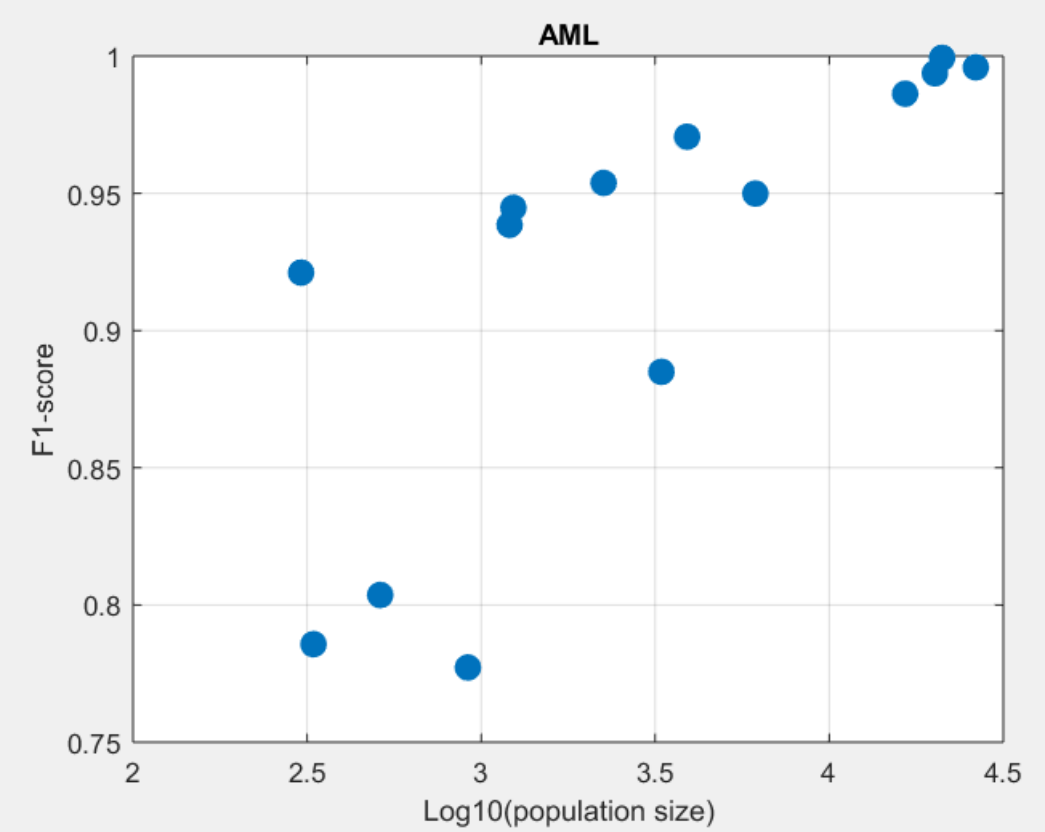
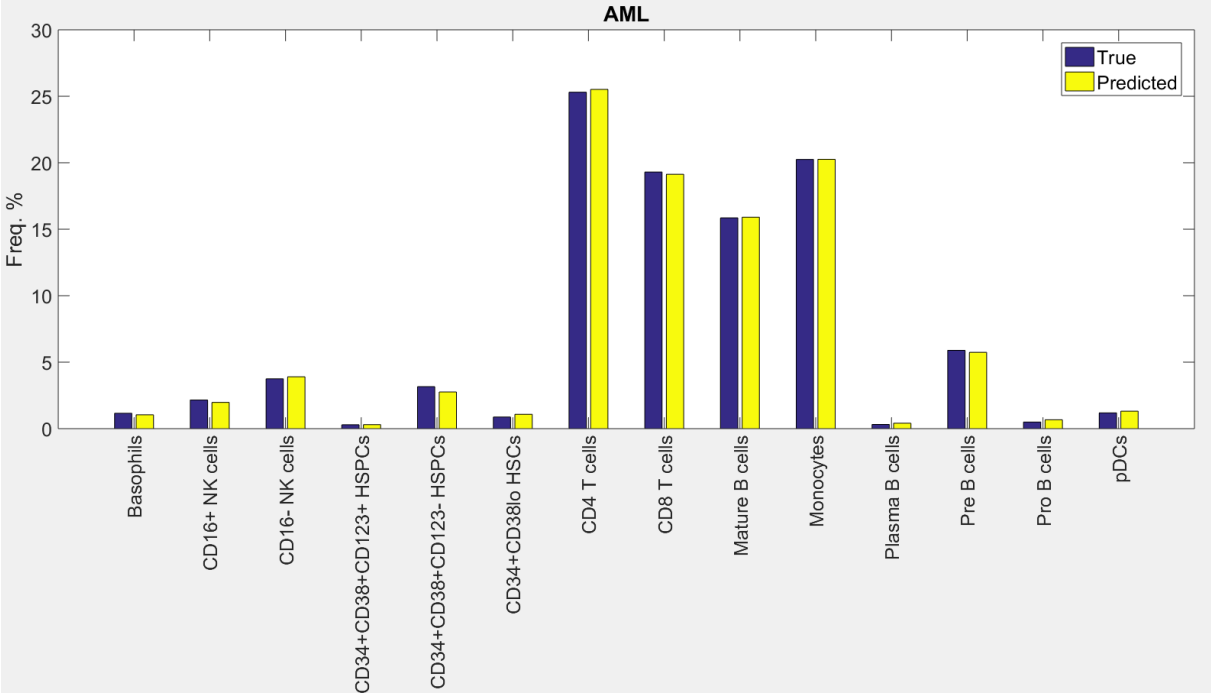
AML Dataset Classification

```
%% Read the Data and Preprocess

>> DataTable = readtable('AML_benchmark.csv');

% remove unneeded columns
>> DataTable.Time=[];
>> DataTable.Cell_length=[];
>> DataTable.DNA1=[];
>> DataTable.DNA2=[];
>> DataTable.Viability=[];
>> DataTable.file_number=[];
>> DataTable.event_number=[];
>> DataTable.subject=[];

% Separate Data points and Labels
>> Labels=DataTable.cell_type;
>> DataTable.cell_type=[];
>> Data = table2array(DataTable);
>> clear DataTable

% clear NotDebrisSinglets
>> Data(strcmp('NotDebrisSinglets',Labels),:)=[];
>> Labels(strcmp('NotDebrisSinglets',Labels))=[];

% Apply arcsinh5 transformation
>> Data=asinh((Data-1)/5);

%% run LDA Classifier with 5-fold cross-validation
>> CVO = cvpartition(Labels,'k',5);
>> Accuracy = zeros(CVO.NumTestSets,1);
>> training_time = zeros(CVO.NumTestSets,1);
>> testing_time = zeros(CVO.NumTestSets,1);
>> CellTypes = unique(Labels);
>> ConfusionMat = zeros(length(CellTypes));
>> for i = 1:CVO.NumTestSets
>>     trIdx = CVO.training(i);
>>     teIdx = CVO.test(i);
>>     tic
>>     classificationLDA = fitcdiscr(...
>>         Data(trIdx,:), ...
>>         Labels(trIdx));
>>     training_time(i)=toc;          %in seconds
>>
>>     tic
>>     Predictor = predict(classificationLDA,Data(teIdx,:));
>>     testing_time(i)=toc;           %in seconds
>>     Accuracy(i) =nnz(strcmp(Predictor,Labels(teIdx))) ...
       /size(Labels(teIdx),1);
>>     ConfusionMat = ConfusionMat + ...
       confusionmat(Labels(teIdx),Predictor,'order',CellTypes);
```

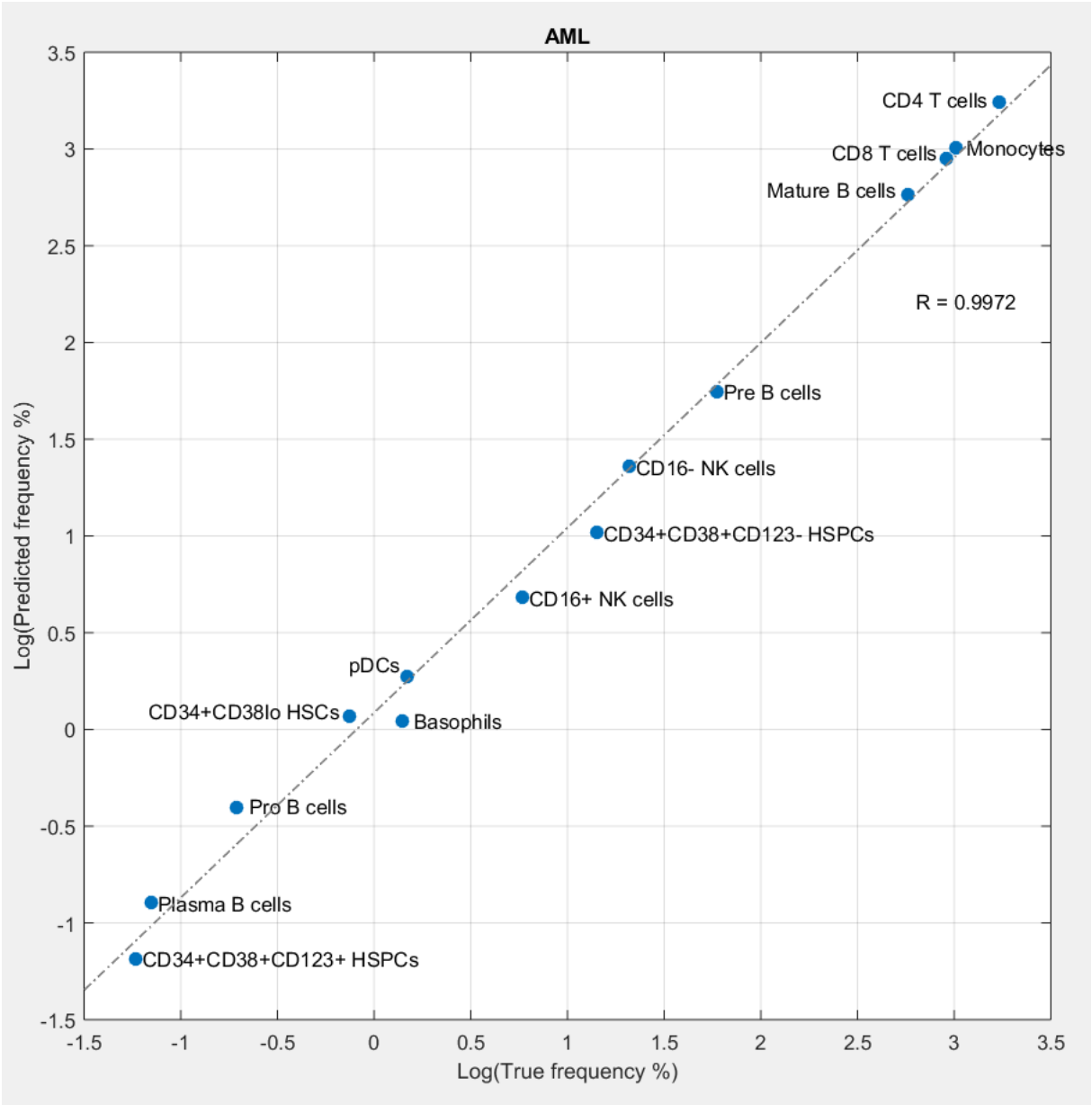
```
>> end
>> Total_time = sum(training_time)+sum(testing_time);
>> training_time = mean(training_time);
>> testing_time = mean(testing_time);
>> cvAcc = mean(Accuracy)*100;
>> cvSTD = std(Accuracy)*100;
>> disp(['LDA Accuracy = ' num2str(cvAcc) ' ' char(177) ' '...
   num2str(cvSTD) ' %'])

LDA Accuracy = 98.1437 ± 0.018051 %

>> clear i Predictor classificationLDA trIdx teIdx CVO Accuracy

%% Performance evaluation
% F1 measure
>> Precision = diag(ConfusionMat)./sum(ConfusionMat,1)';
>> Recall = diag(ConfusionMat)./sum(ConfusionMat,2);
>> Fmeasure = 2 * (Precision.*Recall)./(Precision+Recall);
>> MedianFmeasure = median(Fmeasure);
>> Subset_size = sum(ConfusionMat,2);
>> WeightedFmeasure = (Subset_size./size(Data,1))'*Fmeasure;

>> disp(['Median F1-score = ' num2str(MedianFmeasure)])

Median F1-score = 0.94711

>> figure,scatter(log10(Subset_size),Fmeasure,100,'filled')
>> title('AML')
>> xlabel('Log10(population size)'),ylabel('F1-score')
>> box on, grid on
```




```
%% Population Frequency
>> True_Freq = sum(ConfusionMat,2)./sum(sum(ConfusionMat));
>> Predicted_Freq = sum(ConfusionMat,1)'./sum(sum(ConfusionMat));
>> Max_Freq_diff = max(abs(True_Freq-Predicted_Freq))*100;

>> disp(['delta_f = ' num2str(Max_Freq_diff)])

delta_f = 0.40121

>> figure,bar([True_Freq*100 Predicted_Freq*100])
>> xticklabels(CellTypes)
>> xtickangle(90)
>> set(gca,'FontSize',15)
>> legend({'True','Predicted'},'FontSize',15)
>> legend show
>> ylabel('Freq. %'),title('AML')
```




```
%% Population Frequency scatter plot

>> X=log(True_Freq*100);
>> Y=log(Predicted_Freq*100);
>> figure,scatter(X,Y,50,'filled')
>> box on, grid on
>> xlabel('Log(True frequency %)')
>> ylabel('Log(Predicted frequency %)')
>> title('AML')
>> for k=1:length(CellTypes)
>>     text(X(k),Y(k),CellTypes{k})
>> end
>> lsline
>> text(0,0,['R = ' num2str(corr(X,Y))])
```


AML
CD4 T cells
CD8 T cells
Monocytes
Mature B cells
R = 0.9972
Pre B cells
CD16- NK cells
CD34+CD38+CD123- HSPCs
CD16+ NK cells
pDCs
CD34+CD38lo HSCs
Basophils
Pro B cells
Plasma B cells
CD34+CD38+CD123+ HSPCs
Log(Predicted frequency %)
Log(True frequency %)
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3
3.5