

**T.C. İSTANBUL TİCARET ÜNİVERSİTESİ  
MÜHENDİSLİK VE TASARIM FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BIL458 BİYOİNFORMATİK**

**DNA SEKANS HİZALAMA  
ALGORİTMALARI UYGULAMASI**

**Alperen KÖYLÜ**

**Bilgisayar Mühendisliği Programında Hazırlanan**

**ÖDEV RAPORU**

**Danışman: Arzu Kakışım**

**İSTANBUL, 2020**

Number of used sequence for comparison is 27. Each sequence is compared by first sequence in order to find the best global and local alignment with respect to their score.

## Global Alignment

Needleman-Wunsch algorithm is implemented for finding global alignments score. In the figure below, there are details of the highest scoring pairings found at the end of the Needleman-Wunsch alignment, which operates 104 times using 4 different cost matrices and 26 different sequence pairs.

### MAXIMUM SCORE WITH PROVIDING COST MATRIX

#### DIRECTION MATRIX:

```
[['0' 'H' 'H' 'H' 'H' ... 'H' 'H' 'H' 'H' 'H']
['V' 'D' 'H' 'H' 'H' ... 'H' 'H' 'H' 'H' 'H']
['V' 'V' 'D' 'H' 'H' ... 'H' 'H' 'H' 'H' 'H']
['V' 'V' 'V' 'D' 'H' ... 'H' 'H' 'H' 'H' 'H']
['V' 'V' 'V' 'V' 'D' ... 'H' 'H' 'H' 'H' 'H']
...
['V' 'V' 'V' 'V' 'V' ... 'D' 'H' 'H' 'H' 'H']
['V' 'V' 'V' 'V' 'V' ... 'V' 'D' 'H' 'H' 'H']
['V' 'V' 'V' 'V' 'V' ... 'V' 'V' 'D' 'H' 'H']
['V' 'V' 'V' 'V' 'V' ... 'V' 'V' 'V' 'D' 'H']
['V' 'V' 'V' 'V' 'V' ... 'V' 'V' 'V' 'V' 'D']]
```

#### SCORING MATRIX:

```
[ [ 0. -6. -12. -18. -24. ... -276. -282. -288. -294. -300.]
[ -6. 5. -1. -7. -13. ... -265. -271. -277. -283. -289.]
[ -12. -1. 13. 7. 1. ... -251. -257. -263. -269. -275.]
[ -18. -7. 7. 21. 15. ... -237. -243. -249. -255. -261.]
[ -24. -13. 1. 15. 26. ... -226. -232. -238. -244. -250.]
...
[ -276. -265. -251. -237. -226. ... 216. 210. 204. 198. 192.]
[ -282. -271. -257. -243. -232. ... 210. 224. 218. 212. 206.]
[ -288. -277. -263. -249. -238. ... 204. 218. 228. 222. 216.]
[ -294. -283. -269. -255. -244. ... 198. 212. 222. 236. 230.]
[ -300. -289. -275. -261. -250. ... 192. 206. 216. 230. 240.]]
```

```
ATTAAAGGTTTATACCTTCCCAGGTAACAAACCAACCACTTTCGATCTC
||||| ||||||| ||||||| ||||| |||||||
ATTAGGGGTTTATATCTTCCCAGGTACCAACCCACCAACTTTCGATCTC
```

SCORE: 240

#### COST MATRIX:

```
- A T C G
- -6 -6 -6 -6 -6
A -6 +5 -1 -4 -4
T -6 -1 +8 -3 -4
C -6 -4 -3 +4 -4
G -6 -4 -4 -4 +6
```

# Representation of all values of the score matrix for Needleman-Wunsch alignment algorithm;

(High Resolution Image can be found in the repository [Click for repository](#))

'	0	-6	-12	-18	-24	-30	-36	-42	-48	-54	-60	-66	-72	-78	-84	-90	-96	-102	-108	-114	-120	-126	-132	-138	-144	-150	-156	-162	-168	-174	-180	-186	-192	-198	-204	-210	-216	-222	-228	-234	-240	-246	-252	-258	-264	-270	-276	-282	-288	-294	-300	
◀	-6	5	-1	-7	-13	-19	-25	-31	-37	-43	-49	-55	-61	-67	-73	-79	-85	-91	-97	-103	-109	-115	-121	-127	-133	-139	-145	-151	-157	-163	-169	-175	-181	-187	-193	-199	-205	-211	-217	-223	-229	-235	-241	-247	-253	-259	-265	-271	-277	-283	-289	
├	-12	-1	13	7	1	-5	-11	-17	-23	-29	-35	-41	-47	-53	-59	-65	-71	-77	-83	-89	-95	-101	-107	-113	-119	-125	-131	-137	-143	-149	-155	-161	-167	-173	-179	-185	-191	-197	-203	-209	-215	-221	-227	-233	-239	-245	-251	-257	-263	-269	-275	
├	-18	-7	7	21	15	9	3	-3	-9	-15	-21	-27	-33	-39	-45	-51	-57	-63	-69	-75	-81	-87	-93	-99	-105	-111	-117	-123	-129	-135	-141	-147	-153	-159	-165	-171	-177	-183	-189	-195	-201	-207	-213	-219	-225	-231	-237	-243	-249	-255	-261	
◀	-24	-13	1	15	26	20	14	8	2	4	-10	-16	-22	-28	-34	-40	-46	-52	-58	-64	-70	-76	-82	-88	-94	-100	-106	-112	-118	-124	-130	-136	-142	-148	-154	-160	-166	-172	-178	-184	-190	-196	-202	-208	-214	-220	-226	-232	-238	-244	-250	
◀	-30	-19	-5	9	20	22	16	10	4	1	-5	-11	-17	-23	-29	-35	-41	-47	-53	-59	-65	-71	-77	-83	-89	-95	-101	-107	-113	-119	-125	-131	-137	-143	-149	-155	-161	-167	-173	-179	-185	-191	-197	-203	-209	-215	-221	-227	-233	-239		
◀	-36	-25	-11	3	14	16	18	12	6	3	0	-6	-6	-12	-12	-18	-24	-30	-36	-42	-48	-54	-60	-66	-72	-78	-84	-90	-96	-102	-108	-114	-120	-126	-132	-138	-144	-150	-156	-162	-168	-174	-180	-186	-192	-198	-204	-210	-216	-222	-228	
◊	-42	-31	-17	-3	8	20	22	24	18	12	6	0	-6	-10	-16	-16	-22	-28	-34	-40	-46	-52	-58	-64	-70	-76	-82	-88	-94	-100	-106	-112	-118	-124	-130	-136	-142	-148	-154	-160	-166	-172	-178	-184	-190	-196	-202	-208	-214	-220	-226	-232
◊	-48	-37	-23	-9	2	14	26	28	30	24	18	12	6	0	-6	-12	-18	-24	-30	-36	-42	-48	-54	-60	-66	-72	-78	-84	-90	-96	-102	-108	-114	-120	-126	-132	-138	-144	-150	-156	-162	-168	-174	-180	-186	-192	-198	-204				
├	-54	-43	-29	-15	-4	8	20	22	24	38	32	26	20	14	8	2	-4	-10	-16	-22	-28	-34	-40	-46	-52	-58	-64	-70	-76	-82	-88	-94	-100	-106	-112	-118	-124	-130	-136	-142	-148	-154	-160	-166	-172	-178	-184	-190				
├	-60	-49	-35	-21	-10	2	14	16	18	32	46	40	34	28	22	16	10	4	-2	-8	-14	-20	-26	-32	-38	-44	-50	-56	-62	-68	-74	-80	-86	-92	-98	-104	-110	-116	-122	-128	-134	-140	-146	-152	-158	-164	-170	-176				
├	-66	-55	-41	-27	-16	-4	8	10	12	26	40	54	48	42	36	30	24	18	12	6	0	-6	-12	-18	-24	-30	-36	-42	-48	-54	-60	-66	-72	-78	-84	-90	-96	-102	-108	-114	-120	-117	-114	-120	-126	-132	-138	-144	-150	-156	-162	
◀	-72	-61	-47	-33	-22	-10	2	4	6	20	34	48	59	53	47	41	35	29	23	17	11	5	-1	-7	-13	-19	-25	-31	-37	-43	-49	-55	-61	-67	-73	-79	-85	-91	-97	-103	-109	-115	-118	-115	-121	-127	-133	-139	-145	-151		
├	-78	-67	-53	-39	-28	-16	-4	-2	0	14	28	42	53	67	61	55	49	43	37	31	25	19	13	7	1	5	-11	-17	-23	-29	-35	-41	-47	-53	-59	-65	-71	-77	-83	-89	-95	-101	-107	-110	-116	-122	-128	-119	-125	-131	-137	
◀	-84	-73	-59	-45	-34	-22	-10	-8	-6	8	22	36	47	61	72	66	60	54	48	42	36	30	24	18	12	6	0	-6	-12	-18	-24	-30	-36	-42	-48	-54	-60	-66	-72	-78	-84	-90	-96	-102	-108	-114	-117	-123	-126	-132		
◊	-90	-79	-65	-51	-40	-28	-16	-14	-12	2	16	30	41	55	66	69	70	64	58	52	46	40	34	28	22	16	10	4	-2	-8	-14	-20	-26	-32	-38	-44	-50	-56	-62	-68	-74	-80	-86	-92	-98	-104	-110	-116	-119	-125	-122	
◊	-96	-85	-71	-57	-46	-34	-22	-20	-18	-4	10	24	35	49	60	63	73	67	61	62	56	50	44	38	32	26	20	14	8	2	-4	-10	-16	-22	-28	-34	-40	-46	-52	-58	-64	-70	-76	-82	-88	-94	-100	-106	-112	-118	-121	
├	-102	-91	-77	-63	-52	-40	-28	-26	-24	-10	4	18	29	43	54	68	67	81	75	69	63	57	51	45	39	40	34	28	22	16	10	4	-2	-8	-14	-20	-26	-32	-38	-44	-50	-56	-62	-68	-74	-80	-86	-92	-98	-104	-110	
├	-108	-97	-83	-69	-58	-46	-34	-32	-30	-16	-2	12	23	37	48	62	65	75	89	83	77	71	65	59	53	47	41	35	29	23	17	11	5	-1	-7	-13	-19	-25	-31	-37	-43	-48	-54	-60	-66	-72	-78	-84	-90	-96		
◊	-114	-103	-89	-75	-64	-52	-40	-38	-36	-22	-8	6	17	31	42	56	66	69	83	93	87	81	75	69	63	57	51	45	39	33	27	21	15	9	3	-3	-9	-15	-21	-27	-33	-39	-45	-51	-50	-56	-62	-68	-74	-80	-86	
◊	-120	-109	-95	-81	-70	-58	-46	-44	-42	-28	-14	0	11	25	36	50	60	63	77	87	97	91	85	79	73	67	61	55	49	43	37	31	25	19	13	7	1	-5	-11	-17	-23	-29	-35	-41	-47	-53	-59	-65	-64	-70	-76	
◊	-126	-115	-101	-87	-76	-64	-52	-50	-48	-34	-20	-6	5	19	30	44	54	57	71	81	91	101	95	89	83	77	71	65	59	53	47	41	35	29	23	17	11	5	-1	-7	-13	-19	-25	-31	-37	-43	-49	-55	-61	-67	-66	
◀	-132	-121	-107	-93	-82	-70	-58	-56	-54	-40	-26	-12	-1	13	24	38	48	53	65	75	85	95	106	100	94	88	82	76	70	64	58	52	46	40	34	28	22	16	10	4	-2	-8	-14	-20	-26	-32	-38	-44	-50	-56	-62	
◊	-138	-127	-113	-99	-88	-76	-64	-52	-50	-46	-32	-18	-7	7	18	32	42	47	59	69	79	89	100	112	106	100	94	88	82	76	70	64	58	52	46	40	34	28	22	16	10	4	-2	-8	-14	-20	-26	-32	-38	-44	-50	
◊	-144	-133	-119	-105	-94	-82	-70	-58	-46	-52	-38	-24	-13	1	12	26	36	41	53	63	73	83	94	106	118	112	106	100	94	88	82	76	70	64	58	52	46	40	34	28	22	16	10	4	-2	-8	-14	-20	-26	-32	-38	
├	-150	-139	-125	-111	-100	-88	-76	-64	-52	-38	-44	-30	-19	-5	6	20	30	44	49	57	67	77	88	100	112	126	120	114	108	102	96	90	84	78	72	66	60	54	48	42	36	30	24	18	12	6	0	-6	-12	-18	-24	
◀	-156	-145	-131	-117	-106	-94	-82	-70	-58	-44	-39	-36	-25	-11	0	14	24	38	43	51	61	71	82	94	106	120	131	125	119	113	107	101	95	89	83	77	71	65	59	53	47	41	35	29	23	17	11	5	-1	-7	-13	
◀	-162	-151	-137	-123	-112	-100	-88	-76	-64	-50	-45	-40	-31	-17	-6	8	18	32	37	45	55	65	76	88	100	114	125	127	121	124	118	112	106	100	94	88	82	76	70	64	58	52	46	40	34	28	22	16	10	4	-2	
◊	-168	-157	-143	-129	-118	-106	-94	-82	-70	-56	-51	-46	-37	-23	-12	2	12	26	31	41	49	59	70	82	94	108	119	129	131	125	120	114	116	110	104	98	92	86	80	74	68	62	56	50	44	38	32	26	20	14	8	
◊	-174	-163	-149	-135	-124	-112	-100	-88	-76	-62	-57	-52	-41	-29	-18	-4	6	20	25	35	43	53	64	76	88	102	113	123	125	136	130	125	119	113	107	109	103	97	91	85	79	73	67	61	55	49	43	37	31	25	19	
◀	-180	-169	-155	-141	-130	-118	-106	-94	-82	-68	-63	-58	-47	-35	-24	-10	0	14	19	29	37	47	58	70	82	96	107	117	119	130	141	135	129	123	117	112	106	100	102	96	90	84	78	72	66	60	54	48	42	36	30	
◊	-186	-175	-161	-147	-136	-124	-112	-100	-88	-74	-69	-64	-53	-41	-30	-16	-4	8	13	23	31	41	52	64	76	90	101	111	113	124	135	146	140	134	128	122	116	110	105	107	101	95	89	83	77	71	65	59	53	47	41	
◊	-192	-181	-167	-153	-142	-130	-118	-106	-94	-80	-75	-70	-59	-47	-36	-22	-12	2	7	17	27	35	46	58	70	84	95	105	115	118																						

(High Resolution Image can be found in the repository [Click for repository](#))

[illegible]

## Local Alignment

Smith-Waterman algorithm is implemented for finding local alignments score. In the figure below, there are details of the highest scoring pairings found at the end of the Smith-Waterman alignment, which operates 104 times using 4 different cost matrices and 26 different sequence pairs.

```
MAXIMUM SCORE WITH PROVIDING COST MATRIX

DIRECTION MATRIX:
[['0' 'H' 'H' 'H' 'H' ... 'H' 'H' 'H' 'H' 'H']
 ['V' 'D' 'D' 'D' 'D' ... 'D' 'D' 'D' 'D' 'D']
 ['V' 'D' 'D' 'D' 'H' ... 'D' 'D' 'H' 'D' 'H']
 ['V' 'D' 'D' 'D' 'H' ... 'D' 'D' 'D' 'D' 'H']
 ['V' 'D' 'V' 'V' 'D' ... 'D' 'H' 'D' 'D' 'D']
 ...
 ['V' 'D' 'D' 'D' 'D' ... 'D' 'H' 'H' 'H' 'H']
 ['V' 'D' 'D' 'D' 'V' ... 'V' 'D' 'H' 'D' 'H']
 ['V' 'D' 'V' 'D' 'D' ... 'V' 'V' 'D' 'H' 'D']
 ['V' 'D' 'D' 'D' 'D' ... 'V' 'D' 'V' 'D' 'H']
 ['V' 'D' 'V' 'V' 'D' ... 'V' 'V' 'D' 'V' 'D']]

SCORING MATRIX:
[[ 0.  0.  0.  0.  0.  0. ...  0.  0.  0.  0.  0.]
 [ 0.  5.  0.  0.  5.  5. ...  5.  0.  0.  0.  0.]
 [ 0.  0. 13.  8.  2.  0. ...  0. 13.  7.  8.  2.]
 [ 0.  0.  8. 21. 15.  0. ...  0.  8. 10. 15.  9.]
 [ 0.  5.  2. 15. 26.  9. ...  9.  3.  4.  9. 11.]
 ...
 [ 0.  5.  0.  0.  9. ... 216. 210. 204. 198. 192.]
 [ 0.  0. 13.  8.  3. ... 210. 224. 218. 212. 206.]
 [ 0.  0.  7. 10.  4. ... 204. 218. 228. 222. 216.]
 [ 0.  0.  8. 15.  9. ... 198. 212. 222. 236. 230.]
 [ 0.  0.  2.  9. 11. ... 192. 206. 216. 230. 240.]]

ATTAAAGGTTTATACCTTCCCAGGTAACAAACCAACCACTTTTCGATCTC
||||| ||||||||| ||||||||| ||||||| |||||||||||||||||
ATTAGGGGTTTATATCTTCCCAGGTACCAAACCCACCAACTTTTCGATCTC

SCORE: 240

COST MATRIX:
- A T C G
- -6 -6 -6 -6 -6
A -6 +5 -1 -4 -4
T -6 -1 +8 -3 -4
C -6 -4 -3 +4 -4
G -6 -4 -4 -4 +6
```

(High Resolution Image can be found in the repository [Click for repository](#))

(High Resolution Image can be found in the repository [Click for repository](#))

[illegible]

**Personal Conclusion Comment:**

Since the given dna sequences are very close to each other, that is, there is not much protein difference between them and the gap penalty value (-6) we want to use is smaller than the mismatch penalty in the given cost matrices, the algorithm that is applied prefers to do mismatch instead of gap. In local alignment, this method often gives more accurate results, but in global alignment, accurate results cannot be obtained.

Especially if we look at global alignment and local alignment maximum score matches, almost the same results are obtained due to the errors mentioned above.

In my opinion, either to give the gap penalty value close to 0, or to encourage it to replace the mismatch instead of mismatch by making the gap penalty value less than the gap penalty value, especially for the global alignment to work correctly.

Another arrangement which is not very important but needs to be made is to make the similarities (or lengths) of the given dna sequences as different as possible.