Σημείωσεις για το μάθημα "Προγραμματισμός σε python"

Αλέξανδρος Καντεράκης kantale@ics.forth.gr (mailto:kantale@ics.forth.gr)

Διάλεξη 11η, 14 Ιανουαρίου 2020

Σκοπός αυτής της διάλεξης είναι η εισαγωγή σε μεθόδους για να προσπελαύνουμε υπάρχοντες online βάσεις δεδομένων οι οποίες περιέχουν γενετικές-βιολογικές πληροφορίες.

Καταρχήν πρέπει να αναφέρουμε ότι υπάρχουν χιλιάδες τέτοιες βάσεις οι οποίες περιέχουν πληροφορίες για όλες τις μορφές των -omics data. Μία πολύ καλή λίστα βρίσκεται εδώ: https://www.oxfordjournals.org/nar/database/a/ (https://www.oxfordjournals.org/nar/database/a/) (προφανώς και η λίστα δεν είναι πλήρης).

Πως λοιπόν μπορούμε προγραμματιστικά να προσπελάσουμε αυτές τις βάσεις;

HTTP GET and POST requests

Πριν απαντήσουμε σε αυτό πρέπει να πούμε κάποια βασικά στοιχεία για το HTTP (https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) πρωτόκολλο. Το HTTP είναι ένα πρωτόκολλο που καθορίζει πως δύο υπολογιστές μπορούν να επικοινωνήσουν ώστε ο ένας να "πάρει" κάποια πληροφορία από τον άλλο. Αυτή η "πληροφορία" είναι συνήθως μία HTML σελίδα, αλλά μπορεί να είναι και οποιαδίποτε άλλη μορφή πληροφορίας. Μία πηγή πληροφορίας ορίζεται σύμφωνα με το HTTP ως ένα URL ή αλλιώς ένα link. π.χ: http://bioinfo-grad.gr/eclass/ (http://bioinfo-grad.gr/eclass/).

Ένας υπολογιστής που "ζητάει" πληροφορία συνήθως αναφέρεται ως "client" ενώ ένας υπολογιστής που μπορεί να τη δώσει ονομάζεται ως server. Ένας client για να "ζητήσει" πληροφορία από τον "server" πρέπει να ξέρει το URL του server. Στη συνέχεια εκτελεί ένα "request" (υποθέτω "αίτημα" στα ελληνικά), προς τον server και του δηλώνει ποια είναι η μορφή της πληροφορίας που θέλει. Ο server αποφασίζει αν την έχει (και αν πρέπει να τη δώσει) την πληροφορία και ανάλογα την επιστρέφει στον client (κάτι σαν το ΚΕΠ..). Σε περίπτωση που κάτι δεν πάει καλά, ο server μπορεί να ενημερώσει τον client τι λάθος έγινε επιστρέφοντας έναν κωδικό (https://en.wikipedia.org/wiki/List_of_HTTP_status_codes). Αν ο κωδικός είναι το 200, τότε δεν συνέβει κανένα λάθος. Όταν ένας server "απαντάει" σε ένα request, τότε λέμε ότι έστειλε ένα responce.

Όταν ο client επικοινωνεί με τον server του δηλώνει: Τι θέλει, και σε ποια μορφή το θέλει.

Σχετικά με το "τι θέλω" υπάρχουν ΔΥΟ βασικοί τρόποι δήλωσης: οι "GET" και "POST" (http://www.w3schools.com/tags/ref_httpmethods.asp).

Η GET είναι ένας τρόπος να δηλώσεις τι θέλεις από έναν server, κωδικοποιώντας το αίτημα σου πάνω στο link. Αν υποθέσουμε π.χ. ότι θέλεις πληροροφίες για το προιόν με id=123 και το οποίο ανοίκει στη κατηγορία category=678 τότε το link που εκτελεί ένα GET request θα έχει τη μορφή: http://www.example.com/index.html?id=123&category=678 (http://www.example.com/index.html?id=123&category=678). Για παράδειγμα αν κοιτάξουμε αυτό το εντελώς τυχαίο link: http://news.in.gr/greece/article/?aid=1500029633 (http://news.in.gr/greece/article/?aid=1500029633) θα δούμε ότι είναι ένα GET request στη σελίδα news.in.gr/greece/article όπου ζητάει το "resource" το οποίο έχει aid=1500029633. Το GET request είναι ο κύριος τρόπος επικοινωνίας των browsers με τους web servers.

Τα GET requests έχουν δύο μειονεκτήματα:

- Αυτό που ζητάει ο client "φαίνεται" πάνω στο link. Γενικότερα τα links δεν θεωρούνται "ασφαλή" πληροφορία και είναι προσβάσιμα από πολλούς "ενδιάμεσους". Αν για παράδειγμα ζητάμε κάποια δεδομένα με βάση προσωπικές μας πληροφορίες (π.χ. τηλέφωνο) δεν είναι καλή ιδέα να φαίνεται αυτό στο link.
- Δεδομένου ότι αυτό που ζητάμε βρίσκεται πάνω στο link. Αν αυτό που ζητάμε είναι πολύπλοκο, τότε το link μπορεί να γίνει τεράστιο και μη διαχειρίσιμο. Για παράδειγμα να ένα άσχημο link με πολλά GET πεδία: https://www.amazon.com/Data-Visualization-Python-JavaScript-Transform/dp/1491920513/ref=s9_cartx_gw_g14_i5_r?_encoding=UTF8&fpl=fresh&pf_rd_m=ATVPDKIKX0DER&pf_rd_s=&pf_rd_r=WRMBPSF9K3H62TEWG5DW&pf_rd_t=36701&pf_rd_p=a6aaf593-1ba4-4f4e-bdcc-0febe090b8ed&pf_rd_i=desktop (https://www.amazon.com/Data-Visualization-Python-JavaScript-Transform/dp/1491920513/ref=s9_cartx_gw_g14_i5_r?_encoding=UTF8&fpl=fresh&pf_rd_m=ATVPDKIKX0DER&pf_rd_s=&pf_rd_r=WRMBPSF9K3H62TEWG5DW&pf_rd_t=36701&pf_rd_p=a6aaf593-1ba4-4f4e-bdcc-0febe090b8ed&pf_rd_i=desktop)

Ο δεύτερος τρόπος επικοινωνίας του client με τον server είναι μεσώ του POST request. Όταν κάνουμε POST βάζουμε μέσα σε ένα ειδικό πεδίο τα δεδομένα μας και το HTTP πρωτόκολο στέλνει αυτά τα δεδομένα στον server χωρίς να φαίνονται στο link. Π.χ. όταν κάνετε login στο site: http://bioinfo-grad.gr/eclass/ (http://bioinfo-grad.gr/eclass/) βάζετε το username και το password σας. Στη συνέχεια ο browser στέλνει αυτά τα δεδομένα στον server για επιβεβαίωση. Αυτά τα δεδομένα ΔΕΝ πρέπει προφανώς να φαίνονται στο link. Πως γίνεται αυτό; Αν κοιτάξετε τον κώδικα της σελίδας (π.χ. από Firefox μπορείτε να το δείτε επιλέγοντας Tools -> Web Developer -> Page Source). Θα δείτε σε ένα σημείο να γράφει:

python requests package

Ωραία όλα αυτά, αλλά πως τα κάνουμε μέσω python; Παρόλο που η python έχει βιβλιοθήκες για να κάνουμε GET και POST, θα χρησιμοποιήσουμε ένα εξωτερικό πακέτο το οποίο είναι εξαιρετικά εύχρηστο. Αυτό το πακέτο είναι το requests: http://docs.python-requests.org/en/master/ (http://docs.python-requests.org/en/master/) . Για να το εγκαταστίσετε τρέξτε:

```
pip install requests
```

ΠΡΟΣΟΧΗ! πρέπει να βεβαιωθείτε ότι το πρόγραμμα pip βρίσκεται στο ίδιο σημείο που είναι και η python που τρέχετε!. Για παράδειγμα:

```
$ which pip
/Users/alexandroskanterakis/anaconda3/bin/pip
$ which python
/Users/alexandroskanterakis/anaconda3/bin/python
```

Βλέπουμε δηλαδή ότι το pip και η python που τρέχω είναι στο ίδιο σημείο. Διαφορετικά θα πρέπει να γράψετε κάτι σαν αυτό:

```
/Directory/where/your/python/is/pip install requests
```

Για να βεβαιωθείτε ότι έχει εγκατασταθεί σωστά θα πρέπει να μπορείτε να κάνετε import αυτό το πακέτο χωρίς πρόβλημα:

```
$ python
Python 3.5.2 |Anaconda 4.1.1 (x86_64)| (default, Jul  2 2016, 17:52:12)
[GCC 4.2.1 Compatible Apple LLVM 4.2 (clang-425.0.28)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import requests
>>>
```

Reference Genome

Σήμερα το ανθρώπινο γονιδίωμα είναι διαθέσιμο σε πολλές διαφορετικές μορφές.

Είναι πολύ ενδιαφέρον κάποιος να προσπελάσει το επίσημο FTP site όπου είναι διαθέσιμο με τη παρακάτω σειρά:

ftp://ftp.ncbi.nih.gov/genomes/

ftp://ftp.ncbi.nih.gov/genomes/H_sapiens/

ftp://ftp.ncbi.nih.gov/genomes/H_sapiens/current

ftp://ftp.ncbi.nih.gov/genomes/H_sapiens/current/GCF_000001405.39_GRCh38.p13/

Σε αυτό το directory βρίσκεται το αρχείο: GCF_000001405.39_GRCh38.p13_genomic.fna.gz

Το αρχείο αυτό είναι και η τελευταία έκδοση (Ιανουάριος 2020) του ανθρώπινου DNA (σε fasta format (https://en.wikipedia.org/wiki/FASTA_format)).

Αλλά τι σημαίνει "έκδοση"; Όπως ακριβώς και στο λογισμικό, το ανθρώπινο γωνίδωμα έχει εκδόσεις. Κάθε έκδοση βελτιώνει λάθη της προηγούμενης και βάζει περιοχές που η προηγούμενη δεν είχε. Σήμερα χρησιμοποιούμε κυρίως τη GRCh38 και τη GRCh37.

Εδώ υπάρχει ένα History με τις ημερομηνίες διάθεσης της κάθε έκδοσης (https://genome.ucsc.edu/FAQ/FAQreleases.html#release1)

ENSEMBL

Βασικά

Η ENSEMBL (http://www.ensembl.org/index.html) είναι ένα από τα πιο σημαντικά portals με διάφορες γενομικές πληροφορίες για πολλούς οργανισμούς.

- Κάθε γονδίο έχει κωδικό ENSGXXXXXXXXX Δηλαδή: Ensembl Gene. Παράδειγμα (https://www.ensembl.org/Homo_sapiens/Gene/Summary?db=core;g=ENSG00000151065;r=12:1946053-2004535)
- Κάθε γονίδιο έχει πολλά transcripts. Κάθε transcript έχει κωδικό ENSTXXXXXXXXX Δηλαδή Ensembl Transcript.
- Για πολλά (αλλά όχι για όλα) τα trascripts η ENSEMBL δίνει έναν κωδικό (APPRIS)[http://appris-tools.org/#/ (http://appris-tools.org/#/)]. Το APPRIS είναι μία κατηγοριοποίηση των transcripts όσον αφορά τη σημαντικότητά τους. Διαβάστε περισσότερα εδώ (http://appris-tools.org/#/help/scores)

Προσπελαύνοντας την ENSEMBL

Η ENSEMBL έχει ένα API (Application programming interface (https://en.wikipedia.org/wiki/Application_programming_interface)) το οποίο είναι ένα σύνολο από οδηγίες προς προγραμματιστές για να προσπελαύνουν μία υπηρεσία. Το API της ENSEMBL περιγράφεται εδώ: http://rest.ensembl.org/ (http://rest.ensembl.org/)

Παρατηρήστε ότι κάποιες υπηρεσίες είναι προσβάσιμες μέσω GET και κάποιες μέσω POST. Επίσης παρατηρείστε ότι για όλες τις υπηρεσίες υπάρχει κώδικας παραδείγματα σε python.

Σαν παράδειγμα θα χρησιμοποιήσουμε το API της ENSEMBL το οποίο αναφέρεται στο Variant Effect Predictor (http://www.ensembl.org/info/docs/tools/vep/index.html). Το Variant Effect Predictor μας δίνει διάφορες πληροφορίες σχετικά με το downstream effect και clinical significance ενώς variant. Αλλα πριν κάνουμε αυτό ας δούμε με ποιους τρόπους κωδικοποιούμε έναν variant.

Γενικότερα υπάρχουν δύο τρόποι.

Αν ο variant είναι γνωστός τότε κοιτάμε αν έχει καταχωρηθεί στη dbSNP (https://www.ncbi.nlm.nih.gov/projects/SNP/). Σε αυτή τη περίπτωση ο variant θα έχει εναν κωδικό με το φορμάτ: rsXXXXXX. πχ: rs56116432.

Αν ο variant δεν είναι γνωστός και δεν υπάρχει στη dbSNP τότε μπορούμε να τον περιγράψουμε μέσω του HGVS (http://varnomen.hgvs.org/recommendations/general/) φορμάτ. Η περιγραφή αυτή είναι αρκετά πολύπλοκη και δεφεύγει

GET request για dbSNP variant

Για να πάρουμε πληροφορίες από το Variant Effect Predictor για ένα dbSNP variant, η ENSEMBL δίνει αυτό το API:
http://rest.ensembl.org/documentation/info/vep_id_get (http://rest.ensembl.org/documentation/info/vep_id_get)
Μπορούμε να κάνουμε ένα request σε αυτό το API ως εξής:

In [32]:
```
import requests

# Το URL που πρέπει να χρησιμοποιήσουμε
# υπάρχει στη σελίδα http://rest.ensembl.org/documentation/info/vep_id_get
# Επίσης Αφού είναι GET Request βάζουμε τη πληροφορία που θέλουμε στο URL:
url = 'http://rest.ensembl.org/vep/human/id/rs56116432?'
headers = { "Content-Type" : "application/json"}

# Κάνουμε το GET request
r = requests.get(url, headers=headers)
```

Στη συνέχεια κοιτάμαι αν όλα πήγαν οκ:

In [33]:
```
print (r.ok) # Στην ουσία εδώ κοιτάει αν το response έχει κωδικό 200
```

```
True
```

Παίρνουμε τα JSON δεδομένα:

In [10]:
```
data = r.json()
print (data)
```


```
[{'transcript_consequences': [{'biotype': 'processed_transcript', 'gene_id': '
ENSG00000175164', 'cdna_end': 700, 'hgnc_id': 'HGNC:79', 'gene_symbol_source':
'HGNC', 'cdna_start': 700, 'variant_allele': 'T', 'transcript_id': 'ENST000004
53660', 'consequence_terms': ['non_coding_transcript_exon_variant', 'non_codin
g_transcript_variant'], 'impact': 'MODIFIER', 'strand': -1, 'gene_symbol': 'AB
O'}, {'biotype': 'protein_coding', 'gene_id': 'ENSG00000175164', 'sift_score':
0, 'hgnc_id': 'HGNC:79', 'polyphen_score': 0.997, 'sift_prediction': 'deleteri
ous', 'variant_allele': 'T', 'cds_start': 686, 'codons': 'gGc/gAc', 'transcrip
t_id': 'ENST00000538324', 'cds_end': 686, 'polyphen_prediction': 'probably_dam
aging', 'impact': 'MODERATE', 'cdna_end': 711, 'gene_symbol_source': 'HGNC', '
cdna_start': 711, 'consequence_terms': ['missense_variant'], 'protein_end': 22
9, 'protein_start': 229, 'strand': -1, 'amino_acids': 'G/D', 'gene_symbol': 'A
BO'}, {'biotype': 'protein_coding', 'gene_id': 'ENSG00000175164', 'sift_score
': 0, 'hgnc_id': 'HGNC:79', 'polyphen_score': 0.997, 'sift_prediction': 'delet
erious', 'variant_allele': 'T', 'cds_start': 686, 'codons': 'gGc/gAc', 'transc
ript_id': 'ENST00000611156', 'cds_end': 686, 'polyphen_prediction': 'probably_
damaging', 'impact': 'MODERATE', 'cdna_end': 711, 'gene_symbol_source': 'HGNC
', 'cdna_start': 711, 'consequence_terms': ['missense_variant'], 'protein_end
': 229, 'protein_start': 229, 'strand': -1, 'amino_acids': 'G/D', 'gene_symbol
': 'ABO'}], 'most_severe_consequence': 'missense_variant', 'input': 'rs5611643
2', 'seq_region_name': '9', 'allele_string': 'C/T', 'end': 133256042, 'assembl
y_name': 'GRCh38', 'colocated_variants': [{'exac_amr_maf': 0.004932, 'exac_sas
_maf': 0.001639, 'exac_nfe_maf': 0.005339, 'end': 133256042, 'eas_maf': 0, 'ex
ac_adj_allele': 'T', 'exac_eas_maf': 0, 'exac_fin_maf': 0.02601, 'exac_amr_all
ele': 'T', 'ea_maf': 0.003809, 'exac_oth_allele': 'T', 'minor_allele': 'T', 's
as_allele': 'T', 'amr_allele': 'T', 'exac_afr_allele': 'T', 'start': 13325604
2, 'ea_allele': 'T', 'allele_string': 'C/T', 'exac_eas_allele': 'T', 'afr_maf
': 0, 'amr_maf': 0.0014, 'sas_maf': 0.001, 'eur_allele': 'T', 'exac_fin_allele
': 'T', 'strand': 1, 'afr_allele': 'T', 'exac_maf': 0.003022, 'minor_allele_fr
eq': 0.0026, 'seq_region_name': 9, 'exac_nfe_allele': 'T', 'exac_adj_maf': 0.0
04439, 'exac_afr_maf': 0.0005079, 'aa_maf': 0.0007102, 'exac_sas_allele': 'T',
'eur_maf': 0.0109, 'aa_allele': 'T', 'eas_allele': 'T', 'id': 'rs56116432', 'e
xac_allele': 'T', 'exac_oth_maf': 0.004926}], 'strand': 1, 'id': 'rs56116432',
'start': 133256042}, {'transcript_consequences': [{'biotype': 'protein_coding
', 'gene_id': 'ENSG00000281879', 'sift_score': 0, 'hgnc_id': 'HGNC:79', 'polyp
hen_score': 0.997, 'sift_prediction': 'deleterious', 'variant_allele': 'T', 'c
ds_start': 689, 'codons': 'gGc/gAc', 'transcript_id': 'ENST00000626615', 'cds_
end': 689, 'polyphen_prediction': 'probably_damaging', 'impact': 'MODERATE', '
cdna_end': 714, 'gene_symbol_source': 'HGNC', 'cdna_start': 714, 'consequence_
terms': ['missense_variant'], 'protein_end': 230, 'protein_start': 230, 'stran
d': -1, 'amino_acids': 'G/D', 'gene_symbol': 'ABO'}], 'most_severe_consequence
': 'missense_variant', 'input': 'rs56116432', 'seq_region_name': 'CHR_HG2030_P
ATCH', 'allele_string': 'C/T', 'end': 133256189, 'assembly_name': 'GRCh38', 'c
olocated_variants': [{'allele_string': 'C/T', 'seq_region_name': 'CHR_HG2030_P
ATCH', 'end': 133256189, 'minor_allele': 'T', 'strand': 1, 'id': 'rs56116432',
'minor_allele_freq': 0.0026, 'start': 133256189}], 'strand': 1, 'id': 'rs56116
432', 'start': 133256189}]
```


In [11]:
```
print (len(data))
```

```
2
```

επέστρεψε 2 εγγραφές. Ας πάρουμε την πρώτη:

In [12]:
```
print (data[0])
```


```
{'transcript_consequences': [{'biotype': 'processed_transcript', 'gene_id': 'E
NSG00000175164', 'cdna_end': 700, 'hgnc_id': 'HGNC:79', 'gene_symbol_source':
'HGNC', 'cdna_start': 700, 'variant_allele': 'T', 'transcript_id': 'ENST000004
53660', 'consequence_terms': ['non_coding_transcript_exon_variant', 'non_codin
g_transcript_variant'], 'impact': 'MODIFIER', 'strand': -1, 'gene_symbol': 'AB
O'}, {'biotype': 'protein_coding', 'gene_id': 'ENSG00000175164', 'sift_score':
0, 'hgnc_id': 'HGNC:79', 'polyphen_score': 0.997, 'sift_prediction': 'deleteri
ous', 'variant_allele': 'T', 'cds_start': 686, 'codons': 'gGc/gAc', 'transcrip
t_id': 'ENST00000538324', 'cds_end': 686, 'polyphen_prediction': 'probably_dam
aging', 'impact': 'MODERATE', 'cdna_end': 711, 'gene_symbol_source': 'HGNC', '
cdna_start': 711, 'consequence_terms': ['missense_variant'], 'protein_end': 22
9, 'protein_start': 229, 'strand': -1, 'amino_acids': 'G/D', 'gene_symbol': 'A
BO'}, {'biotype': 'protein_coding', 'gene_id': 'ENSG00000175164', 'sift_score
': 0, 'hgnc_id': 'HGNC:79', 'polyphen_score': 0.997, 'sift_prediction': 'delet
erious', 'variant_allele': 'T', 'cds_start': 686, 'codons': 'gGc/gAc', 'transc
ript_id': 'ENST00000611156', 'cds_end': 686, 'polyphen_prediction': 'probably_
damaging', 'impact': 'MODERATE', 'cdna_end': 711, 'gene_symbol_source': 'HGNC
', 'cdna_start': 711, 'consequence_terms': ['missense_variant'], 'protein_end
': 229, 'protein_start': 229, 'strand': -1, 'amino_acids': 'G/D', 'gene_symbol
': 'ABO'}], 'most_severe_consequence': 'missense_variant', 'input': 'rs5611643
2', 'seq_region_name': '9', 'allele_string': 'C/T', 'end': 133256042, 'assembl
y_name': 'GRCh38', 'colocated_variants': [{'exac_amr_maf': 0.004932, 'exac_sas
_maf': 0.001639, 'exac_nfe_maf': 0.005339, 'end': 133256042, 'eas_maf': 0, 'ex
ac_adj_allele': 'T', 'exac_eas_maf': 0, 'exac_fin_maf': 0.02601, 'exac_amr_all
ele': 'T', 'ea_maf': 0.003809, 'exac_oth_allele': 'T', 'minor_allele': 'T', 's
as_allele': 'T', 'amr_allele': 'T', 'exac_afr_allele': 'T', 'start': 13325604
2, 'ea_allele': 'T', 'allele_string': 'C/T', 'exac_eas_allele': 'T', 'afr_maf
': 0, 'amr_maf': 0.0014, 'sas_maf': 0.001, 'eur_allele': 'T', 'exac_fin_allele
': 'T', 'strand': 1, 'afr_allele': 'T', 'exac_maf': 0.003022, 'minor_allele_fr
eq': 0.0026, 'seq_region_name': 9, 'exac_nfe_allele': 'T', 'exac_adj_maf': 0.0
04439, 'exac_afr_maf': 0.0005079, 'aa_maf': 0.0007102, 'exac_sas_allele': 'T',
'eur_maf': 0.0109, 'aa_allele': 'T', 'eas_allele': 'T', 'id': 'rs56116432', 'e
xac_allele': 'T', 'exac_oth_maf': 0.004926}], 'strand': 1, 'id': 'rs56116432',
'start': 133256042}
```


In [13]:
```
print (len(data[0]["transcript_consequences"]))
```

```
3
```

Η πρώτη εγγραφή έχει consequences σε 3 transcripts. Ας πάρουμε το 2ο:

In [14]:
```
print (data[0]["transcript_consequences"][1])
```


```
{'biotype': 'protein_coding', 'gene_id': 'ENSG00000175164', 'sift_score': 0, '
hgnc_id': 'HGNC:79', 'polyphen_score': 0.997, 'sift_prediction': 'deleterious
', 'variant_allele': 'T', 'cds_start': 686, 'codons': 'gGc/gAc', 'transcript_i
d': 'ENST00000538324', 'cds_end': 686, 'polyphen_prediction': 'probably_damagi
ng', 'impact': 'MODERATE', 'cdna_end': 711, 'gene_symbol_source': 'HGNC', 'cdn
a_start': 711, 'consequence_terms': ['missense_variant'], 'protein_end': 229,
'protein_start': 229, 'strand': -1, 'amino_acids': 'G/D', 'gene_symbol': 'ABO
'}
```


Ας δούμε ποιο είναι το consequence αυτού του mutation σε αυτό το transcript σύμφωνα με το [polyphen](http://genetics.bwh.harvard.edu/pph2/):

In [15]:
```
print (data[0]["transcript_consequences"][1]["polyphen_prediction"])
```

```
probably_damaging
```

POST request για dbSNP variant

Θα κάνουμε τώρα το ίδιο αλλά θα χρησιμοποιήσουμε το API για POST request: http://rest.ensembl.org/documentation/info/vep_id_post (http://rest.ensembl.org/documentation/info/vep_id_post)

In [17]:
```
url = 'http://rest.ensembl.org/vep/human/id'

# δηλώνουμε ότι το αποτέλεσμα θέλουμε να είναι σε JSON φορματ (Accept)
headers = { "Content-Type" : "application/json", "Accept" : "application/json"}

# Αφού είναι POST βάζουμε τα data ξεχωριστά:
data = { "ids" : ["rs56116432", "COSM476" ] }

# ΠΡΟΣΟΧΗ!!!
# Τα data είναι ένα dictionary το οποίο πρέπει να το μετατρέψουμε σε JSON!
import json
data_json = json.dumps(data)

# Κάνουμε το POST request:
r = requests.post(url, headers=headers, data=data_json)
```

Τσεκάρουμε ότι όλα πήγαν οκ:

In [18]:
```
print (r.ok)
```

```
True
```

παίρνουμε τα data:

In [19]:
```
data = r.json()
print (data)
```


```
[{'transcript_consequences': [{'biotype': 'processed_transcript', 'gene_id': '
ENSG00000175164', 'cdna_end': 700, 'hgnc_id': 'HGNC:79', 'gene_symbol_source':
'HGNC', 'cdna_start': 700, 'variant_allele': 'T', 'transcript_id': 'ENST000004
53660', 'consequence_terms': ['non_coding_transcript_exon_variant', 'non_codin
g_transcript_variant'], 'impact': 'MODIFIER', 'strand': -1, 'gene_symbol': 'AB
O'}, {'biotype': 'protein_coding', 'gene_id': 'ENSG00000175164', 'sift_score':
0, 'hgnc_id': 'HGNC:79', 'polyphen_score': 0.997, 'sift_prediction': 'deleteri
ous', 'variant_allele': 'T', 'cds_start': 686, 'codons': 'gGc/gAc', 'transcrip
t_id': 'ENST00000538324', 'cds_end': 686, 'polyphen_prediction': 'probably_dam
aging', 'impact': 'MODERATE', 'cdna_end': 711, 'gene_symbol_source': 'HGNC', '
cdna_start': 711, 'consequence_terms': ['missense_variant'], 'protein_end': 22
9, 'protein_start': 229, 'strand': -1, 'amino_acids': 'G/D', 'gene_symbol': 'A
BO'}, {'biotype': 'protein_coding', 'gene_id': 'ENSG00000175164', 'sift_score
': 0, 'hgnc_id': 'HGNC:79', 'polyphen_score': 0.997, 'sift_prediction': 'delet
erious', 'variant_allele': 'T', 'cds_start': 686, 'codons': 'gGc/gAc', 'transc
ript_id': 'ENST00000611156', 'cds_end': 686, 'polyphen_prediction': 'probably_
damaging', 'impact': 'MODERATE', 'cdna_end': 711, 'gene_symbol_source': 'HGNC
', 'cdna_start': 711, 'consequence_terms': ['missense_variant'], 'protein_end
': 229, 'protein_start': 229, 'strand': -1, 'amino_acids': 'G/D', 'gene_symbol
': 'ABO'}], 'most_severe_consequence': 'missense_variant', 'input': 'rs5611643
2', 'seq_region_name': '9', 'allele_string': 'C/T', 'end': 133256042, 'assembl
y_name': 'GRCh38', 'colocated_variants': [{'exac_amr_maf': 0.004932, 'exac_sas
_maf': 0.001639, 'exac_nfe_maf': 0.005339, 'end': 133256042, 'eas_maf': 0, 'ex
ac_adj_allele': 'T', 'exac_eas_maf': 0, 'exac_fin_maf': 0.02601, 'exac_amr_all
ele': 'T', 'ea_maf': 0.003809, 'exac_oth_allele': 'T', 'minor_allele': 'T', 's
as_allele': 'T', 'amr_allele': 'T', 'exac_afr_allele': 'T', 'start': 13325604
2, 'ea_allele': 'T', 'allele_string': 'C/T', 'exac_eas_allele': 'T', 'afr_maf
': 0, 'amr_maf': 0.0014, 'sas_maf': 0.001, 'eur_allele': 'T', 'exac_fin_allele
': 'T', 'strand': 1, 'afr_allele': 'T', 'exac_maf': 0.003022, 'minor_allele_fr
eq': 0.0026, 'seq_region_name': 9, 'exac_nfe_allele': 'T', 'exac_adj_maf': 0.0
04439, 'exac_afr_maf': 0.0005079, 'aa_maf': 0.0007102, 'exac_sas_allele': 'T',
'eur_maf': 0.0109, 'aa_allele': 'T', 'eas_allele': 'T', 'id': 'rs56116432', 'e
xac_allele': 'T', 'exac_oth_maf': 0.004926}], 'strand': 1, 'id': 'rs56116432',
'start': 133256042}, {'transcript_consequences': [{'biotype': 'protein_coding
', 'gene_id': 'ENSG00000281879', 'sift_score': 0, 'hgnc_id': 'HGNC:79', 'polyp
hen_score': 0.997, 'sift_prediction': 'deleterious', 'variant_allele': 'T', 'c
ds_start': 689, 'codons': 'gGc/gAc', 'transcript_id': 'ENST00000626615', 'cds_
end': 689, 'polyphen_prediction': 'probably_damaging', 'impact': 'MODERATE', '
cdna_end': 714, 'gene_symbol_source': 'HGNC', 'cdna_start': 714, 'consequence_
terms': ['missense_variant'], 'protein_end': 230, 'protein_start': 230, 'stran
d': -1, 'amino_acids': 'G/D', 'gene_symbol': 'ABO'}], 'most_severe_consequence
': 'missense_variant', 'input': 'rs56116432', 'seq_region_name': 'CHR_HG2030_P
ATCH', 'allele_string': 'C/T', 'end': 133256189, 'assembly_name': 'GRCh38', 'c
olocated_variants': [{'allele_string': 'C/T', 'seq_region_name': 'CHR_HG2030_P
ATCH', 'end': 133256189, 'minor_allele': 'T', 'strand': 1, 'id': 'rs56116432',
'minor_allele_freq': 0.0026, 'start': 133256189}], 'strand': 1, 'id': 'rs56116
432', 'start': 133256189}, {'most_severe_consequence': '?', 'allele_string': '
COSMIC_MUTATION', 'input': 'COSM476', 'seq_region_name': '7', 'end': 14075333
6, 'assembly_name': 'GRCh38', 'colocated_variants': [{'allele_string': 'HGMD_M
UTATION', 'seq_region_name': 7, 'phenotype_or_disease': 1, 'strand': 1, 'id':
'CM112509', 'end': 140753336, 'start': 140753336}, {'allele_string': 'COSMIC_M
UTATION', 'seq_region_name': 7, 'phenotype_or_disease': 1, 'start': 140753336,
'strand': 1, 'id': 'COSM18443', 'end': 140753336, 'somatic': 1}, {'allele_stri
ng': 'COSMIC_MUTATION', 'seq_region_name': 7, 'phenotype_or_disease': 1, 'star
t': 140753336, 'strand': 1, 'id': 'COSM476', 'end': 140753336, 'somatic': 1},
{'allele_string': 'COSMIC_MUTATION', 'seq_region_name': 7, 'phenotype_or_disea
se': 1, 'start': 140753336, 'strand': 1, 'id': 'COSM6137', 'end': 140753336, '
somatic': 1}], 'strand': 1, 'id': 'COSM476', 'start': 140753336}]
```


Είναι τα ίδια με πριν:

In [20]:
```
print (data[0]["transcript_consequences"][1]["polyphen_prediction"])
```

```
probably_damaging
```

GET request για HGVS variant

Η ENSEMBL μας δίνει ένα API για να πάρουμε πληροφορίες από το Variant Effect Predictor για ένα HGVS variant μέσω GET request: http://rest.ensembl.org/documentation/info/vep_hgvs_get (http://rest.ensembl.org/documentation/info/vep_hgvs_get)

In [21]:
```
url = 'http://rest.ensembl.org/vep/human/hgvs/9:g.22125504G>C?'
headers={ "Content-Type" : "application/json"}

r = requests.get(url, headers=headers)
```

In [22]:
```
print (r.ok)
```

```
True
```

In [23]:
```
data = r.json()
```

In [24]:
```
print (data)
```


```
[{'transcript_consequences': [{'consequence_terms': ['downstream_gene_variant
'], 'impact': 'MODIFIER', 'biotype': 'antisense', 'gene_id': 'ENSG00000240498
', 'hgnc_id': 'HGNC:34341', 'distance': 4407, 'gene_symbol': 'CDKN2B-AS1', 'ge
ne_symbol_source': 'HGNC', 'strand': 1, 'variant_allele': 'C', 'transcript_id
': 'ENST00000422420'}, {'consequence_terms': ['downstream_gene_variant'], 'imp
act': 'MODIFIER', 'biotype': 'antisense', 'gene_id': 'ENSG00000240498', 'hgnc_
id': 'HGNC:34341', 'distance': 4409, 'gene_symbol': 'CDKN2B-AS1', 'gene_symbol
_source': 'HGNC', 'strand': 1, 'variant_allele': 'C', 'transcript_id': 'ENST00
000428597'}, {'consequence_terms': ['downstream_gene_variant'], 'impact': 'MOD
IFIER', 'biotype': 'antisense', 'gene_id': 'ENSG00000240498', 'hgnc_id': 'HGN
C:34341', 'distance': 4932, 'gene_symbol': 'CDKN2B-AS1', 'gene_symbol_source':
'HGNC', 'strand': 1, 'variant_allele': 'C', 'transcript_id': 'ENST00000577551
'}, {'consequence_terms': ['downstream_gene_variant'], 'impact': 'MODIFIER', '
biotype': 'antisense', 'gene_id': 'ENSG00000240498', 'hgnc_id': 'HGNC:34341',
'distance': 4858, 'gene_symbol': 'CDKN2B-AS1', 'gene_symbol_source': 'HGNC', '
strand': 1, 'variant_allele': 'C', 'transcript_id': 'ENST00000580576'}, {'cons
equence_terms': ['downstream_gene_variant'], 'impact': 'MODIFIER', 'biotype':
'antisense', 'gene_id': 'ENSG00000240498', 'hgnc_id': 'HGNC:34341', 'distance
': 4932, 'gene_symbol': 'CDKN2B-AS1', 'gene_symbol_source': 'HGNC', 'strand':
1, 'variant_allele': 'C', 'transcript_id': 'ENST00000581051'}, {'consequence_t
erms': ['downstream_gene_variant'], 'impact': 'MODIFIER', 'biotype': 'antisens
e', 'gene_id': 'ENSG00000240498', 'hgnc_id': 'HGNC:34341', 'distance': 4932, '
gene_symbol': 'CDKN2B-AS1', 'gene_symbol_source': 'HGNC', 'strand': 1, 'varian
t_allele': 'C', 'transcript_id': 'ENST00000582072'}, {'consequence_terms': ['d
ownstream_gene_variant'], 'impact': 'MODIFIER', 'biotype': 'antisense', 'gene_
id': 'ENSG00000240498', 'hgnc_id': 'HGNC:34341', 'distance': 4932, 'gene_symbo
l': 'CDKN2B-AS1', 'gene_symbol_source': 'HGNC', 'strand': 1, 'variant_allele':
'C', 'transcript_id': 'ENST00000584020'}, {'consequence_terms': ['downstream_g
ene_variant'], 'impact': 'MODIFIER', 'biotype': 'antisense', 'gene_id': 'ENSG0
0000240498', 'hgnc_id': 'HGNC:34341', 'distance': 4932, 'gene_symbol': 'CDKN2B
-AS1', 'gene_symbol_source': 'HGNC', 'strand': 1, 'variant_allele': 'C', 'tran
script_id': 'ENST00000584637'}, {'consequence_terms': ['downstream_gene_varian
t'], 'impact': 'MODIFIER', 'biotype': 'antisense', 'gene_id': 'ENSG00000240498
', 'hgnc_id': 'HGNC:34341', 'distance': 4932, 'gene_symbol': 'CDKN2B-AS1', 'ge
ne_symbol_source': 'HGNC', 'strand': 1, 'variant_allele': 'C', 'transcript_id
': 'ENST00000584816'}, {'consequence_terms': ['downstream_gene_variant'], 'imp
act': 'MODIFIER', 'biotype': 'antisense', 'gene_id': 'ENSG00000240498', 'hgnc_
id': 'HGNC:34341', 'distance': 4960, 'gene_symbol': 'CDKN2B-AS1', 'gene_symbol
_source': 'HGNC', 'strand': 1, 'variant_allele': 'C', 'transcript_id': 'ENST00
000585267'}], 'most_severe_consequence': 'downstream_gene_variant', 'input': '
9:g.22125504G>C', 'seq_region_name': '9', 'allele_string': 'G/C', 'end': 22125
504, 'assembly_name': 'GRCh38', 'colocated_variants': [{'allele_string': 'G/C
', 'eur_maf': 0.4722, 'amr_maf': 0.4553, 'sas_maf': 0.4908, 'eur_allele': 'C',
'end': 22125504, 'eas_maf': 0.5367, 'afr_allele': 'C', 'minor_allele': 'C', 's
eq_region_name': 9, 'minor_allele_freq': 0.4181, 'strand': 1, 'pubmed': '24262
325,19501493,22042884,21860704,21149552,20159871,19474294,21894447,21971053,21
804106,20502693,22199011,18224312,22400124,18533027,18852197,21297524,2240324
0,22856518,23963167,19343170,20386740,21400687,24728607,20017983,24573017,2460
7648,20549515,22144573,22623978,22029572,18362232,19173706,19214202,26252781,2
0435227,21606135,19924713,17554300,19955471,19956433,25717410,24098343,1878030
2,18675980,19475673,20231156,20858905,21152093,21698238,24906238,17634449,1897
9498,19164808,19207022,19750184,20098575,20981302,21242481,21369780,22295058,2
2848412,25617895,23729007,18469204,20605023,21372283,22429504,26483964,2387019
5,18704761,23587283,24926413,19463184,24676469,21424681,20175863,22505696,1955
9344,19578366,23142796,19171343,24246088,18987759,19819472,19926059,21375403,2
1385355,21705410,24777168,25105296,19888323,23454037,18264662,18599798,1865294
6,18654002,18925945,18957718,19135198,19319159,19329499,19548844,19664850,1970
9660,19885677,19956784,20031540,20031580,20031606,20230275,20335276,20395598,2
040077', 'sas_allele': 'C', 'phenotype_or_disease': 1, 'amr_allele': 'C', 'afr
_maf': 0.2133, 'eas_allele': 'C', 'id': 'rs1333049', 'start': 22125504}], 'str
and': 1, 'id': '9:g.22125504G>C', 'start': 22125504}]
```

In [25]:
```
print (len(data))
```

```
1
```

Επέστρεψε μόνο μία εγγρφή.

In [26]:
```
data[0].keys()
```

Out[26]:
```
dict_keys(['transcript_consequences', 'most_severe_consequence', 'input', 'seq
_region_name', 'allele_string', 'end', 'assembly_name', 'colocated_variants',
'strand', 'id', 'start'])
```

Ας δούμε τα transcript consequences:

In [27]:
```
data[0]["transcript_consequences"]
```

Out[27]:

```
[{'biotype': 'antisense',
  'consequence_terms': ['downstream_gene_variant'],
  'distance': 4407,
  'gene_id': 'ENSG00000240498',
  'gene_symbol': 'CDKN2B-AS1',
  'gene_symbol_source': 'HGNC',
  'hgnc_id': 'HGNC:34341',
  'impact': 'MODIFIER',
  'strand': 1,
  'transcript_id': 'ENST00000422420',
  'variant_allele': 'C'},
 {'biotype': 'antisense',
  'consequence_terms': ['downstream_gene_variant'],
  'distance': 4409,
  'gene_id': 'ENSG00000240498',
  'gene_symbol': 'CDKN2B-AS1',
  'gene_symbol_source': 'HGNC',
  'hgnc_id': 'HGNC:34341',
  'impact': 'MODIFIER',
  'strand': 1,
  'transcript_id': 'ENST00000428597',
  'variant_allele': 'C'},
 {'biotype': 'antisense',
  'consequence_terms': ['downstream_gene_variant'],
  'distance': 4932,
  'gene_id': 'ENSG00000240498',
  'gene_symbol': 'CDKN2B-AS1',
  'gene_symbol_source': 'HGNC',
  'hgnc_id': 'HGNC:34341',
  'impact': 'MODIFIER',
  'strand': 1,
  'transcript_id': 'ENST00000577551',
  'variant_allele': 'C'},
 {'biotype': 'antisense',
  'consequence_terms': ['downstream_gene_variant'],
  'distance': 4858,
  'gene_id': 'ENSG00000240498',
  'gene_symbol': 'CDKN2B-AS1',
  'gene_symbol_source': 'HGNC',
  'hgnc_id': 'HGNC:34341',
  'impact': 'MODIFIER',
  'strand': 1,
  'transcript_id': 'ENST00000580576',
  'variant_allele': 'C'},
 {'biotype': 'antisense',
  'consequence_terms': ['downstream_gene_variant'],
  'distance': 4932,
  'gene_id': 'ENSG00000240498',
  'gene_symbol': 'CDKN2B-AS1',
  'gene_symbol_source': 'HGNC',
  'hgnc_id': 'HGNC:34341',
  'impact': 'MODIFIER',
  'strand': 1,
  'transcript_id': 'ENST00000581051',
  'variant_allele': 'C'},
 {'biotype': 'antisense',
  'consequence_terms': ['downstream_gene_variant'],
  'distance': 4932,
  'gene_id': 'ENSG00000240498',
  'gene_symbol': 'CDKN2B-AS1',
  'gene_symbol_source': 'HGNC',
  'hgnc_id': 'HGNC:34341',
  'impact': 'MODIFIER',
  'strand': 1,
  'transcript_id': 'ENST00000582072',
  'variant_allele': 'C'},
 {'biotype': 'antisense',
  'consequence_terms': ['downstream_gene_variant'],
```

POST request για HGVS variant

Μπορούμε να κάνουμε το ίδιο, κάνοντας ένα POST request: http://rest.ensembl.org/documentation/info/vep_hgvs_post (http://rest.ensembl.org/documentation/info/vep_hgvs_post)

In [28]:
```
url = 'http://rest.ensembl.org/vep/human/hgvs'
headers = { "Content-Type" : "application/json", "Accept" : "application/json"}
data = { "hgvs_notations" : ["9:g.22125504G>C"] } # Παρατηρείστε ότι αυτό είναι μία λίστα.
                                                  # Μπορούμε να βάλουμε (σχεδόν) όσα HGVS variants θέλουμε

data_json = json.dumps(data)

r = requests.post(url, headers=headers, data=data_json)
```

In [29]:
```
print (r.ok)
```

```
True
```

In [30]:
```
returned_data = r.json()
print (returned_data)
```


```
[{'transcript_consequences': [{'consequence_terms': ['downstream_gene_variant
'], 'impact': 'MODIFIER', 'biotype': 'antisense', 'gene_id': 'ENSG00000240498
', 'hgnc_id': 'HGNC:34341', 'distance': 4407, 'gene_symbol': 'CDKN2B-AS1', 'ge
ne_symbol_source': 'HGNC', 'strand': 1, 'variant_allele': 'C', 'transcript_id
': 'ENST00000422420'}, {'consequence_terms': ['downstream_gene_variant'], 'imp
act': 'MODIFIER', 'biotype': 'antisense', 'gene_id': 'ENSG00000240498', 'hgnc_
id': 'HGNC:34341', 'distance': 4409, 'gene_symbol': 'CDKN2B-AS1', 'gene_symbol
_source': 'HGNC', 'strand': 1, 'variant_allele': 'C', 'transcript_id': 'ENST00
000428597'}, {'consequence_terms': ['downstream_gene_variant'], 'impact': 'MOD
IFIER', 'biotype': 'antisense', 'gene_id': 'ENSG00000240498', 'hgnc_id': 'HGN
C:34341', 'distance': 4932, 'gene_symbol': 'CDKN2B-AS1', 'gene_symbol_source':
'HGNC', 'strand': 1, 'variant_allele': 'C', 'transcript_id': 'ENST00000577551
'}, {'consequence_terms': ['downstream_gene_variant'], 'impact': 'MODIFIER', '
biotype': 'antisense', 'gene_id': 'ENSG00000240498', 'hgnc_id': 'HGNC:34341',
'distance': 4858, 'gene_symbol': 'CDKN2B-AS1', 'gene_symbol_source': 'HGNC', '
strand': 1, 'variant_allele': 'C', 'transcript_id': 'ENST00000580576'}, {'cons
equence_terms': ['downstream_gene_variant'], 'impact': 'MODIFIER', 'biotype':
'antisense', 'gene_id': 'ENSG00000240498', 'hgnc_id': 'HGNC:34341', 'distance
': 4932, 'gene_symbol': 'CDKN2B-AS1', 'gene_symbol_source': 'HGNC', 'strand':
1, 'variant_allele': 'C', 'transcript_id': 'ENST00000581051'}, {'consequence_t
erms': ['downstream_gene_variant'], 'impact': 'MODIFIER', 'biotype': 'antisens
e', 'gene_id': 'ENSG00000240498', 'hgnc_id': 'HGNC:34341', 'distance': 4932, '
gene_symbol': 'CDKN2B-AS1', 'gene_symbol_source': 'HGNC', 'strand': 1, 'varian
t_allele': 'C', 'transcript_id': 'ENST00000582072'}, {'consequence_terms': ['d
ownstream_gene_variant'], 'impact': 'MODIFIER', 'biotype': 'antisense', 'gene_
id': 'ENSG00000240498', 'hgnc_id': 'HGNC:34341', 'distance': 4932, 'gene_symbo
l': 'CDKN2B-AS1', 'gene_symbol_source': 'HGNC', 'strand': 1, 'variant_allele':
'C', 'transcript_id': 'ENST00000584020'}, {'consequence_terms': ['downstream_g
ene_variant'], 'impact': 'MODIFIER', 'biotype': 'antisense', 'gene_id': 'ENSG0
0000240498', 'hgnc_id': 'HGNC:34341', 'distance': 4932, 'gene_symbol': 'CDKN2B
-AS1', 'gene_symbol_source': 'HGNC', 'strand': 1, 'variant_allele': 'C', 'tran
script_id': 'ENST00000584637'}, {'consequence_terms': ['downstream_gene_varian
t'], 'impact': 'MODIFIER', 'biotype': 'antisense', 'gene_id': 'ENSG00000240498
', 'hgnc_id': 'HGNC:34341', 'distance': 4932, 'gene_symbol': 'CDKN2B-AS1', 'ge
ne_symbol_source': 'HGNC', 'strand': 1, 'variant_allele': 'C', 'transcript_id
': 'ENST00000584816'}, {'consequence_terms': ['downstream_gene_variant'], 'imp
act': 'MODIFIER', 'biotype': 'antisense', 'gene_id': 'ENSG00000240498', 'hgnc_
id': 'HGNC:34341', 'distance': 4960, 'gene_symbol': 'CDKN2B-AS1', 'gene_symbol
_source': 'HGNC', 'strand': 1, 'variant_allele': 'C', 'transcript_id': 'ENST00
000585267'}], 'most_severe_consequence': 'downstream_gene_variant', 'input': '
9:g.22125504G>C', 'seq_region_name': '9', 'allele_string': 'G/C', 'end': 22125
504, 'assembly_name': 'GRCh38', 'colocated_variants': [{'allele_string': 'G/C
', 'eur_maf': 0.4722, 'amr_maf': 0.4553, 'sas_maf': 0.4908, 'eur_allele': 'C',
'end': 22125504, 'eas_maf': 0.5367, 'afr_allele': 'C', 'minor_allele': 'C', 's
eq_region_name': 9, 'minor_allele_freq': 0.4181, 'strand': 1, 'pubmed': '24262
325,19501493,22042884,21860704,21149552,20159871,19474294,21894447,21971053,21
804106,20502693,22199011,18224312,22400124,18533027,18852197,21297524,2240324
0,22856518,23963167,19343170,20386740,21400687,24728607,20017983,24573017,2460
7648,20549515,22144573,22623978,22029572,18362232,19173706,19214202,26252781,2
0435227,21606135,19924713,17554300,19955471,19956433,25717410,24098343,1878030
2,18675980,19475673,20231156,20858905,21152093,21698238,24906238,17634449,1897
9498,19164808,19207022,19750184,20098575,20981302,21242481,21369780,22295058,2
2848412,25617895,23729007,18469204,20605023,21372283,22429504,26483964,2387019
5,18704761,23587283,24926413,19463184,24676469,21424681,20175863,22505696,1955
9344,19578366,23142796,19171343,24246088,18987759,19819472,19926059,21375403,2
1385355,21705410,24777168,25105296,19888323,23454037,18264662,18599798,1865294
6,18654002,18925945,18957718,19135198,19319159,19329499,19548844,19664850,1970
9660,19885677,19956784,20031540,20031580,20031606,20230275,20335276,20395598,2
040077', 'sas_allele': 'C', 'phenotype_or_disease': 1, 'amr_allele': 'C', 'afr
_maf': 0.2133, 'eas_allele': 'C', 'id': 'rs1333049', 'start': 22125504}], 'str
and': 1, 'id': '9:g.22125504G>C', 'start': 22125504}]
```

NCBI / Entrez

ENTREZ (https://www.ncbi.nlm.nih.gov/gquery/). Η ENTREZ είναι μία ομογενοποιημένη βάση δεδομένων από το NCBI (Το αμερικάνικο EMBL). Δυστυχώς το API της ENTREZ δεν είναι τόσο φιλικό όσο της ENSEMBL. Το documentation (https://www.ncbi.nlm.nih.gov/books/NBK25501/) είναι διάσπαρτο και δεν υπάρχουν πολλά παραδείγματα (Και για αυτά που υπάρχουν είναι σε perl (https://www.ncbi.nlm.nih.gov/books/NBK25498/) 😱). Ευτυχώς η biopython περιέχει μεθόδους για πρόσβαση σε αυτό το API: http://biopython.org/DIST/docs/tutorial/Tutorial.html#htoc110 (http://biopython.org/DIST/docs/tutorial/Tutorial.html#htoc110)

Η RefSeq (https://en.wikipedia.org/wiki/RefSeq) είναι η βάση δεδομένων με ακολουθίες του NCBI. Κάθε ακολουθία έχει κωδικό XX_YYYYY.Z. π.χ: NM_178510.2 . Το XX είναι το είδος της ακολουθίας (δείτε περισσότερα στο TABLE 1, εδώ (https://www.ncbi.nlm.nih.gov/books/NBK21091/)). Το YYYYY είναι ένας αριθμός και το Z είναι η έκδοση του transcipt. Παράδειγμα εγγραφής στη RefSeq: https://www.ncbi.nlm.nih.gov/nuccore/1653962517 (https://www.ncbi.nlm.nih.gov/nuccore/1653962517)

Clinvar

https://www.ncbi.nlm.nih.gov/clinvar (https://www.ncbi.nlm.nih.gov/clinvar)

Περιέχει πληροφορίες για πιθανές κλινικές επιπτώσεις των μεταλλάξεων.

- Η clinvar έχει ένα σύστημα βαθμολόγησης για το πόσο σημαντική είναι η συσχέτιση μίας μετάλλαξης με μία ασθένεια. Το σύστημα αυτό έχει να κάνει με το πόσο εμπεριστατομένη είναι η συσχέτιση από ένα πάνελ ειδικών και κωδικοποιείται με ένα σύστημα από 4 (το πολύ) αστεράκια. https://www.ncbi.nlm.nih.gov/clinvar/docs/review_status/ (https://www.ncbi.nlm.nih.gov/clinvar/docs/review_status/)
- Κάθε μετάλλαξη έχει έναν κωδικό VCVXXXXX.Y Π.χ. VCV000002105.1 (https://www.ncbi.nlm.nih.gov/clinvar/variation/2105/)
- Κάθε μεταλλάξη συσχετίζεται με πολλές ασθένειες. Κάθε τέτοια συσχέτιση έχει έναν κωδικό: RCVXXXXXXX.Y . Για παράδειγμα RCV000211297.1 (https://www.ncbi.nlm.nih.gov/clinvar/RCV000211297.1/)
- Κάθε RCV κωδικός έχει πολλά evidences. Ή αλλιώς πειράματα τα οποία επιβεβαιώνουν (ή όχι..) τη συσχέτιση της μετάλλαξης με την ασθένα. Ένα evidence έχει κωδικό SCVXXXXXX . Για παράδειγμα SCV000268223
- περισσότερα σχετικά με το μοντέλο της clinvar (https://www.ncbi.nlm.nih.gov/clinvar/docs/identifiers/)

mygene.info (http://docs.mygene.info/en/latest/doc/data.html)

Το mygene.info είναι μία από τις πιο δημοφιλής, γρήγορη και σύγχρονη πηγή (μετα)μεταπληροφορία για γονίδια. Περιέχει πληροφορία για τη θέση, λειτουργία, μονοπάτια κτλ για όλα τα γονίδια από πολλούς οργανισμούς.

Υπάρχει αναλυτική περιγραφή στο documentation του mygene.info για το πως μπορούμε να κάνουμε GET requests για να πάρουμε διάφορες πληροφορίες για γονίδια (http://docs.mygene.info/en/latest/doc/query_service.html).

Για παράδειγμα έστω ότι θέλουμε να βρούμε ποια είναι η θέση του γονιδίου TPMT. Μέσω της `requests`:

```
import requests

parameters = {
  'fields': 'genomic_pos',
  'species': 'human',
  'q' : 'symbol:tpmt',
}
url = 'http://mygene.info/v3/query'

response = requests.get(url, params=parameters)
```

Στη συνέχεια μπορούμε να δούμε αν όλα πήγαν καλά:

```
response.ok # Αυτό πρέπει να είναι True
```

Αφού επιβεβαιώσουμε ότι το `response.ok` είναι `True`, μπορούμε να πάρουμε το αποτέλεσμα σε μορφή json:

```
data = response.json()
print (data)
```

```
{'max_score': 88.12873,
 'took': 4,
 'total': 1,
 'hits': [{'_id': '7172',
   '_score': 88.12873,
   'genomic_pos': {'chr': '6',
    'end': 18155074,
    'ensemblgene': 'ENSG00000137364',
    'start': 18128311,
    'strand': -1}}]}
```

Βλέπουμε ότι υπάρχει ένα `start` και ένα `end`. Μπορούμε να τα προσπελάσουμε:

```
chromosome = data['hits'][0]['genomic_pos']['chr']
start = data['hits'][0]['genomic_pos']['start']
end = data['hits'][0]['genomic_pos']['end']
print (chromosome, start, pos)
# Τυπώνει: ('6', 18128311, 18155074)
```

myvariant.info (http://docs.myvariant.info/en/latest/)

Παρόμοια με το mygene.info υπάρχει και το myvariant.info το οποίο παρέχει ένα API για πληροφορίες για μεταλλάξεις (http://docs.myvariant.info/en/latest/). Η πρόσβαση γίνεται μέσω GET requests. Για παράδειγμα ποια είναι η θέση της μετάλλαξης rs58991260;

```
url = 'http://myvariant.info/v1/query'
parameters = {
    'q': 'rs58991260',
    'fields': 'dbsnp',
}
```

In [4]:
```
import requests

r = requests.get("http://myvariant.info/v1/query?q=rs1800497", headers={ "Content-Type" : "application/json"})
```

In [5]:
```
r.ok
```

Out[5]:
```
True
```

In [6]:
```
data = r.json()
```

In [7]:
```
conditions = [x['conditions']['name'] for x in data['hits'][0]['clinvar']['rcv']]
```

In [8]:
```
conditions
```

Out[8]:
```
['Dopamine receptor d2, reduced brain density of',
 'ethanol response - Toxicity/ADR',
 'clozapine response - Toxicity/ADR',
 'bupropion response - Efficacy',
 'olanzapine response - Toxicity/ADR',
 'antipsychotics response - Toxicity/ADR',
 'risperidone response - Toxicity/ADR',
 'not specified']
```

GENCODE

Η GENCODE (https://www.gencodegenes.org/) είναι μία προσπάθεια ώστε να καταγραφούν όλα τα "λειτουργικά" κομμάτια του ανθρώπινου γονιδιώματος. Χρησιμοποιώντας διάφορες μεθοδολογίες (https://en.wikipedia.org/wiki/GENCODE#Methodology) (αυτόματες και με ανθρώπινο curation), καταγράφει σε ποιες περιοχές υπάρχουν γονίδια και πως τα γονίδια αυτά χωρίζονται σε υπο-περιοχές (introns, exons, CDS, stop codons, start codon, 5' UTR και 3'UTR).

Σε αυτό το ftp site: `ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_29/` υπάρχουν όλα τα αρχεία με τη τελευταία έκδοση της GENCODE. Ειδικά σε αυτό το αρχείο: `ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_29/_README.TXT` περιγράφεται τι περιέχει το κάθε αρχείο. Εκεί διαβάζουμε:

```
1. gencode.vX.annotation.{gtf,gff3}.gz:
  Main file, gene annotation on reference chromosomes in GTF and GFF3 file formats.
  These are the main GENCODE gene annotation files. They contain annotation (genes,
  transcripts, exons, start_codon, stop_codon, UTRs, CDS) on the reference chromoso
mes,
  which are chr1-22, X, Y, M in human and chr1-19, X, Y, M in mouse.
```

Μπορούμε να χρησιμοποιήσουμε το `gencode.v29.annotation.gff3.gz` το οποίο μπορούμε να κατεβάσουμε απο εδώ: ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_29/gencode.v29.annotation.gff3.gz Επίσης μπορείτε να μελετήσετε το format gff3 (http://gmod.org/wiki/GFF3) το οποίο χρησιμοποιεί η GENCODE. Η ίδια πληροφορία υπάρχει και σε ένα άλλο format το οποίο λέγεται GTF, αλλά το GFF3 θεωρείται λίγο πιο εξελιγμένο. Μπορείτε να διαβάσετε περισσότερα εδώ (https://www.biostars.org/p/99462/).

Ας κατεβάσουμε λοιπόν το αρχείο από το command line:

```
wget ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_29/gencode.v2
9.annotation.gff3.gz
```

Μέσα σε αυτό το αρχείο βρίσκουμε αυτές τις γραμμές: (Εχω κρατήσει μόνο ένα συγκεκριμμένο μήκος για κάθε γραμμή)

```
chr1    ENSEMBL transcript      2586750 2591467 .       +       .       ID=ENST0000
0444521.6;Parent=ENSG00000157870.15;
chr1    ENSEMBL exon    2586750 2586948 .       +       .       ID=exon:ENST0000044
4521.6:1;Parent=ENST00000444521.6;
chr1    ENSEMBL CDS     2586796 2586948 .       +       0       ID=CDS:ENST00000444
521.6;Parent=ENST00000444521.6;
chr1    ENSEMBL start_codon     2586796 2586798 .       +       0       ID=start_co
don:ENST00000444521.6;Parent=ENST00000444521.6;
chr1    ENSEMBL exon    2587091 2587295 .       +       .       ID=exon:ENST0000044
4521.6:2;Parent=ENST00000444521.6;
chr1    ENSEMBL CDS     2587091 2587295 .       +       0       ID=CDS:ENST00000444
521.6;Parent=ENST00000444521.6;
chr1    ENSEMBL exon    2587741 2587792 .       +       .       ID=exon:ENST0000044
4521.6:3;Parent=ENST00000444521.6;
chr1    ENSEMBL CDS     2587741 2587792 .       +       2       ID=CDS:ENST00000444
521.6;Parent=ENST00000444521.6;
chr1    ENSEMBL exon    2588336 2588453 .       +       .       ID=exon:ENST0000044
4521.6:4;Parent=ENST00000444521.6;
chr1    ENSEMBL CDS     2588336 2588453 .       +       1       ID=CDS:ENST00000444
521.6;Parent=ENST00000444521.6;
chr1    ENSEMBL exon    2588550 2588625 .       +       .       ID=exon:ENST0000044
4521.6:5;Parent=ENST00000444521.6;
chr1    ENSEMBL CDS     2588550 2588625 .       +       0       ID=CDS:ENST00000444
521.6;Parent=ENST00000444521.6;
chr1    ENSEMBL exon    2588922 2589040 .       +       .       ID=exon:ENST0000044
4521.6:6;Parent=ENST00000444521.6;
chr1    ENSEMBL CDS     2588922 2589040 .       +       2       ID=CDS:ENST00000444
```

Άλλες βάσεις δεδομένων

Γενικότερα το τοπίο με της online βάσεις γενομικών δεδομένων είναι αρκετά δυναμικό. Καινούργιες βάσεις προστίθονται, νέα APIs κτλ. Εκτός από την ENSEMBL άλλες υπάρχουσες βάσεις είναι:

- gnonAD (https://gnomad.broadinstitute.org/)
- 1000 Genomes Project (https://en.wikipedia.org/wiki/1000_Genomes_Project) Το #3 πιο σημαντικό project στη γενετική (από ιστορικής άποψης) μετά το Human Genome Project (https://en.wikipedia.org/wiki/Human_Genome_Project) και το HapMap projet (https://en.wikipedia.org/wiki/International_HapMap_Project). Περιέχει το DNA sequence και τους γονότυπους από 2500 ανθρώπους.
- KEGG (https://www.genome.jp/kegg/), wikipedia (https://en.wikipedia.org/wiki/KEGG), rest API (https://www.kegg.jp/kegg/rest/keggapi.html)

In [ ]: