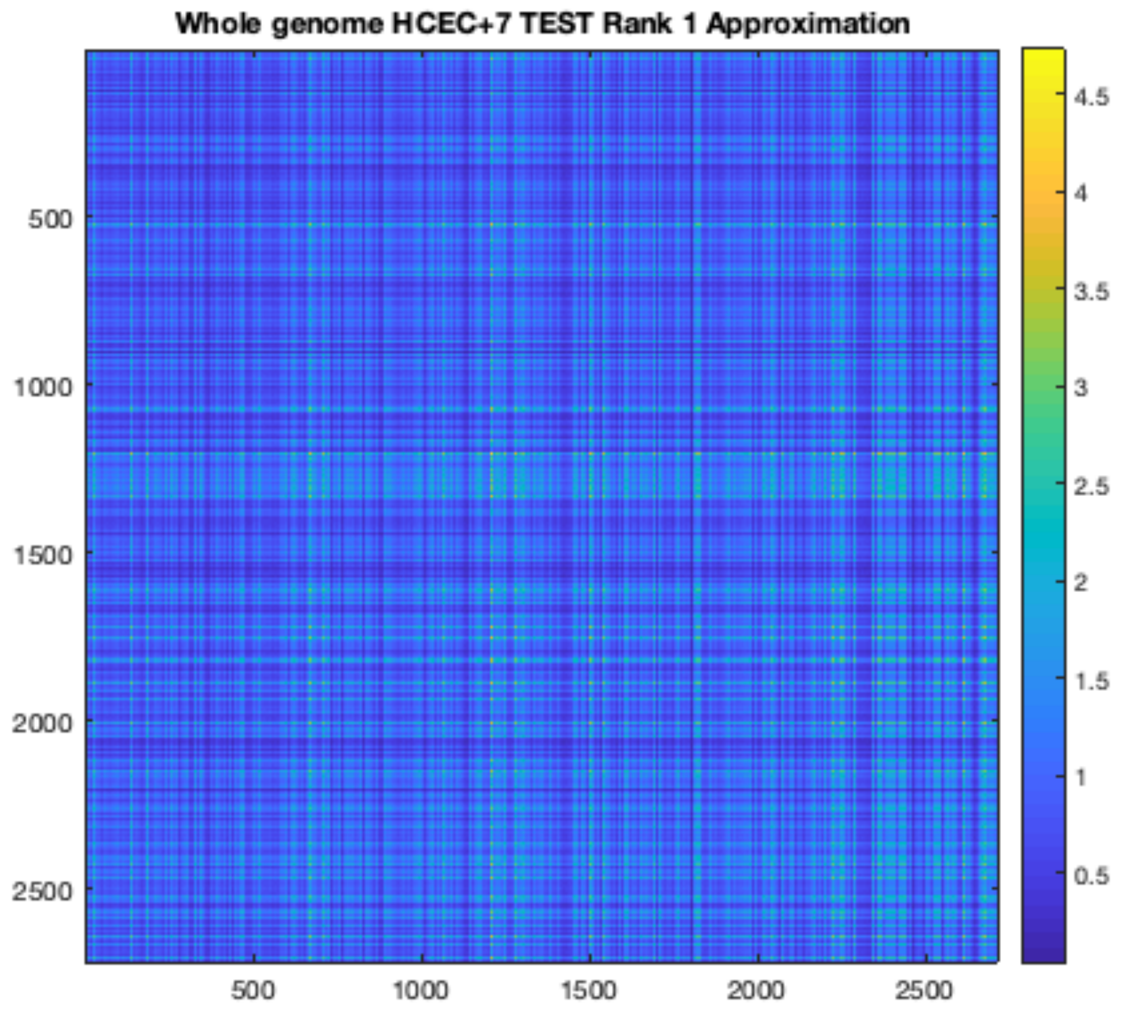


Compare Rank1 approximation

```
fig(10) = figure(10);
set(fig(10), 'unit', 'normalized', 'Position', [.15 .15 .7 .4])

for k = 1:2
    ax(k) = subplot(1,2,k);
end

% Control
subplot(ax(1));
A1 = sigma1(1)*U1(:,1)*V1(:,1)';
imagesc(A1)
title('Whole genome HCEC Control Rank 1 Approximation')
colorbar
axis('square')

% Test
subplot(ax(2));
A2 = sigma2(1)*U2(:,1)*V2(:,1)';
imagesc(A2)
title('Whole genome HCEC+7 TEST Rank 1 Approximation')
colorbar
axis('square')
```

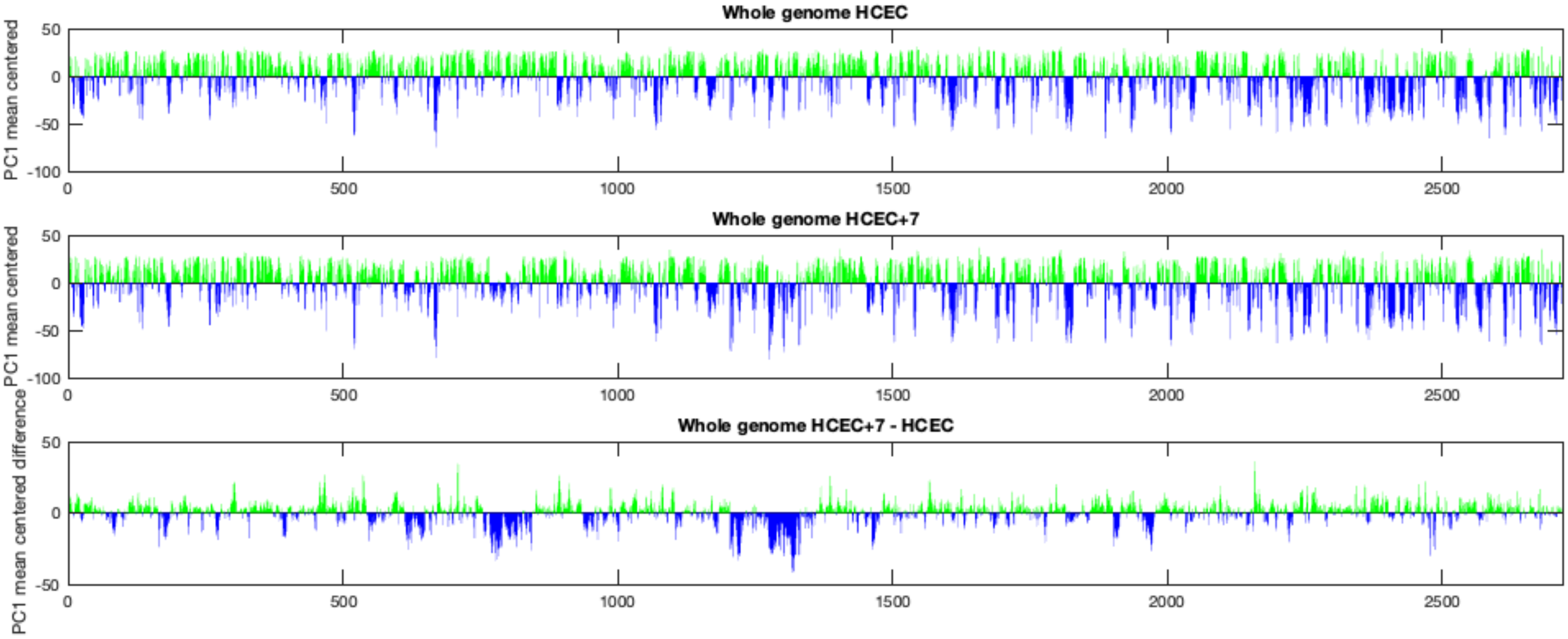



Whole genome HCEC Control Rank 1 Approximation


Whole genome HCEC+7 TEST Rank 1 Approximation

Compare difference between first principle component

```
% PC1 of control
% mean center (on rows in this case since we want low dimensional)
center = hcecHic - mean(hcecHic,2);
[U,S,V] = svd(center','econ');
svdScores = U * S;
PC1_control = svdScores(:,1);
x_values = 1:length(PC1_control);
pos_index_control = PC1_control >= 0;
neg_index_control = PC1_control < 0;

% PC1 of test
center = hcec7Hic - mean(hcec7Hic,2);
[U,S,V] = svd(center','econ');
svdScores = U * S;
PC1_test = svdScores(:,1);
pos_index_test = PC1_test >= 0;
neg_index_test = PC1_test < 0;

% difference in PC1
PC1_difference = PC1_test - PC1_control;
pos_index_difference = PC1_difference >= 0;
neg_index_difference = PC1_difference < 0;

fig(11) = figure(11);
set(fig(11), 'unit', 'normalized', 'Position', [.15 .15 .7 .4])

for k = 1:3
    ax(k) = subplot(3,1,k);
end

% make barplot
subplot(ax(1));
bar(x_values(pos_index_control), PC1_control(pos_index_control), 'g', 'BarWidth', 1)
hold on
bar(x_values(neg_index_control), PC1_control(neg_index_control), 'b', 'BarWidth', 1)
ylabel('PC1 mean centered')
title('Whole genome HCEC')

subplot(ax(2));
bar(x_values(pos_index_test), PC1_test(pos_index_test), 'g', 'BarWidth', 1)
hold on
bar(x_values(neg_index_test), PC1_test(neg_index_test), 'b', 'BarWidth', 1)
ylabel('PC1 mean centered')
title('Whole genome HCEC+7')

subplot(ax(3));
bar(x_values(pos_index_difference), PC1_difference(pos_index_difference), 'g', 'BarWidth', 1)
hold on
bar(x_values(neg_index_difference), PC1_difference(neg_index_difference), 'b', 'BarWidth', 1)
ylabel('PC1 mean centered difference')
title('Whole genome HCEC+7 - HCEC')
```



Published with MATLAB® R2019a