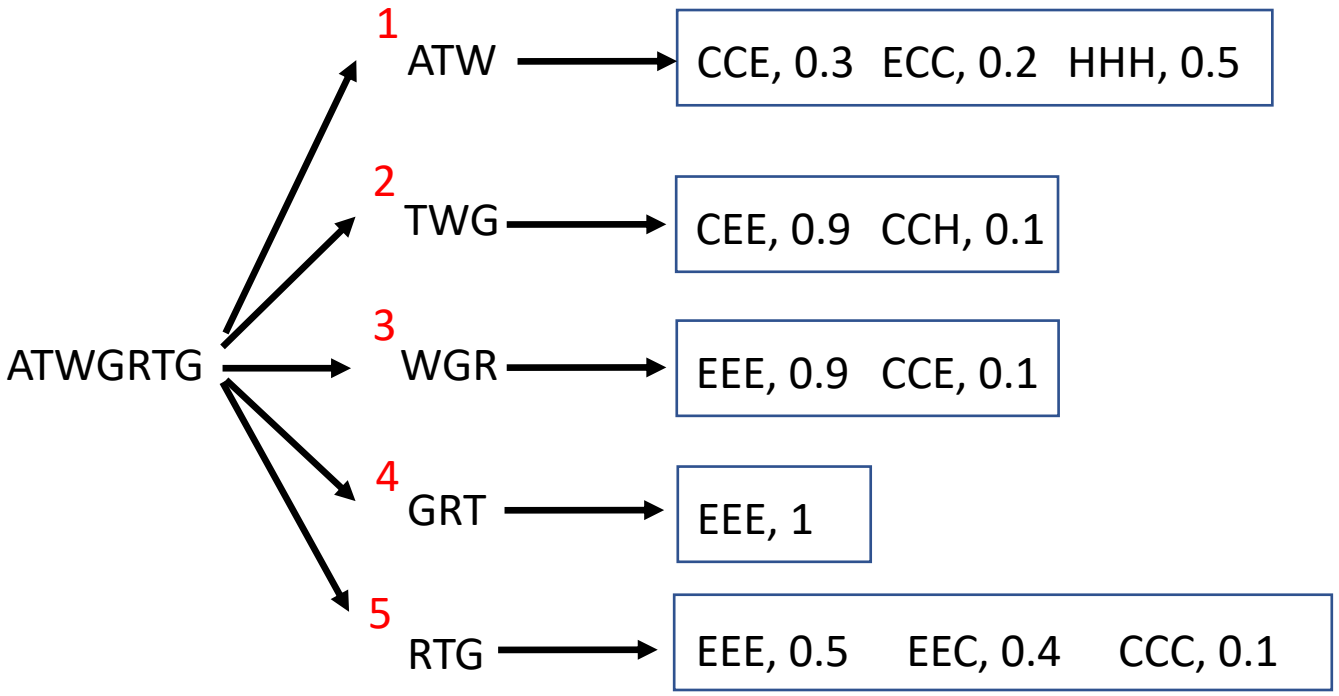
RTG → EEE, 0.5 EEC, 0.4 CCC, 0.1

seq = ATWGRTG
K = 3



Length(sequence) - (k-1) = # layers

7 - (3-1) = 5

seq = ATWGRTG
K = 3

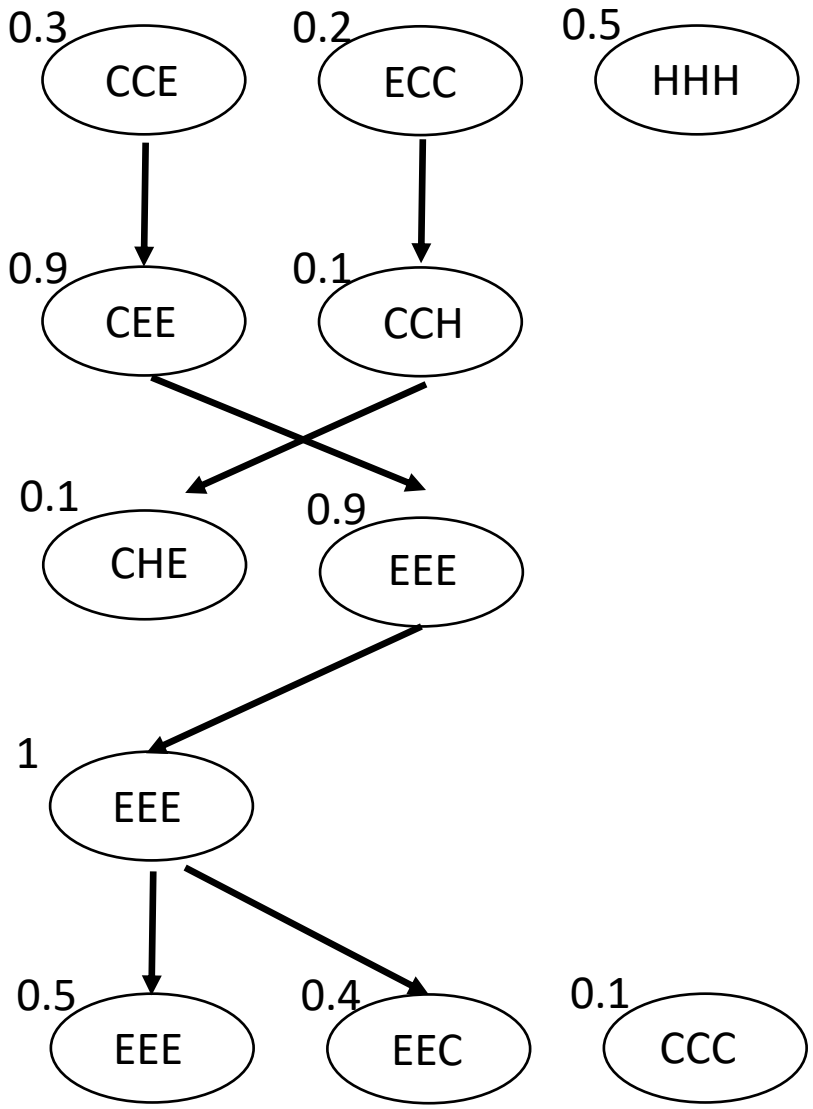
1 ATW → CCE, 0.3 ECC, 0.2 HHH, 0.5

2 TWG → CEE, 0.9 CCH, 0.1

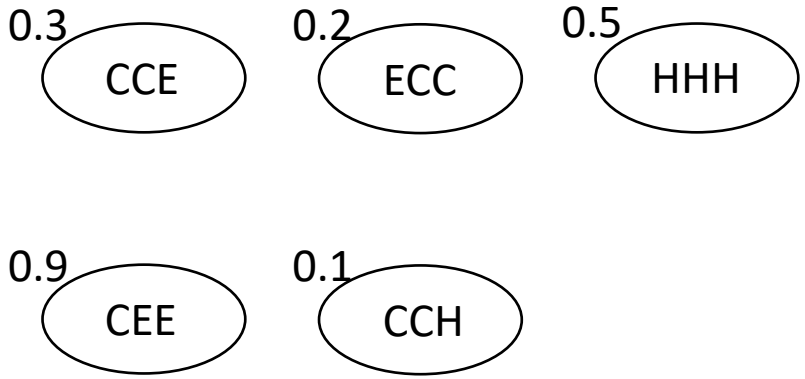
3 WGR → CHE, 0.1 EEE, 0.9

4 GRT → EEE, 1

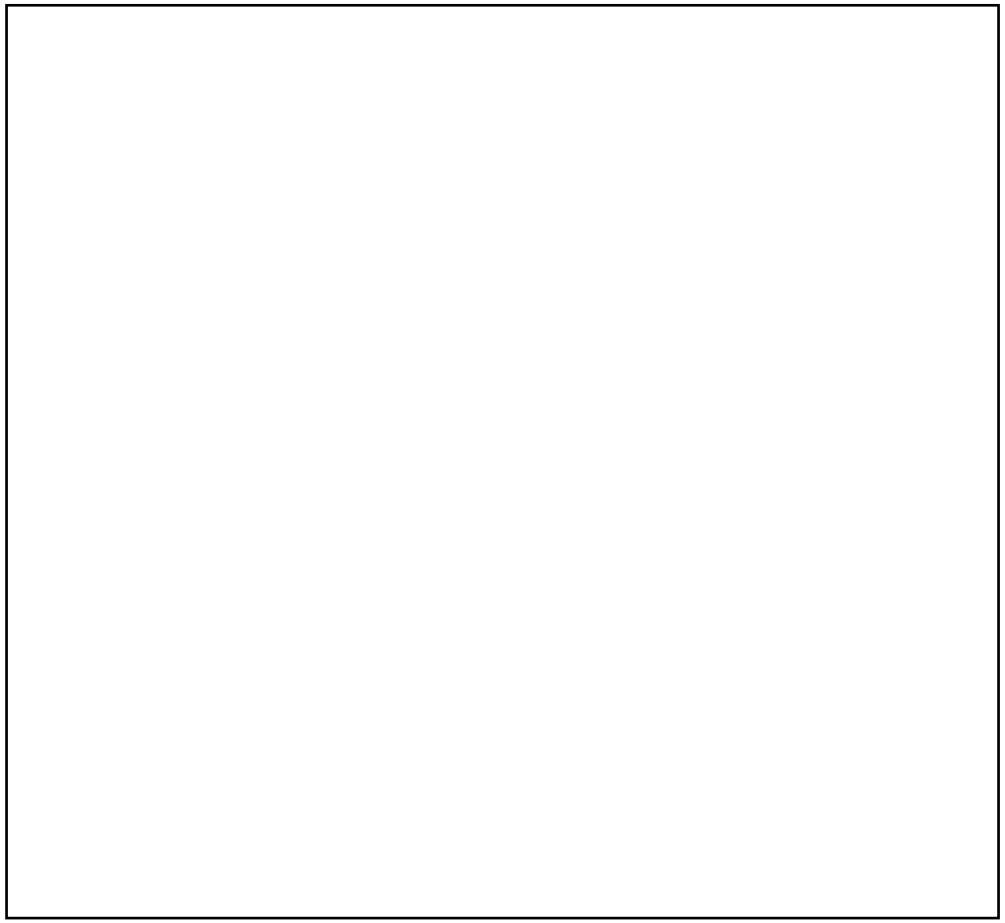
5 RTG → EEE, 0.5 EEC, 0.4 CCC, 0.1



seq = ATWGRTG
K = 3



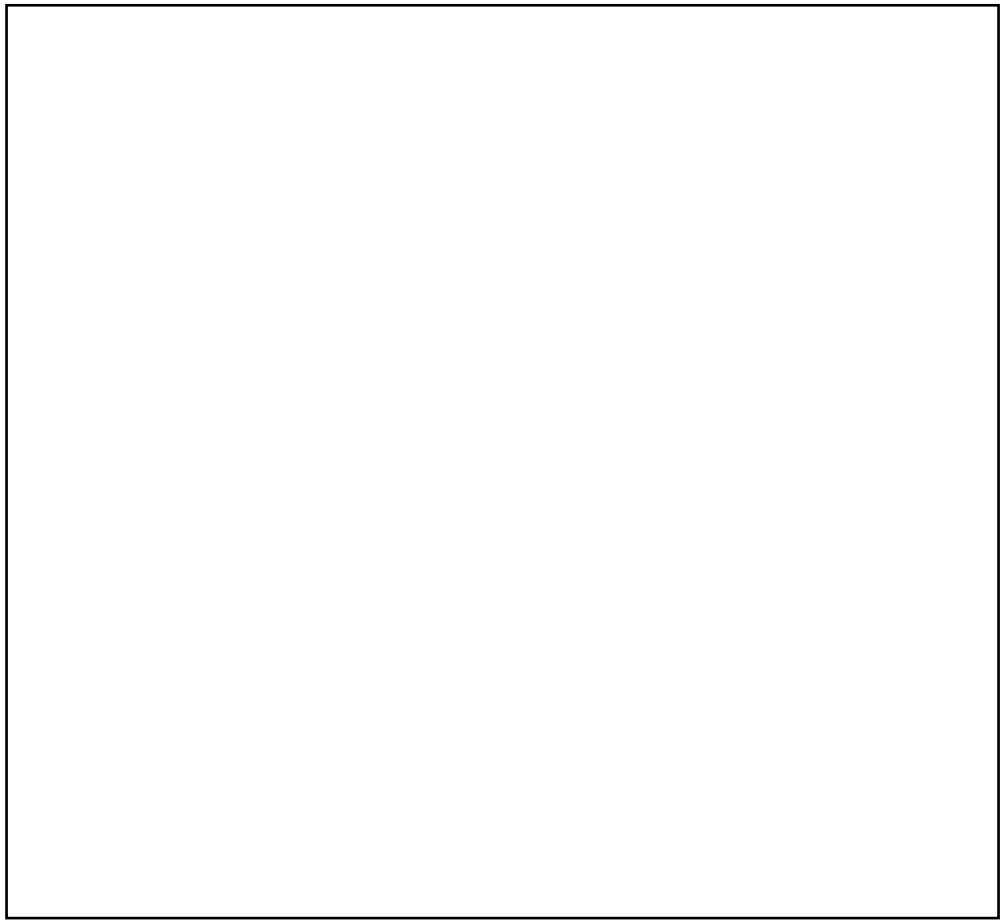
stitchextend_dict

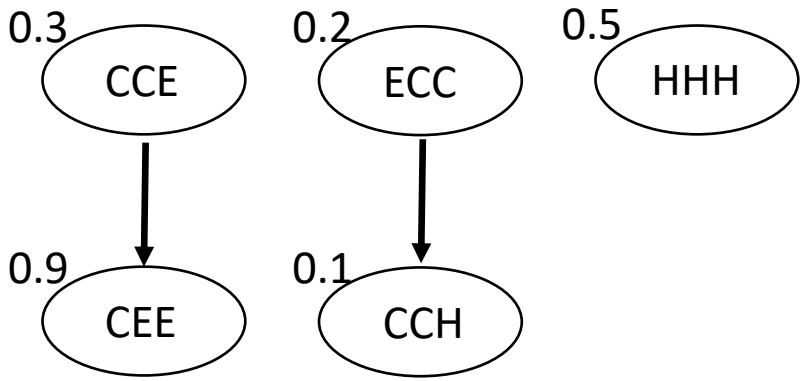


Layer 1, Layer 2

seq = ATWGRTG
K = 3

stitchextend_dict





CCE
CEE

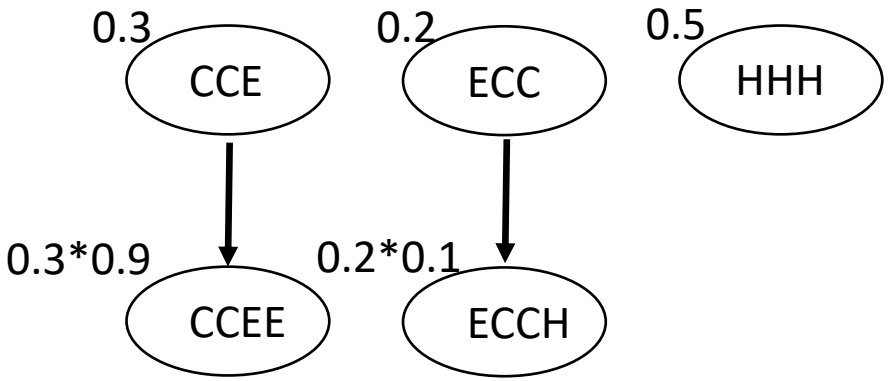
ECC
CCH

Layer 1, Layer 2

seq = ATWGRTG
K = 3

stitchextend_dict

CCEE	0.3*0.9
ECCH	0.2*0.1



CCE
CEE

ECC
CCH

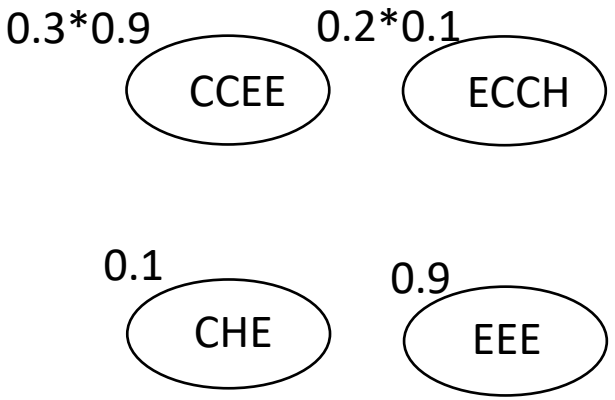
Edge Contraction

Layer 1, Layer 2

seq = ATWGRTG
K = 3

stitchextend_dict

CCEE	0.3*0.9
ECCH	0.2*0.1

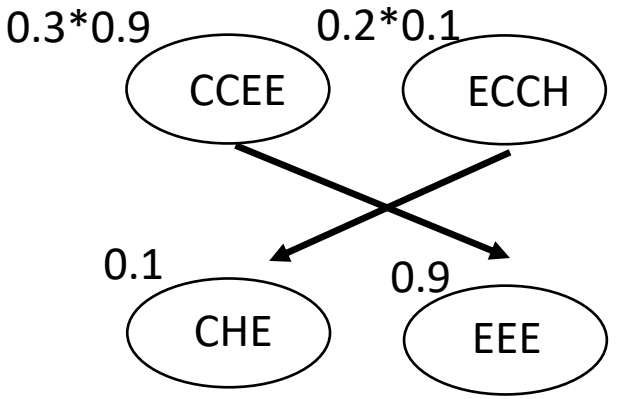


Layer 2, Layer 3

seq = ATWGRTG
K = 3

stitchextend_dict

CCEE	0.3*0.9
ECCH	0.2*0.1



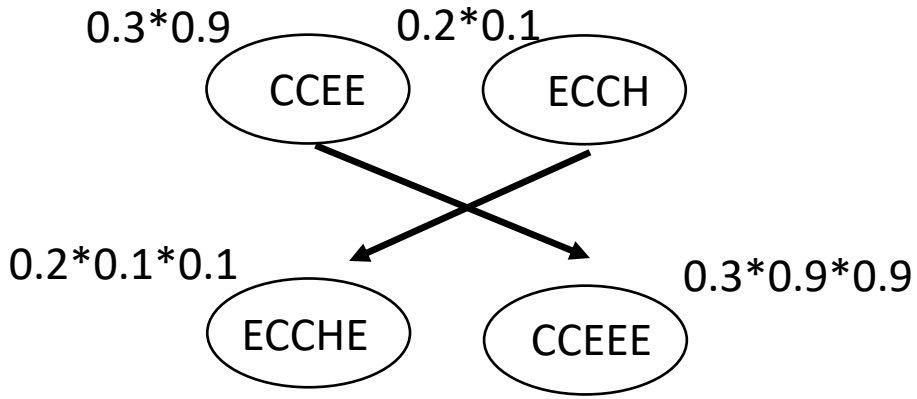
CCEE ECCH
EEE CHE

Layer 2, Layer 3

seq = ATWGRTG
K = 3

stitchextend_dict

CCEE	0.3*0.9	Del
ECCH	0.2*0.1	Del
ECCHE	0.2*0.1*0.1	Add
CCEEE	0.3*0.9*0.9	Add



CCEE
EEE

ECCH
CHE

Edge Contraction

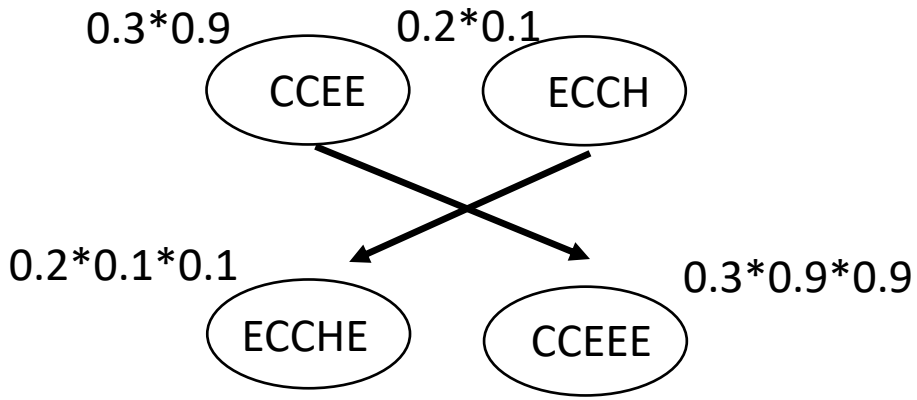
Layer 2, Layer 3

seq = ATWGRTG
K = 3

stitchextend_dict

ECCHE 0.2*0.1*0.1

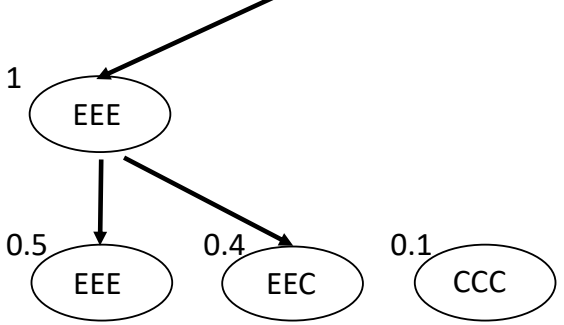
CCEEE 0.3*0.9*0.9



Edge Contraction

Layer 2, Layer 3

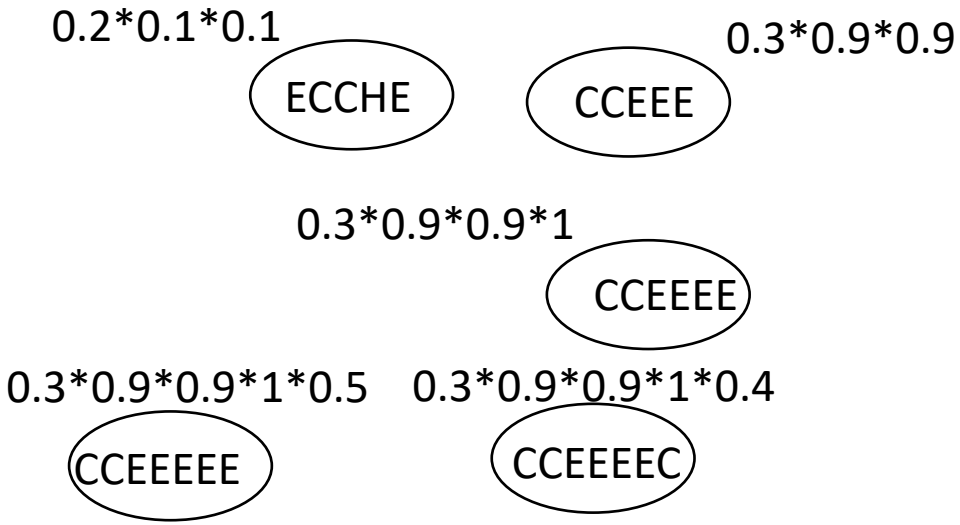
seq = ATWGRTG
K = 3



stitchextend_dict

ECCHE 0.2*0.1*0.1

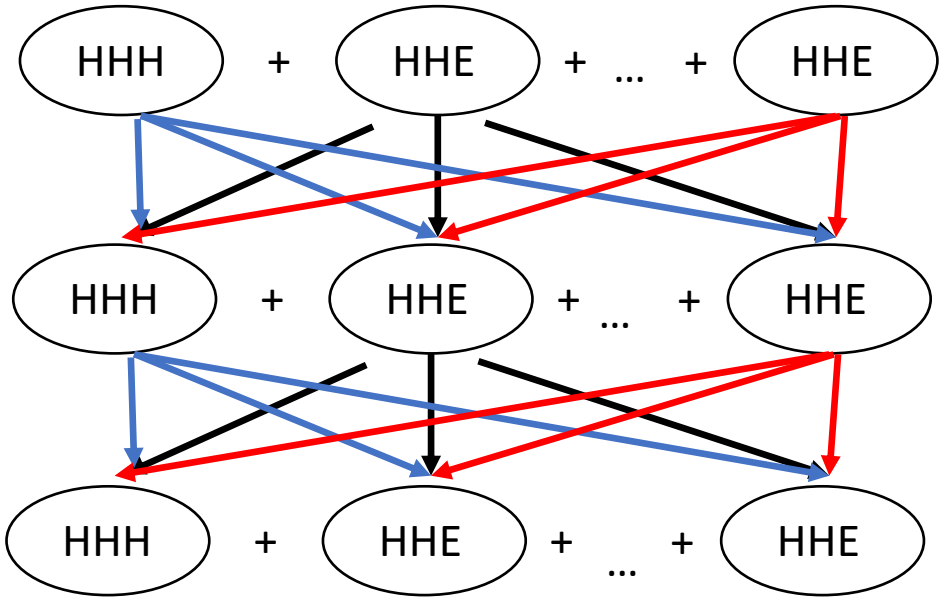
CCEEE 0.3*0.9*0.9



Edge Contraction

Layer 2, Layer 3

0.3*0.9*0.9*1*0.5



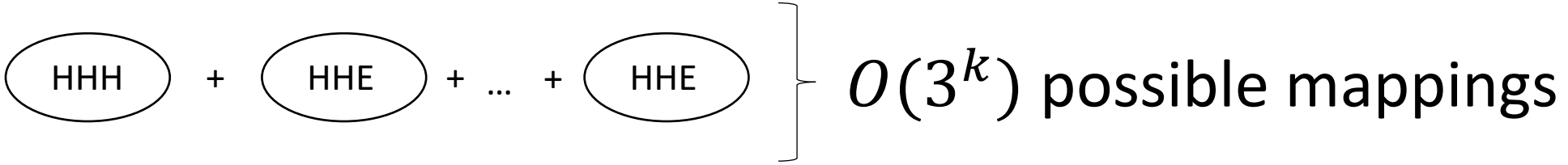
*

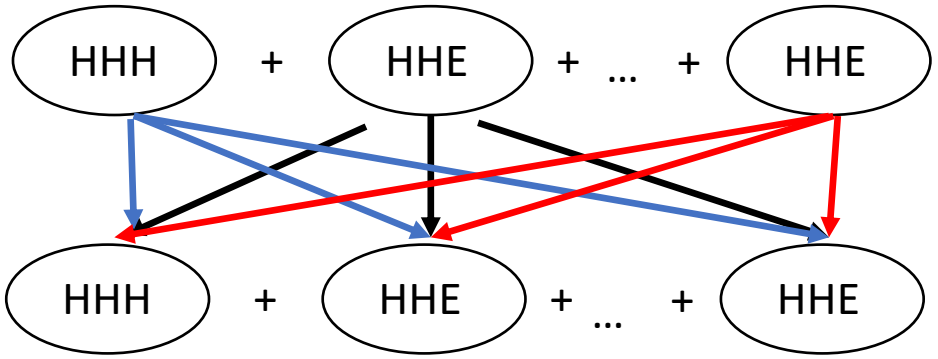
*

*

HHH + HHE + + HHE

$O(\ell - \mathrm{k} + 1)$ Layers in the graph


HHH
+
HHE
+ … +
HHE
$O(3^k)$ possible mappings



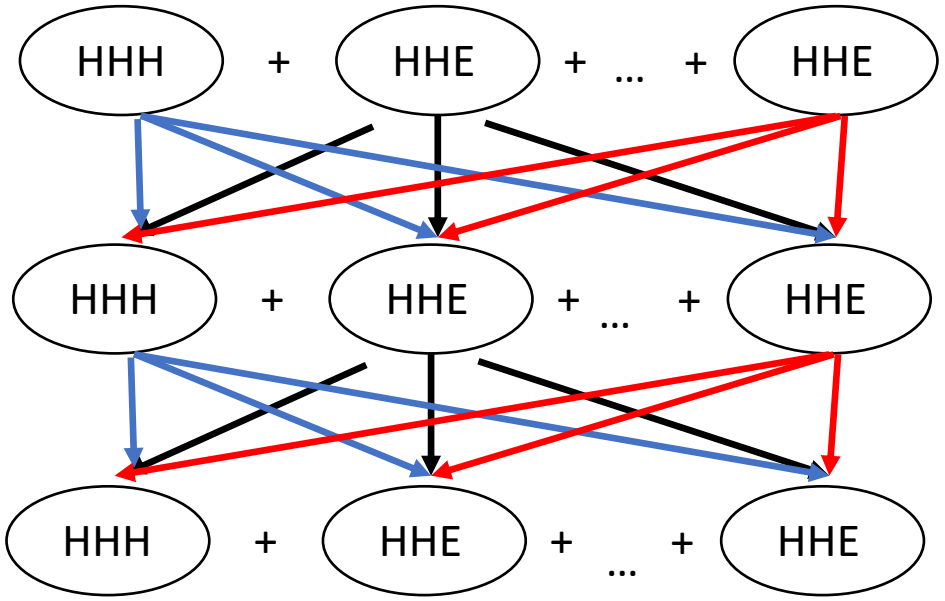
At most 3 out edges per node

E..C..C
C..CH

E..C..C
C..CC

E..C..C
C..CE

$k-1$ overlap



$At\ most\ 3\ out\ edges\ per\ node$

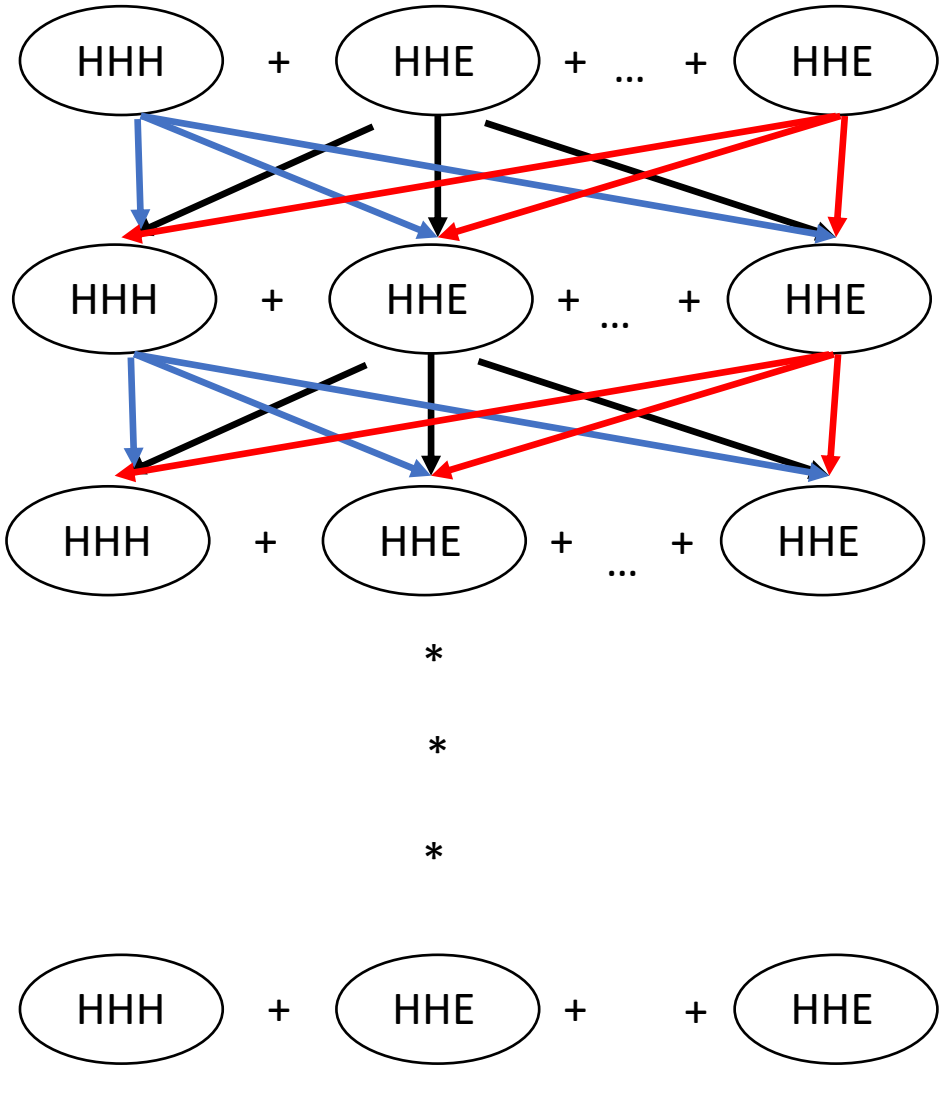
$At\ most\ 3 * 3\ possible\ out\ edges\ per\ node$
after extension

*
*
*

*
*
*

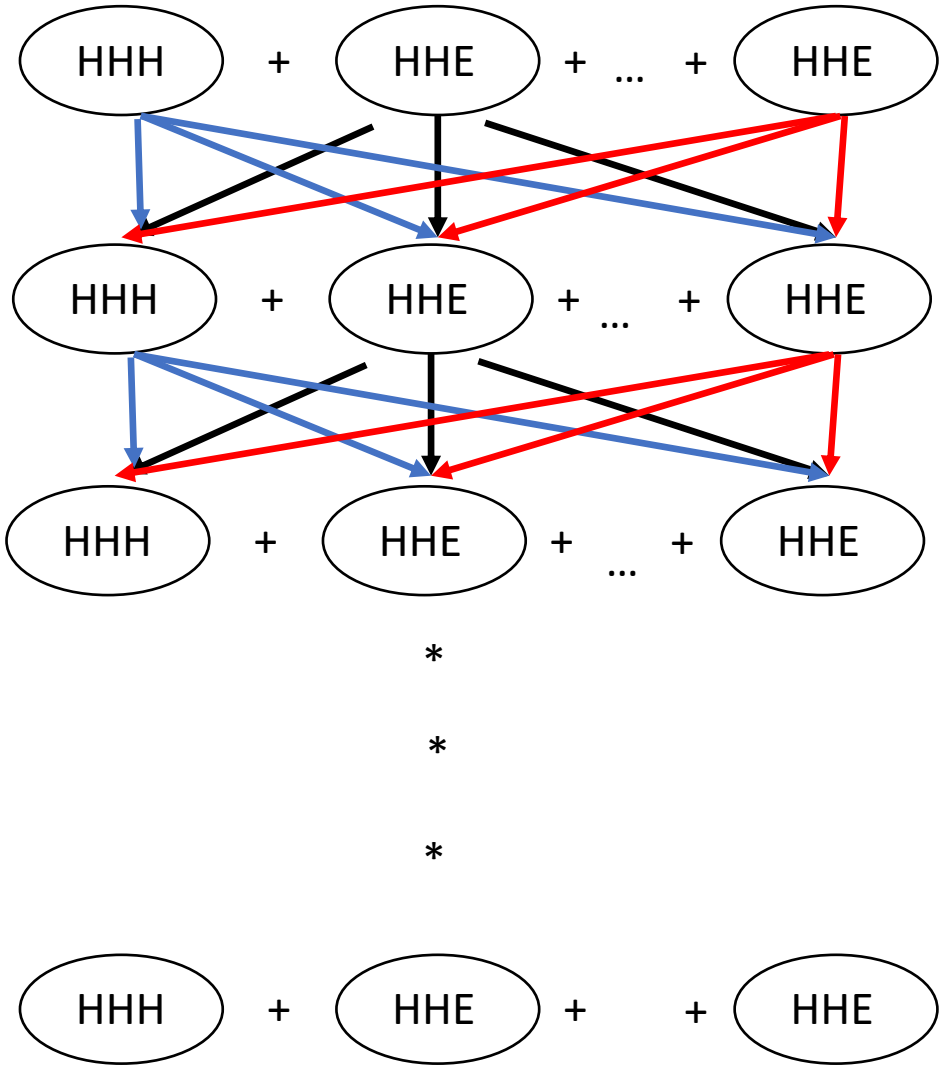
$O(3^{\ell-k+1})$ possible paths by the end

(upper bound)



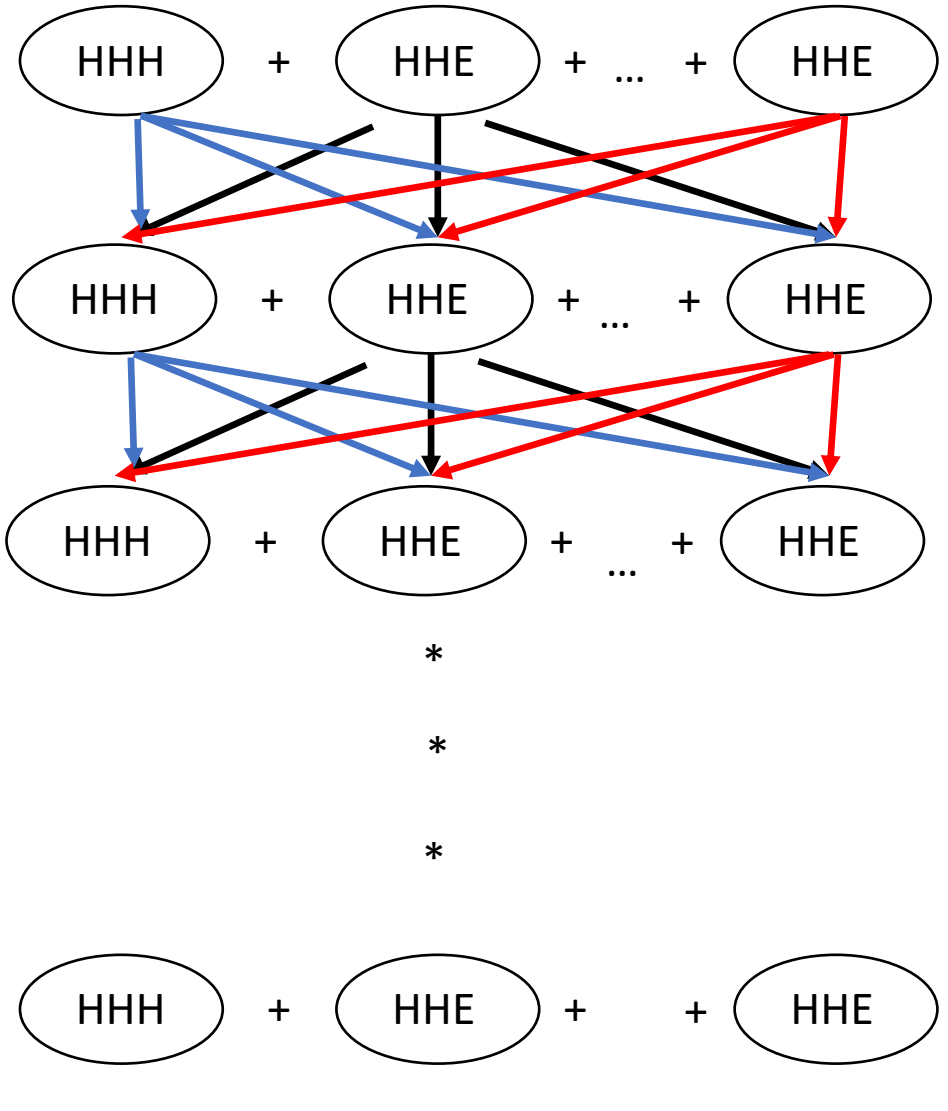
$$O(3^{\ell-k+1} * 3^{k} * (\ell - k + 1))$$

$$O(3^{\ell+1} * (\ell - k + 1))$$



Heuristic:

Limit the number of extended sequences kept after each iteration through layers. Only take the top X probable extended sequences.



$At\ most\ 3\ out\ edges\ per\ node$

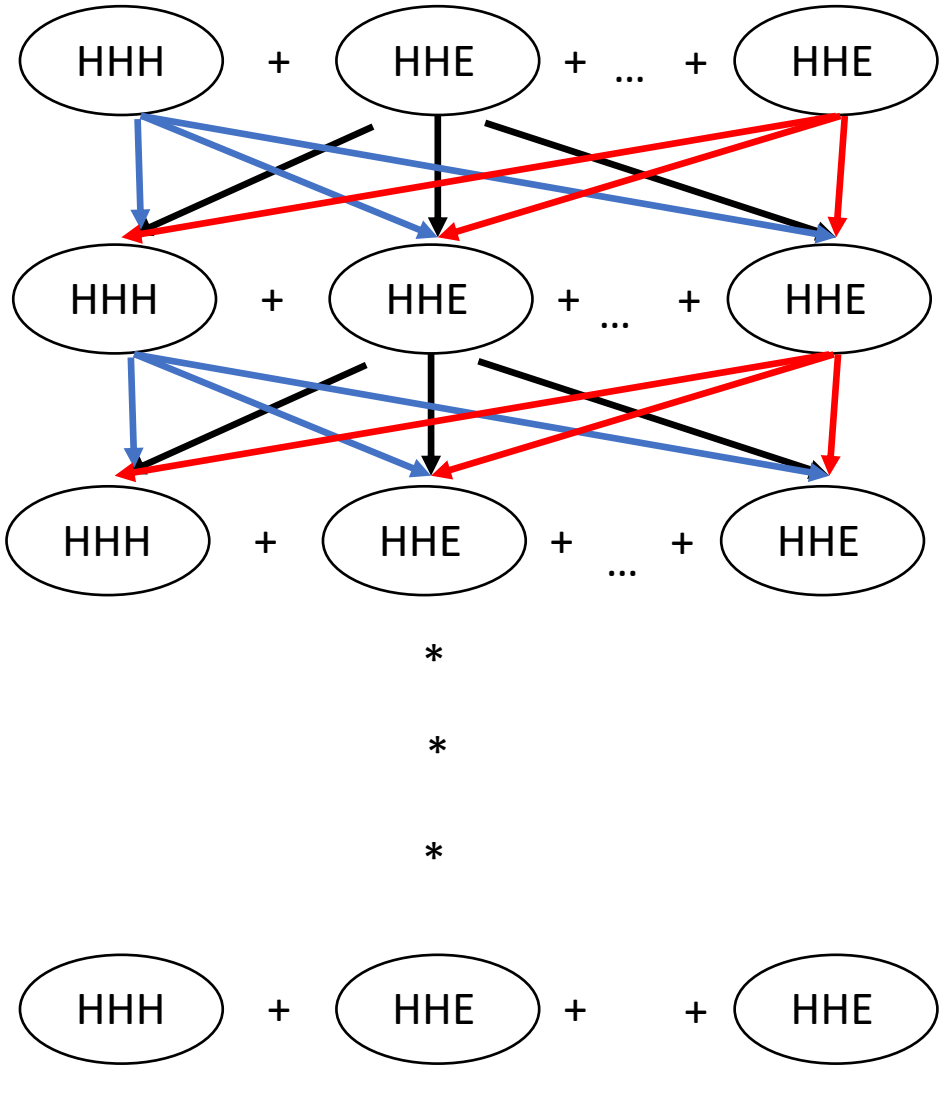
$At\ most\ 3 * 3\ possible\ out\ edges\ per\ node$
after extension

*
*
*

*
*
*

$O(1)$ possible paths by the end

Since paths are limited as we are extending



Heuristic:

$$O(3^k * (\ell - \mathrm{k} + 1))$$

$O(\ell - \mathrm{k} + 1)$ Layers in the graph

INPUT:

Protein Sequence = ATWGRTG

$\ell = 7$

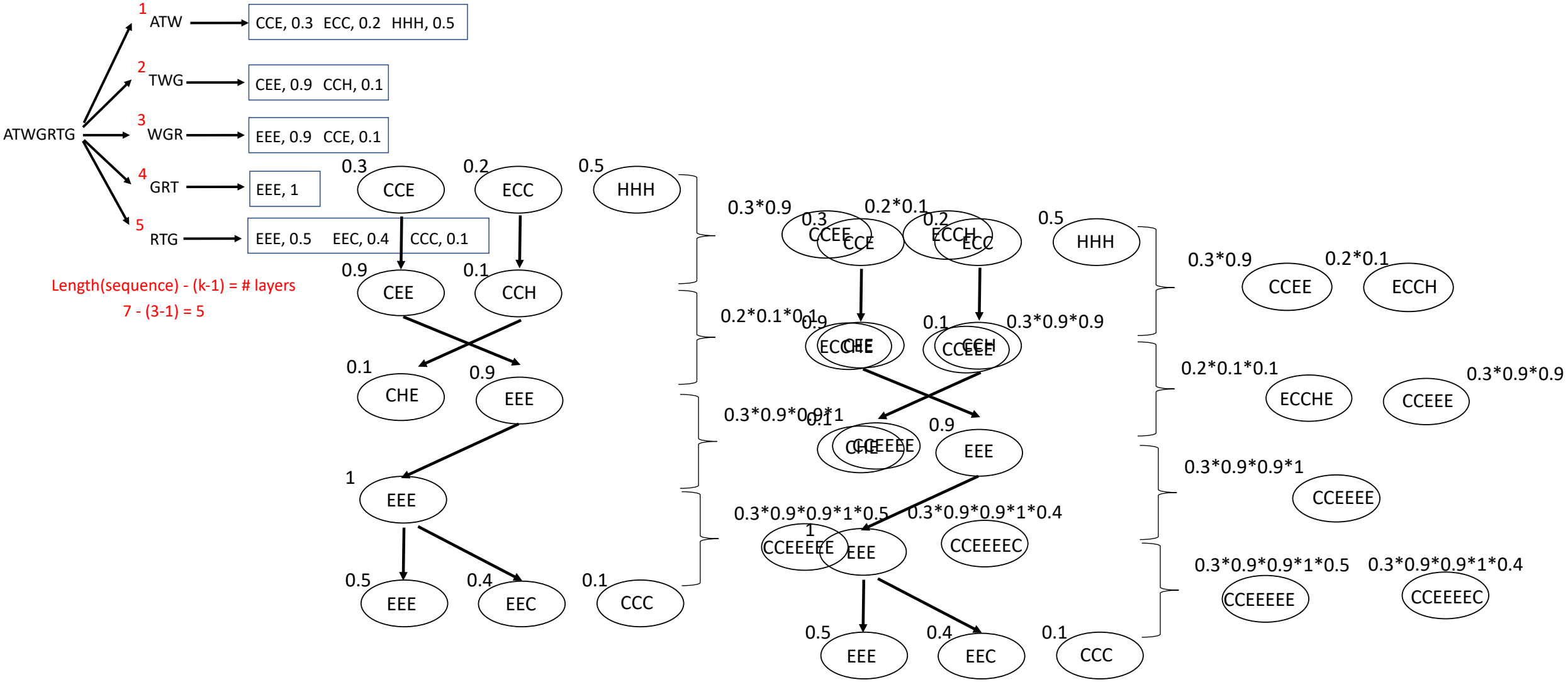
k = 3

DebruijnExtend

$$O(3^{\ell+1} * (\ell - k + 1))$$

Output (most probable):

CCEEEEE Prob: 0.1215



INPUT:
Protein Sequence = ATWGRTG
$\ell = 7$
k = 3

DebruijnExtend

Dreycey Albin, Angela Folz

OUTPUT (most probable):
CCEEEEE Prob: 0.1215
TC: $O(3^{\ell+1} * (\ell - \mathrm{k} + 1))$

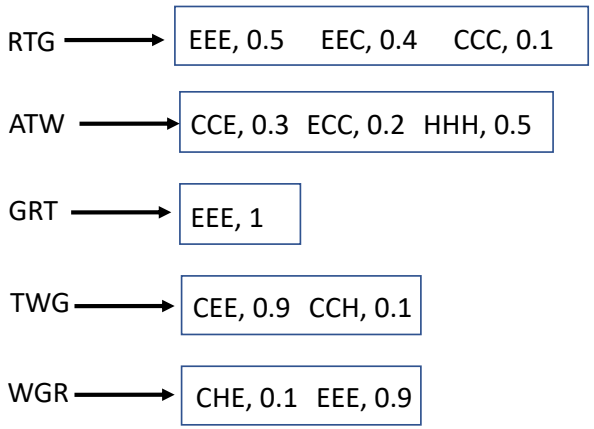
Example of using software (CMD line)

python DebruijnExtend.py gfp.fasta 4 gfp.ss3

```
CCCCCCCCCCCEEEEEEEEEEECCEEEEEEEEEE
ECCCCEEEEEEECCCCCCCCHHHHCCCCCCHHHC
-log(P)=231.1
```

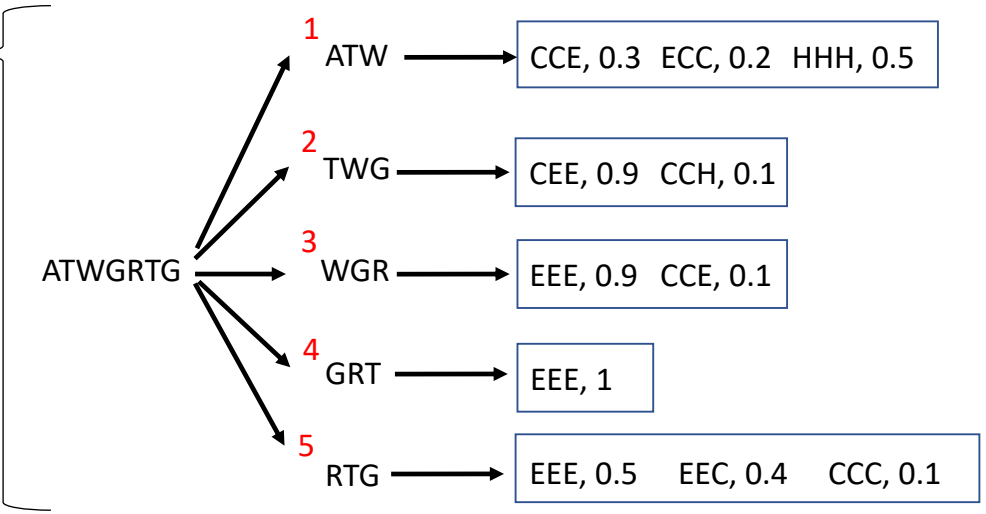
STEP 0 – Hash Table (Training)

For each k-mer in a training database, find every possible secondary structure and its probability.



STEP 1 – K-mer Mapping

Look up the corresponding set of secondary structures for each k-mer using the precomputed hash table.



STEP 2 – Stitch-Extend

Use BFS to traverse the Debruijn graph to find the highest weighted path. This is implemented using a dynamic programming method where the subproblem is matching the contracted nodes to the nodes of the next layer.