

Multi-Center Dataset Classification

```
%% Read the Data and Preprocess

>> VarNames = {'CCR6', 'CD20', 'CD45', 'CD14', 'CD16', 'CD8', ...
    'CD3', 'CD4'};
>> SamplesData=struct('Data', [], 'Labels', {});

>> H=dir(fullfile('Samples\', '*.csv'));
>> SamplesFiles = cellstr(char(H(1:end).name));

>> H=dir(fullfile('Labels\', '*.csv'));
>> LabelsFiles = cellstr(char(H(1:end).name));
>> clear H

>> for i=1:length(SamplesFiles)
>>     SamplesData(i).Data = csvread(['Samples\' SamplesFiles{i}]);
>>     SamplesData(i).Labels = csvread(['Labels\' LabelsFiles{i}]);
>> end
>> clear i SamplesFiles LabelsFiles

>> Labels = [];
>> for i=1:length(SamplesData)
>>     % Apply arcsinh5 transformation
>>     SamplesData(i).Data = asinh((SamplesData(i).Data-1)/5);
>>     Labels = [Labels; SamplesData(i).Labels];
>> end
>> clear i

%% run LDA Classifier with 4-fold cross-validation on samples

>> CVO = cvpartition(1:1:16, 'k', 4);
>> Accuracy = zeros(length(SamplesData), 1);
>> training_time = zeros(CVO.NumTestSets, 1);
>> testing_time = zeros(length(SamplesData), 1);
>> CellTypes = unique(Labels);
>> ConfusionMat = zeros(length(CellTypes));
>> WeightedFmeasure = zeros(length(SamplesData), 1);
>> for i = 1:CVO.NumTestSets
>>     trIdx = find(CVO.training(i));
>>     teIdx = find(CVO.test(i));

>>     DataTrain=[];
>>     LabelsTrain=[];
>>     for j=1:length(trIdx)
>>         DataTrain = [DataTrain; SamplesData(trIdx(j)). ...
            Data(SamplesData(trIdx(j)).Labels~=0, :)];
>>         LabelsTrain = [LabelsTrain; SamplesData(trIdx(j)). ...
            Labels(SamplesData(trIdx(j)).Labels~=0)];
>>     end
>>     clear j
```

```

>> tic
>> classificationLDA = fitcdiscr(...
    DataTrain, ...
    LabelsTrain);
>> training_time(i)=toc;           %in seconds

>> for j=1:length(teIdx)
>>     tic
>>     [Predictor,scores] = predict(classificationLDA, ...
        SamplesData(teIdx(j)).Data);
>>     Current_Scores = max(scores,[],2);
>>     Predictor(Current_Scores < 0.4)=0;
>>     testing_time(teIdx(j))=toc;           %in seconds
>>     Accuracy(teIdx(j)) = nnz(Predictor(SamplesData ...
        (teIdx(j)).Labels~=0)==SamplesData(teIdx(j)). ...
        Labels(SamplesData(teIdx(j)).Labels~=0)) ...
        /size(SamplesData(teIdx(j)).Labels(SamplesData ...
        (teIdx(j)).Labels~=0),1);
>>     ConfusionMat = ConfusionMat + confusionmat(SamplesData...
        (teIdx(j)).Labels,Predictor,'order',CellTypes);
>> end
>> clear j
>> 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 = 98.4426 ± 1.6561 %

```

>> clear i Predictor classificationLDA trIdx teIdx CVO Accuracy
    DataTrain LabelsTrain

```

```

%% Performance evaluation

```

```

>> col1 = ConfusionMat(2:end,1);
>> ConfusionMat = ConfusionMat(2:end,2:end);
% F1 measure
>> Precision = diag(ConfusionMat)./sum(ConfusionMat,1)';
>> Recall = diag(ConfusionMat)./(sum(ConfusionMat,2)+col1);
>> Fmeasure = 2 * (Precision.*Recall)./(Precision+Recall);
>> MedianFmeasure = median(Fmeasure);
>> Subset_size = sum(ConfusionMat,2)+col1;
>> WeightedFmeasure = (Subset_size./sum(Subset_size))*Fmeasure;

>> disp(['Weighted F1-score = ' num2str(WeightedFmeasure)])

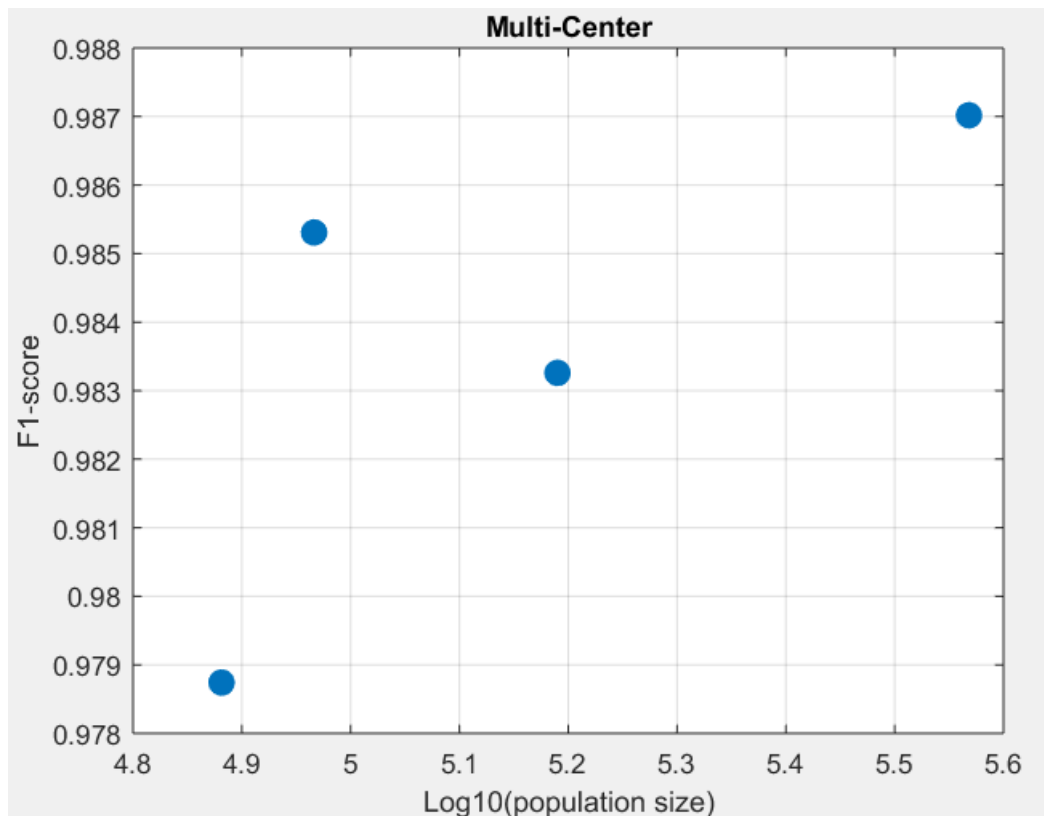
```

Weighted F1-score = 0.98504

```

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

```



```
%% Population Frequency
```

```
>> True_Freq = (sum(ConfusionMat,2)+col1)./ ...
    (sum(sum(ConfusionMat))+sum(col1));
>> Predicted_Freq = sum(ConfusionMat,1)'./ ...
    (sum(sum(ConfusionMat))+sum(col1));
>> Max_Freq_diff = max(abs(True_Freq-Predicted_Freq))*100;
```

```
delta_f = 0.83083
```

```
>> figure,bar([True_Freq*100 Predicted_Freq*100])
>> xticklabels({'B Cells','CD4+ T Cells', ...
    'CD8+ T Cells','Monocytes'})
>> set(gca,'FontSize',20)
>> legend({'True','Predicted'},'FontSize',15)
>> legend show
>> ylabel('Freq. %'),title('Multi-Center')
```

