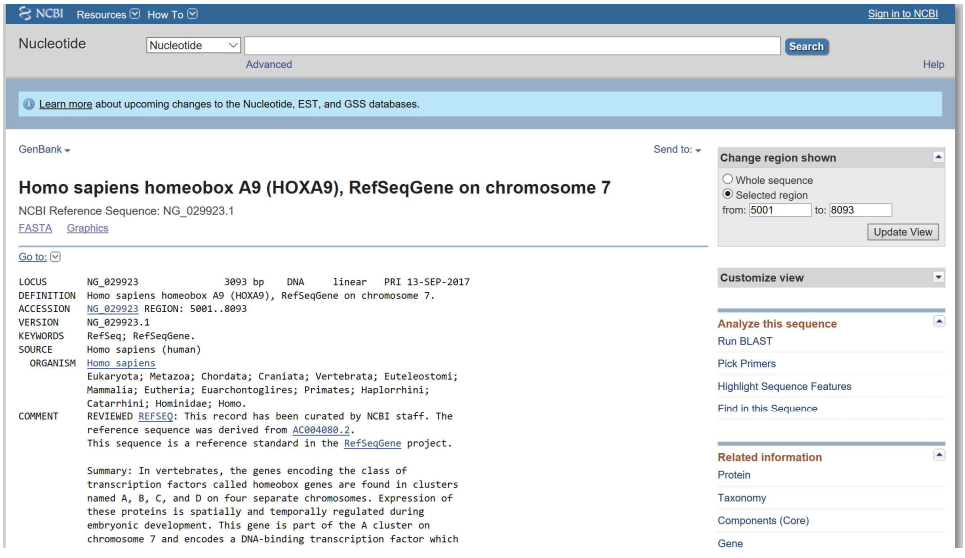
Sequências

- Sequências: formato GenBank
- BioPython: objetos SeqIO, SeqRecord, SeqFeature

NCBI- format GenBank



Formato GenBank

- Começa com um header iniciado pela palavra chave LOCUS e que tem um conjunto de outros campos identificados por um título em maiúsculas
- No meio tem a tabela de "FEATURES" com a anotação, i.e. a informação biológica relevante
- No final tem a sequência iniciada pela palavra ORIGIN; no início de cada linha tem o nº da posição
- Cada registo termina com // (terminador)

```
LOCUS       NT_037436             64721 bp    DNA     linear
DEFINITION  Drosophila melanogaster chromosome 3L.
ACCESSION   NT_037436 REGION: complement(1036369..1101089)
VERSION     NT_037436.4
DBLINK      BioProject: PRJNA164
            BioSample: SAMN02803731
            Assembly: GCF_000001215.4
KEYWORDS    RefSeq.
SOURCE      Drosophila melanogaster (fruit fly)
  ORGANISM  Drosophila melanogaster
            Eukaryota; Metazoa; Ecdysozoa; Arthropoda; Hexapoda;
```

```
FEATURES             Location/Qualifiers
     source          1..64721
                     /organism="Drosophila melanogaster"
                     /mol_type="genomic DNA"
                     /db_xref="taxon:7227"
                     /chromosome="3L"
                     /genotype="y[1]; Gr22b[1] Gr22d[1] cn[1] CG33964[R4.2]
                     bw[1] sp[1]; LysC[1] MstProx[1] GstD5[1] Rh6[1]"
     gene            1..64721
                     /gene="bab1"
                     /locus_tag="Dmel_CG9097"
                     /gene_synonym="añon-WO0118547.639; bab; BAB; BAB-1; bab-I
                     Bab1; BAB1; bric-a-brac; CG13910; CG9097; Dmel\CG9097"
                     /note="bric a brac 1"
                     /gen_map="3-0.5 cM"
                     /map="61E2-61F1"
                     /db_xref="FLYBASE:FBgn0004870"
```

```
ORIGIN
        1 ctcgtgctga gaagactttt ggccctgttt gttggccaat acagccatag aacactga
       61 ccgaaacaga aacagaaacg gaactgaaag cacacactga ataacactgg acacgacg
      121 gagaaccgaa tcgaactgaa agaacgaggc ggcgggctaa ttgccaaatg gccacgta
      181 gcgcgtaacg taacgaatac gttaaacacc catacacaca cgcccagaca cacacaca
      241 cacacacaca cacagagaga cagaggcacg cacacagata cgcacatgta catgccca
      301 ctcagttgag cgtcgtaatt gtcgcggatt cttttgaatg cttattcctt aaacgcta
      361 tgaagagcga atcgcgtaac gaataacgat cgcgtaacgt ttacgatcgc cggtgatc
      421 acgatcaacg atcaacgatc agctgcagac aaaatggcgt cggcgcaggc ggagacga
      481 gtcggcttgg cgtccgaaca gggaccagtg gctcagaggc agcgcaaagg gacgggat
```


Formato GenBank

LOCUS:

LOCUS DMU54469 2881 bp DNA linear INV 22-FEB-1998

- Locus name
- Tamanho da sequência
- tipo
- divisão
- Data submissão

Divisões do NCBI:
BCT – bactérias
INV – invertebrados
MAM – outros mamíferos
PRI – primatas
PLN – plantas
…

DEFINITION:

DEFINITION Drosophila melanogaster eukaryotic initiation factor 4E (eIF4E) gene, alternative splice products, complete cds.

Na linha de definição temos um sumário do conteúdo biológico do registo

Formato GenBank

ACESSION:

ACCESSION U54469

Código do registo (chave primária)

VERSION:

U54469.1 GI:1322283

Acession.version
GI:geninfo identifier; cada versão de um registo tem um GI diferente

KEYWORDS:

KEYWORDS .

Este campo serve para colocar palavras chave sobre o registo. O seu uso é desencorajado por muitos.

SOURCE:

Organismo e informação taxonómica

SOURCE Drosophila melanogaster (fruit fly)
ORGANISM Drosophila melanogaster
Eukaryota; Metazoa; Arthropoda; Hexapoda; Insecta;
Pterygota; Neoptera; Endopterygota; Diptera; Brachycera;
Muscomorpha; Ephydroidea; Drosophilidae; Drosophila.

Formato GenBank

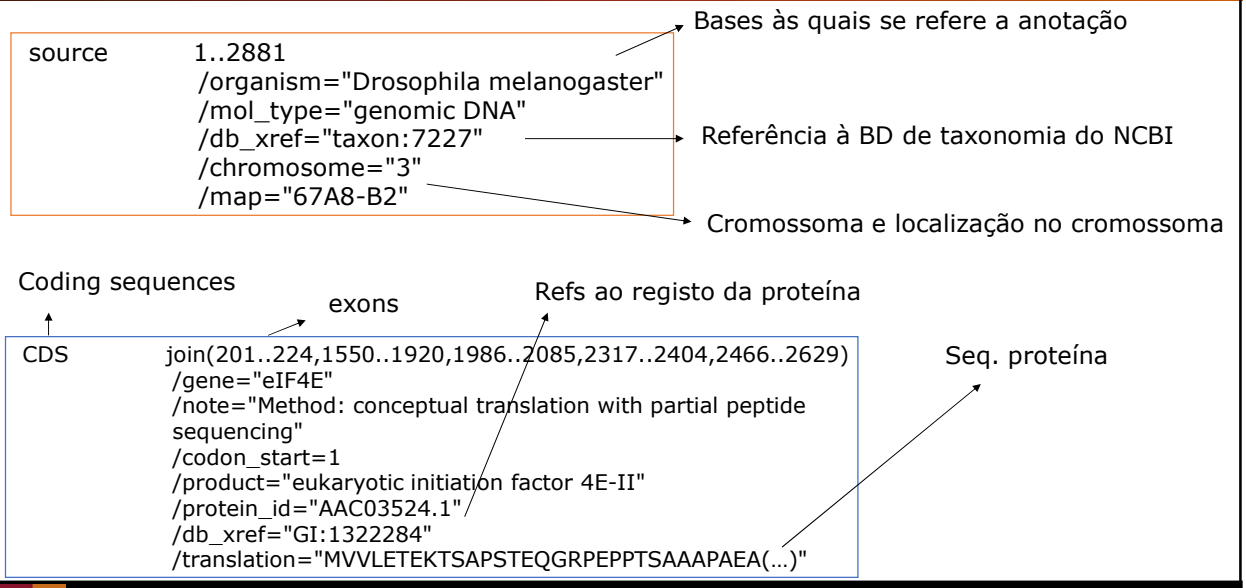
REFERENCE: Referências bibliográficas relacionadas com o registo

REFERENCE 1 (bases 1 to 2881)
AUTHORS Lavoie,C.A., Lachance,P.E., Sonenberg,N. and Lasko,P.
TITLE Alternatively spliced transcripts from the Drosophila eIF4E gene
produce two different Cap-binding proteins
JOURNAL J. Biol. Chem. 271 (27), 16393-16398 (1996)
PUBMED 8663200
(…)

FEATURES:

Esta secção do registo contém as anotações biológicas sendo iniciada com a palavra chave FEATURES. Está organizada em pares chave/ localização:
Na 1ª coluna temos as chaves (tipo de anotação) que pode tomar valores como: source, gene, mRNA, CDS, etc.
Na 2ª colunas temos a informação respeitante a esta chave.

Formato GenBank



Biopython

Objeto *SeqIO*

- SeqIO fornece um conjunto de interfaces para trabalhar com vários formatos de sequências biológicas
- As funções disponíveis permitem ler e escrever ficheiros com sequências em diversos formatos distintos
- As sequências são guardadas como objetos do tipo ***SeqRecord***

Objeto *SeqRecord*

- Permite associar às sequências biológicas aspetos relacionados com a sua anotação, i.e. características das sequências e das entidades que estas representam
- Objeto **SeqRecord** é o tipo base para o input/output de sequências (a tratar mais adiante)

Campos do objeto *SeqRecord*

- Um objeto ***SeqRecord*** tem os seguintes campos:
  - **seq**: a sequência propriamente dita
  - **id**: identificador da sequências
  - **name**: nome da sequência
  - **description**: descrição
  - **letter_annotations**: anotações por letra (posição) da sequência
  - **annotations**: anotações globais da sequência (não estruturadas)
  - **features**: anotações estruturadas da sequência (lista de objetos ***SeqFeature***)
  - **dbxrefs**: referências a bases de dados (externas)

Criação de um *SeqRecord* manualmente

```
>>> from Bio.Seq import Seq
>>> seq = Seq("ATGAATGATAGCTGAT")
>>> from Bio.SeqRecord import SeqRecord
>>> seqRec = SeqRecord(seq)
>>> seqRec.id
'<unknown id>'
>>> seqRec.id = "AB12345"
>>> seqRec.id
'AB12345'
>>> seqRec.description = "Minha sequencia"
>>> seqRec.description
'Minha sequencia'
>>> seqRec.seq
Seq('ATGAATGATAGCTGAT')
>>> seqRec.annotations["role"] = "unknown"
>>> seqRec.annotations
{'role': 'unknown'}
```

Objeto base/simples para guardar uma sequência.

Adição de valores aos campos de anotação manualmente

Objeto *SeqIO*

- O objeto SeqIO permite realizar operações de leitura e escrita de ficheiros com sequências em diversos formatos

- Normalmente, ao ler as sequências o resultado será um objeto ***SeqRecord***, ou um iterador que retorna estes objetos se o ficheiro tiver mais do que uma sequência

```
$ wget https://nextcloud.bio.di.uminho.pt/s/AZtWXdtmL36jGwW/download -O data.zip
```


SeqIO: função *read*

- A função read permite ler ficheiros com um único registo
- Exemplo: ficheiro Fasta "NC_005816.fna":

```
>gi|45478711|ref|NC_005816.1| Yersinia pestis biovar Microtus ... pPCP1, complete sequence
TGTAACGAACGGTGCAATAGTGATCCACACCCAACGCCTGAAATCAGATCCAGGGGGTAATCTGCTCTCC
...
```

http://www.ncbi.nlm.nih.gov/nuccore/NC_005816

```
>>> from Bio import SeqIO
>>> record = SeqIO.read("NC_005816.fna", "fasta")
>>> record
SeqRecord(seq=Seq('TGTAACGAACGGTGCAATAGTGATCCACACCCAACGCCTGA...CTG'),
id='gi|45478711|ref|NC_005816.1|',
name='gi|45478711|ref|NC_005816.1|',
description='gi|45478711|ref|NC_005816.1| Yersinia pestis biovar Microtus ... sequence',
dbxrefs=[])
```

SeqIO: função *read*

```
>>> record.seq
Seq('TGTAACGAACGGTGCAATAGTGATCCACACCCAACGCCTGAAATCAGATCCAGG...CTG')
>>> record.id
'gi|45478711|ref|NC_005816.1|'
>>> record.name
'gi|45478711|ref|NC_005816.1|'
>>> record.description
'gi|45478711|ref|NC_005816.1| Yersinia pestis biovar Microtus ... pPCP1, complete sequence'
>>> record.dbxrefs
[]
>>> record.annotations
{}
>>> record.letter_annotations
{}
>>> record.features
[]
```


SeqIO: leitura de um ficheiro NCBI GenBank

```
LOCUS       NC_005816                9609 bp    DNA     circular  BCT 21
DEFINITION  Yersinia pestis biovar Microtus str. 91001 plasmid pPCP1, complete
            sequence.
ACCESSION   NC_005816
VERSION     NC_005816.1 GI:45478711
PROJECT     GenomeProject:10638
 ...
```

Formato GenBank.

```
>>> from Bio import SeqIO
>>> record = SeqIO.read("NC_005816.gb", "genbank")
>>> record
SeqRecord(seq=Seq('TGTAACGAACGGTGCAATAGTGATCCACACCCAACGCCTGA...CTG',
IUPACAmbiguousDNA()), id='NC_005816.1', name='NC_005816',
description='Yersinia pestis biovar Microtus str. 91001 plasmid pPCP1, complete
sequence.',
dbxrefs=['Project:10638'])
```

SeqIO: leitura de um ficheiro NCBI GenBank (cont.)

```
>>> record.seq
Seq('TGTAACGAACGGTGCAATAGTGATCCACACCCAACGCCTGAAATC...CTG')
>>> record.id
'NC_005816.1'
>>> record.name
'NC_005816'
>>> record.description
'Yersinia pestis biovar Microtus str. 91001 plasmid pPCP1, complete sequence.'
>>> len(record.annotations)
11
>>> record.annotations["source"]
'Yersinia pestis biovar Microtus str. 91001'
>>> record.dbxrefs
['Project:10638']
>>> len(record.features)
29
```

Acesso aos campos:
- id
- name
- description
- annotations
- dbxrefs (referências externas)

Objeto *SeqFeature*

- Permite guardar informação sobre features (anotações) das sequências de forma estruturada
- Estrutura baseada no formato GenBank / EMBL, na sua tabela de features
- Principais atributos de um objeto SeqFeature:
  - **location**: localização da feature na sequência (pode ser uma posição, um intervalo, etc.)
  - **type**: diz o tipo da feature (string)
  - **qualifiers**: informação adicional (dicionário)

FeatureLocation

```
>>> from Bio import SeqFeature
>>> start_pos = SeqFeature.AfterPosition(5)
>>> end_pos = SeqFeature.BetweenPosition(9, left=8, right=9)
>>> my_location = SeqFeature.FeatureLocation(start_pos, end_pos)
>>> print (my_location)
[>5:(8^9)]
>>> int(my_location.start)
5
>>> int(my_location.end)
9
```

Cria um objeto SeqFeacture do tipo gene com a localização entre a posição 5 e 18.

Extrai da sequência original a sequência de nucleótidos referente ao gene.

```
>>> example_parent = Seq("ACCGAGACGGCAAAGGCTAGCATAGGTATGAGACTT")
>>> from Bio.SeqFeature import SeqFeature, FeatureLocation
>>> example_feature = SeqFeature(FeatureLocation(5, 18, strand=-1), type="gene")
>>> feature_seq = example_feature.extract(example_parent)
>>> print (feature_seq)
AGCCTTTGCCGTC
```


SeqFeature

```
for feat in record.features:
    print (feat)
```

Listar todas as features e devidos atributos

```
featcds = [ ]
for i in range(len(record.features)):
    if record.features[i].type == "CDS":
        featcds.append(i)
for k in featcds:
    print (record.features[k].location)
for k in featcds:
    print (record.features[k].extract(record.seq))
```

Identificar features que são do tipo CDS
Identificar a sua localização
Identificar sub-sequência do DNA afetada pela feature

SeqIO: parse

- função: *Bio.SeqIO.parse(),* com argumentos:
  - Handle ou nome do ficheiro
  - Formato do ficheiro (uma string)
  - Opcionalmente, o alfabeto para a sequência a ler
- Retorna iterador sobre objetos *SeqRecord*

```
from Bio import SeqIO
for seq_record in SeqIO.parse("ls_orchid.fasta", "fasta"):
        print (seq_record.id )
        print (seq_record.seq)
        print (len(seq_record))
```

```
from Bio import SeqIO
for seq_record in SeqIO.parse("ls_orchid.gbk", ”genbank"):
        print( seq_record.id )
        print (seq_record.seq)
        print (len(seq_record))
```

http://biopython.org/DIST/docs/tutorial/examples/ls_orchid.fasta

SeqIO.read

- Semelhante à função ***parser***
- Usada quando o ficheiro tem apenas uma única sequência, caso contrário uma excepção será lançada.
- Retorna um objeto do tipo ***SeqRecord***

```
>>> from Bio import SeqIO
>>> seq_record = SeqIO.read("ls_orchid.fasta", "fasta”)

Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "C:\Users\sara\Anaconda3\lib\site-packages\Bio\SeqIO\__init__.py", line 712, in read
    raise ValueError("More than one record found in handle")
ValueError: More than one record found in handle
```

SeqIO: parse

```
>>> from Bio import SeqIO
>>> record_iterator = SeqIO.parse("ls_orchid.gbk", "genbank")
>>> first_record = next(record_iterator)
>>> print (first_record)
ID: Z78533.1
Name: Z78533
Description: C.irapeanum 5.8S rRNA gene and ITS1 and ITS2 DNA.
Number of features: 5
/sequence_version=1
/source=Cypripedium irapeanum
/taxonomy=['Eukaryota', 'Viridiplantae', 'Streptophyta', ..., 'Cypripedium']
/keywords=['5.8S ribosomal RNA', '5.8S rRNA gene', ..., 'ITS1', 'ITS2']
/references=[...]
/accessions=['Z78533']
/data_file_division=PLN
...
```


SeqIO: parse (cont)

```
>>> from Bio import SeqIO
>>> record_iterator = SeqIO.parse("ls_orchid.gbk", "genbank")
>>> first_record = next(record_iterator)
>>> print (first_record)
ID: Z78533.1
Name: Z78533
Description: C.irapeanum 5.8S rRNA gene and ITS1 and ITS2 DNA.
Number of features: 5
/sequence_version=1
/source=Cypripedium irapeanum
/taxonomy=['Eukaryota', 'Viridiplantae', 'Streptophyta', ..., 'Cypripedium']
/keywords=['5.8S ribosomal RNA', '5.8S rRNA gene', ..., 'ITS1', 'ITS2']
/references=[...]
/accessions=['Z78533']
/data_file_division=PLN
...
```

Ficheiro com várias sequências no formato GenBank

SeqIO: parse (cont)

```
>>> print (first_record.annotations["source"])
Cypripedium irapeanum

>>> from Bio import SeqIO
>>> all_species = []
>>> for seq_record in SeqIO.parse("ls_orchid.gbk", "genbank"):
        all_species.append(seq_record.annotations["organism"])
>>> print (all_species)
['Cypripedium irapeanum', 'Cypripedium californicum', ..., 'Paphiopedilum barbatum']
```

.parse() retorna um iterador sobre os objetos SeqRecord

Sequências retiradas da web

- FASTA

```
from Bio import Entrez
from Bio import SeqIO
Entrez.email = "...@example.com"
handle = Entrez.efetch(db="nucleotide", rettype="fasta", retmode="text", id="6273291")
seq_record = SeqIO.read(handle, "fasta")
handle.close()
print (seq_record.id, " com ", len(seq_record.features), " features ")
```

Package Entrez permite acesso ao NCBI através da web

Nota: https://biopython.org/DIST/docs/api/Bio.Entrez-module.html

Sequências retiradas da web

- Genbank

```
from Bio import Entrez
from Bio import SeqIO
Entrez.email = ”...@example.com"
handle = Entrez.efetch(db="nucleotide", rettype="gb", retmode="text",
id="6273291,6273290,6273289")
for seq_record in SeqIO.parse(handle, "gb"):
        print (seq_record.id, seq_record.description[:100], "...")
        print ("Sequence length: ", len(seq_record))
        print (len(seq_record.features), " features" )
        print ("from: ", seq_record.annotations["source"] )
handle.close()
```


Escrita de sequências: *write*

- Função ***write***: permite escrever sequências para ficheiros em diversos formatos
- Argumentos:
  - Lista de objetos SeqRecord (que representam as sequências e sua anotação que se pretende escrever)
  - Handle ou nome do ficheiro
  - Formato (string)

Escrita de sequências: exemplo

```
>>> from Bio.Alphabet import generic_protein
>>> s1 = SeqRecord(Seq("MMYQQGCFAGGTV", generic_protein),
           id = "gi|14150838|gb|AAK54648.1|AF376133_1",
          description="chalcone synthase [Cucumis sativus]")

>>> s2 = SeqRecord(Seq("MVTVEEFRRAQ", generic_protein),
          id="gi|13919613|gb|AAK33142.1|",
          description="chalcone synthase [Fragaria vesca]")

>>> lr = [s1, s2]
>>> from Bio import SeqIO
>>> SeqIO.write(lr, “ex.faa", "fasta")
2
```

1 - Construção manual de dois objetos seqRecord.

2- Criar lista com duas sequências

3- Gravar no ficheiro

Conversão de formatos

- Ler as sequências e gravar em outro formato

```
from Bio import SeqIO
records = SeqIO.parse("ls_orchid.gbk", "genbank")
count = SeqIO.write(records, "my_example.fasta", "fasta")
print ("Convertidos %i registos" % count )
```

- Usar uma única função para converter os formatos

```
from Bio import SeqIO
count = SeqIO.convert("ls_orchid.gbk", "genbank", "my_example.fasta", "fasta")
print ("Convertidos %i registos" % count )
```