HIV life cycle and mechanism
Antiretroviral therapy

Blocking HIV replication by ART

DNA of a helper cell
T-helper cell
HIV-DNA
HIV-RNA
mRNA
integrate
HIV-DNA
integrate
reverse transcriptase
reverse transcriptase inhibitors
HIV
gp120
CD4-receptor
protease inhibitors
budding
protein synthesis construction of HIV
protease inhibitors
fission/fusion inhibitors

(after AS Fauci)
HIV-protease cleavage site

Knowledge of the mechanism of HIV protease cleavage specificity is critical to the design of specific and effective HIV inhibitors. Searching for an accurate, robust, and rapid method to correctly predict the cleavage sites in proteins is crucial when searching for possible HIV inhibitors.

Scope is to predict if a sequence of aminoacids will constitute a cleavage site.

Learning patterns in cleavage sites

Accurate prediction of **known** cleavage and non-cleavage sites

Identifying unknown sites.
Candidate sites

Possible candidate sites are represented by an **octamer** within a protein sequence.

An octamer is a sequence of 8 essential amino acids
Data

There are 2 datasets available:
- 746
- 1625

Possible sites are represented as sequence of 8 letters among 20 (‘ARNDCQEGHILKMFPSWTYV’ representing different aminoacids)

The known cleavage sites have label 1
The known non-cleavage sites have label -1
Problem 1: load the data

Octamer are in alphabetic form: they cannot be directly loaded in Matlab!!

1) Scan each line in the file
2) Extract the character sequence
3) Provide a numerical code for each aminoacid
4) Extract the cleavage label
Problem 1: load the data

% Use Matlab I/O c-like routines

% Open I/O file stream
datafile='725Data.txt';
F=fopen(datafile);

% Read one line at a time until end of file
count=0;
while(~feof(F))
    count=count+1;
    s=fgets(F);
    data(count,:)=sscanf(a,'%c%c%c%c%c%c%c,%i\n')';
    count=count+1;
end;
Code the sequences

Now you have load all data in a 725x9 matrix:
- The first 8 numbers of each rows are the ASCII code of a letter representing an aminoacid
- The last number in each row is the label
- Think of other possible numerical coding for the 20 different aminoacids that you can use
Problem 2: train a linear classifier

Design a linear classifier to predict the cleavage sites.
Evaluate the training error

1. Extract the octamere code $x_i$
2. Extract the label: $l(i)$
3. Create design matrix $D$ (adding the bias to each data point) and the label vector $L$
4. Estimate weight vector $w = D \setminus L$
5. Classify each data point
\[
\hat{l}(i) = w^T \tilde{x}_i = w^T \begin{bmatrix} 1 \\ x_i \end{bmatrix}
\]
Problem 3: estimate $Err_{CV}$

Run a 10-fold cross validation for the classification.

1) Divide the dataset into 10 folds

1) At each cross-validation iteration
   1) Use the current fold for test
   2) Use the other 9 folds for train
   3) Evaluate the classification error on the test fold
   4) Store the test error

2) Evaluate mean and standard deviation of the test error
Problem 3: randomize the folds

% Shuffle the data
r = rand(size(data,1),1);
[dummy,ind]=sort(r);
data_shuffle=data(ind,1:8);
label_shuffle=data(ind,9);

% Evaluate number of data per fold
N_fold=10;
fold_data=fix(size(data,1)/N_fold);
Problem 3: cross validate

% Cross validation
for cv=1:10
    % Find indexes of test data
    ntest=(cv-1)*N_fold+1:cv*N_fold;
    data_test=data_shuffle(ntest,:);

    % Find indexes of train data
    ind=ones(size(data,1),1);
    ind(ntest)=0;
    ntrain=find(ind);
    data_train=data_shuffle(ntrain,:);

    % Learn the classifier on the training data
    % Evaluate the error on the test data
    classifier = ...
    train_err(cv)=...
    test_err(cv)=...
end;
Problem 4: change dataset

1) Run the same cross-validation procedure on the 1625 dataset (*1625Data.txt*)

2) Run the learning on the 725 dataset and the test on the 1625 data set

3) Run the learning on the 1625 dataset and the test on the 725 data set

4) Evaluate and compare the difference errors (cross validation within the same data set, validation using the other data set)