Aufgabe 6

##Erstellen Sie einen Fasta File Parser:

Schreiben Sie eine Funktion, die ein Fasta File (WS2016/examples/sequence.fasta) einliest, parst und in ein Daten-Objekt einfügt.

Fasta Format:

Das Fasta oder Pearson Format ist ein simples textbasiertes Format zur Darstellung und Speicherung der Primärstruktur von Nukleinsäuren oder Arminosequenzen.

Eine Sequenz im FASTA-Format beginnt mit einer einzeiligen Beschreibung, dann folgen die Sequenzdaten. Eine Sequenz endet mit dem Auftauchen einer weiteren Kopfzeile oder dem Ende des Files.

Die Kopfzeile wird mit einem `>` Zeichen eingeleitet.

```
> gi|1032940704|ref|XM_007620033.2| PREDICTED: Cricetulus griseus zinc
finger protein 862 (Znf862), transcript variant X7, mRNA
TTGTATGGGAACCAAAGATGAGCCCATATGTTTCCAGTCCCACACTGAGCCAGGGTCTAAGCAAGTCGAC
CCACAAAGGAGGGACCTGTGTATGGAGTGTGCAGGTGCTCAGCCACTTGCAGGGAGGAACTGGAGACAGG
AGAACTATCCTGGAGCAGCGA
> gi|1032940689|ref|XM_007620031.2| PREDICTED: Cricetulus griseus GTPase,
IMAP family member 6 (Gimap6), mRNA
GGAGAGGACACATGAGTCTGGTGTTCTCGGTGTACGAATGTTATGGTGTTTGTGGAGATCTCATAAGAAA
GAAAAAACCTCCCTCAGTCTCTCATACCTACAGATGAAGGAAGATTATTGACAGGTGGATGACAGCACTG
ATGCTGAAGACACGGCTGAGCTTGCCATTTTTATTAGCACTATTGATGACAAGTATAACATCCCTGAAGA
AATGGCATCTTAGTGCCATTAAAACACACAACTGGGGACCAGGAGGTGGCTCCGCAGTTTTCAAATGGA
```

Beschreibung:

Bilden Sie Daten in einer Sequenz von Dictionaries ab. Bauen Sie das Daten Objekt wie folgt auf. Erzeugen Sie pro Sequenz ein Feld **'raw'** indem Sie den Originalinhalt der Sequenz speichern. Weiters erzeugen Sie ein Feld **'id'** in der Sie den ersten Teil der Kopfzeile (bis zum ersten Leerzeichen) ohne dem führenden `>` speichern. Den zweiten Teil der Kopfzeile speichern Sie im Feld **'description'** ab. Die Sequenz selbst speichern Sie im Feld **'sequence'** ab und entfernen jede Art von ***whitespace*** Zeichen.

```
[{'raw':">gi|1032940704... ...ATGA",
   'id':"gi|1032940704..",
   'description':"PREDICTED: Cricetulus grise...",
   'sequence':"TTGTATGGG..."},
 {'raw':">gi|1032940689... ...ATGA", ...}]
```

Zum Einlesen eines Text Files verwenden sie die Builtin Funktion `open()` der Sie den File Pfad und den Mode übergeben. Für den Mode verwenden sie **"r"** (Read) zum Lesen und **"w"** (write) wenn Sie ein File zum Schreiben öffnen.

```
file_handle = open("filename", "r")
```

Den File Handle können Sie dann einfach in einer for Schleife zeilenweise durchlaufen.

```
for line in file_handle:
    print(line)
```

Um zu überprüfen, ob es sich bei der Zeile um eine Kopfzeile oder Sequenz handelt verwenden Sie die `startswith()` Methode der line Zeichenkette.

Übergeben Sie den Filenamen als Parameter und geben sie als Rückgabewert das Daten Objekt zurück.

```
seq_data = parse_fasta("sequence.fasta")
```

Erstellen Sie einen Genbank File Parser:

Schreiben Sie eine Funktion die ein Genbank File (WS2016/examples/sequence.gb) einliest, parst und in ein Daten Objekt einfügt.

Genbank Format:

Das Genbank Format enthält weit mehr Informationen als das simple FASTA-Format und ist auch dementsprechend viel schwerer zu parsen. Eine Beschreibung des Formats finden Sie in http://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html.

```
LOCUS       XM_007620033            7009 bp    mRNA    linear   ROD 27-MAY-2016
DEFINITION  PREDICTED: Cricetulus griseus zinc finger protein 862 (Znf862),
            transcript variant X7, mRNA.
ACCESSION   XM_007620033
VERSION     XM_007620033.2  GI:1032940704
DBLINK      BioProject: PRJNA239316
KEYWORDS    RefSeq.
SOURCE      Cricetulus griseus (Chinese hamster)
  ORGANISM  Cricetulus griseus
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Euarchontoglires; Glires; Rodentia;
            Sciurognathi; Muroidea; Cricetidae; Cricetinae; Cricetulus.
COMMENT     MODEL REFSEQ:  This record is predicted by automated computational
            analysis. This record is derived from a genomic sequence
            (NW_006878416.1) annotated using gene prediction method: Gnomon.
            Also see:
                Documentation of NCBI's Annotation Process

            On May 27, 2016 this sequence version replaced gi:625253769.

            ##Genome-Annotation-Data-START##
            Annotation Provider         :: NCBI
            Annotation Status           :: Full annotation
            Annotation Version          :: Cricetulus griseus Annotation
                                           Release 102
            Annotation Pipeline         :: NCBI eukaryotic genome annotation
                                           pipeline
            Annotation Software Version :: 7.0
            Annotation Method           :: Best-placed RefSeq; Gnomon
            Features Annotated          :: Gene; mRNA; CDS; ncRNA
            ##Genome-Annotation-Data-END##
FEATURES             Location/Qualifiers
     source          1..7009
                     /organism="Cricetulus griseus"
                     /mol_type="mRNA"
                     /db_xref="taxon:10029"
                     /chromosome="Unknown"
                     /country="China"
                     /collection_date="Feb-2011"
     gene            1..7009
                     /gene="Znf862"
                     /note="Derived by automated computational analysis using
                     gene prediction method: Gnomon. Supporting evidence
                     includes similarity to: 10 Proteins, and 100% coverage of
                     the annotated genomic feature by RNAseq alignments,
                     including 19 samples with support for all annotated
                     introns"
                     /db_xref="GeneID:100766003"
     CDS             5249..5644
                     /gene="Znf862"
                     /codon_start=1
                     /product="zinc finger protein 862 isoform X4"
```

```
                     /protein_id="XP_007618223.1"
                     /db_xref="GI:625253770"
                     /db_xref="GeneID:100766003"
                     /translation="MNQRRFAWSRACATGITLRMLTALLQDVHLLGESAVRPSNPPEQ
                     SIAHPVMTAHSFALPVIIFTTFWGLIGIAGPWFVPKGPNRGVIITMLVATAVCCYLFW
                     LIAILAQLNPLFGPQLKNETIWYVRFLWE"
ORIGIN
        1 ttgtatggga accaaagatg agcccatatg tttccagtcc cacactgagc cagggtctaa
       61 gcaagtcgac ccacaaagga gggacctgtg tatggagtgt gcaggtgctc agccacttgc
      121 agggaggaac tggagacagg agaactatcc tggagcagcg tgtgaccgag ctagagagat
//
```

Beschreibung:

Erzeugen Sie wie im vorigen Beispiel ein Daten-Objekt und parsen Sie den Inhalt in die entsprechenden Felder. Verwenden Sie das Feld **ACCESSION** im Feld **'id'**. **DEFINITION** im Feld **'description'** und **ORIGIN** im Feld **'sequence'**. Zusätzlich generieren sie ein neues Feld **'features'** im Daten-Objekt und parsen Sie alle **gene** Einträge in eine Sequenz aus Dictionaries mit den Feldern **'position'**, **'name'**, **'description'** und **'id'** und ein Feld **'organism'** in das den Organismus hinein parsen.

```
[{'raw':"LOCUS       XM_007620033... ...//",
  'id':"XM_007620033",
  'description':"PREDICTED: Cricetulus grise...",
  'sequence':"TTGTATGGG...",
  'organism':"Cricetulus griseus",
  'features': [{'position':"1..7009",
                'name':"Znf862",
                'description':"Derived by automated...",
                'id': "GeneID:100766003"}]},
...]
```

Erkunden Sie die nötigen String Methoden (https://docs.python.org/3/library/stdtypes.html#str) und verfahren Sie wie im vorigen Beispiel.

```
seq_data = parse_gb("sequence.gb")
```

Erstellen Sie nützliche Funktionen für das Sequenz-Objekt

Jetzt wo wir die Daten in einem für Python brauchbaren Format haben, erstellen wir nun diverse Funktionen um mit den Daten zu arbeiten.

Erstellen Sie folgende Funktionen:

- **get_raw(db, index))** - gibt den ***raw*** String des indizierten Sequenz-Objekt zurück
- **get_id(db, index))** - gibt die ***id*** des indizierten Sequenz-Objekt zurück
- **get_description(db, index))** - gibt die ***description*** des indizierten Sequenz-Objekt zurück
- **get_sequence(db, index)** - gibt die ***sequence*** des indizierten Sequenz-Objekt zurück
- **get_fasta(db, index)** - Kreiert aus ***id***, ***description*** und ***sequence*** eine FASTA-Sequenz und gibt diese als String zurück. Die Zeilenlänge soll bei der Sequenz nicht 80 Zeichen überschreiten. Fügen Sie einen `\n` *new line* Zeichen nach jeweils 80 Zeichen ein um einen mehrzeiligen String zu generieren.
- **get_feature(db, index, feature)** - Gibt das gesuchte Feature zurück
- **add_feature(db, index, feature, value)** - Fügt ein neues Feature zu einem bestehenden Daten-Objekt hinzu. Zum Beispiel ein **'organism'** Feature zu einer FASTA-Sequenz.
- **add_sequence_object(db, id, description, sequence, **features)** - Fügt ein komplett neues Daten-Objekt, ohne zuvor ein File zu parsen, hinzu.
- **get_gc_content(db, index)** - Berechnet den GC-Gehalt von Nucleotid Sequenzen.
- **get_output(db, index, type='markdown')** - Formatiert den Output zum Beispiel als markdown, html oder csv output (Advanced)

Wobei **db** unser Daten-Objekt von den vorherigen Aufgaben ist. **index** der Index des gesuchten Sequenz-Objekts in **db** und **features** ein beliebiger Feature Name.