Project report

1 Linear gap penalty

1.1 Data testing

The algorithm of linear gap penalty is contained in linear_gap __penalty.py under the folder of codes. To testing this algorithm, I choose input1.txt, input3.txt, parameter1.txt and parameter3.txt as the testing data because their score for initiating a gap are 0, so we just need to fill in a V table. The test results are as follows:

Input1:

input1.txt/parameter1.txt								
seq1=tgacaatccc								
seq2=tgagcatggt								
		4:	running					
	score	entries	time/s					
DP	10	121	0.0013					
banded-DP	10	65	0.0013					
X-drop	10	43	0.0024					

Input3:

Input3.txt/parameter3.txt

seq1=gcggcgcaagatggcccaggagaaccccaagatgcacaactcggagatca gcaagcgcctgggcgccgagtggaaacttttgtcggagacggagaagcggccgttca tcgacgaggctaa

seq2=gcggcgcaagatggcccaagagaaccccaagatgcacaactcggagatcag caagcgcctgggcgcgagtggaaacttttgtcggagacggagaagcggccgttcat cgacgaggccaagcggctgcga

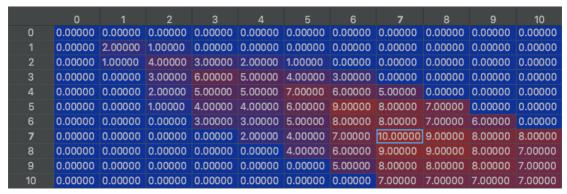
	score	entries	running time/s
DP	234	15851	0.3374
banded-DP	234	841	0.1931
X-drop	234	602	0.6111

1.2 Table check

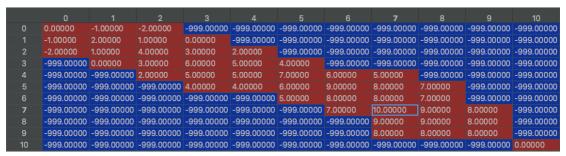
The V table we've filled in are as follows:

	0	1	2	3	4	5	6	7	8	9	10
0	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
1	0.00000	2.00000	1.00000	0.00000	0.00000	0.00000	0.00000	2.00000	1.00000	0.00000	2.00000
2	0.00000	1.00000	4.00000	3.00000	2.00000	1.00000	0.00000	1.00000	4.00000	3.00000	2.00000
3	0.00000	0.00000	3.00000	6.00000	5.00000	4.00000	3.00000	2.00000	3.00000	3.00000	2.00000
4	0.00000	0.00000	2.00000	5.00000	5.00000	7.00000	6.00000	5.00000	4.00000	3.00000	2.00000
5	0.00000	0.00000	1.00000	4.00000	4.00000	6.00000	9.00000	8.00000	7.00000	6.00000	5.00000
6	0.00000	0.00000	0.00000	3.00000	3.00000	5.00000	8.00000	8.00000	7.00000	6.00000	5.00000
7	0.00000	2.00000	1.00000	2.00000	2.00000	4.00000	7.00000	10.00000	9.00000	8.00000	8.00000
8	0.00000	1.00000	1.00000	1.00000	1.00000	4.00000	6.00000	9.00000	9.00000	8.00000	7.00000
9	0.00000	0.00000	0.00000	0.00000	0.00000	3.00000	5.00000	8.00000	8.00000	8.00000	7.00000
10	0.00000	0.00000	0.00000	0.00000	0.00000	2.00000	4.00000	7.00000	7.00000	7.00000	7.00000

DP(input1)



banded-DP(input1)



X-drop(input1)

2 Affine gap penalty

2.1 Data testing

The algorithm of affine gap penalty is contained in affine_gap __penalty.py under the folder of codes. To testing this algorithm, I choose input2.txt and parameter2.txt as the testing data because their score for initiating a gap are not 0, so we need to fill in V, E and F three tables. The test results are as follows:

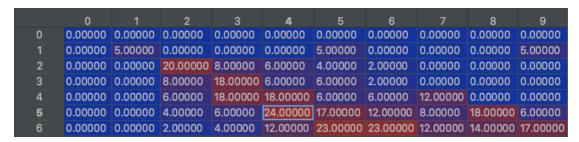
Input2:

Input2.txt/parameter2.txt							
seq1=AWGHEE							
seq2=AWHEAEHEA							
	GOONO	entries	running				
	score	entries	time/s				
DP	24	70	0.0011				
banded-DP	24	43	0.0013				
X-drop	24	21	0.0035				

2.2 Table check

The V, E and F table we've filled in are as follows:

DP:



DP(input2, V table)



DP(input2, E table)

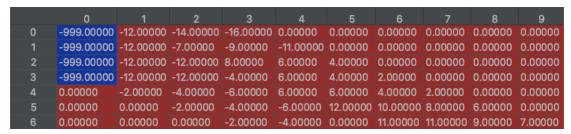
					4				8	9
0	-999.00000	-999.00000	-999.00000	-999.00000	-999.00000	-999.00000	-999.00000	-999.00000	-999.00000	-999.00000
1	-12.00000	-12.00000	-12.00000	-12.00000	-12.00000	-12.00000	-12.00000	-12.00000	-12.00000	-12.00000
2	-14.00000	-7.00000	-12.00000	-12.00000	-12.00000	-7.00000	-12.00000	-12.00000	-12.00000	-7.00000
3	-16.00000	-9.00000	8.00000	-4.00000	-6.00000	-8.00000	-10.00000	-12.00000	-12.00000	-9.00000
4	-18.00000	-11.00000	6.00000	6.00000	-6.00000	-6.00000	-10.00000	-12.00000	-12.00000	-11.00000
5	-20.00000	-12.00000	4.00000	6.00000	6.00000	-6.00000	-6.00000	0.00000	-12.00000	-12.00000
6	-22.00000	-12.00000	2.00000	4.00000	12.00000	5.00000	0.00000	-2.00000	6.00000	-6.00000

DP(input2, F table)

banded-DP:

	0	1	2	3	4	5	6	7	8	9
0	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
1	0.00000	5.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.00000	0.00000	20.00000	8.00000	6.00000	4.00000	0.00000	0.00000	0.00000	0.00000
3	0.00000	0.00000	8.00000	18.00000	6.00000	6.00000	2.00000	0.00000	0.00000	0.00000
4	0.00000	0.00000	6.00000	18.00000	18.00000	6.00000	6.00000	12.00000	0.00000	0.00000
5	0.00000	0.00000	4.00000	6.00000	24.00000	17.00000	12.00000	8.00000	18.00000	0.00000
6	0.00000	0.00000	0.00000	4.00000	12.00000	23.00000	23.00000	12.00000	14.00000	17.00000

banded-DP(input2, V table)

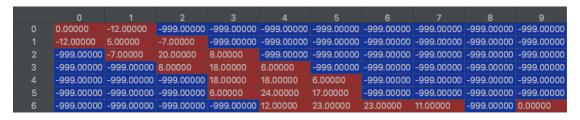


banded-DP(input2, E table)

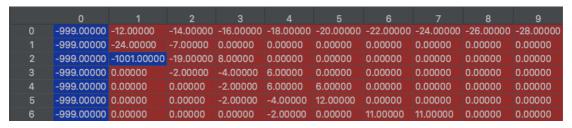
	0		2	3	4		6	7	8	9
0	-999.00000	-999.00000	-999.00000	-999.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
1	-12.00000	-12.00000	-12.00000	-12.00000	-2.00000	0.00000	0.00000	0.00000	0.00000	0.00000
2	-14.00000	-7.00000	-12.00000	-12.00000	-4.00000	-2.00000	0.00000	0.00000	0.00000	0.00000
3	-16.00000	-9.00000	8.00000	-4.00000	-6.00000	-4.00000	-2.00000	0.00000	0.00000	0.00000
4	0.00000	-11.00000	6.00000	6.00000	-6.00000	-6.00000	-4.00000	-2.00000	0.00000	0.00000
5	0.00000	0.00000	4.00000	6.00000	6.00000	-6.00000	-6.00000	0.00000	-2.00000	0.00000
6	0.00000	0.00000	0.00000	4.00000	12.00000	5.00000	0.00000	-2.00000	6.00000	-2.00000

banded-DP(input2, F table)

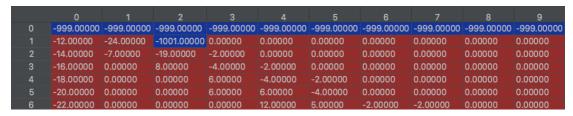
X-drop:



X-drop(input2, V table)



X-drop(input2, E table)



X-drop(input2, F table)

2.3 Other sequences' analysises

For more analysises, I download four pairs of homology protein sequences from http://eggnogdb.embl.de/#/app/downloads. The sequence file are under the folder of analytical_input. The test results are as follows:

analysis_input1.txt:

analysis_input1.txt/parameter.txt

 $seq1 = MGLSTKTIHVHRHQQHLKWSRDIPPVLTVNSGETVTFDALDGSNGMIARDSGDAILKFDVARADPGFGPIYVNEAEPG\\ DVLKVEILDLQLGDWAWTAIVPNFGLLADEFPEARLKIWDIDPSLPYVKFKEGVHIRKAPFLGIMAVAPGVEGEFSTIPPLET\\ GGNMDCRYLTAGSTLYLPIRTKGALFSCGDGHTAQGDGEVCGTAIETTLTASLRLTVVKNQPWVTSPQFQTPPLKQMLSAADIEADKGEYATMGIDCDLLEATRKATRNMIEWLTRTKDLTREEAYMLTSVAGCLKMAEVVDMPNYAIVMTIPLNIFVAPDWPNGTTT\\ WPNGTTT$

 $seq2 = MGLSTKTIHVHRHQQHLKWSRDIPPVLTIDSGETVTFDALDGSNGMITRESGDEAILNFDVERADPGFGPIYVNDAEPG\\ DVLKVDILDLQLGDWAWTAIVPGFGLLADDFLEARLKIWDIDSNLPYVKFKEGVHIRKAPFLGIMAVAPGVEGEFSTIPPLET\\ GGNMDCRYLTTGSTLYLPIRTKGALFSCGDGHTAQGDGEVCGTAIETTLTASLRLTVCKNQPWVTSPQFQTPPLKQTLSASD\\ IESDKGEYATMGIDSDLLEATRKATRSMIEWLTCTKDLTREEAYMLTSVAGCLKMAEVVDMPNYAIVMTMPLSIFVASDRP\\ NGTTT\\ \\$

	score	entries	running time/s
DP	2032	109230	6.1001
banded-DP	2032	2301	4.7067
X-drop	2032	991	11.005

analysis input2.txt:

analysis_input2.txt/parameter.txt

seq1=MNKKRFILSLLFIVFVIASLLIGVSSELTLKSLLSGNKEAYNVLFRSRMPRTLAIILSASSLSIAGLIMQSLSRNKFISPQTA GTNDAAAFGVLISFIFLGSIPSYYRLIFSFVFALISSLLFIFILNKIKFKNIIYVPLIGLMYGGLLSAVTRLIAYETNTLQLLSSLNL\\ GTFSHINVLNGSMLLILVPVLLIAIIFAEGFNVASLGEEFATNLGVNYKKVLFLGLIVTSLIAASTYLVVGPLPFLGLIIPNIVSM\\ YYGDNLKKNIIDIALLGSIFVLANDIISRLVIFPFELSVGFTMGITGAIIFILLIFKQVKNND

 $seq2 = MNKKRFLIIALFFILLIISLLIGVSNSVTIGALLKGDKNAWFIFLESRVPRTLVIVLTASSLSIAGLIMQAISRNKFISPSTAG\\ TTNAAVLGVLIGFLFFGTQSLYIRFIFAFVFALLSSILFITVLNKIKVKNVIYVPLIGMMYGTLISSITTLIAQRFNALQTVSMLN\\ LGSFSHITRLNSTLLIVLIPSLVLAVIFASQFNIVSLGEDFSKNLGVNYKRIMFIGLIVISVISASTYIVVGPLPFLGLIIPNIVTVYY\\ GDNLKRNLFDVAIFGATFVLLNDIISRLLIFPYEISVGFTMGISGAIIFLILIFRKVKQK$

	score	entries	running time/s
DP	1389	102720	5.3258
banded-DP	1389	2231	4.0357
X-drop	410	329	5.8234

Comparing the two table above we can find that the banded-DP is the fast, because it just need to calculate the entries within the bandwith, the DP is the second one and the X-drop is the slowest one because if the max value is the last entry of the V table, this algorithm not just need to calculate the whole table, but also need to judge if there is a consistent entry. But for the number of entries, the X-drop always be the fewest, because the number of consistent entry is few, and for DP and banded-DP, both of them need to fill in a specified number of entries.

But we can also find that in the second result table, the X-drop get a small score, is that wrong? No, because the X-drop is a extentional algorithm, and the extention depends on the threshold value X, once it judge there is no more consistent entry then it will stop, so the choose of threshold's value X is very important.

analysis_input3.txt:

analysis_input3.txt/parameter.txt

seq1=MAGSHPVAENPPAVQVASPFGFVRALEWAAAERLPQMATYWEIILYIQENLAMANGAVRPLEALKNWGDDFDPFAR RAYSWLLVHHESNSSESAIAGISHTRPRLRQQIMDSLASHERINRNTNVKGAVDQTFPDSIKSEVDFWSQEAITKSINNRTPAA SAIADIPQTRQQQAVYIREMFDALKNTQDIIEKASNHKVGTVKSTSGRVFQDIAWVLYREARNLQQGKPGVKPWCTSFKYK QYPTMRARWNDMVEFFQTSKAAVANLIVAHSEKRFAGNPTKEKQRKTVNDKGNKKKATKMKDNEEKAARLDKVVKDA KGKAAAAKSAPDQGTNEEWDAEASEDLGGDGEAEETDYEAEEFDDSAREREDNDHESDDEATHLLQNAQQDVDRYDDIP EDQDSDDLKFQFGRAMGDSAASTVPLAGTSQPHAPAYPPLPGSISHRRQVRAIFPDEGHPPSIMPSNDHESYHPCYSGGLVSP NVQGQCIDDARSNDLGSSPLSGQPMDYLEHTLHRYISPSSPELTSGIGFMGQAQPQEGRGSDTAVGTSRIAHMPLNTGTSGV GGGAQNNSDPQDSTNSSRSRARRNRHSSEDGSKTRAEAPPRRRPRI

seq2 = Magnhpiagsppvvqvaspfgfvralecaaaemlpqmstyweiilyiqenlamangavrplealknwgddfdpfarr ayswllvhhesnssesaiagishtrprlrqqimdslasherinrnthvkgavdqtfpdsiksevdfwsqeaitksinnrtpaas aiadiprtrqeqavyiremfdalkntqdiiekasnhkvgtvkstsgrvlqdlawvlyrearnlqqgkpgvnpwctsfkykq yptlrarwndmveffhtskaavanlivthsekrfagnptkekqrktvndkcnkkkatrmkdneekaatldkvvkdakg kvataestpnkgideerdaedsvdlddeaeefddsegepkdtdnesddeathllqnaqqdvgryddikeeqesddlnpqsg qamgdsaasttafaasshphapayppspgligrrrqvraispdeghhssimpsiehesygpfhaggvvsqndhdryvddang ndrgslqlpsqpvnypeqplhqhtspsppelpsgigfvkeeeipqsvvttshiapmplntgtsgvgggaqdntdlrdstnssrs rarrnrhssedssdshaqararrpri

	score	entries	running time/s
DP	3117	362355	72.9901
banded-DP	2115	4159	73.5448
X-drop	2102	1141	97.8880

analysis_input4.txt:

analysis input4.txt/parameter.txt

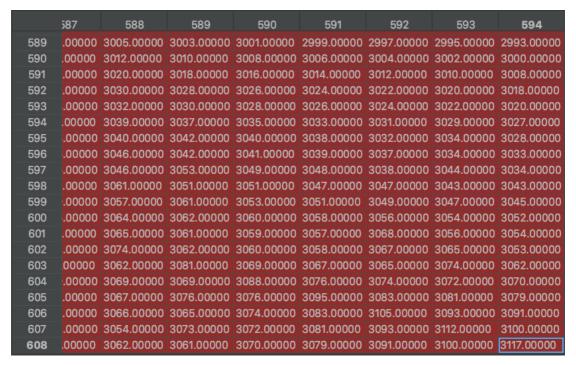
seq1=MLKFIKNNKWWVAIISVFAVFLSSFGIFAKSFVDSNKQKVVNKVENYIKASSYAVQSRILKETENLNEDYLNQKIGKKT
LLNEFNNNFIWQPNNNKTTSLDTISDLWKTYFGSTDNVLRKDLQIQYQNNNKLENIKSSKGEITPKNIDFLLTISKSLEKFLNG
FAPSLASLGISFLQSTVLQHRDDPNFEKYKSGITSVANAIEDNRETFNYLGKVLTPANLDENYYKNLTVKQAITKNINKLASVI
TNNKDFASQTDVDKLPEALDKLLVDLELDSITNIFSDIDFKNFKAEKISEIFPKIKNLFNNGTFTKLKQKSLEILNKITPHLATY
LYSELFFGLYYVANSDLKNPEDLLKQKVDDSVFIAFTKNKLDLNVLVDGLVKVLKNKKDFERLYDFIFKRFDQNKIFNNINT
LGTDTGTDSLLYDLINWLEKKLYNVSNTLSTIIRFAELALNDANIQKTIKEKLVEFIKSKLPKISVGSWIVNFNKDELEVSLKIY
LGFIPLRTPLQLKADFFGKTGLINNLINILKSFNSSINYAFEEFFKYIKNTLYLNTNKKFTLSPFVELVNNLNAFLKNDKNVYIS
LAKGLIGDLEIKTIYDFITLPYNKEFLNSLVQKYAGKDLQPVLDKIKAFLESLKTYGFIKEPEKLKEQFPQYLENLSKHLATYE
NNDLLNFNLLDSLYEGNIISDFISKWINFLTKDITKEDNPILPILRAINKSDKLEKLEQIKNQWTSKVSDLAKKIDNYSNISKIRN
IKLQLPKELVEQFGLQSLDGLNITELLEGLSNYIKDYLKANPNKVIGFNISSIGMMLYALTVKVGVEFKKELSKNNFLYNKNT
KQDKSKTVLKALADGFDSHDNSSDVGRDSTLNRKDQSYYNWDKIYFYINGFDKPYILDRTNLKEEFSYSPLHMLIGINPDKT
TYFKGSIGYAIGSLFGGLNTTDPNYNLSIENKNDATGILNVFNYVLDQKDKELKKHEDQIATQYYDKDAWETKVINSSEDEI
NYELIRLKSSKTQESKQLGSRFKVKLAKKKHSSYWEITQIIAVDYKAA

seq2=MLKFIKNSKWWVIIISIFAIFFSSFGIFAKSYVDSNKQKVVNKIQNYVQASSYAVQSRILKETENLNEDYLNEKIGKKSLL
NEFNSDFIWRPNNSKTTSSDTISDLWKTYFGSSNNVLNKNLQIQYKNNNEYKNIESSKGEITPKNIDFLLSISKSAEKFLNGFAP
SLASLGLSFLQNKVLNNRDDSKLQSYKDGINKFADVIENNKDLFSYLGKLFTPEPLEKDYYKDLTVQQALVKNINQIAAVIA
DNKEFAKETEVDKIPEALDKILAELGLDSLGEIIGELISSKNGFQNLGKIFSKIKNIFTSQNFQKLKTKGLELINKITPYLVTYLY
SEIFFGLYYVTHEEFKNPSDLTKQKVNVSDFSALVNNKIDLNVLINGFLKVLKDKKSFDRFYNFIFKRFNENEIFNSKNNIGSN
NGVGNLIFDLINWVEKKLHNFSNLLETLIKFVDFAMSDEKIKKTFEENIKNFITKKLNELGTPLGKWHVEIKNGILNISASSLW
LGSLTAKIEVFSKDGIIPKIIDVLKKIHGIINDSSETIIKHIKEIFYLAKDNILDFSITSKNISEIIMTFKELLINKKILNVNVKALFIK
LLDISSIYDILELPYSNSALKWILERFAKDKIEPHLKKIKSISETLQKNKFIQNADKIKEQLPKYLDLFSSHFKKYEKSSDLKFNL
NISLYKGNIIEDLILKWINFLLQDAENKENPLLPIIRLITKKNSLKTLDDIKNEWVSKITNISKKIESFQNISKIRDKKIDIPKELLK
YFGLESINNQTILRLLEILGKYFDDYLSKNPNKVVGVNISSLGKVLTALTIKVGVEYPPENKDKNFLYSKDIKENKTKTILKAL
VYGFDTHDNSSDVGSDSIKNRKPESYYNWDKIHFYINGSSQAITLDRTKLKDDSSYSPLHMLLGINPDKSSYLKDSLGYVFGT
LFGGLPASDSNYQLSIENKTDVTSILNVFNYVLDKKDKELKKQEEQIATKYYDKNAWSTKILSSNENEIIYQLIRLKSSDTKES
KRLGTRFEVRLLKNKNNPYWSINRVIALGYKAV

	score	entries	running time/s
DP	3954	1103550	767.9343
banded-DP	3954	7341	732.0790
X-drop	1386	862	903.6858

Comparing the last two table above we can find that with the extention of the sequence, all of the algorithm becoming very slow, but there is also reasonable, because the matrix is become bigger and bigger, which leads to the algorithm's running time increasing in an exponential way.

Another interesting thing we can find is that the score of banded-DP also become smaller like the X-drop we've talked above, is that wrong? Of course not. As we all know, the standard Smith Waterman dynamic programing need to fill in the whole table, so whatever the dimension of the matrix is, it can always get the max entry. But for banded-DP algorithm, it just need to calculate the entries within the bandwith section, so what if the two sequence are not the same length and the max entry is the last one of the matrix, it means the max entry is not in the section of bandwith, so it can't find the correct max entry. Here is an example below:



The real max entry is the last one of this matrix



The bandwith section dosen't contains the real max entry

So we can see, the choose of the bandwith value is important for banded-DP algorithm, if the two sequence are not the same length, you might need to set a bigger value of bandwith to make sure you will not miss the max entry.