ex2

April 19, 2017

In [35]:
```
import numpy as np

class HMM(object):
    def __init__(self, transition, initial, emission, states, alpha):
        self._A = transition
        self._I = initial
        self._E = emission
        self.Q = states
        self.sig = alpha
        self.smap = m

    def viterbi(self, emissions):
        emissions = map(self.smap.get)
        rows, cols = len(self.Q), len(emissions)
        probs = np.zeros(shape=(rows,cols), dtype=float)
        traceback = np.zeros(shape=(rows,cols), dtype=int)
        for i in xrange(0, rows):
            probs[i, 0] = self.E[i, x[0]] * self.I[i]
        for j in xrange(1, cols):
            for i in xrange(0, rows):
                em_p = self.E[i, x[j]]
                mx, mxi = probs[0, j-1] * self.A[0, i] * em_p, 0
                for i2 in xrange(1, rows):
                    pr = probs[i2, j-1] * self.A[i2, i] * em_p
                    if pr > mx:
                        mx, mxi = pr, i2
                probs[i, j], traceback[i, j] = mx, mxi
        omx, omxi = probs[0, cols-1], 0
        for i in xrange(1, cols):
            if probs[i, cols-1] > omx:
                omx, omxi = probs[i, cols-1], i
        i, p = omxi, [omxi]
        for j in xrange(cols-1, 0, -1):
            i = traceback[i, j]
            p.append(i)
        p = ''.join(map(lambda emissisions: self.Q[emissions], p[::-1]))
        return omx, p
```

```
In [36]: sequence = 'TTACGCGCGCGCGATATTT'

         states = ('I', 'N')
         alphabet = ('A', 'C', 'T', 'G')
         start_p = np.array([0.4, 0.6])
         trans_p = np.array([[0.8, 0.2], [0.3, 0.7]])
         emission_p = np.array([[0.1, 0.4, 0.4, 0.1],[0.3, 0.2, 0.2, 0.3]])
         hmm.viterbi(sequence)


        ---------------------------------------------------------------------------

        AttributeError                            Traceback (most recent call last)

        <ipython-input-36-64753432e877> in <module>()
          8 ma = ({'IA': 0.1, 'IC': 0.4, 'IG':0.4, 'IT':0.1}, {'NA':0.3, 'NC': 0.2,
          9 hmm = HMM(trans_p, start_p, emission_p, states, alphabet, ma)
    ---> 10 hmm.viterbi(sequence)


        <ipython-input-35-4cb66c784316> in viterbi(self, emissions)
         11
         12     def viterbi(self, emissions):
    ---> 13         emissions = map(self.smap.get)
         14         rows, cols = len(self.Q), len(emissions)
         15         probs = np.zeros(shape=(rows,cols), dtype=float)


        AttributeError: 'tuple' object has no attribute 'get'
```

model

```
In [ ]:

In [ ]:

In [ ]:
```