Table of Contents

Problem 1 - RNA-seq

```
%Initialization (initial E step)
Y = [1 0 1 1 1;1 1 0 0 1;1 1 1 0 0]
readA= Y(:,1)/sum(Y(:,1));
readB= Y(:,2)/sum(Y(:,2));
readC= Y(:,3)/sum(Y(:,3));
readD= Y(:,4)/sum(Y(:,4));
readE= Y(:,5)/sum(Y(:,5));
INTreadA= Y(:,1)/sum(Y(:,1));
INTreadB= Y(:,2)/sum(Y(:,2));
INTreadC= Y(:,3)/sum(Y(:,3));
INTreadD= Y(:,4)/sum(Y(:,4));
INTreadE= Y(:,5)/sum(Y(:,5));
count = 0;
dParameter1 = 1;
```

```
Y =

     1     0     1     1     1
     1     1     0     0     1
     1     1     1     0     0
```

Section (1A):

```
while dParameter1 > 0.01
    iterA = readA;
    iterB = readB;
    iterC = readC;
    iterD = readD;
    iterE = readE;
    %Single iteration prep (M step)
    totalChance =
 (readA(1,:)+readB(1,:)+readC(1,:)+readD(1,:)+readE(1,:))+(readA(2,:)+readB(2,:)+r
    pred = (readA(1,:)+readB(1,:)+readC(1,:)+readD(1,:)+readE(1,:))/
totalChance;
    pgreen = (readA(2,:)+readB(2,:)+readC(2,:)+readD(2,:)+readE(2,:))/
totalChance;
```

```
    pblue = (readA(3,:)+readB(3,:)+readC(3,:)+readD(3,:)+readE(3,:))/
totalChance;
    %Set new read percentag
    readA(1,:) = INTreadA(1,:)*pred;
    readB(1,:) = INTreadB(1,:)*pred;
    readC(1,:) = INTreadC(1,:)*pred;
    readD(1,:) = INTreadD(1,:)*pred;
    readE(1,:) = INTreadE(1,:)*pred;
    readA(2,:) = INTreadA(2,:)*pgreen;
    readB(2,:) = INTreadB(2,:)*pgreen;
    readC(2,:) = INTreadC(2,:)*pgreen;
    readD(2,:) = INTreadD(2,:)*pgreen;
    readE(2,:) = INTreadE(2,:)*pgreen;
    readA(3,:) = INTreadA(3,:)*pblue;
    readB(3,:) = INTreadB(3,:)*pblue;
    readC(3,:) = INTreadC(3,:)*pblue;
    readD(3,:) = INTreadD(3,:)*pblue;
    readE(3,:) = INTreadE(3,:)*pblue;
    %Normalize read percentages
    readA = readA/sum(readA);
    readB = readB/sum(readB);
    readC = readC/sum(readC);
    readD = readD/sum(readD);
    readE = readE/sum(readE);
    %change in parameter
    dParaMatrix = zeros(3,5);
    dParaMatrix(1,1) = abs(readA(1,:)-iterA(1,:));
    dParaMatrix(2,1) = abs(readA(2,:)-iterA(2,:));
    dParaMatrix(3,1) = abs(readA(3,:)-iterA(3,:));
    dParaMatrix(1,2) = abs(readB(1,:)-iterB(1,:));
    dParaMatrix(2,2) = abs(readB(2,:)-iterB(2,:));
    dParaMatrix(3,2) = abs(readB(3,:)-iterB(3,:));
    dParaMatrix(1,3) = abs(readC(1,:)-iterC(1,:));
    dParaMatrix(2,3) = abs(readC(2,:)-iterC(2,:));
    dParaMatrix(3,3) = abs(readC(3,:)-iterC(3,:));
    dParaMatrix(1,4) = abs(readD(1,:)-iterD(1,:));
    dParaMatrix(2,4) = abs(readD(2,:)-iterD(2,:));
    dParaMatrix(3,4) = abs(readD(3,:)-iterD(3,:));
    dParaMatrix(1,5) = abs(readE(1,:)-iterE(1,:));
    dParaMatrix(2,5) = abs(readE(2,:)-iterE(2,:));
    dParaMatrix(3,5) = abs(readE(3,:)-iterE(3,:));
    dParameterRow = max(dParaMatrix);
    dParameter1 = max(dParameterRow);
    count = count + 1;
end
dParameter1
pred
pgreen
pblue
count
% This algorithm requires 6 iteratons to convergence. This is the same
% solution from class.
```

```
dParameter1 =

    0.0061


pred =

    0.6343


pgreen =

    0.1828


pblue =

    0.1828


count =

     6
```

Section (1B)

```
Y = [1 0 1 1 1;1 1 0 0 1;1 1 1 0 0]
readA= Y(:,1)/sum(Y(:,1));
readB= Y(:,2)/sum(Y(:,2));
readC= Y(:,3)/sum(Y(:,3));
readD= Y(:,4)/sum(Y(:,4));
readE= Y(:,5)/sum(Y(:,5));
INTreadA= Y(:,1)/sum(Y(:,1));
INTreadB= Y(:,2)/sum(Y(:,2));
INTreadC= Y(:,3)/sum(Y(:,3));
INTreadD= Y(:,4)/sum(Y(:,4));
INTreadE= Y(:,5)/sum(Y(:,5));
count = 0;
dParameter1 = 1;
while dParameter1 > 0.001
    iterA = readA;
    iterB = readB;
    iterC = readC;
    iterD = readD;
    iterE = readE;
    %Single iteration prep (M step)
    totalChance =
 (readA(1,:)+readB(1,:)+readC(1,:)+readD(1,:)+readE(1,:))+(readA(2,:)+readB(2,:)+r
    pred = (readA(1,:)+readB(1,:)+readC(1,:)+readD(1,:)+readE(1,:))/
totalChance;
    pgreen = (readA(2,:)+readB(2,:)+readC(2,:)+readD(2,:)+readE(2,:))/
totalChance;
```

```
    pblue = (readA(3,:)+readB(3,:)+readC(3,:)+readD(3,:)+readE(3,:))/
totalChance;
    %Set new read percentag
    readA(1,:) = INTreadA(1,:)*pred;
    readB(1,:) = INTreadB(1,:)*pred;
    readC(1,:) = INTreadC(1,:)*pred;
    readD(1,:) = INTreadD(1,:)*pred;
    readE(1,:) = INTreadE(1,:)*pred;
    readA(2,:) = INTreadA(2,:)*pgreen;
    readB(2,:) = INTreadB(2,:)*pgreen;
    readC(2,:) = INTreadC(2,:)*pgreen;
    readD(2,:) = INTreadD(2,:)*pgreen;
    readE(2,:) = INTreadE(2,:)*pgreen;
    readA(3,:) = INTreadA(3,:)*pblue;
    readB(3,:) = INTreadB(3,:)*pblue;
    readC(3,:) = INTreadC(3,:)*pblue;
    readD(3,:) = INTreadD(3,:)*pblue;
    readE(3,:) = INTreadE(3,:)*pblue;
    %Normalize read percentages
    readA = readA/sum(readA);
    readB = readB/sum(readB);
    readC = readC/sum(readC);
    readD = readD/sum(readD);
    readE = readE/sum(readE);
    %change in parameter
    dParaMatrix = zeros(3,5);
    dParaMatrix(1,1) = abs(readA(1,:)-iterA(1,:));
    dParaMatrix(2,1) = abs(readA(2,:)-iterA(2,:));
    dParaMatrix(3,1) = abs(readA(3,:)-iterA(3,:));
    dParaMatrix(1,2) = abs(readB(1,:)-iterB(1,:));
    dParaMatrix(2,2) = abs(readB(2,:)-iterB(2,:));
    dParaMatrix(3,2) = abs(readB(3,:)-iterB(3,:));
    dParaMatrix(1,3) = abs(readC(1,:)-iterC(1,:));
    dParaMatrix(2,3) = abs(readC(2,:)-iterC(2,:));
    dParaMatrix(3,3) = abs(readC(3,:)-iterC(3,:));
    dParaMatrix(1,4) = abs(readD(1,:)-iterD(1,:));
    dParaMatrix(2,4) = abs(readD(2,:)-iterD(2,:));
    dParaMatrix(3,4) = abs(readD(3,:)-iterD(3,:));
    dParaMatrix(1,5) = abs(readE(1,:)-iterE(1,:));
    dParaMatrix(2,5) = abs(readE(2,:)-iterE(2,:));
    dParaMatrix(3,5) = abs(readE(3,:)-iterE(3,:));
    dParameterRow = max(dParaMatrix);
    dParameter1 = max(dParameterRow);
    count = count + 1;
end
dParameter1
pred
pgreen
pblue
count
% 9 iterations are required.


Y =
```

```
     1     0     1     1     1
     1     1     0     0     1
     1     1     1     0     0


dParameter1 =

   7.5360e-04


pred =

    0.6396


pgreen =

    0.1802


pblue =

    0.1802


count =

     9
```

Section (1C)

```
%New Initialization 1:
Y = [0 0 1 1 1;1 1 0 0 1;1 1 1 0 0]
readA= Y(:,1)/sum(Y(:,1));
readB= Y(:,2)/sum(Y(:,2));
readC= Y(:,3)/sum(Y(:,3));
readD= Y(:,4)/sum(Y(:,4));
readE= Y(:,5)/sum(Y(:,5));
INTreadA= Y(:,1)/sum(Y(:,1));
INTreadB= Y(:,2)/sum(Y(:,2));
INTreadC= Y(:,3)/sum(Y(:,3));
INTreadD= Y(:,4)/sum(Y(:,4));
INTreadE= Y(:,5)/sum(Y(:,5));
count = 0;
dParameter1 = 1;
while dParameter1 > 0.01
    iterA = readA;
    iterB = readB;
    iterC = readC;
    iterD = readD;
    iterE = readE;
```

```
    %Single iteration prep (M step)
    totalChance =
 (readA(1,:)+readB(1,:)+readC(1,:)+readD(1,:)+readE(1,:))+(readA(2,:)+readB(2,:)+r
    pred = (readA(1,:)+readB(1,:)+readC(1,:)+readD(1,:)+readE(1,:))/
totalChance;
    pgreen = (readA(2,:)+readB(2,:)+readC(2,:)+readD(2,:)+readE(2,:))/
totalChance;
    pblue = (readA(3,:)+readB(3,:)+readC(3,:)+readD(3,:)+readE(3,:))/
totalChance;
    %Set new read percentag
    readA(1,:) = INTreadA(1,:)*pred;
    readB(1,:) = INTreadB(1,:)*pred;
    readC(1,:) = INTreadC(1,:)*pred;
    readD(1,:) = INTreadD(1,:)*pred;
    readE(1,:) = INTreadE(1,:)*pred;
    readA(2,:) = INTreadA(2,:)*pgreen;
    readB(2,:) = INTreadB(2,:)*pgreen;
    readC(2,:) = INTreadC(2,:)*pgreen;
    readD(2,:) = INTreadD(2,:)*pgreen;
    readE(2,:) = INTreadE(2,:)*pgreen;
    readA(3,:) = INTreadA(3,:)*pblue;
    readB(3,:) = INTreadB(3,:)*pblue;
    readC(3,:) = INTreadC(3,:)*pblue;
    readD(3,:) = INTreadD(3,:)*pblue;
    readE(3,:) = INTreadE(3,:)*pblue;
    %Normalize read percentages
    readA = readA/sum(readA);
    readB = readB/sum(readB);
    readC = readC/sum(readC);
    readD = readD/sum(readD);
    readE = readE/sum(readE);
    %change in parameter
    dParaMatrix = zeros(3,5);
    dParaMatrix(1,1) = abs(readA(1,:)-iterA(1,:));
    dParaMatrix(2,1) = abs(readA(2,:)-iterA(2,:));
    dParaMatrix(3,1) = abs(readA(3,:)-iterA(3,:));
    dParaMatrix(1,2) = abs(readB(1,:)-iterB(1,:));
    dParaMatrix(2,2) = abs(readB(2,:)-iterB(2,:));
    dParaMatrix(3,2) = abs(readB(3,:)-iterB(3,:));
    dParaMatrix(1,3) = abs(readC(1,:)-iterC(1,:));
    dParaMatrix(2,3) = abs(readC(2,:)-iterC(2,:));
    dParaMatrix(3,3) = abs(readC(3,:)-iterC(3,:));
    dParaMatrix(1,4) = abs(readD(1,:)-iterD(1,:));
    dParaMatrix(2,4) = abs(readD(2,:)-iterD(2,:));
    dParaMatrix(3,4) = abs(readD(3,:)-iterD(3,:));
    dParaMatrix(1,5) = abs(readE(1,:)-iterE(1,:));
    dParaMatrix(2,5) = abs(readE(2,:)-iterE(2,:));
    dParaMatrix(3,5) = abs(readE(3,:)-iterE(3,:));
    dParameterRow = max(dParaMatrix);
    dParameter1 = max(dParameterRow);
    count = count + 1;
end
dParameter1
pred;
```

```
pgreen;
pblue;
X = [pred pgreen pblue];
labels = {'pred','pgreen','pblue'};
figure
pie(X,labels)
snapnow
count
Y = [0 0 1 1 1;1 1 0 0 1;1 1 1 0 0]
readA= Y(:,1)/sum(Y(:,1));
readB= Y(:,2)/sum(Y(:,2));
readC= Y(:,3)/sum(Y(:,3));
readD= Y(:,4)/sum(Y(:,4));
readE= Y(:,5)/sum(Y(:,5));
INTreadA= Y(:,1)/sum(Y(:,1));
INTreadB= Y(:,2)/sum(Y(:,2));
INTreadC= Y(:,3)/sum(Y(:,3));
INTreadD= Y(:,4)/sum(Y(:,4));
INTreadE= Y(:,5)/sum(Y(:,5));
count = 0;
dParameter1 = 1;
while dParameter1 > 0.001
    iterA = readA;
    iterB = readB;
    iterC = readC;
    iterD = readD;
    iterE = readE;
    %Single iteration prep (M step)
    totalChance =
 (readA(1,:)+readB(1,:)+readC(1,:)+readD(1,:)+readE(1,:))+(readA(2,:)+readB(2,:)+r
    pred = (readA(1,:)+readB(1,:)+readC(1,:)+readD(1,:)+readE(1,:))/
totalChance;
    pgreen = (readA(2,:)+readB(2,:)+readC(2,:)+readD(2,:)+readE(2,:))/
totalChance;
    pblue = (readA(3,:)+readB(3,:)+readC(3,:)+readD(3,:)+readE(3,:))/
totalChance;
    %Set new read percentag
    readA(1,:) = INTreadA(1,:)*pred;
    readB(1,:) = INTreadB(1,:)*pred;
    readC(1,:) = INTreadC(1,:)*pred;
    readD(1,:) = INTreadD(1,:)*pred;
    readE(1,:) = INTreadE(1,:)*pred;
    readA(2,:) = INTreadA(2,:)*pgreen;
    readB(2,:) = INTreadB(2,:)*pgreen;
    readC(2,:) = INTreadC(2,:)*pgreen;
    readD(2,:) = INTreadD(2,:)*pgreen;
    readE(2,:) = INTreadE(2,:)*pgreen;
    readA(3,:) = INTreadA(3,:)*pblue;
    readB(3,:) = INTreadB(3,:)*pblue;
    readC(3,:) = INTreadC(3,:)*pblue;
    readD(3,:) = INTreadD(3,:)*pblue;
    readE(3,:) = INTreadE(3,:)*pblue;
    %Normalize read percentages
    readA = readA/sum(readA);
```

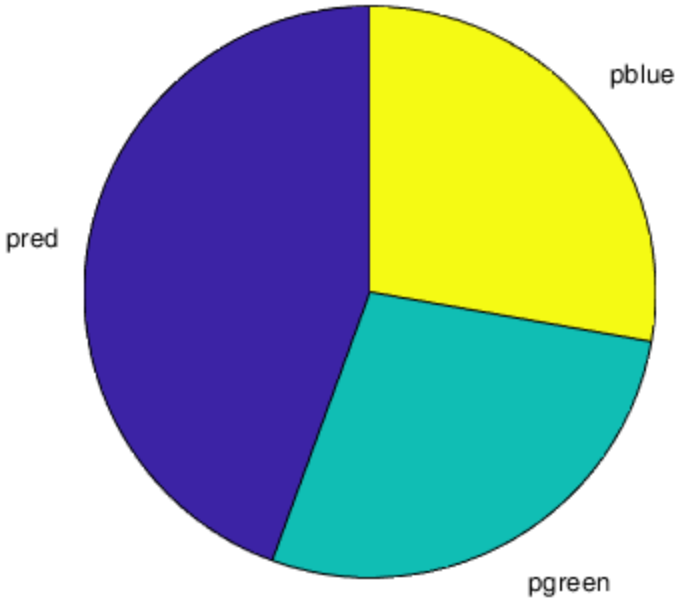
```
    readB = readB/sum(readB);
    readC = readC/sum(readC);
    readD = readD/sum(readD);
    readE = readE/sum(readE);
    %change in parameter
    dParaMatrix = zeros(3,5);
    dParaMatrix(1,1) = abs(readA(1,:)-iterA(1,:));
    dParaMatrix(2,1) = abs(readA(2,:)-iterA(2,:));
    dParaMatrix(3,1) = abs(readA(3,:)-iterA(3,:));
    dParaMatrix(1,2) = abs(readB(1,:)-iterB(1,:));
    dParaMatrix(2,2) = abs(readB(2,:)-iterB(2,:));
    dParaMatrix(3,2) = abs(readB(3,:)-iterB(3,:));
    dParaMatrix(1,3) = abs(readC(1,:)-iterC(1,:));
    dParaMatrix(2,3) = abs(readC(2,:)-iterC(2,:));
    dParaMatrix(3,3) = abs(readC(3,:)-iterC(3,:));
    dParaMatrix(1,4) = abs(readD(1,:)-iterD(1,:));
    dParaMatrix(2,4) = abs(readD(2,:)-iterD(2,:));
    dParaMatrix(3,4) = abs(readD(3,:)-iterD(3,:));
    dParaMatrix(1,5) = abs(readE(1,:)-iterE(1,:));
    dParaMatrix(2,5) = abs(readE(2,:)-iterE(2,:));
    dParaMatrix(3,5) = abs(readE(3,:)-iterE(3,:));
    dParameterRow = max(dParaMatrix);
    dParameter1 = max(dParameterRow);
    count = count + 1;
    if count < 5
        X = [pred pgreen pblue];
        labels = {'pred','pgreen','pblue'};
        figure
        pie(X,labels)
    end
end
dParameter1
pred;
pgreen;
pblue;
X = [pred pgreen pblue];
labels = {'pred','pgreen','pblue'};
figure
pie(X,labels)
snapnow
count
%This set converges for both 0.01 and 0.001 cutoffs.
Y = [0 0 1 1 1;0 1 0 0 1;1 1 1 0 0]
readA= Y(:,1)/sum(Y(:,1));
readB= Y(:,2)/sum(Y(:,2));
readC= Y(:,3)/sum(Y(:,3));
readD= Y(:,4)/sum(Y(:,4));
readE= Y(:,5)/sum(Y(:,5));
INTreadA= Y(:,1)/sum(Y(:,1));
INTreadB= Y(:,2)/sum(Y(:,2));
INTreadC= Y(:,3)/sum(Y(:,3));
INTreadD= Y(:,4)/sum(Y(:,4));
INTreadE= Y(:,5)/sum(Y(:,5));
count = 0;
```

```
dParameter1 = 1;
while dParameter1 > 0.01
    iterA = readA;
    iterB = readB;
    iterC = readC;
    iterD = readD;
    iterE = readE;
    %Single iteration prep (M step)
    totalChance =
 (readA(1,:)+readB(1,:)+readC(1,:)+readD(1,:)+readE(1,:))+(readA(2,:)+readB(2,:)+r
    pred = (readA(1,:)+readB(1,:)+readC(1,:)+readD(1,:)+readE(1,:))/
totalChance;
    pgreen = (readA(2,:)+readB(2,:)+readC(2,:)+readD(2,:)+readE(2,:))/
totalChance;
    pblue = (readA(3,:)+readB(3,:)+readC(3,:)+readD(3,:)+readE(3,:))/
totalChance;
    %Set new read percentag
    readA(1,:) = INTreadA(1,:)*pred;
    readB(1,:) = INTreadB(1,:)*pred;
    readC(1,:) = INTreadC(1,:)*pred;
    readD(1,:) = INTreadD(1,:)*pred;
    readE(1,:) = INTreadE(1,:)*pred;
    readA(2,:) = INTreadA(2,:)*pgreen;
    readB(2,:) = INTreadB(2,:)*pgreen;
    readC(2,:) = INTreadC(2,:)*pgreen;
    readD(2,:) = INTreadD(2,:)*pgreen;
    readE(2,:) = INTreadE(2,:)*pgreen;
    readA(3,:) = INTreadA(3,:)*pblue;
    readB(3,:) = INTreadB(3,:)*pblue;
    readC(3,:) = INTreadC(3,:)*pblue;
    readD(3,:) = INTreadD(3,:)*pblue;
    readE(3,:) = INTreadE(3,:)*pblue;
    %Normalize read percentages
    readA = readA/sum(readA);
    readB = readB/sum(readB);
    readC = readC/sum(readC);
    readD = readD/sum(readD);
    readE = readE/sum(readE);
    %change in parameter
    dParaMatrix = zeros(3,5);
    dParaMatrix(1,1) = abs(readA(1,:)-iterA(1,:));
    dParaMatrix(2,1) = abs(readA(2,:)-iterA(2,:));
    dParaMatrix(3,1) = abs(readA(3,:)-iterA(3,:));
    dParaMatrix(1,2) = abs(readB(1,:)-iterB(1,:));
    dParaMatrix(2,2) = abs(readB(2,:)-iterB(2,:));
    dParaMatrix(3,2) = abs(readB(3,:)-iterB(3,:));
    dParaMatrix(1,3) = abs(readC(1,:)-iterC(1,:));
    dParaMatrix(2,3) = abs(readC(2,:)-iterC(2,:));
    dParaMatrix(3,3) = abs(readC(3,:)-iterC(3,:));
    dParaMatrix(1,4) = abs(readD(1,:)-iterD(1,:));
    dParaMatrix(2,4) = abs(readD(2,:)-iterD(2,:));
    dParaMatrix(3,4) = abs(readD(3,:)-iterD(3,:));
    dParaMatrix(1,5) = abs(readE(1,:)-iterE(1,:));
    dParaMatrix(2,5) = abs(readE(2,:)-iterE(2,:));
```

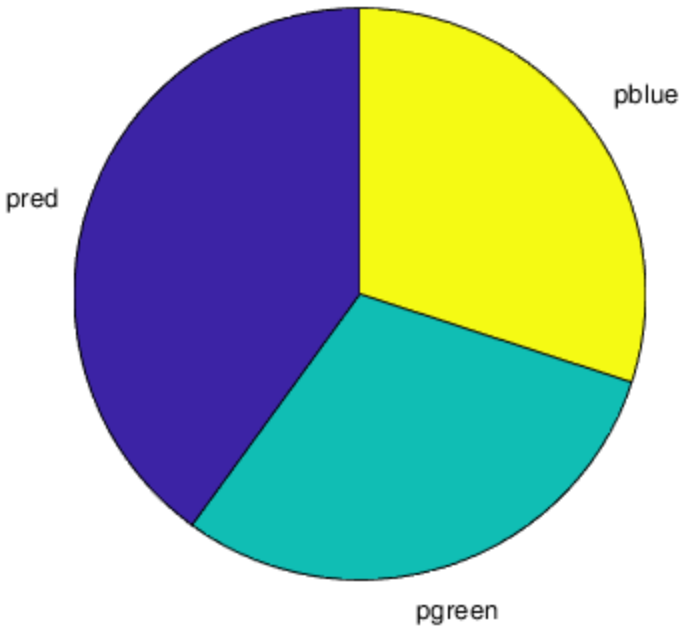
```
    dParaMatrix(3,5) = abs(readE(3,:)-iterE(3,:));
    dParameterRow = max(dParaMatrix);
    dParameter1 = max(dParameterRow);
    count = count + 1;
end
dParameter1
pred;
pgreen;
pblue;
X = [pred pgreen pblue];
labels = {'pred','pgreen','pblue'};
figure
pie(X,labels)
snapnow
count
Y = [0 0 1 1 1;0 1 0 0 1;1 1 1 0 0]
readA= Y(:,1)/sum(Y(:,1));
readB= Y(:,2)/sum(Y(:,2));
readC= Y(:,3)/sum(Y(:,3));
readD= Y(:,4)/sum(Y(:,4));
readE= Y(:,5)/sum(Y(:,5));
INTreadA= Y(:,1)/sum(Y(:,1));
INTreadB= Y(:,2)/sum(Y(:,2));
INTreadC= Y(:,3)/sum(Y(:,3));
INTreadD= Y(:,4)/sum(Y(:,4));
INTreadE= Y(:,5)/sum(Y(:,5));
count = 0;
dParameter1 = 1;
while dParameter1 > 0.001
    iterA = readA;
    iterB = readB;
    iterC = readC;
    iterD = readD;
    iterE = readE;
    %Single iteration prep (M step)
    totalChance =
 (readA(1,:)+readB(1,:)+readC(1,:)+readD(1,:)+readE(1,:))+(readA(2,:)+readB(2,:)+r
    pred = (readA(1,:)+readB(1,:)+readC(1,:)+readD(1,:)+readE(1,:))/
totalChance;
    pgreen = (readA(2,:)+readB(2,:)+readC(2,:)+readD(2,:)+readE(2,:))/
totalChance;
    pblue = (readA(3,:)+readB(3,:)+readC(3,:)+readD(3,:)+readE(3,:))/
totalChance;
    %Set new read percentag
    readA(1,:) = INTreadA(1,:)*pred;
    readB(1,:) = INTreadB(1,:)*pred;
    readC(1,:) = INTreadC(1,:)*pred;
    readD(1,:) = INTreadD(1,:)*pred;
    readE(1,:) = INTreadE(1,:)*pred;
    readA(2,:) = INTreadA(2,:)*pgreen;
    readB(2,:) = INTreadB(2,:)*pgreen;
    readC(2,:) = INTreadC(2,:)*pgreen;
    readD(2,:) = INTreadD(2,:)*pgreen;
    readE(2,:) = INTreadE(2,:)*pgreen;
```

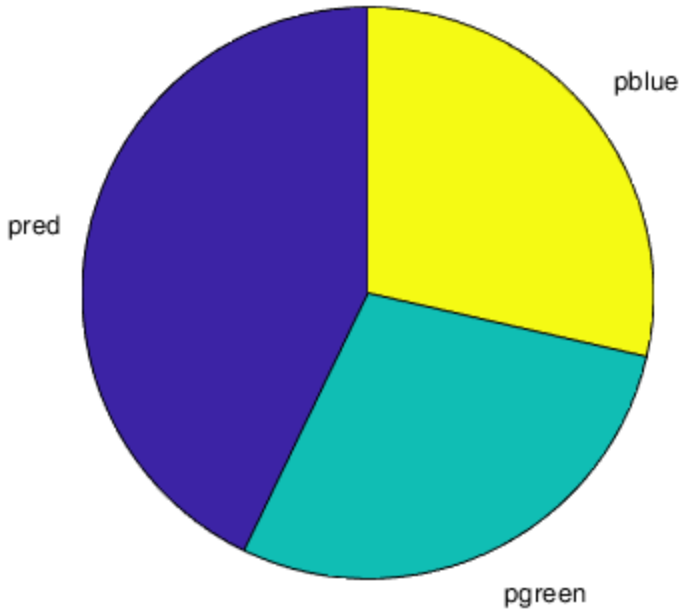
```
    readA(3,:) = INTreadA(3,:)*pblue;
    readB(3,:) = INTreadB(3,:)*pblue;
    readC(3,:) = INTreadC(3,:)*pblue;
    readD(3,:) = INTreadD(3,:)*pblue;
    readE(3,:) = INTreadE(3,:)*pblue;
    %Normalize read percentages
    readA = readA/sum(readA);
    readB = readB/sum(readB);
    readC = readC/sum(readC);
    readD = readD/sum(readD);
    readE = readE/sum(readE);
    %change in parameter
    dParaMatrix = zeros(3,5);
    dParaMatrix(1,1) = abs(readA(1,:)-iterA(1,:));
    dParaMatrix(2,1) = abs(readA(2,:)-iterA(2,:));
    dParaMatrix(3,1) = abs(readA(3,:)-iterA(3,:));
    dParaMatrix(1,2) = abs(readB(1,:)-iterB(1,:));
    dParaMatrix(2,2) = abs(readB(2,:)-iterB(2,:));
    dParaMatrix(3,2) = abs(readB(3,:)-iterB(3,:));
    dParaMatrix(1,3) = abs(readC(1,:)-iterC(1,:));
    dParaMatrix(2,3) = abs(readC(2,:)-iterC(2,:));
    dParaMatrix(3,3) = abs(readC(3,:)-iterC(3,:));
    dParaMatrix(1,4) = abs(readD(1,:)-iterD(1,:));
    dParaMatrix(2,4) = abs(readD(2,:)-iterD(2,:));
    dParaMatrix(3,4) = abs(readD(3,:)-iterD(3,:));
    dParaMatrix(1,5) = abs(readE(1,:)-iterE(1,:));
    dParaMatrix(2,5) = abs(readE(2,:)-iterE(2,:));
    dParaMatrix(3,5) = abs(readE(3,:)-iterE(3,:));
    dParameterRow = max(dParaMatrix);
    dParameter1 = max(dParameterRow);
    count = count + 1;
    if count < 5
        X = [pred pgreen pblue];
        labels = {'pred','pgreen','pblue'};
        figure
        pie(X,labels)
        snapnow
    end
end
dParameter1
pred;
pgreen;
pblue;
X = [pred pgreen pblue];
labels = {'pred','pgreen','pblue'};
figure
pie(X,labels)
snapnow
count
%This initialization converges to be without any green.


Y =
```

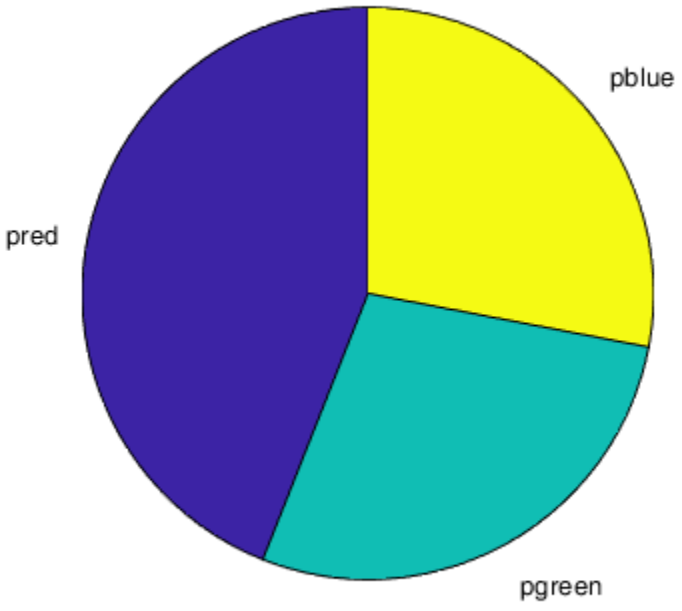
```
     0     0     1     1     1
     1     1     0     0     1
     1     1     1     0     0

dParameter1 =

    0.0043
```

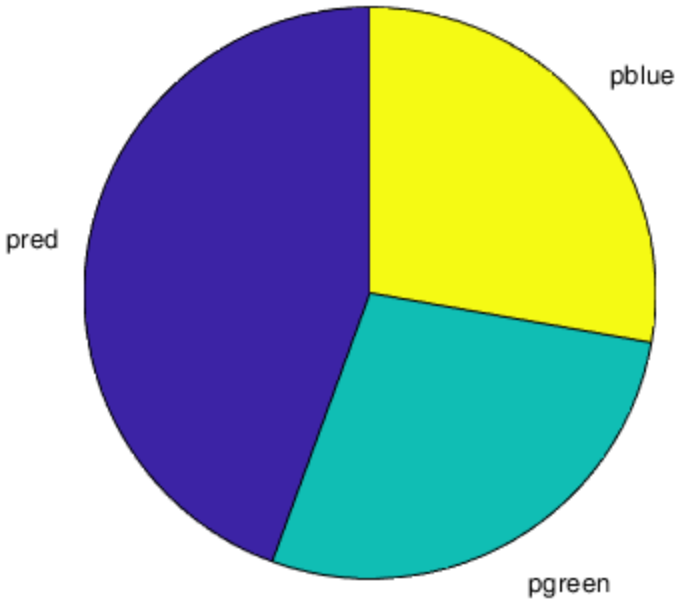


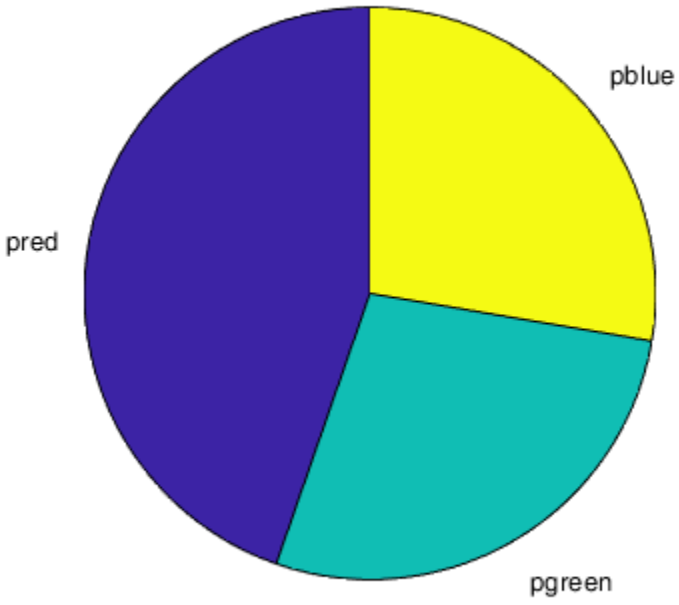

```
count =

     4

Y =

     0     0     1     1     1
     1     1     0     0     1
     1     1     1     0     0

dParameter1 =

   6.2578e-04
```


pblue
pred
pgreen

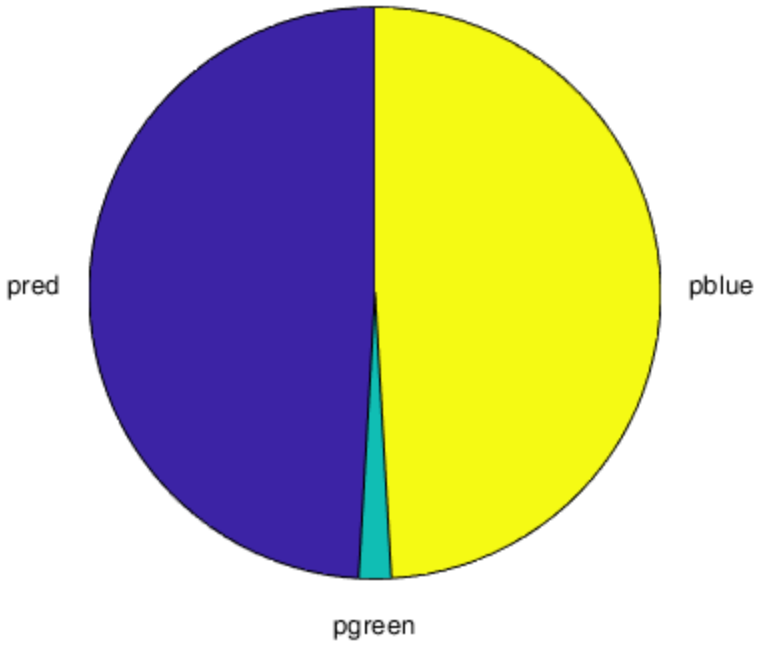

pblue
pred
pgreen


pblue
pred
pgreen


pblue
pred
pgreen

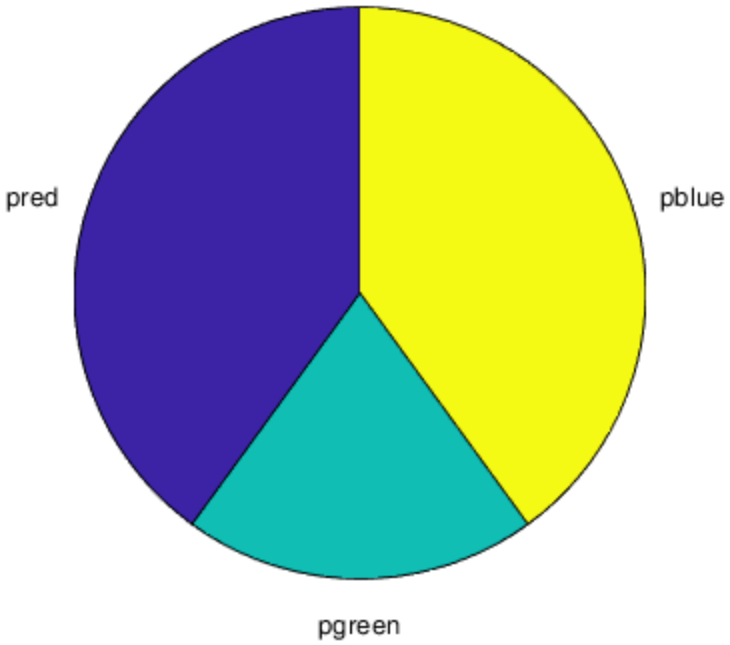

pblue
pred
pgreen

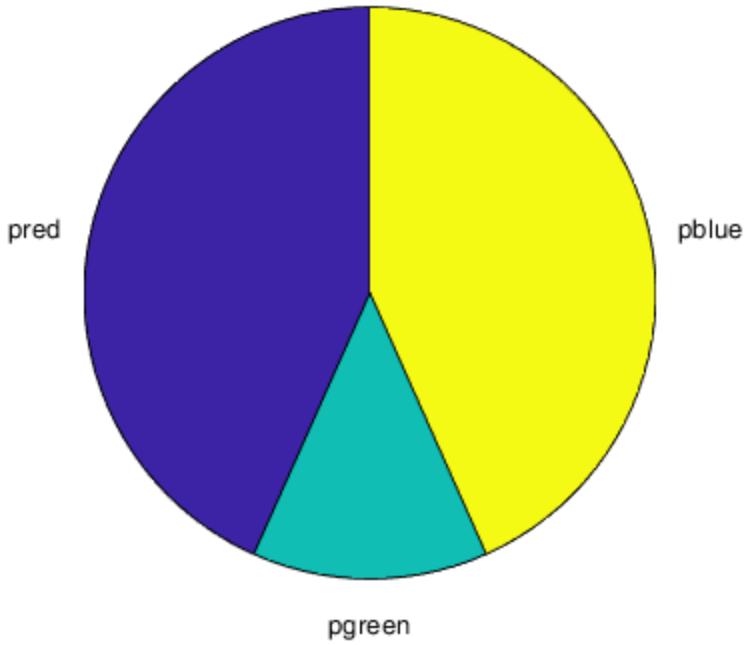
```
count =

     6


Y =

     0     0     1     1     1
     0     1     0     0     1
     1     1     1     0     0


dParameter1 =

    0.0098
```

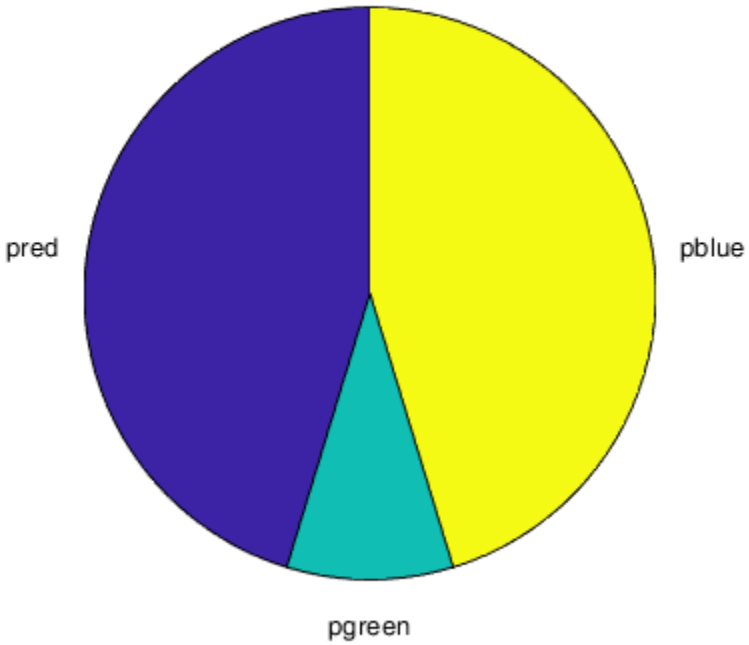


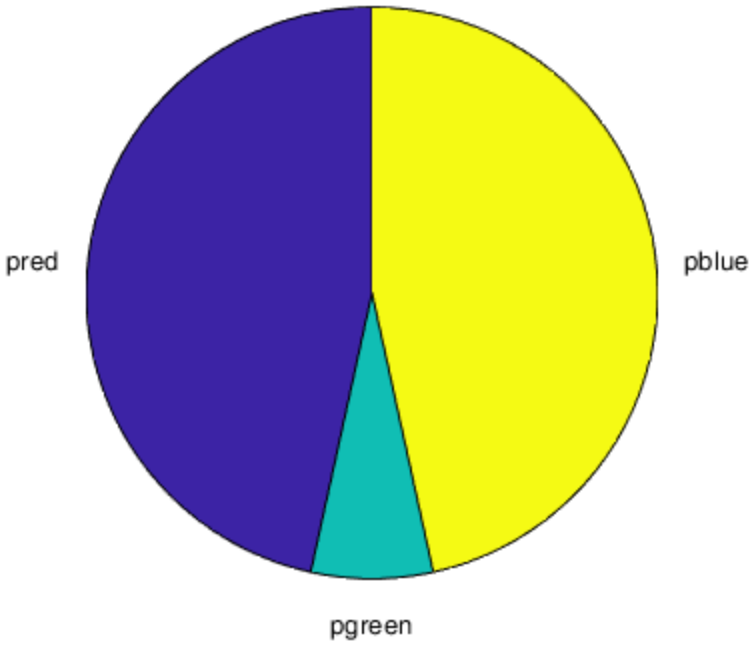

```
count =

     9


Y =
```

```
0     0     1     1     1
0     1     0     0     1
1     1     1     0     0
```

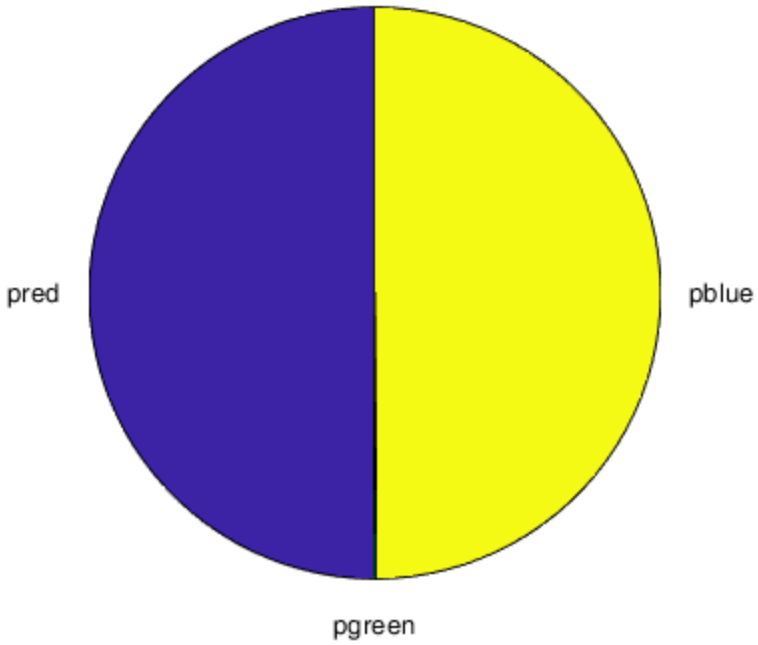



pred
pblue
pgreen


pred
pblue
pgreen




```
dParameter1 =

   9.1550e-04
```




```
count =

    19
```

Section (1D)

New likelihood function: L(pred,pgreen,pblue)= PI^7/i=1(3Ek=1 yk,iPk) New log likelihood. The new expectation is the sum of the logs of the likelihoods of the data appearing given the hidden matrix and the parameters pred, pgreen and pblue. E = log(pgreen + pblue) + log(pred + pblue)+ log(pred) + log(pred +pgreen)+log(pred+pgreen+pblue) + log(pgreen) where pred+pgreen+pblue = 1. The new likelihood function is still concave since it converges to a local maximum.

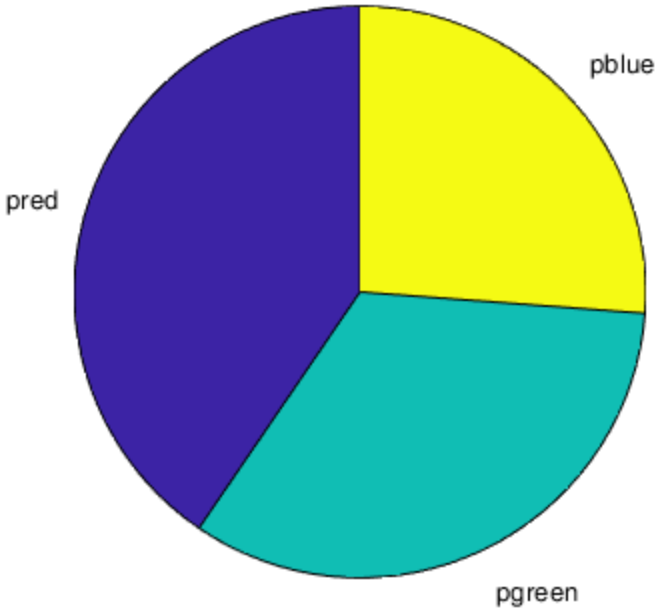
Section(1E)

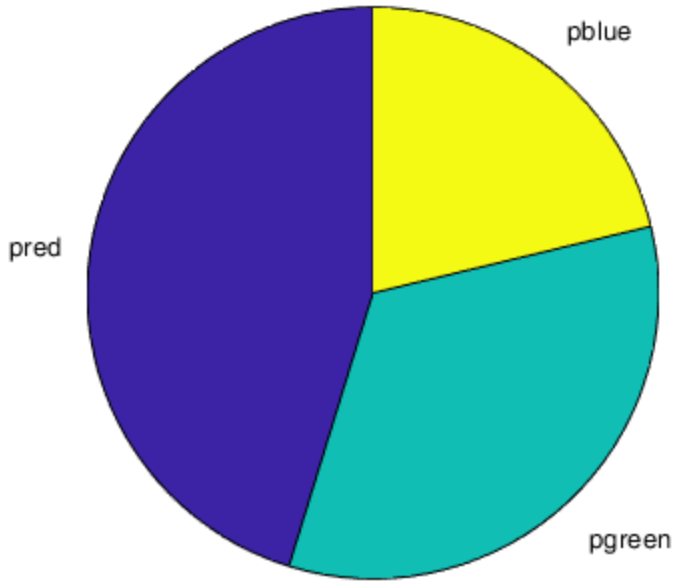
```
Y = [1 0 1 1 1 1 0;0 1 0 0 1 1 1;1 1 1 0 0 1 0]
readA= Y(:,1)/sum(Y(:,1));
readB= Y(:,2)/sum(Y(:,2));
readC= Y(:,3)/sum(Y(:,3));
readD= Y(:,4)/sum(Y(:,4));
readE= Y(:,5)/sum(Y(:,5));
readF= Y(:,6)/sum(Y(:,6));
readG= Y(:,7)/sum(Y(:,7));
INTreadA= Y(:,1)/sum(Y(:,1));
```

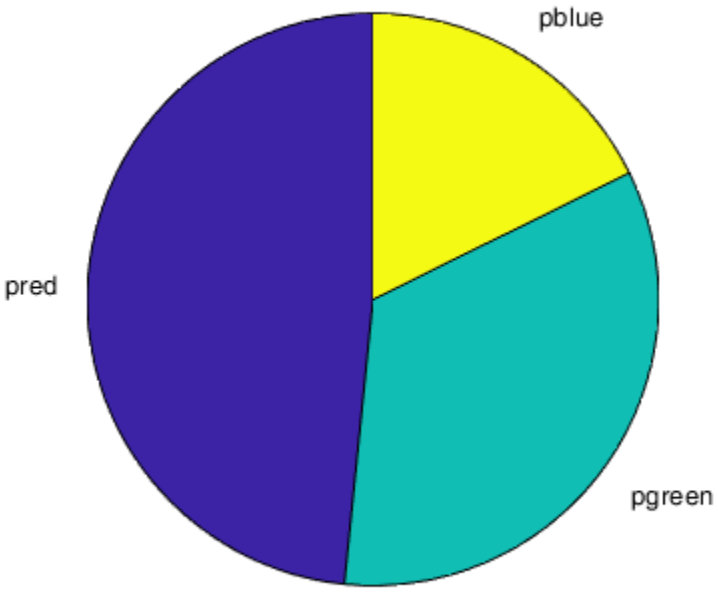
```
INTreadB= Y(:,2)/sum(Y(:,2));
INTreadC= Y(:,3)/sum(Y(:,3));
INTreadD= Y(:,4)/sum(Y(:,4));
INTreadE= Y(:,5)/sum(Y(:,5));
INTreadF= Y(:,6)/sum(Y(:,6));
INTreadG= Y(:,7)/sum(Y(:,7));
count = 0;
dParameter1 = 1;
while dParameter1 > 0.001
    iterA = readA;
    iterB = readB;
    iterC = readC;
    iterD = readD;
    iterE = readE;
    iterF = readF;
    iterG = readG;
    %Single iteration prep (M step)
    totalChance =
 (readA(1,:)+readB(1,:)+readC(1,:)+readD(1,:)+readE(1,:)+readF(1,:)+readG(1,:))+(r
    pred =
 (readA(1,:)+readB(1,:)+readC(1,:)+readD(1,:)+readE(1,:)+readF(1,:)+readG(1,:))/
totalChance;
    pgreen =
 (readA(2,:)+readB(2,:)+readC(2,:)+readD(2,:)+readE(2,:)+readF(2,:)+readG(2,:))/
totalChance;
    pblue =
 (readA(3,:)+readB(3,:)+readC(3,:)+readD(3,:)+readE(3,:)+readF(3,:)+readG(3,:))/
totalChance;
    %Set new read percentag
    readA(1,:) = INTreadA(1,:)*pred;
    readB(1,:) = INTreadB(1,:)*pred;
    readC(1,:) = INTreadC(1,:)*pred;
    readD(1,:) = INTreadD(1,:)*pred;
    readE(1,:) = INTreadE(1,:)*pred;
    readF(1,:) = INTreadF(1,:)*pred;
    readG(1,:) = INTreadG(1,:)*pred;
    readA(2,:) = INTreadA(2,:)*pgreen;
    readB(2,:) = INTreadB(2,:)*pgreen;
    readC(2,:) = INTreadC(2,:)*pgreen;
    readD(2,:) = INTreadD(2,:)*pgreen;
    readE(2,:) = INTreadE(2,:)*pgreen;
    readF(2,:) = INTreadF(2,:)*pgreen;
    readG(2,:) = INTreadG(2,:)*pgreen;
    readA(3,:) = INTreadA(3,:)*pblue;
    readB(3,:) = INTreadB(3,:)*pblue;
    readC(3,:) = INTreadC(3,:)*pblue;
    readD(3,:) = INTreadD(3,:)*pblue;
    readE(3,:) = INTreadE(3,:)*pblue;
    readF(3,:) = INTreadF(3,:)*pblue;
    readG(3,:) = INTreadG(3,:)*pblue;
    %Normalize read percentages
    readA = readA/sum(readA);
    readB = readB/sum(readB);
    readC = readC/sum(readC);
```

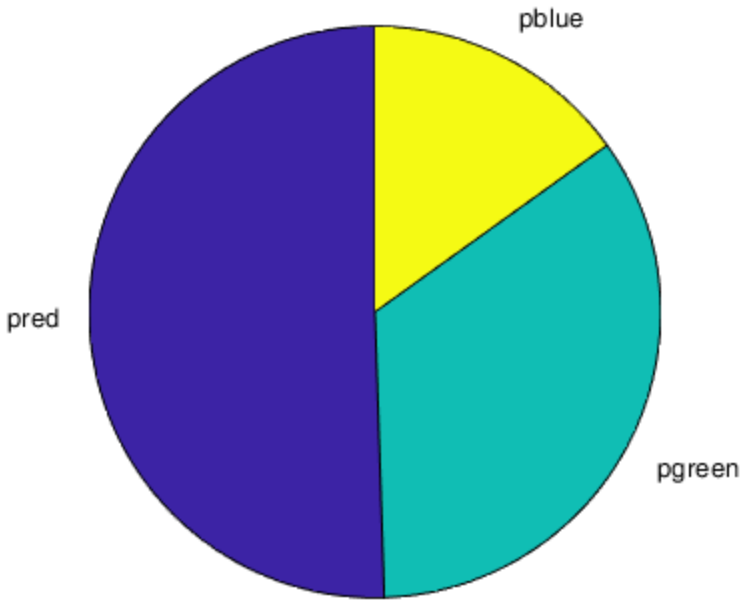
```
    readD = readD/sum(readD);
    readE = readE/sum(readE);
    readF = readF/sum(readF);
    readG = readG/sum(readG);
    %change in parameter
    dParaMatrix = zeros(3,7);
    dParaMatrix(1,1) = abs(readA(1,:)-iterA(1,:));
    dParaMatrix(2,1) = abs(readA(2,:)-iterA(2,:));
    dParaMatrix(3,1) = abs(readA(3,:)-iterA(3,:));
    dParaMatrix(1,2) = abs(readB(1,:)-iterB(1,:));
    dParaMatrix(2,2) = abs(readB(2,:)-iterB(2,:));
    dParaMatrix(3,2) = abs(readB(3,:)-iterB(3,:));
    dParaMatrix(1,3) = abs(readC(1,:)-iterC(1,:));
    dParaMatrix(2,3) = abs(readC(2,:)-iterC(2,:));
    dParaMatrix(3,3) = abs(readC(3,:)-iterC(3,:));
    dParaMatrix(1,4) = abs(readD(1,:)-iterD(1,:));
    dParaMatrix(2,4) = abs(readD(2,:)-iterD(2,:));
    dParaMatrix(3,4) = abs(readD(3,:)-iterD(3,:));
    dParaMatrix(1,5) = abs(readE(1,:)-iterE(1,:));
    dParaMatrix(2,5) = abs(readE(2,:)-iterE(2,:));
    dParaMatrix(3,5) = abs(readE(3,:)-iterE(3,:));
    dParaMatrix(1,6) = abs(readF(1,:)-iterF(1,:));
    dParaMatrix(2,6) = abs(readF(2,:)-iterF(2,:));
    dParaMatrix(3,6) = abs(readF(3,:)-iterF(3,:));
    dParaMatrix(1,7) = abs(readG(1,:)-iterG(1,:));
    dParaMatrix(2,7) = abs(readG(2,:)-iterG(2,:));
    dParaMatrix(3,7) = abs(readG(3,:)-iterG(3,:));
    dParameterRow = max(dParaMatrix);
    dParameter1 = max(dParameterRow);
    count = count + 1;
    if count < 5
        X = [pred pgreen pblue];
        labels = {'pred','pgreen','pblue'};
        figure
        pie(X,labels)
        snapnow
    end
end
dParameter1
pred
pgreen
pblue
count
%system converges from multiple intializations to minimize pblue.
 starting
%with green and blue at 50/50 instead of 3/3rds causes convergence to
 10%
%blue instead of 0.2%. red/blue 50/50 similarly converges blue to only
 1%.


Y =

     1     0     1     1     1     1     0
```

```
0     1     0     0     1     1     1
1     1     1     0     0     1     0
```




pblue
pred
pgreen


pblue
pred
pgreen




```
dParameter1 =

   9.6863e-04

pred =

    0.5935

pgreen =

    0.3950

pblue =

    0.0115

count =

    48
```

Section (1F)

i.Likelihood: L(Theta | X) = Prob(X=x| Theta = read column/sum(read))

```
%Log Likelihood: pkt = [1400Sumi=1 Zkit]/[3Sumj=1 1400Sumi=1 Zjit]
% Zkit = Yki*pk(t-1)/[3Sumk=1 yki*pk(t-1)]
% Zjit = Yji*pj(t-1)/[3Sumj=1 yji*pj(t-1)]
aReads = [1; 1; 1];
aReads = repmat(aReads,1,100);
bReads = [0; 1; 1];
bReads = repmat(bReads,1,150);
cReads = [1; 0; 1];
cReads = repmat(cReads,1,200);
dReads = [1; 0; 0];
dReads = repmat(dReads,1,250);
eReads = [1; 1; 0];
eReads = repmat(eReads,1,300);
gReads = [0; 1; 0];
gReads = repmat(gReads,1,400);

Y = [aReads bReads cReads dReads eReads gReads];
dY = Y;
count = 0;
dP = 1;
while dP > 0.001
    iterY = dY;
    totalchance = sum(dY,'all');
    pred = sum(dY(1,:),'all')./totalchance;
    pgreen = sum(dY(2,:),'all')./totalchance;
    pblue = sum(dY(3,:),'all')./totalchance;
    pChart = [pred pgreen pblue];
    for k = 1:3
        dY(k,:) = Y(k,:).*pChart(k);
    end
    for i = 1:1400
            dY(:,i) = dY(:,i)./sum(dY(:,i),'all');
    end
    dParaMatrix = zeros(3,1400);
    for k = 1:3
        for i = 1:1400
            dParaMatrix(k,i) = abs(dY(k,i)-iterY(k,i));
        end
    end
    dParameterRow = max(dParaMatrix);
    dP = max(dParameterRow);
    count = count + 1;
end
pChart
dP
count
% ii. The results are similar as the single read sample. This likely
 comes about from a lack of blue only reads, so the likelihood
 function is drawn over times to more prevalent readings.
```

```
% iii. The number of reads does not increase operational complexity.
 While
% the structure of the code would give the appearance of going from
 O(n) to
% O(n^2), both were O(n^2) in order to make the parameter difference
% matrix.
```

```
pChart =

    0.4497    0.5499    0.0004
```

```
dP =

   5.9756e-04
```

```
count =

    12
```

Published with MATLAB® R2020a