**Needleman-Wunsch Algorithm**

def computeFMatrix(Alignment A, Alignment B, F(i,j), d):

for i=0 to len(A):

F(i,0) = d \* i

for j = 0 to len(B):

F(0,j) = d \* j

for i=1 to len(A):

for j=1 to len(B):

Match = F(i-1,j-1) + S(Ai,Bj)

Delete = F(i-1,j) + d

Insert = F(i,j-1) + d

F(i,j) = max(Match,Insert,Delete)

def NW(Alignment A, Alignment B, F(i,j)):

Alignment A = “ “

Alignment B = “ “

i = len(A)

j = len(B)

while (i > 0 or j > 0):

# Case 1: the index i on Alignment A matches the index j on # Alignment B (match)

if (i > 0 and j > 0 and F(i,j) = F(i-1,j-1) + S(Ai,Bj)):

Alignment A = Ai + Alignment A

Alignment B = Bj + Alignment B

i = i-1

j = j-1

# Case 2: the index i on Alignment A does not match

# the index j on Alignment B (mismatch)

elif (i > 0 and F(i,j) = F(i-1,j)+d):

Alignment A = Ai + Alignment A

Alignment B = “-“ + Alignment B

i = i-1

# Case 3: the index i or j on Alignments A or B

# is missing (gap/ indel)

else:

Alignment A = “-“ + Alignment A

Alignment B = Bj + Alignment B

j = j-1