Comp 555 Homework 3

Elliott Hauser

November 7, 2012

1 Problem 8.6

The shortest common superstring of the 8 3-mers given in the problem is given in Figure 1. This is the shortest possible subsequence because, for a set like this one where the minimum Hamming distance between strings is 1, the minimum theoretical length of a superstring is the number of substrings plus 2 (i.e. the Hamming distance of the two end pieces). For the distance to be less than this, one or more of the substrings would have to be identical to one another. To be a longer superstring, more fragments would have to be incompatible with other substrings' prefixes or suffixes.

A	G	T	A	A	A	C	T	T	T
A	G	T							
	G	T	A						
		T	A	A					
			A	A	A				
				A	A	C			
					A	C	T		
						C	T	T	
							T	T	T

Figure 1: The shortest common superstring of 8 3-mers.

The Hamiltonian path approach to this problem is shown in Figure 2. The path does not visit every edge in the graph (since the Hamiltonian path is defined as visiting every node once and only once, regardless of edges), and the superstring represented by the path is the same, AGTAAACTTT.

The Eulerian Path approach to this problem is shown in Figure 3. There is only one possible Eulerian path where all nodes are balanced (have indegree=outdegree) except for up to two semibalanced nodes, in this case AG and TT. The Eulerian path corresponds to the same superstring we saw above, AGTAAACTTT.

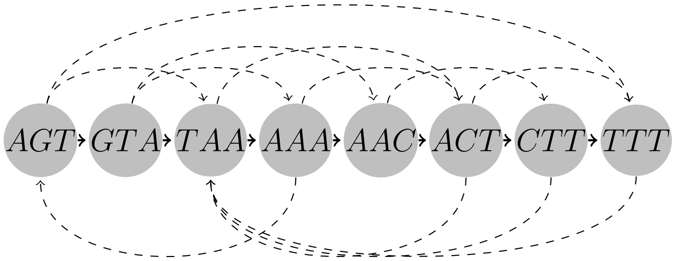


Figure 2: The Hamiltonian Path approach to finding the shortest common superstring. The 8 node Hamiltonian path is shown in solid, while other edges are shown dashed.

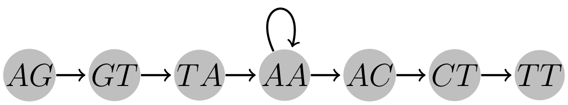


Figure 3: The Eulerian Path approach to finding the shortest common superstring.

2 Problem 8.7

If we reframe this problem as a Eulerian path problem through a graph with digits as the nodes, and edges in the graph representing the 2-digit numbers, we can see in Figure 4 that it forms a complete graph with 10 nodes. This string could be generated by finding a Eulerian cycle (being a complete graph, all nodes are balanced) in the graph.This means that there are many strings of the minimal length $10^2 + 1 = 101$characters that form the desired supersting.

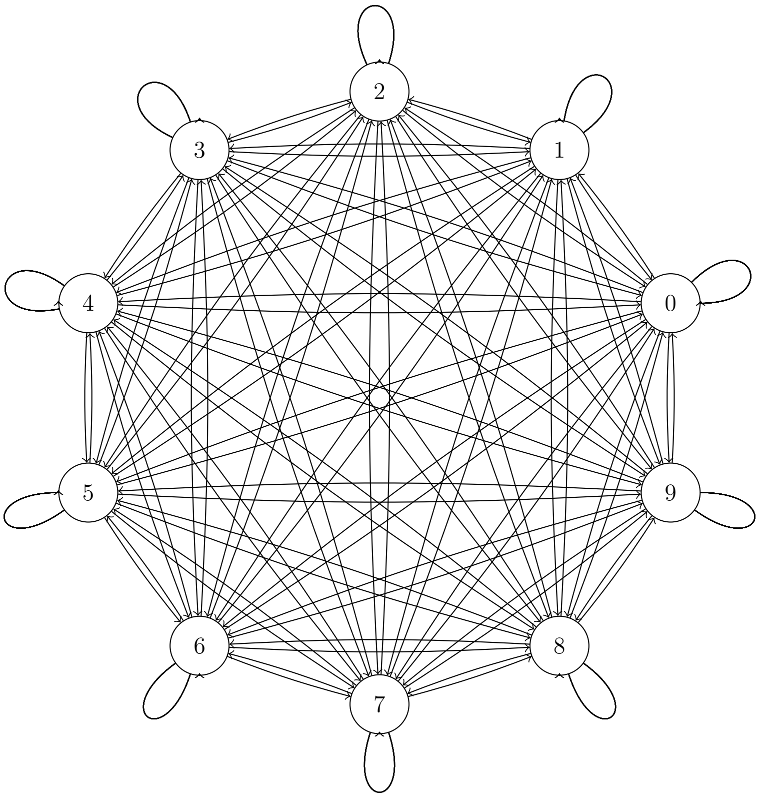


Figure 4: A Eulerian path approach to finding the shortest superstring problem of the set of 2 digit decimal numbers. LaTeX code adapted from http://www.texample.net/tikz/examples/complete-graph/, by Jean-Noël Quintin

3 Problem 8.9

A Eulerian graph corresponding to $S=$ {ATG, GGG, GGT, GTA, GTG, TAT, TGG} is

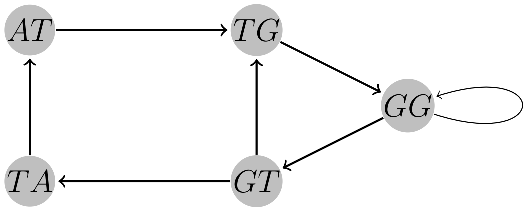
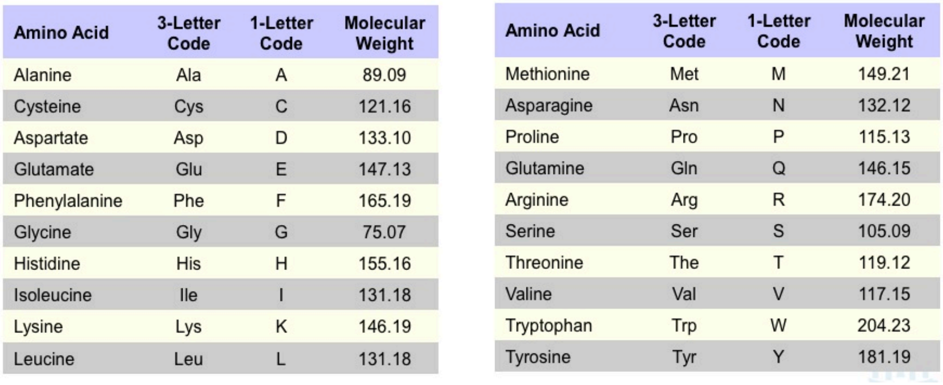


Figure 5: The Eulerian Path approach to finding the shortest common superstring in problem 8.9.

The three Eulerian paths of this graph correspond to GTATGGGTG and GTGGGTATG. Both of these paths start at the GT node but travel in slightly different paths to visit all edges of the graph. Interestingly, the strings are reversals of each other.

4 PEPTID



Amino Acid	3-Letter Code	1-Letter Code	Molecular Weight
Alanine	Ala	A	89.09
Cysteine	Cys	C	121.16
Aspartate	Asp	D	133.10
Glutamate	Glu	E	147.13
Phenylalanine	Phe	F	165.19
Glycine	Gly	G	75.07
Histidine	His	H	155.16
Isoleucine	Ile	I	131.18
Lysine	Lys	K	146.19
Leucine	Leu	L	131.18

Amino Acid	3-Letter Code	1-Letter Code	Molecular Weight
Methionine	Met	M	149.21
Asparagine	Asn	N	132.12
Proline	Pro	P	115.13
Glutamine	Gln	Q	146.15
Arginine	Arg	R	174.20
Serine	Ser	S	105.09
Threonine	The	T	119.12
Valine	Val	V	117.15
Tryptophan	Trp	W	204.23
Tyrosine	Tyr	Y	181.19

Figure 6: The list of amino acid weights and codes from Lecture 15, slide 4.

4a The theoretical MS/MS spectrum of PEPTID is shown below in Table 1.

Wt								Wt	Calculation	
760	P	E	P	T	I	D	$\vert$	-	$115 + 147 + 115 + 119 + 131 + 133$	
627	P	E	P	T	I	$\vert$	D	133	$115 + 147 + 115 + 119 + 131$	133
496	P	E	P	T	$\vert$	I	D	264	$115 + 147 + 115 + 119$	$131 + 133$
377	P	E	P	$\vert$	T	I	D	383	$115 + 147 + 115$	$119 + 131 + 133$
262	P	E	$\vert$	P	T	I	D	498	$115 + 147$	$115 + 119 + 131 + 133$
115	P	$\vert$	E	P	T	I	D	645	115	$147 + 115 + 119 + 131 + 133$

$$S1 = \{115, 133, 262, 264, 377, 383, 496, 498, 627, 645, 760\}$$

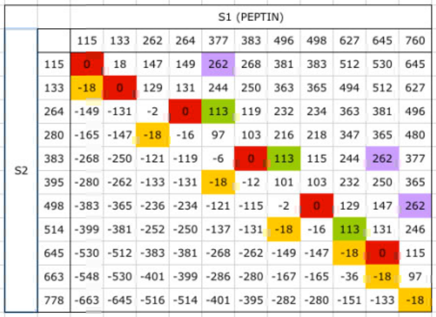
Table 1: The theoretical MS/MS spectrum of PEPTID, with calculations

4b The Shared Peak Counts (SPC) of $S2$ and $S3$ with $S1$ are given below in Table 2.

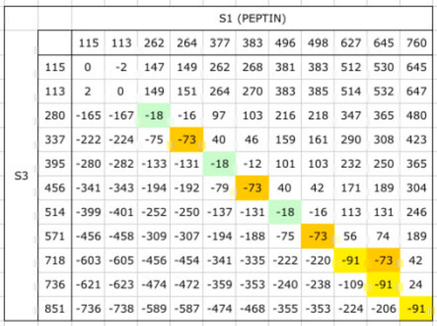
4c The spectral convolutions of $S2$ and $S3$, respectively, with $S1$, are given in Figure 7. As can immediately be seen from the colorings, $S1$ and $S2$ are more similar by

$$S1 = \{115, 133, 262, 264, 377, 383, 496, 498, 627, 645, 760\}$$
$$S2 = \{\mathbf{115{,}133{,}264}, 280, \mathbf{383}, 395\mathbf{498}, 514, \mathbf{645}, 663, 778\} \quad \text{SPC} = 6$$
$$S3 = \{\mathbf{115{,}133}, 280, 337, 395, 456, 514, 571, 718, 736, 851\} \quad \text{SPC} = 2$$

Table 2: The Shared Peak Counts (SPC) of $S2$ and $S3$ with $S1$. Values of $S2$ and $S3$ found in $S1$ are highlighted.



		S1 (PEPTIN)										
		115	133	262	264	377	383	496	498	627	645	760
S2	115	0	18	147	149	262	268	381	383	512	530	645
	133	-18	0	129	131	244	250	363	365	494	512	627
	264	-149	-131	-2	0	113	119	232	234	363	381	496
	280	-165	-147	-18	-16	97	103	216	218	347	365	480
	383	-268	-250	-121	-119	-6	0	113	115	244	262	377
	395	-280	-262	-133	-131	-18	-12	101	103	232	250	365
	498	-383	-365	-236	-234	-121	-115	-2	0	129	147	262
	514	-399	-381	-252	-250	-137	-131	-18	-16	113	131	246
	645	-530	-512	-383	-381	-268	-262	-149	-147	-18	0	115
	663	-548	-530	-401	-399	-286	-280	-167	-165	-36	-18	97
	778	-663	-645	-516	-514	-401	-395	-282	-280	-151	-133	-18



		S1 (PEPTIN)										
		115	113	262	264	377	383	496	498	627	645	760
S3	115	0	-2	147	149	262	268	381	383	512	530	645
	113	2	0	149	151	264	270	383	385	514	532	647
	280	-165	-167	-18	-16	97	103	216	218	347	365	480
	337	-222	-224	-75	-73	40	46	159	161	290	308	423
	395	-280	-282	-133	-131	-18	-12	101	103	232	250	365
	456	-341	-343	-194	-192	-79	-73	40	42	171	189	304
	514	-399	-401	-252	-250	-137	-131	-18	-16	113	131	246
	571	-456	-458	-309	-307	-194	-188	-75	-73	56	74	189
	718	-603	-605	-456	-454	-341	-335	-222	-220	-91	-73	42
	736	-621	-623	-474	-472	-359	-353	-240	-238	-109	-91	24
	851	-736	-738	-589	-587	-474	-468	-355	-353	-224	-206	-91

Figure 7: The spectral convolutions of $S2$ and $S3$, respectively, with $S1$. D(k) for k>2 are highlighted.

this method of comparison than are $S1$ and $S3$. The $S1 \ominus S2$ convolution has peak heights of 5 and 6, whereas the highest peak in $S1 \ominus S3$ is only 4.

4d To determine the residue substitutions that might have given rise to $S2$ and $S3$, we'll need a difference matrix of all the amino acids. This is given in Figure 8.

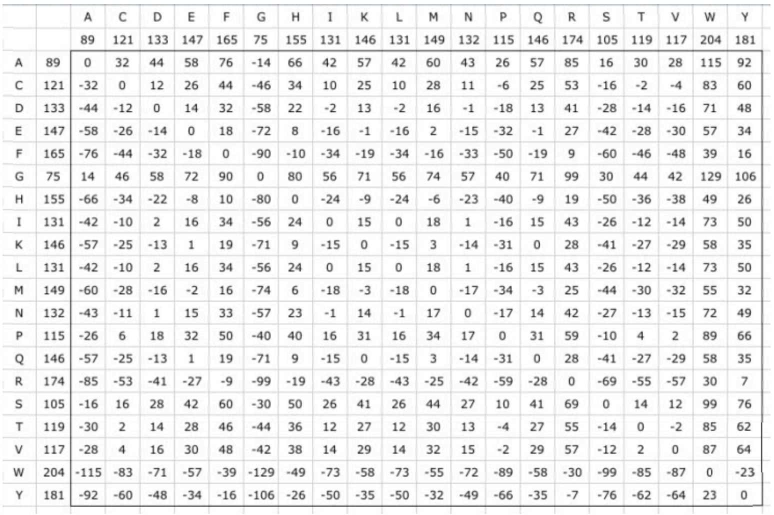
The possible residues changes for $S1 \ominus S3$ with a mass difference of 18 are:

- E → F PFPTID
- P → D DEDTID
- I → M PEPTMD

The only possible residues change for $S1 \ominus S3$ with a mass difference of 73 are:

- I → W PEPTWD

We can combine these observations with the theoretical spectrum calculated above for $S1$, PEPTID, to determine the residue substitutions for $S3$. Table 3 shows that PFPTWD is a possible residue substition for PEPTID consistent with spectrum $S3$.



		A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
		89	121	133	147	165	75	155	131	146	131	149	132	115	146	174	105	119	117	204	181
A	89	0	32	44	58	76	-14	66	42	57	42	60	43	26	57	85	16	30	28	115	92
C	121	-32	0	12	26	44	-46	34	10	25	10	28	11	-6	25	53	-16	-2	-4	83	60
D	133	-44	-12	0	14	32	-58	22	-2	13	-2	16	-1	-18	13	41	-28	-14	-16	71	48
E	147	-58	-26	-14	0	18	-72	8	-16	-1	-16	2	-15	-32	-1	27	-42	-28	-30	57	34
F	165	-76	-44	-32	-18	0	-90	-10	-34	-19	-34	-16	-33	-50	-19	9	-60	-46	-48	39	16
G	75	14	46	58	72	90	0	80	56	71	56	74	57	40	71	99	30	44	42	129	106
H	155	-66	-34	-22	-8	10	-80	0	-24	-9	-24	-6	-23	-40	-9	19	-50	-36	-38	49	26
I	131	-42	-10	2	16	34	-56	24	0	15	0	18	1	-16	15	43	-26	-12	-14	73	50
K	146	-57	-25	-13	1	19	-71	9	-15	0	-15	3	-14	-31	0	28	-41	-27	-29	58	35
L	131	-42	-10	2	16	34	-56	24	0	15	0	18	1	-16	15	43	-26	-12	-14	73	50
M	149	-60	-28	-16	-2	16	-74	6	-18	-3	-18	0	-17	-34	-3	25	-44	-30	-32	55	32
N	132	-43	-11	1	15	33	-57	23	-1	14	-1	17	0	-17	14	42	-27	-13	-15	72	49
P	115	-26	6	18	32	50	-40	40	16	31	16	34	17	0	31	59	-10	4	2	89	66
Q	146	-57	-25	-13	1	19	-71	9	-15	0	-15	3	-14	-31	0	28	-41	-27	-29	58	35
R	174	-85	-53	-41	-27	-9	-99	-19	-43	-28	-43	-25	-42	-59	-28	0	-69	-55	-57	30	7
S	105	-16	16	28	42	60	-30	50	26	41	26	44	27	10	41	69	0	14	12	99	76
T	119	-30	2	14	28	46	-44	36	12	27	12	30	13	-4	27	55	-14	0	-2	85	62
V	117	-28	4	16	30	48	-42	38	14	29	14	32	15	-2	29	57	-12	2	0	87	64
W	204	-115	-83	-71	-57	-39	-129	-49	-73	-58	-73	-55	-72	-89	-58	-30	-99	-85	-87	0	-23
Y	181	-92	-60	-48	-34	-16	-106	-26	-50	-35	-50	-32	-49	-66	-35	-7	-76	-62	-64	23	0

Figure 8: A difference matrix of all the amino acids

Wt								Wt	Wt								Wt
760	P	E	P	T	I	D	|	-	851	P	F	P	T	W	D	|	-
627	P	E	P	T	I	|	D	133	718	P	F	P	T	I	|	D	133
496	P	E	P	T	|	I	D	264	514	P	F	P	T	|	W	D	337
377	P	E	P	|	T	I	D	383	395	P	F	P	|	T	W	D	456
262	P	E	|	P	T	I	D	498	280	P	F	|	P	T	W	D	571
115	P	|	E	P	T	I	D	645	115	P	|	F	P	T	W	D	736

Table 3: Calculations of a possible residue substation from PEPTID to PFPTWD

The substitution picture with $S2$ is somewhat more complicated, if only because of the higher numbers of D(k) for $k > 2$. In the end, though, a single substitution, E → F, accounts for all of the observed spectral differences. As can be seen in Figure 8, F is the only acid with a weight difference of 18 with E.

Wt								Wt	Wt								Wt
760	P	E	P	T	I	D	|	-	778	P	F	P	T	I	D	|	-
627	P	E	P	T	I	|	D	133	645	P	F	P	T	I	|	D	133
496	P	E	P	T	|	I	D	264	514	P	F	P	T	|	I	D	264
377	P	E	P	|	T	I	D	383	395	P	F	P	|	T	I	D	383
262	P	E	|	P	T	I	D	498	280	P	F	|	P	T	I	D	498
115	P	|	E	P	T	I	D	645	115	P	|	F	P	T	I	D	663

Table 4: Calculations of a possible residue substition from PEPTID to PFPTID

5 Programming Exercise

See also electronic submission.

```
from sys import exit
import networkx as nx
from random import choice

”””A␣program␣to␣find␣a␣Eulerian␣path␣through␣a␣directed␣graph”””

# Construct graph from input
# input=eval(raw_input(”enter your graph”))
# G=nx.DiGraph()
# for edge in input:
#  G.add_edge(edge[0],edge[1], label=edge[2])

n = 0

def in_out_balance(node):
    balance=G.in_degree(node) − G.out_degree(node)
    return balance

def set_semibalanced(node):
    global unbalanced
    global n
    G.node[node][’semibalanced’] = 1
    unbalanced = 1
    n = n + 1

def none():
    print ”None”
    return None
    exit()

def find_balance(G):
    global start
    global finish
    global unbalanced
    unbalanced = 0
    n = 0
    for node in G:
        balance = in_out_balance(node)
        if abs(balance) > 1:
            none()
        if abs(balance) == 1:
            set_semibalanced(node)
            if balance == 1:
                finish=node
            else:
                start=node
        if n > 2:
            none()
    # Connect start and finish if unbalanced
    # Otherwise, pick a random starting point
    if unbalanced==1:
        G.add_edge(finish,start, label=None)
    else:
        start=choice(G.nodes())

def euler(G):
    global start
    global finish
    find_balance(G)
    path=[]
    if nx.is_eulerian(G):
        circuit=list(nx.eulerian_circuit(G, start))
        for edge in circuit:
            current=G.edge[edge[0]][edge[1]]['label']
            if current != None:
                path.append(current)
        if unbalanced==0:
            finish=circuit.pop()[1]
        answer=start, path, finish
        print answer

    else:
        none()

euler(G)
```