RNA-CDS Mapping

Purpose: Mapping for RNA(with NM, XM ID) and CDS(with NP, XP)

Program : parsing_mapping.py

Input :

(1) "GCF_000001405.39_GRCh38.p13_rna_from_genomic.fna"

(2) "GCF_000001405.39_GRCh38.p13_cds_from_genomic.fna"

(3) "GCF_000001405.39_GRCh38.p13_rna.fna"

(4) "GCF_000001405.39_GRCh38.p13_genomic.gff" : Annotation file

* **rna_from_genomic vs rna**

- 전자는 후자의 각각의 특정 NM, XM ID에 대한 여러 버전의 Sequence들을 정렬해놓음.

- NM_ID가 같아도 seq_ID가 다르다면, Sequence 내용도 조금씩 다름

- 즉, 하나의 sequence의 식별키 =, [seq_id, gene, Transcript_ID]

* "**GCF_000001405.39_GRCh38.p13_genomic.gff"**

- rna_from_genomic과 cds_from_genomic에 대한 bijective mapping을 위한 reference

-즉, 모든 CDS에 대한 RNA matching 가능함. – Full matching

Output : RNA_CDS_map(final).csv

-RNA_ID와 CDS_ID을 matching을 시키고, 각각의 ID에 맞는 서열을 불러와,

RNA 속, CDS region을 찾은 내용.

Function: (1)annotation_file(file_name) : annotation원본 파일 전처리(NM, XM, NP, XP 만 필터링)

(2)parsing_fasta(file_name) : fasta 파일 dataframe화

(3)RNA_CDS_Mapping(rna, cds, rna_seq, annotation)

Algorithm:

(1) annotation_file

-annotation을 이용한 mapping을 위해서는, annotation에서 sequence에 대한 seq_id, gene, Protein_ID(RNA는 Transcript_ID)를 가져와야한다.

-이때, annotation에서 CDS에 대한 정보만 가져와서 확인해본 결과, 해당 CDS에 대한 "Parent", 즉, RNA sequence의 ID 정보까지 포함하고 있다.

-CDS의 annotation data 상의 Transcript_ID(Protein_ID의 Parent)의 종류 개수와 RNA annotation data 상의 Transcript_ID 종류 개수가 같고, 각각의 gene pool이 동일하다면, 모든 CDS에 대한 RNA full mapping이 가능하다는 것을 나타낸다.

-annotation_file output : annotation file, 이후 annotation만든 결과, full matching이 가능하다는 것이 확인 되면 다음과 같은 문구가 나온다.

-"Yes, there is full match in cds annotation : [mapping 가능한 seq 개수] sequences"

-"Yes, genes in both annotation are also matched! : mapping이 되는 gene 개수)

-Column : seq_id Gene Parent Protein_ID Transcript_ID

(2) parsing_fasta

-**output :** fasta_di(Dictionary)

-fasta 파일 원본을 DataFrame형태로 바꾸는 function

-Biopython module을 이용해서 Sequence ID, Description, Sequence를 parsing

-RNA_from_genomic, CDS_from_genomic, RNA 의 각각의 DataFrame을 딕셔너리로 묶음.

(3) RNA_CDS_Mapping

-**output:** mapping(pandas dataframe)

-Annoation은 CDS_from_genomic의 NP, XP서열을 모두 full matching 하도록 만든 파일

- mapping base를 Annotation으로 설정

- 서열 ID(Transcript_ID, Protein_ID)에 맞는 서열들을 fasta 파일 dataframe 에서 찾아온다.

-fasta Dataframe 속 서열을 불러오기 위한 row index 찾기 : groupby 이용

-fasta Dataframe과 Annotation을 concat 후, ["seq_id", "Gene", "Trnascript_ID"]로

-groupy를 하면, 각각 특정된 row의 index가 하나씩만 가져오게 된다.

-이 중, fasta Dataframe의 row index만 가져와서 해당 index의 Sequence를 추출

-Column : Gene seq_id Transcript_ID Protein_ID CDS_seq Full_seq Full_length

CDS_length CDS_start CDS_end RNA_seq RNA_length start end

Discussion

(1) df.to_excel : character length limit(32467)

-parsing_fasta 이후, fasta_di(Dictionary) 속 dataframe을 excel(.xlsx)로 변환시키면 길이가 32467(2^15)을 넘는 서열은 초과한 부분이 잘린 상태로 저장됨.

-이후, mapping시 cds region을 못찾는 경우가 발생함.

-solution : df.to_csv를 통해서 저장

(2) NCBI Database RNA-CDS Sequence Problem

```
In [114]: mapping.loc[mapping["CDS_start"] == 0][col]
Out[114]:
        Gene   Transcript_ID      Protein_ID  Full_length  CDS_start
71509   OAZ1  NM_001301020.1  NP_001287949.1         1175          0
71510   OAZ1     NM_004152.3     NP_004143.1         1181          0
71511   OAZ2  NM_001301302.1  NP_001288231.1         1931          0
71512   OAZ2     NM_002537.3     NP_002528.1         1934          0
71513   OAZ3  NM_001134939.1  NP_001128411.1          852          0
71514   OAZ3  NM_001301371.1  NP_001288300.1          739          0
71515   OAZ3     NM_016178.2     NP_057262.2          887          0
75917  PEG10  NM_001172437.2  NP_001165908.1         6629          0
75919  PEG10  NM_001184961.1  NP_001171890.1         6618          0
75921  PEG10     NM_015068.3     NP_055883.2         6618          0
```

다음과 같이 "CDS_start"가 "0"인 data들은 RNA에서 CDS region을 찾지 못한 경우를 보여준다.

- **Ribosomal Framshifting**

Reason 1] Gene : OAZ1, OAZ2, OAZ3 인 서열들 : a+1 ribosomal Framshifing

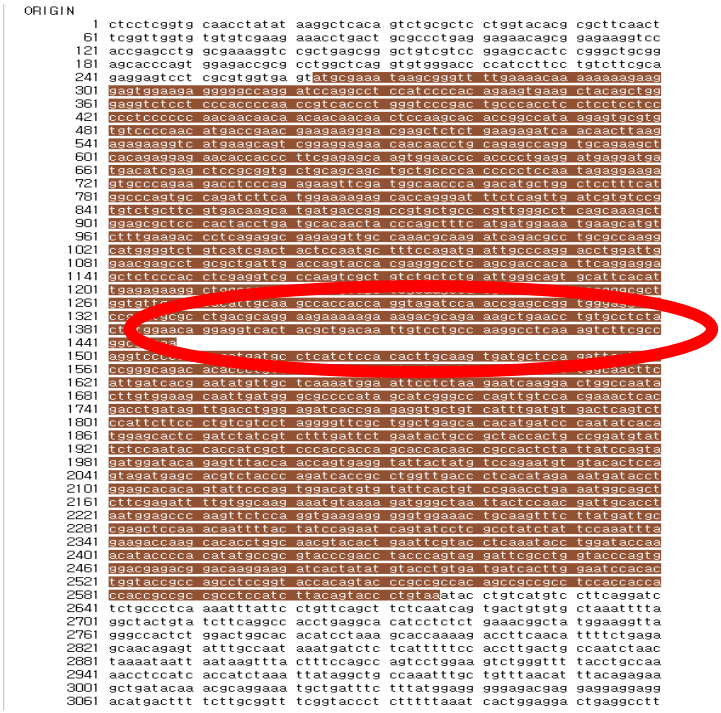
```
ORIGIN
        1 agcatctata aaggcgggcg gcggcagagg cgccattttg cgaacggcga gcagcggcgg
       61 cggcgcggag agacgcagcg gaggttttcc tggtttcgga ccccagcggc cggatggtga
      121 aatcctccct gcagcggatc ctcaatagcc actgcttcgc cagagagaag gaaggggata
      181 aacccagcgc [illegible]tccac gccagccgca ccatgccgct cctaagcctg cacagccgcg
      241 gcggcagc[illegible] cagtc[illegible]gagg gtctccctcc actgctgtag taacccgggt ccggggcctc
      301 ggtggtgc[illegible] ctgat[illegible]ccc tcacccaccc ctgaagatcc caggtgggcg agggaatagt
      361 cagagggat[illegible] acaa[illegible]tttc agctaactta ttctactccg atgatcggct gaatgtaaca
      421 gaggaactaa [illegible]ccaacga caagacgagg attctcaacg tccagtccag gctcacagac
      481 gccaaacgca ttaactggcg aacagtgctg agtggcggca gcctctacat cgagatcccg
      541 ggcggcgcgc tgcccgaggg gagcaaggac agctttgcag ttctcctgga gttcgctgag
      601 gagcagctgc gagccgacca tgtcttcatt tgcttccaca agaaccgcga ggacagagcc
      661 gccttgctcc gaaccttcag ctttttgggc tttgagattg tgagaccggg gcatcccctt
      721 gtccccaaga gacccgacgc ttgcttcatg gcctacacgt tcgagagaga gtcttcggga
      781 gaggaggagg agtaggccg cctcggggct gggcatccgg cccctggggc cacccctgt
      841 cagccgggtg ggtaggaacc gtagactcgc tcatctcgcc tgggtttgtc cgcatgttgt
      901 aatcgtgcaa ataaacgctc actccgaatt agcggtgtat ttcttgaagt ttaatattgt
      961 gtttgtgata ctgaagtatt tgctttaatt ctaaataaaa atttatattt tacttttta
     1021 ttgctggttt aagatgattc agattatcct tgtactttga ggagaagttt cttatttgga
     1081 gtcttttgga aacagtctta gtcttttaac ttggaaagat gaggtattaa tcccctccat
     1141 tgctctccaa aagccaataa agtgattaca cccgaaaaaa aaaaaaaaa
//
```

전체 RNA 서열과 CDS 서열이 NCBI database와 일치하나, 다음과 같이 CDS region이 중간 "T"를 기점으로 분리되어있음.(T의 염기서열 삽입)

단, CDS fasta 파일의 CDS 서열은 해당 "T"를 제외하고 분리된 CDS region을 연속하게 끔 만든 서열. (즉, CDS 서열에 저 "T"가 빠져있음.)

따라서, 정확한 mapping이 가능함에도, CDS 서열의 부정확함(?)때문에 해당 7개 서열이 mapping 불가능.

Reason 2] Gene: PEG10 인 서열 : a-1 ribosomal frameshifting



NCBI 전체 RNA 서열 상, 공백이 있는 것으로 보임. 이것은 특정 인덱스로 분리된 전후 CDS 서열이 한 염기서열을 중복으로 겹쳐서 존재하기 때문(다음사진은 1447번째 서열을 기준으로 전과 후로 나뉜 각각의 CDS 서열에 1447번째 염기가 중복하여 존재) 그러나, CDS fasta 서열 상, 중복된 염기서열를 기준으로 분리된 CDS로 존재하는 것이 아닌 연속된 하나의 서열로 존재함.

따라서, 하나의 연속된 RNA서열상에서 염기서열 중복을 고려하지 않은 연속된 CDS 서열의 CDS region을 찾는 것은 불가능