Bioinformatics Programming in Python

http://taspython.eu/

Βικάτος Παντελεήμων

Σύνοψη

1. *Γιατί να χρησιμοποιούμε python ;*
2. Python modules
3. Biopython
4. Παραδείγματα

Χαρακτηριστικά

- ✓ Διερμηνευόμενη ,υψηλού επιπέδου Γ.Π.
- ✓ Ανοιχτού κώδικα
- ✓ Εύκολη
  - ➢ Εκμάθηση
  - ➢ Αναγνωσιμότητα
  - ➢ Συντήρηση
- ✓ Εύπλαστη
- ✓ Παίζει παντού (Cross Platform)
- ✓ Συνεργάσιμη
- ✓ Ώριμη
- ✓ Όχι πια segmentation fault

Γιατί python?

Ερώτημα : Διευκολύνει τους μηχανικούς που ασχολούνται με Bioinformatics ;

Γιατί python?

Ερώτημα : Διευκολύνει τους μηχανικούς που ασχολούνται με Bioinformatics ;

Απάντηση : Με βεβαιότητα ΝΑΙ !!!!

Γιατί python?

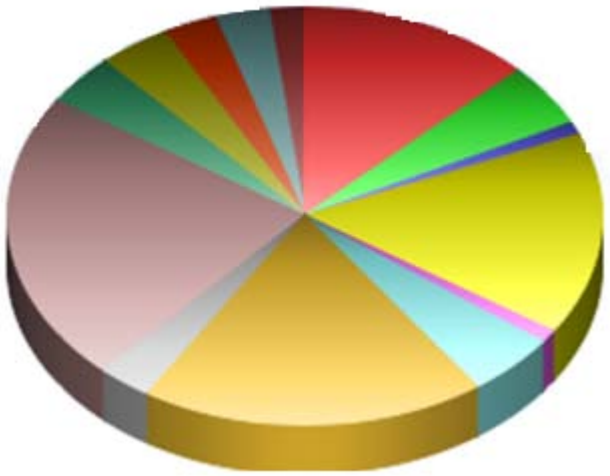
Ερώτημα : Διευκολύνει τους μηχανικούς που ασχολούνται με Bioinformatics ;

Απάντηση : Με βεβαιότητα ΝΑΙ !!!!

Λόγος : Δεν ανησυχείς για τα παρακάτω :

- Παράξενα σύμβολα (~=, <>, eq, '\n', {}...)
- Εναλλακτική σύνταξη για να κάνει την ίδια λειτουργία
- Ορισμός τύπος μεταβλητών
- Διαχείριση μνήμης
- IO, call by reference/value κτλ…

Ποια γλώσσα χρησιμοποιείται ;



- C/C++ 13%
- C#/VB/.NET 5%
- FORTRAN 1%
- Java/BioJava 16%
- Mathematica 1%
- Matlab 5%
- Perl/BioPerl 19%
- PHP/BioPHP 3%
- Python/BioPython 23%
- Ruby/BioRuby 4%
- R/S-Plus 4%
- SQL 3%
- Unix/Linux Shell 3%
- None of above 2%

http://www.bioinformatics.org/poll/index.php?dispid=16

Τι είναι η Biopython?

BioPython : μια συλλογή τυποποιημένων libraries σε python για τη βιοπληροφορική .

- Ανοιχτού κώδικα (Open Source,)
- Cross platform:
  - ✓ Linux, Windows, Mac OS X, …
- Συναφή projects
  - ✓ BioPerl, BioRuby, BioJava, …

Τι είναι η Biopython?

Πλεονεκτήματα χρήσης open source libraries :

- Αναπαραγωγιμότητα
- Ευκολία σύγκρισης αποτελεσμάτων
- Λιγότερα λάθη
- Λιγότερος χρόνος υλοποίησης

Εφαρμογές της Biopython

- Διαχείριση και επεξεργασία ακολουθιών
- BLAST (τοπική και online)
- Web databases (NCBI's EUtils)
- Επιλογή command line διεπαφών (e.g. clustalw)
- Ομαδοποίηση (Bio.Cluster)
- Φυλογενετική (Bio.Nexus)
- Δομή Πρωτεϊνών (Bio.PDB)
- Υποστήριξη βάσεων (Bio.SQL)
- Γενετική Πληθυσμού (Bio.PopGen)

Επιπλέον modules

NumPy

- N-dimensional μητρώα
- Συναρτήσεις γραμμικής άλγεβρας
- Μετασχηματισμούς Fourier
- Γεννήτορες τυχαίων αριθμών

SciPy

- Στατιστικά πακέτα
- Αριθμητική ολοκλήρωση
- Γραμμική άλγεβρα
- Επεξεργασία σημάτων
- Επεξεργασία εικόνας
- Γενετικούς αλγόριθμους
- Επιλυτές Διαφορικών εξισώσεων

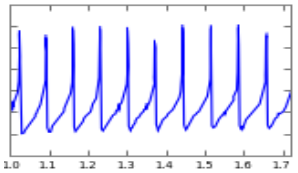
Επιπλέον modules

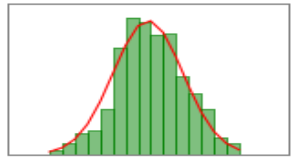
Matplotlib

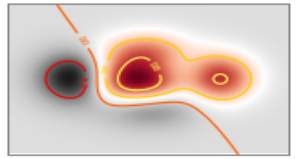
- Βιβλιοθήκη για το σχεδιασμό 2D και 3D διαγραμμάτων .

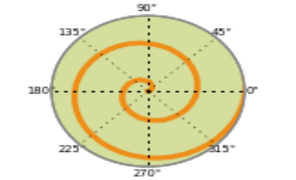
Πλεονεκτήματα

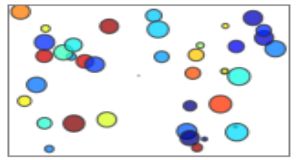
- ✓Ευκολία χρήσης
- ✓Documentation και tutorials
- ✓Αποδοτικό visualization.

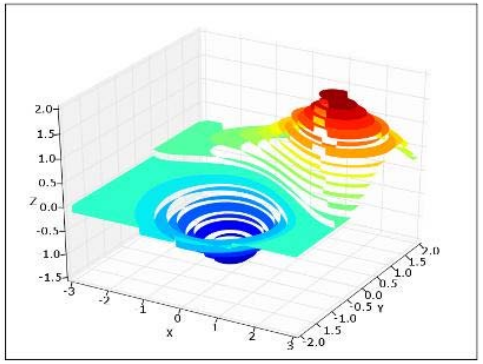












http://taspython.eu/

Επιπλέον modules

NLTK(Natural Language Toolkit)

Corpus readers	interfaces to many corpora
Tokenizers	whitespace, newline, blankline, word, treebank, sexpr, regexp, Punkt sentence segmenter
Stemmers	Porter, Lancaster, regexp
Taggers	regexp, n-gram, backoff, Brill, HMM, TnT
Chunkers	regexp, n-gram, named-entity
Parsers	recursive descent, shift-reduce, chart, feature-based, probabilistic, dependency, ccg, ...
Semantic interpretation	untyped lambda calculus, first-order models, DRT, glue semantics, hole semantics, parser interface
WordNet	WordNet interface, lexical relations, similarity, interactive browser
Classifiers	decision tree, maximum entropy, naive Bayes, Weka interface, megam
Clusterers	expectation maximization, agglomerative, k-means
Metrics	accuracy, precision, recall, windowdiff, distance metrics, inter-annotator agreement coefficients, word association measures, rank correlation
Estimation	uniform, maximum likelihood, Lidstone, Laplace, expected likelihood, heldout, cross-validation, Good-Turing, Witten-Bell
Miscellaneous	unification, chatbots, many utilities
NLTK-Contrib (less mature)	categorial grammar (Lambek, CCG), finite-state automata, hadoop (MapReduce), kimmo, readability, textual entailment, timex, TnT interface, inter-annotator agreement
Browse the source code: http://code.google.com/p/nltk/source/browse/trunk/nltk	
BuildBot: automatic testing of NLTK code: http://buildbot.nltk.org/	

Άλλες εφαρμογές και βιβλιοθήκες

Django	(Web frameworks)
Plone	(Content Management System)
ReportLab	(PDF generation)
MPI for Python	(Παράλληλος Προγραμματισμός)
SymPy	(Συμβολικά Μαθηματικά)
Python/R interface	(στατιστική ανάλυση)
SWIG	(Simplified Wrapper and Interface Generator)
Pygr	(βάση δεδομένων γραφικών)
PysCeS	(Προσομοίωση των κυτταρικών συστημάτων)
SloppyCell	(Προσομοίωση βιομοριακών δικτύων)

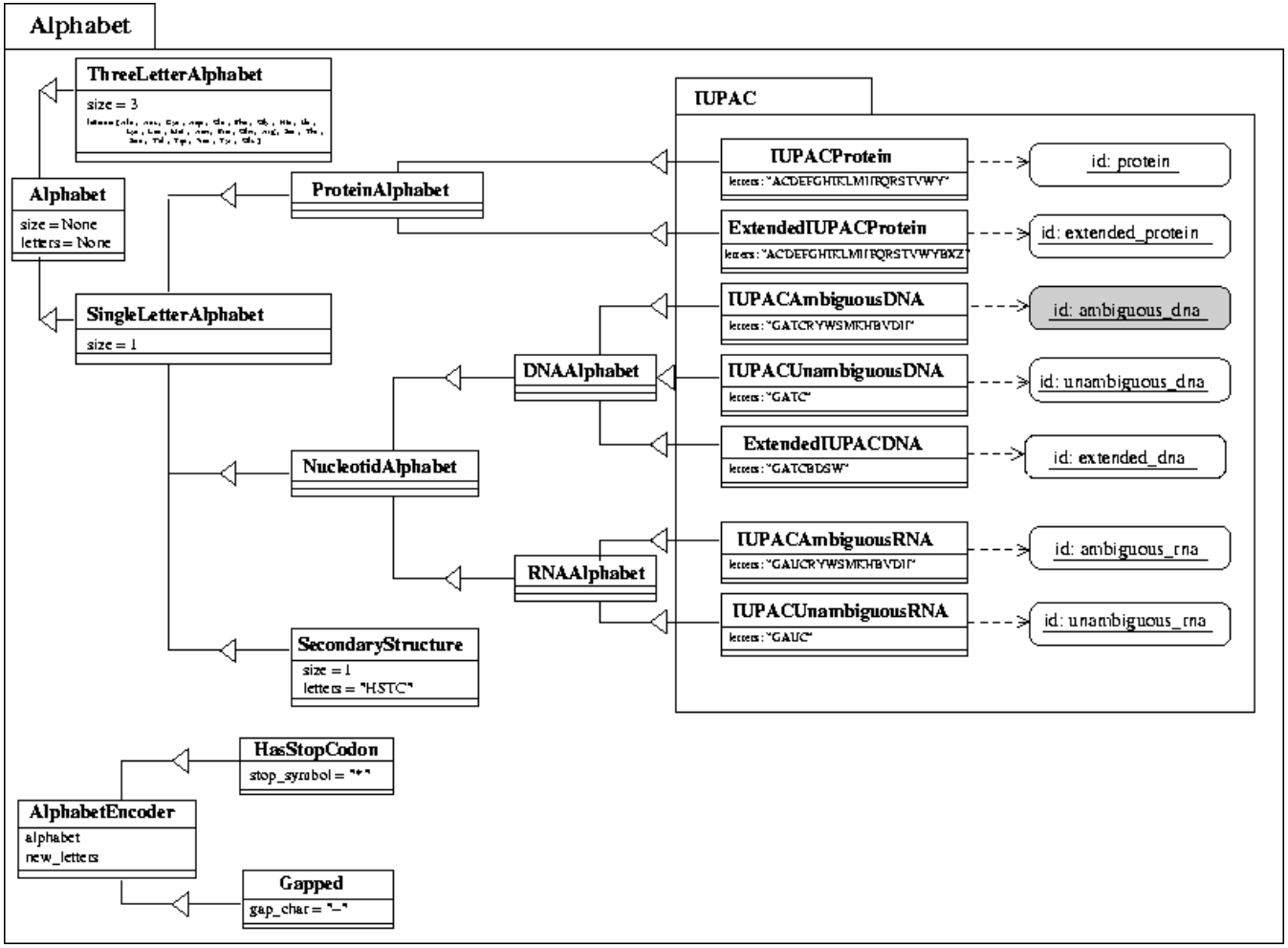
...

Biopython – Sequence objects

```
>>> from Bio.Seq import Seq
>>> my_seq = Seq("AGTACACTGGT")
>>> my_seq
Seq('AGTACACTGGT', Alphabet())
>>> print my_seq
AGTACACTGGT
>>> my_seq.alphabet
Alphabet()
```

- Λειτουργούν ως strings αλλά έχουν περισσότερες ιδιότητες

Biopython - Alphabet



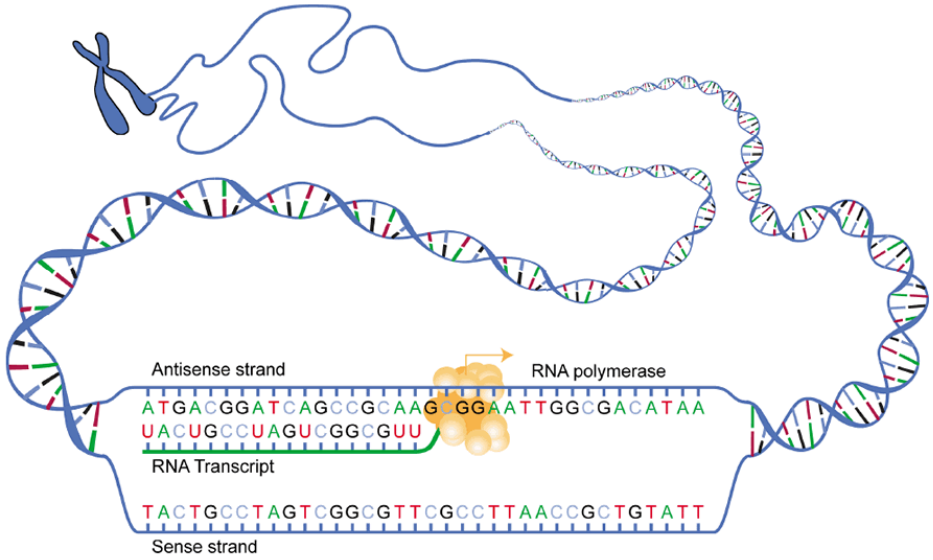
Biopython – Seq Functions

Βασικές συναρτήσεις

complement()	:συμπληρωματική
reverse_complement()	: αντίστροφη συμπληρωματική
transcribe())	: DNA to RNA
back_transcribe())	: RNA to DNA
translate()	: DNA to protein

Biopython – Seq Functions

Transcription



Biopython – Seq Functions

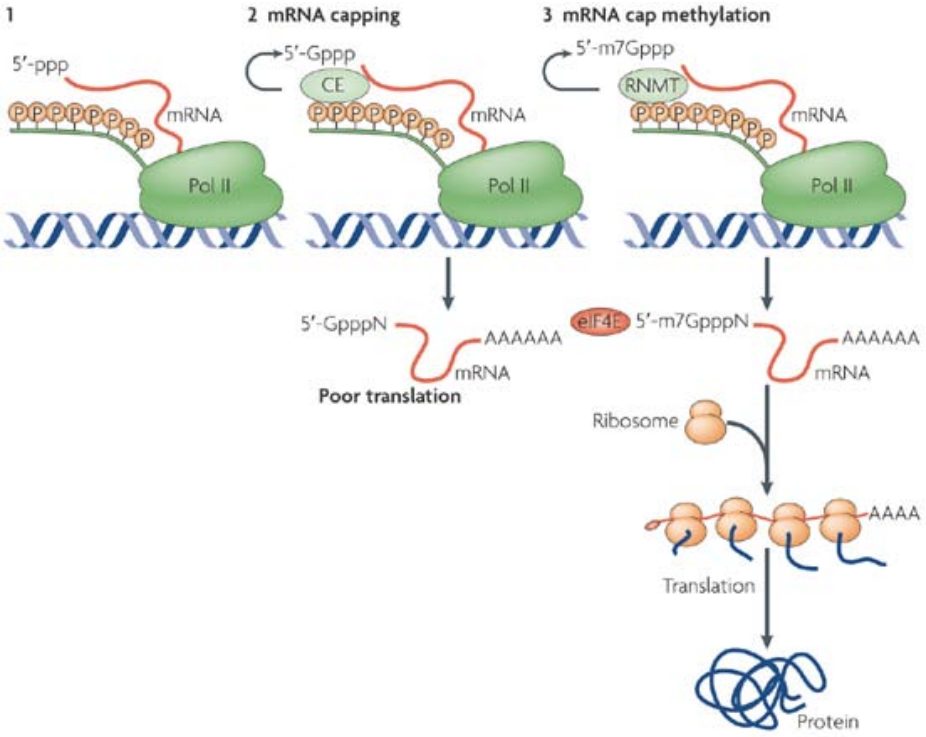
Transcription

```
>>> from Bio.Seq import Seq
>>> from Bio.Alphabet import IUPAC
>>> coding_dna =
Seq("ATGGCCATTGTAATGGGCCGCTGAAAGGGTGCCCGATAG",
IUPAC.unambiguous_dna)
>>> coding_dna
Seq('ATGGCCATTGTAATGGGCCGCTGAAAGGGTGCCCGATAG',
IUPACUnambiguousDNA())

>>> messenger_rna = coding_dna.transcribe()
>>> messenger_rna
Seq('AUGGCCAUUGUAAUGGGCCGCUGAAAGGGUGCCCGAUAG',
IUPACUnambiguousRNA())
```

Biopython – Seq Functions

Translation



Biopython – Seq Functions

Translation

```
>>> from Bio.Seq import Seq
>>> from Bio.Alphabet import IUPAC
>>> messenger_rna =
Seq("AUGGCCAUUGUAAUGGGCCGCUGAAAGGGUGCCCGAUAG",
IUPAC.unambiguous_rna)
>>> messenger_rna
Seq('AUGGCCAUUGUAAUGGGCCGCUGAAAGGGUGCCCGAUAG',
IUPACUnambiguousRNA())

>>> messenger_rna.translate()
Seq('MAIVMGR*KGAR*', HasStopCodon(IUPACProtein(), '*'))
```

Biopython – Seq Functions

Translation Tables

First Position	Seconed Position: U				Seconed Position: C				Seconed Position: A				Seconed Position: G				Third Position
	code	Amino Acid	code	Amino Acid	code	Amino Acid	code	Amino Acid	code	Amino Acid	code	Amino Acid	code	Amino Acid			
U	UUU	phe	UCU	ser	UAU	tyr	UGU	cys									U
	UUC		UCC		UAC		UGC										C
	UUA	leu	UCA		UAA	STOP	UGA	STOP									A
	UUG		UCG		UAG	STOP	UGG	trp									G
C	CUU	leu	CCU	pro	CAU	his	CGU	arg									U
	CUC		CCC		CAC		CGC										C
	CUA		CCA		CAA	gln	CGA										A
	CUG		CCG		CAG		CGG										G
A	AUU	ile	ACU	thr	AAU	asn	AGU	ser									U
	AUC		ACC		AAC		AGC										C
	AUA		ACA		AAA	lys	AGA	arg									A
	AUG	met	ACG		AAG		AGG										G
G	GUU	val	GCU	ala	GAU	asp	GGU	gly									U
	GUC		GCC		GAC		GGC										C
	GUA		GCA		GAA	glu	GGA										A
	GUG		GCG		GAG		GGG										G

```
>>> from Bio.Data import CodonTable
>>> standard_table = CodonTable.unambiguous_dna_by_name["Standard"]
```

Biopython – SeqIO

Βασικές λειτουργίες :

parse	:	όλων των στοιχείων ενός βιολογικού αρχείου
read	:	διάβασμα ενός στοιχείου
write	:	εγγραφή στοιχείων στο αρχείο
convert	:	μετατροπή αρχείου από την μια μορφή στην άλλη

Biopython – SeqIO

Βασικές λειτουργίες :

parse : όλων των στοιχείων ενός βιολογικού αρχείου
read : διάβασμα ενός στοιχείου
write : εγγραφή στοιχείων στο αρχείο
convert : μετατροπή αρχείου από την μια μορφή στην άλλη

File Formats :

ace	gb (genbank)	pir
clustal	ig	stockholm
Embl	nexus	swiss
fasta	phd	tab
fastq	phylip	qual

και για 3D δομές : pdb

Biopython – SeqIO

Parsing & read από αρχείο

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

Biopython – SeqIO

Parsing & read από αρχείο

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

```
gi|2765658|emb|Z78533.1|CIZ78533
Seq('CGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATTGATGAGACCGTGG...CGC',
    SingleLetterAlphabet())
740
...
gi|2765564|emb|Z78439.1|PBZ78439
Seq('CATTGTTGAGATCACATAATAATTGATCGAGTTAATCTGGAGGATCTGTTTACT...GCC',
    SingleLetterAlphabet())
592
```

Biopython – SeqIO

Parsing & read από αρχείο με iterator

```
from Bio import SeqIO
handle = open(“ls_orchid.fasta")
record_iterator = SeqIO.parse(handle, "fasta")
first_record = record_iterator.next()
print first_record.id
print first_record.description
second_record = record_iterator.next()
print second_record.id
print second_record.description
```

Biopython – SeqIO

Parsing & read από αρχείο με iterator

```
from Bio import SeqIO
handle = open("ls_orchid.fasta")
record_iterator = SeqIO.parse(handle, "fasta")
first_record = record_iterator.next()
print first_record.id
print first_record.description
second_record = record_iterator.next()
print second_record.id
print second_record.description
```

```
gi|2765658|emb|Z78533.1|CIZ78533
gi|2765658|emb|Z78533.1|CIZ78533 C.irapeanum 5.8S rRNA gene and ITS1 and
ITS2 DNA
gi|2765657|emb|Z78532.1|CCZ78532
gi|2765657|emb|Z78532.1|CCZ78532 C.californicum 5.8S rRNA gene and ITS1 and
ITS2 DNA
```

Biopython – SeqIO

Parsing & read από το διαδίκτυο

```
from Bio import Entrez
from Bio import SeqIO
Entrez.email = "A.N.Other@example.com"
handle = Entrez.efetch(db="nucleotide", rettype="fasta", id="6273291")
seq_record = SeqIO.read(handle, "fasta")
handle.close()
print "%s with %i features" % (seq_record.id, len(seq_record.features))
```

Biopython – SeqIO

Parsing & read από το διαδίκτυο

```
from Bio import Entrez
from Bio import SeqIO
Entrez.email = "A.N.Other@example.com"
handle = Entrez.efetch(db="nucleotide", rettype="fasta", id="6273291")
seq_record = SeqIO.read(handle, "fasta")
handle.close()
print "%s with %i features" % (seq_record.id, len(seq_record.features))
```

```
gi|6273291|gb|AF191665.1|AF191665 with 0 features
```

Biopython – SeqIO

Μετατροπή αρχείων διαφορετικό format

```
from Bio import SeqIO
from StringIO import StringIO
handle1 =open(“my_example.fasta")
handle2 =open(“ls_orchid.gbk")
count = SeqIO.convert(handle2, "genbank", handle1, "fasta")
handle1.close()
handle2.close()
```

Biopython – SeqRecord

Εγγραφές βιολογικών κειμένων

- SeqRecord = Seq object + metadata

metadata :

- id
- name
- description
- annotations
- features
- dbxrefs

Biopython – SeqRecord

Επιλογή στοιχείων ενός Record

```
>>> from Bio import SeqIO
>>> record = SeqIO.read("NC_005816.fna", "fasta")
>>> record
SeqRecord(seq=Seq('TGTAACGAACGGTGCAATAGTGATCCACACCCAACGCCTGAAATCAGATCCAGG..
    .CTG',
SingleLetterAlphabet()), 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=[])
```

Biopython – SeqRecord

Επιλογή στοιχείων ενός Record

```
>>> from Bio import SeqIO
>>> record = SeqIO.read("NC_005816.fna", "fasta")
>>> record
SeqRecord(seq=Seq('TGTAACGAACGGTGCAATAGTGATCCACACCCAACGCCTGAAATCAGATCCAGG..
    .CTG',
SingleLetterAlphabet()), 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=[])
```

```
>>> 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'
```

Biopython – SeqRecord

Δημιουργία Record και Format

```
from Bio.Seq import Seq
from Bio.SeqRecord import SeqRecord
from Bio.Alphabet import generic_protein
record =
    SeqRecord(Seq("MMYQQGCFAGGTVLRLAKDLAENNRGARVLVVCSEITAVTFRGPSETHLDSMVGQA
    LFGD" \
+"GAGAVIVGSDPDLSVERPLYELVWTGATLLPDSEGAIDGHLREVGLTFHLLKDVPGLISK" \
+"NIEKSLKEAFTPLGISDWNSTFWIAHPGGPAILDQVEAKLGLKEEKMRATREVLSEYGNM" \
+"SSAC", generic_protein),
id="gi|14150838|gb|AAK54648.1|AF376133_1",
description="chalcone synthase [Cucumis sativus]")
print record.format("fasta")
```

Biopython – SeqRecord

Δημιουργία Record και Format

```
from Bio.Seq import Seq
from Bio.SeqRecord import SeqRecord
from Bio.Alphabet import generic_protein
record =
    SeqRecord(Seq("MMYQQGCFAGGTVLRLAKDLAENNRGARVLVVCSEITAVTFRGPSETHLDSMVGQA
    LFGD" \
+"GAGAVIVGSDPDLSVERPLYELVWTGATLLPDSEGAIDGHLREVGLTFHLLKDVPGLISK" \
+"NIEKSLKEAFTPLGISDWNSTFWIAHPGGPAILDQVEAKLGLKEEKMRATREVLSEYGNM" \
+"SSAC", generic_protein),
id="gi|14150838|gb|AAK54648.1|AF376133_1",
description="chalcone synthase [Cucumis sativus]")
print record.format("fasta")
```

```
>gi|14150838|gb|AAK54648.1|AF376133_1 chalcone synthase [Cucumis sativus]
MMYQQGCFAGGTVLRLAKDLAENNRGARVLVVCSEITAVTFRGPSETHLDSMVGQALFGD
GAGAVIVGSDPDLSVERPLYELVWTGATLLPDSEGAIDGHLREVGLTFHLLKDVPGLISK
NIEKSLKEAFTPLGISDWNSTFWIAHPGGPAILDQVEAKLGLKEEKMRATREVLSEYGNM
SSAC
```

Biopython – SeqRecord

Εγγραφή Record σε αρχείο

```
from Bio import SeqIO
handle = open(“my_example.fasta")
SeqIO.write(my_records, handle,"fasta")
handle.close()
```

Biopython – BLAST

Basic **L**ocal **A**lignment **S**earch **T**ool :

- **Βάση δεδομένων και Web Service**
  - ✓ **Online και τοπική**

Τρόπος χρήσης :

1. Αναζήτηση με την function qblast()
2. Επιλογή blast προγράμματος
3. Δήλωση βάσης δεδομένων
4. Αναζήτηση query

   - Επιστρέφει XML αρχείο με πληροφορίες για το alignment .

Biopython – BLAST

Χρησιμοποίηση της online BLAST

```
from Bio.Blast import NCBIWWW
from Bio import SeqIO
handle = open(“m_cold.fasta")
save_file = open(“my_blast.xml", "w")
record = SeqIO.read(handle, format="fasta")
result_handle = NCBIWWW.qblast("blastn", "nr", record.seq)
save_file.write(result_handle.read())
save_file.close()
handle.close()
```

Biopython – BLAST

BLAST Record και Στοίχιση

```
from Bio.Blast import NCBIXML
save_file = open(“my_blast.xml")
blast_record = NCBIXML.read(save_file)
E_VALUE_THRESH = 0.04
for alignment in blast_record.alignments:
    for hsp in alignment.hsps:
        if hsp.expect < E_VALUE_THRESH:
            print "****Alignment****"
            print "sequence:", alignment.title
            print "length:", alignment.length
            print "e value:", hsp.expect
            print hsp.query[0:75] + "..."
            print hsp.match[0:75] + "..."
            print hsp.sbjct[0:75] + "..."
```

Biopython – BLAST

BLAST Record και Στοίχιση

```
****Alignment****
sequence: gi|42460737|emb|BX828212.1| Arabidopsis thaliana Full-length cDNA Complete sequence from
clone GSLTPGH63ZH10 of Hormone Treated Callus of strain col-0 of Arabidopsis thaliana (thale cress)
length: 910
e value: 1.34878e-25
AAAATGGGGAGAGAAATGAAGTACTTGGCCATGAAAACTGATCAATTGGCCGTGGCTAATATGATCGATTCCGAT...
||||||||  ||  |   |||  ||  ||||||||||| |||||  |   |  ||  ||||||  |||||  ||||||||...
AAAATGGGAAGGGG--TGA-GTTTTTGGCCATGAAGACTGAGGA---GAACGCGGCTAACCTGATCAATTCCGAT...
```

Biopython – NCBI's Entrez

Entrez : Σύστημα ανάκτησης πληροφορίας από τις βάσεις δεδομένων της NCBI.

```
from Bio import Entrez
Entrez.email = "A.N.Other@example.com"
handle = Entrez.einfo()
record = Entrez.read(handle)
print record["DbList"]
```

Biopython – NCBI's Entrez

Entrez : Σύστημα ανάκτησης πληροφορίας από τις βάσεις δεδομένων της NCBI.

```
from Bio import Entrez
Entrez.email = "A.N.Other@example.com"
handle = Entrez.einfo()
record = Entrez.read(handle)
print record["DbList"]
```

Περιεχόμενα βάσης :

['pubmed', 'protein', 'nucleotide', 'nuccore', 'nucgss', 'nucest', 'structure', 'genome', 'books', 'cancerchromosomes', 'cdd', 'gap', 'domains', 'gene', 'genomeprj', 'gensat', 'geo', 'gds', 'homologene', 'journals', 'mesh', 'ncbisearch', 'nlmcatalog', 'omia', 'omim', 'pmc', 'popset', 'probe', 'proteinclusters', 'pcassay', 'pccompound', 'pcsubstance', 'snp', 'taxonomy', 'toolkit', 'unigene', 'unists']

Biopython – NCBI's Entrez

Αναζήτηση στην βάση

```
from Bio import Entrez
Entrez.email = "A.N.Other@example.com"
handle = Entrez.esearch(db="nucleotide",term="Cypripedioideae[Orgn] AND matK[Gene]")
record = Entrez.read(handle)
print record["Count"]
print record["IdList"]
```

Biopython – NCBI's Entrez

Αναζήτηση στην βάση

```
from Bio import Entrez
Entrez.email = "A.N.Other@example.com"
handle = Entrez.esearch(db="nucleotide",term="Cypripedioideae[Orgn] AND matK[Gene]")
record = Entrez.read(handle)
print record["Count"]
print record["IdList"]
```

```
'25'
['126789333', '61585496', '61585494', '61585492', '61585490', '61585488',
'61585486', '61585484', '61585482', '51831269', '51831267', '48527434',
'37222967', '37222966', '37222965', '18027046', '246973149', '246655146',
'164513137', '164513135']
```

Biopython – NCBI's Entrez

Ανάκτηση στοιχείων από το Entrez

```
from Bio import Entrez
Entrez.email = "A.N.Other@example.com”
handle = Entrez.efetch(db="nucleotide", id="186972394", rettype="gb")
print handle.read()
```

Τυπώνει το ζητούμενο αρχείο σε μορφή genbank.

Biopython – NCBI's Entrez

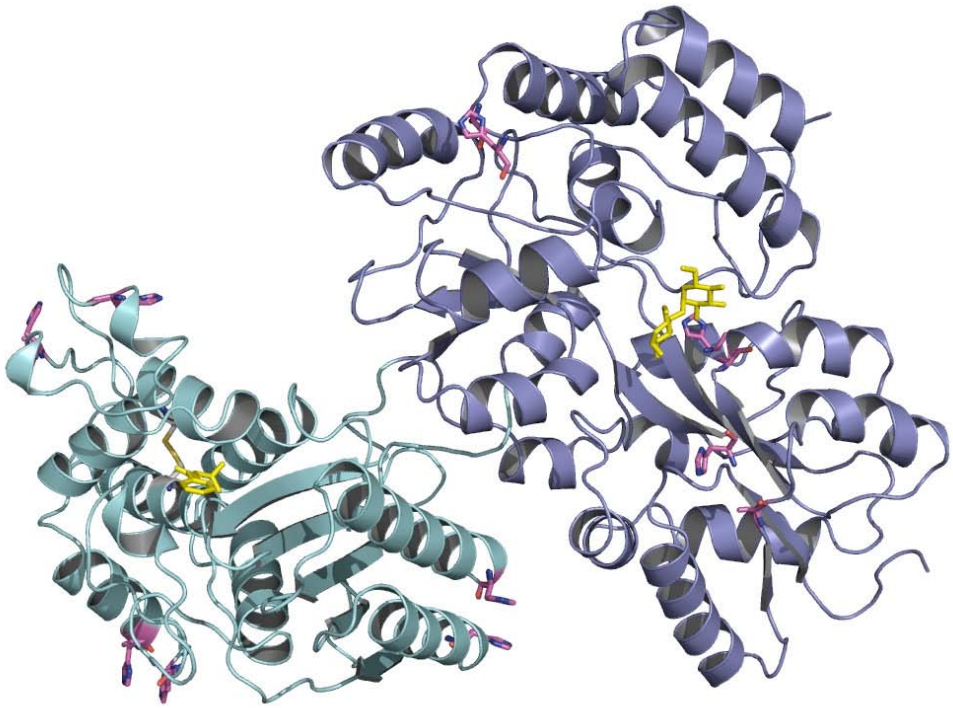
Επιπλέον συναρτήσεις

ELink : αναζήτηση για σχετικά αντικείμενα στην NCBI Entrez
EGQuery : αναζήτηση σε όλες τις βάσεις(global search)
ESummary : ανάκτηση περιλήψεων από τα primary IDs

Biopython – PDBParser

Διαχείριση αρχείων PDB

- ✓ Περιγραφή της 3D αναπαράστασης μακρομορίων



Biopython κ.α.

Population genetics

- Bio.PopGen

Supervised learning methods

- LogisticRegression ,kNN, NaiveBayes
- Bio.MarkovModel

Genome

- Bio.Graphics , GenomeDiagram

Support & Tutorials

Υποστήριξη

- Open Bioinformatics Foundation
- Διεθνής ομάδα από εθελοντές προγραμματιστές
- Πλήρης οδηγός “Biopython Tutorial & Cookbook”
- Εκτενείς λεπτομέρειες στο www.biopython.org

Πηγές

Βιβλιογραφία

- Bioinformatics Programming in Python: A Practical Course for Beginners Ruediger-Marcus Flaig
- Bioinformatics Programming Using Python , Mitcell L. Model
- Python for Bioinformatics , Sebastian Bassi

Links:

- *http://biopython.org/wiki/Main_Page*
- *http://biopython.org/DIST/docs/tutorial/Tutorial.html*
- *http://www.pasteur.fr/recherche/unites/sis/formation/python/index.html*

Thank You !

http://taspython.eu/