# Sample Classification With Taxonomy

Given a taxonomy we seek to classify samples. The most common scenario is that of classifying patients as either healthy or having a disease. Only certain datasets had class labels for samples available. Here we use the Segerstolpe data which labels samples as either having type 2 diabetes or being healthy. Since the Segerstolpe dataset only included 10 real samples, we randomly generated 20 samples (10 from each class) using the original dataset.

## Build Dendrogram

We start off with the overall similarity matrix, $\mathbf{B}$. This $\mathbf{B}$ was constructed using OT-RMC. See end of document to find code that builds $\mathbf{B}$. See OTRMC_SA.mlx for details on the construction of $\mathbf{B}$.

```
A = real(-log(Biggammaij)); %Biggammaij is our overall similarity matrix
A= A/ max(max(A));
B = squareform(A);
tree=linkage(B,'ward');
cellClusters=cluster(tree,"MaxClust", numcells); %This generates the metaclusters
```

## Arrange Cluster Proportions

We use the meta-clusters to build a proportion matrix which will use to predict a categorical class for each sample. This proportion matrix will have as many rows as the number of samples available and as many columns as the number of metaclusters. The entries in column $i$ represent the proportions of the clusters that belong to metacluster $i$.

```
%predCellMat is a matrix of indices. For each row (a sample) we have the
%metacluster label for all of its clusters
predCellMat = zeros(numPat,numcells);
for i = 1:numPat
    start = sum(stride(1:i-1))+1;
    predCellMat(i,1:stride(i))=cellClusters(start:start+stride(i)-1);
end

%We star
propsmatNew = zeros(numPat, numcells);
for i=1:numPat
    start = sum(stride(1:i-1))+1;
    props=ww(start:start+stride(i)-1);
    patclusters=predCellMat(i,:); %these are the metacluster labels for clusters in sample i
    propindex = 0;
    for j=1:numcells
        if ismember(patclusters(j), 1:numcells) ==1 %if cluster belongs to a metacluster. False
            propindex = propindex+1;
            if propsmatNew(i,patclusters(j))==0
                propsmatNew(i,patclusters(j)) = props(propindex);
            else %ie. two clusters within same sample are assigned to same metacluster.
                propsmatNew(i,patclusters(j)) = propsmatNew(i,patclusters(j))+props(propindex);
            end
        end
```

```
    end
end
```

## Prediction

We can now implement Random Forest:

```
%Y= Y-1; %Only needed for vector with 2s and 1s
propsmat=propsmatNew;
label2=zeros(numPat,1);
score2=zeros(numPat,1);
for i=1:numPat
    testpropsmat=propsmat(i,:);
    trainpropsmat=propsmat([1:(i-1),(i+1):numPat], :);
    trainY=Y([1:(i-1),(i+1):numPat]);
    rf = TreeBagger(150, trainpropsmat, trainY, 'OOBPrediction', 'on', 'MinLeafSize', 1);  %ran
    [lb,s] = predict(rf,testpropsmat);
    score2(i)=s(2); % posterior for class 1
    label2(i)=str2num(lb{1});
end
% classification accuracy
accu2=sum(label2==Y)/length(Y);
% AUC
[xa,ya,ta,aucv]=perfcurve(Y,score2,1); % aucv1 is Area Under Curve value,
%ta1 is the threshold at which xa1, ya1 are obtained
aucv
```

```
aucv = 0.9900
```

```
accu2
```

```
accu2 = 0.9000
```