

Υπολογιστική ανακάλυψη φαρμάκων με χρήση της πλατφόρμας TorchDrug



Παπικινός Θωμάς

PKM202025, c20papi@ionio.gr

*ΠΜΣ “Ψηφιακές Εφαρμογές και Καινοτομία”
Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο*

Πίνακας περιεχομένων

Περίληψη.....	3
1. Εισαγωγή.....	3
2. Σκοπός και εργαλεία που θα χρησιμοποιηθούν.....	5
2.1 Σκοπός.....	5
2.2 Περιβάλλον προγραμματισμού.....	6
2.3 Η πλατφόρμα TorchDrug.....	6
2.4 Βασικά στοιχεία του TorchDrug.....	6
3. Πρόβλεψη ιδιοτήτων.....	9
3.1 Διαδικασία.....	9
3.2 Dataset.....	9
3.3 Προετοιμασία του dataset.....	10
3.4 Επιλογή μοντέλου.....	10
3.5 Εκπαίδευση και αξιολόγηση του μοντέλου.....	10
3.6 Παραδείγματα προβλέψεων.....	12
4. Δημιουργία μορίων.....	13
4.1 Περί της διαδικασίας.....	13
4.2 Το dataset της προεκπαίδευσης.....	13
4.3 Εκτέλεση της προεκπαίδευσης.....	14
4.4 QED score ενός μορίου.....	17
4.5 Δημιουργία του τελικού μοντέλου.....	19
5. Βιοιατρικές αναπαραστάσεις γνώσης.....	21
5.1 Σημασιολογικά δίκτυα.....	21
5.2 Η βάση δεδομένων Hetionet.....	22
5.3 Παράδειγμα χρήσης.....	25
5.4 Διερεύνηση της σχέσης της AD με τον διαβήτη.....	27

6. Συμπεράσματα	29
Παράρτημα.....	31
Google Colab notebooks	31
Κώδικας του κεφαλαίου “Πρόβλεψη ιδιοτήτων”	31
Κώδικας του κεφαλαίου “Δημιουργία μορίων”	35
Κώδικας του κεφαλαίου “Βιοιατρικές αναπαραστάσεις γνώσης”	40
Βιβλιογραφία.....	42

Περίληψη

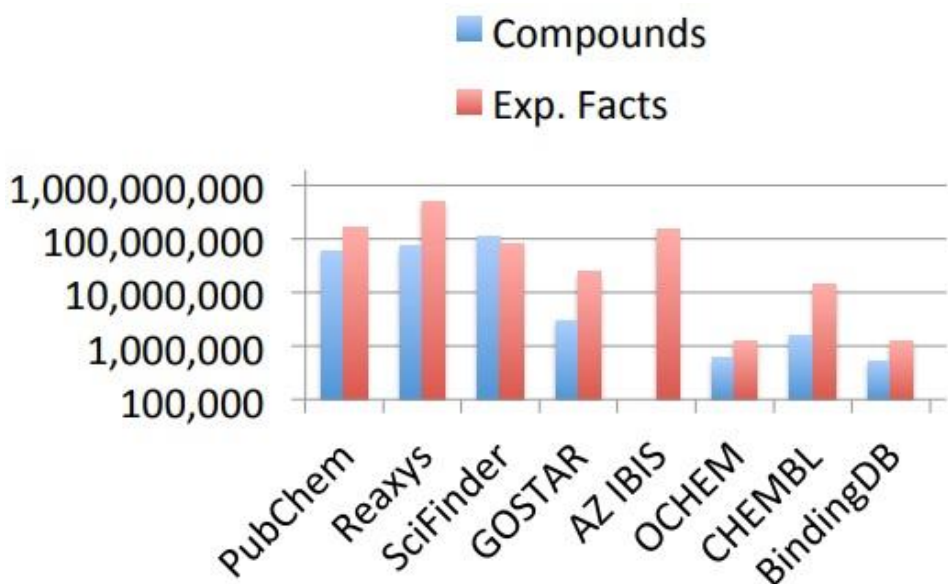
Η παρούσα εργασία αποτελεί μια επίδειξη του πώς μπορούν να εκτελεστούν διάφορα βήματα της διαδικασίας ανακάλυψης φαρμάκων (όπως για παράδειγμα η πρόβλεψη ιδιοτήτων ενός μορίου) χρησιμοποιώντας την πλατφόρμα torchdrug. Το TorchDrug είναι βασισμένο στη βιβλιοθήκη μηχανικής μάθησης PyTorch για την γλώσσα Python και υλοποιεί τεχνικές από τη θεωρία της μηχανικής μάθησης γράφων.

1. Εισαγωγή

Η διαδικασία της σχεδίασης, παραγωγής και δοκιμής ενός φαρμάκου είναι εξαιρετικά κοστοβόρα και χρονοβόρα. Πράγματι, η μέση διάρκεια από τη στιγμή της έναρξης της διαδικασίας σχεδίασης ενός φαρμάκου μέχρι την τελική έγκριση και διάθεσή του στην αγορά παίρνει κατά μέσο όρο 14 χρόνια (Myers & Baker, 2001) ενώ το συνολικό κόστος εκτιμάται ότι βρίσκεται μεταξύ 0.8 με 1 δισεκατομμύριο δολάρια (Moses, Dorsey, Matheson, & Tier, 2005).

Επιπλέον, τα τελευταία χρόνια η ποικιλία, ταχύτητα και ο όγκος των δεδομένων που συλλέγονται στις γενομικές έρευνες έχουν αυξηθεί ραγδαία δημιουργώντας την έννοια “big data”. Ενδεικτικά, τεχνικές όπως η next generation sequencing (NGS) επιτρέπουν την αλληλούχηση του ανθρώπινου γονιδιώματος σε μία μόνο μέρα, σε αντίθεση με την προηγούμενη τεχνική Sanger που χρειάστηκε πάνω από μια δεκαετία (Behjati & Tarpey, 2013). Η κύρια διαφορά τους είναι ότι η Sanger sequencing αλληλουχεί μόνο μερικά κομμάτια γενετικού υλικού τη φορά ενώ η NGS προσφέρει parallelization, τη δυνατότητα δηλαδή αλληλούχησης εκατομμυρίων κομματιών γενετικού υλικού τη φορά.

Στον τομέα της υπολογιστής χημείας έχουν σημειωθεί επίσης ραγδαίες εξελίξεις που έχουν ως αποτέλεσμα και εκεί τη συσσώρευση τεράστιου όγκου δεδομένων. Στην παρακάτω εικόνα παρατίθενται τα μεγέθη μερικών δημοφιλών χημικών βάσεων δεδομένων (Tetko, 2016).



Εικόνα 1: μέγεθος δημοφιλών χημικών βάσεων δεδομένων

Για να γίνει ακόμα πιο κατανοητό το μέγεθος το όγκου των βιοχημικών δεδομένων που έχει συσσωρευθεί στις μέρες μας, παραθέτουμε επίσης μερικά παραδείγματα big data προερχόμενα από τον τομέα της βιοχημείας (MilmanII & Zhurkovich, 2020).

Πίνακας 1: μέγεθος διαφόρων βιοχημικών δεδομένων

Data array	Data volume
Results of sequencing of human DNA	750 Mb
Genome data of the United States	222 Pb
Mass spectrometry databases in proteomics	>1 Tb
Accumulated data in metabolomics	around 4 Tb
Chemical database, information about tens of millions of compounds	≥2 Tb
A chromatogram with tandem mass spectra	10–100 Mb
A mass spectrometry image of a tissue cut, imaging (biosamples)	up to hundreds Gb
Library of one million tandem mass spectra	hundreds Gb

Έτσι λοιπόν, οι ανάγκες για μείωση του κόστους και χρόνου για την ανάπτυξη νέων φαρμάκων καθώς και για την αξιοποίηση του νέου αυξημένου όγκου δεδομένων καθιστά τη χρήση υπολογιστικών μεθόδων για τη σχεδιάσή τους αναγκαία δημιουργώντας τον κλάδο του computer-aided drug design (CADD).

2. Σκοπός και εργαλεία που θα χρησιμοποιηθούν

2.1 Σκοπός

Στην παρούσα εργασία θα χρησιμοποιήσουμε την πλατφόρμα TorchDrug για την εκτέλεση διάφορων εργασιών που συσχετίζονται με την διαδικασία ανακάλυψης νέων φαρμάκων. Συγκεκριμένα, θα χρησιμοποιηθούν τεχνικές μηχανικής μάθησης και έννοιες από τη θεωρία γράφων για τη μελέτη μορίων, την πρόβλεψη των ιδιοτήτων τους καθώς και την παραγωγή τους.

2.2 Περιβάλλον προγραμματισμού

Η γλώσσα προγραμματισμού που θα χρησιμοποιηθεί είναι η Python σε έκδοση 3 και διάφορα libraries για αυτήν όπως το PyTorch και το Matplotlib. Ο κώδικας θα είναι στο Google Colab το οποίο είναι ένα Jupyter notebook που βρίσκεται στο cloud. Το Google Colab παρέχει δωρεάν υπολογιστικές πηγές συμπεριλαμβανομένων GPUs και TPUs (tensor processing units), κάτι που θα μας φανεί πολύ χρήσιμο διότι οι εργασίες που θα εκτελεστούν είναι πολύ βαριές υπολογιστικά λόγω του ότι εμπλέκονται νευρωνικά δίκτυα.

Το Google Colab έχει το πλεονέκτημα του ότι δε χρειάζεται καμία εγκατάσταση καθώς τρέχει εξ' ολοκλήρου στους servers της Google, έχει ήδη προεγκατεστημένες τις πιο δημοφιλείς βιβλιοθήκες και τα notebooks του μπορούν να διαμοιραστούν με οποιονδήποτε έχει ένα Google account δίχως κάποια άλλη προϋπόθεση. Έτσι, τα όποια αποτελέσματα εξάγονται από έναν ερευνητή είναι πλήρως αναπαραγωγίσιμα (reproducible results).

2.3 Η πλατφόρμα TorchDrug

Το κύριο εργαλείο που θα χρησιμοποιηθεί είναι το TorchDrug. Πρόκειται για μια open source βιβλιοθήκη για την Python που είναι χτισμένη πάνω στη βιβλιοθήκη μηχανικής μάθησης PyTorch. Δημιουργήθηκε από την ερευνητική ομάδα MilaGraph με ηγέτη τον καθηγητή Jian Tang στο Mila-Quebec AI Institute. Ο σκοπός του TorchDrug είναι η ανακάλυψη φαρμάκων αξιοποιώντας τη θεωρία των νευρωνικών δικτύων γράφων (graph neural networks) με ελάχιστες απαιτήσεις γνώσεων πάνω στο πεδίο ανακάλυψης φαρμάκων (minimal domain knowledge required) (Zhu, et al., 2021).

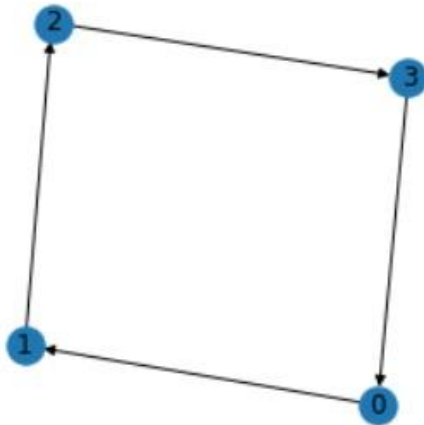
2.4 Βασικά στοιχεία του TorchDrug

Η βασική δομή δεδομένων του TorchDrug είναι οι γράφοι. Χρησιμοποιώντας την υποδομή της βιβλιοθήκης μηχανικής μάθησης PyTorch πάνω στην οποία είναι βασισμένο, το TorchDrug αναπαριστά τους γράφους με τη μορφή τένσορα επιτρέποντας έτσι τη χρησιμοποίηση GPU και TPU για δραματική αύξηση στην ταχύτητα υπολογισμών όπως και την αυτόματη παραγωγή.

Αυτά τα δύο στοιχεία είναι κρίσιμης σημασίας για τα νευρωνικά δίκτυα διότι η βασική μορφή αναπαράστασής τους είναι οι τένσορες των οποίων οι υπολογισμοί επιταχύνονται δραματικά χρησιμοποιώντας GPU ή TPU (Caulfield, 2009) και η παραγωγή είναι απαραίτητη στη διαδικασία εκπαίδευσης νευρωνικών δικτύων.

Για την αναπαράσταση ενός γράφου, αρκεί να εισάγουμε μια λίστα με τις ακμές που επιθυμούμε.

```
edge_list = [[0, 1], [1, 2], [2, 3], [3, 0]]  
graph = data.Graph(edge_list, num_node=4)  
graph.visualize()
```



Εικόνα 2: ένας απλός γράφος

Επίσης είναι δυνατή η αναπαράσταση ενός μορίου με την ακολουθία SMILES (χημικός συμβολισμός που επιτρέπει την αναπαράσταση ενός μορίου με χαρακτήρες ASCII (U.S. Environmental Protection Agency, 2016)) που αντιστοιχεί σε αυτό. Σημειώνουμε ότι το αποτέλεσμα θα είναι και αυτό στην ουσία ένας γράφος.

Θα δείξουμε τώρα πως μπορεί να εισαχθεί μια χημική ουσία όπως για παράδειγμα η παρακεταμόλη στο TorchDrug. Αρχικά πλοηγούμαστε στο <https://pubchem.ncbi.nlm.nih.gov/>, εισάγουμε “paracetamol” στο πεδίο αναζήτησης και εντοπίζουμε την αναπαράσταση SMILES.

SEARCH FOR

paracetamol

Treating this as a text search.

COMPOUND BEST MATCH



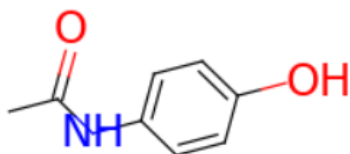
Acetaminophen; Paracetamol; 4-Acetamidophenol; 103-90-2
Compound CID: 1983
MF: $C_8H_9NO_2$ MW: 151.16g/mol
IUPAC Name: N-(4-hydroxyphenyl)acetamide
Isomeric SMILES: CC(=O)NC1=CC=C(C=C1)O
InChIKey: RZVAJINKPMORJF-UHFFFAOYSA-N
InChI: InChI=1S/C8H9NO2/c1-6(10)9-7-2-4-8(11)5-3-7/h2-5,11H,1H3,(H,9,10)
Create Date: 2004-09-16

Εικόνα 3: εύρεση της αναπαράστασης SMILES της παρακεταμόλης στη βάση δεδομένων PubChem

Εφόσον έχουμε την ακολουθία SMILES μιας χημικής ουσίας, η εισαγωγή της στο TorchDrug είναι απλή.

```
mol = data.Molecule.from_smiles("CC(=O)NC1=CC=C(C=C1)O")  
mol.visualize("Το μόριο της παρακεταμόλης",figure_size=(5, 5))
```

Το μόριο της παρακεταμόλης



Εικόνα 4: εισαγωγή του μορίου της παρακεταμόλης στο TorchDrug και οπτικοποίησή του

3. Πρόβλεψη ιδιοτήτων

3.1 Διαδικασία

Βασικό βήμα στη διαδικασία ανακάλυψης φαρμάκων είναι η δυνατότητα πρόβλεψης συγκεκριμένων ιδιοτήτων ενός μορίου. Για παράδειγμα δοθείσας μιας ουσίας θα θέλαμε να μπορούμε να προβλέψουμε αν είναι τοξική ή όχι.

3.2 Dataset

Η ιδιότητα που θα επιχειρήσουμε να προβλέψουμε στην παρούσα εργασία είναι το αν ένα μόριο διαπερνά ή όχι τον αιματοεγκεφαλικό φραγμό (blood-brain barrier ή αλλιώς BBB) και για αυτό θα χρησιμοποιήσουμε το dataset BBBP που

συμπεριλαμβάνεται στο TorchDrug. Το dataset BBBP περιέχει 2051 χημικές ουσίες με ετικέτα 0 ή 1 ανάλογα αν διαπερνούν τον BBB ή όχι. Είναι ένα csv αρχείο με 4 στήλες, αύξοντα αριθμό, όνομα ουσίας, 0 ή 1 ανάλογα αν διαπερνά τον BBBP ή όχι και την αναπαράσταση SMILES της. Παρακάτω βλέπουμε πώς διαχειρίζεται το TorchDrug τα datasets, στην ουσία τα αντιμετωπίζει σα μια list με dictionaries.

```
Ο τύπος του dataset BBBP είναι:  
<class 'torchdrug.datasets.bbbp.BBBP'>.  
Το πρώτο στοιχείο του BBBP είναι:  
{'graph': Molecule(num_node=20, num_edge=40, num_relation=4), 'p_np': 1}.
```

Εικόνα 5: η μορφή των datasets στο TorchDrug

3.3 Προετοιμασία του dataset

Αφού φορτώσουμε το dataset, θα χρησιμοποιήσουμε τη βιβλιοθήκη PyTorch (στην οποία είναι βασισμένο το TorchDrug) για να το διαχωρίσουμε σε training/validation/test σύνολα σε αναλογίες 80%/10%/10% αντίστοιχα.

3.4 Επιλογή μοντέλου

Το μοντέλο που θα χρησιμοποιήσουμε είναι το Graph Isomorphism Network (GIN) που προτάθηκε στο άρθρο “How Powerful are Graph Neural Networks?” (Xu, Hu, Leskovec, & Jegelka, 2019).

Το νευρωνικό μας δίκτυο θα έχει 4 κρυφά επίπεδα (hidden layers) με 256 νευρώνες (neurons) το καθένα. Το κριτήριο επιλογής είναι η binary cross entropy (“bce”) και τα μέτρα αξιολόγησης είναι τα area under the precision-recall curve και area under an ROC (receiver operating characteristic) curve (“auprc” και “auroc”).

3.5 Εκπαίδευση και αξιολόγηση του μοντέλου

3.6 Παραδείγματα προβλέψεων

Σε αυτό το σημείο θα εφαρμόσουμε στην πράξη το μοντέλο που δημιουργήσαμε σε μερικές γνωστές φαρμακευτικές ουσίες. Η αναπαράσταση SMILES του κάθε φαρμάκου ανακτήθηκε από το αρχείο BBBP.csv που βρίσκεται στην τοποθεσία <http://deepchem.io.s3-website-us-west-1.amazonaws.com/datasets/BBBP.csv> και έπειτα εισήχθη στο μοντέλο. Η activation function που χρησιμοποιήθηκε (μέσω του PyTorch) ήταν η sigmoid αλλά υπάρχει η δυνατότητα επιλογής και άλλων δημοφιλών συναρτήσεων όπως η ReLu. Στον παρακάτω πίνακα παραθέτουμε τις προβλέψεις του μοντέλου μας σε αντιπαράθεση με την πραγματική τιμή, 0 αν η ουσία δε διαπερνά τον BBB και 1 αλλιώς.

Πίνακας 2: εφαρμογή του μοντέλου μας σε ενδεικτικές ουσίες

Όνομα ουσίας	Χρήση	Διαπερνά τον BBB?	Πρόβλεψη του μοντέλου	Αναπαράσταση SMILES
Amitriptyline	Κατάθλιψη	1	0.9779	<chem>CN(C)CCC=C1c2ccccc2CCc3ccccc13</chem>
Ciprofloxacin	Αντιβιοτικό	0	0.3304	<chem>OC(=O)C1=CN(C2CC2)c3cc(N4CCNCC4)c(F)cc3C1=O</chem>
Morphine	Παυσίτονο	1	0.9885	<chem>CN1CC[C@]23[C@H]4Oc5c(O)ccc(C[C@@H]1[C@@H]2C=CC4O)c35</chem>
Ropinirole	Νόσος του Parkinson	1	0.8806	<chem>C2=C(C1=C(NC(C1)=O)C=C2)CCN(CCC)CCC</chem>
Gabapentin	Επιληψία	1	0.9971	<chem>NCC1(CCCCC1)CC(O)=O</chem>
Prednisone	Ανοσοκατασταλτικό	0	0.0571	<chem>C[C@]12CC(=O)[C@H]3[C@@H](CCC4=CC(=O)C=C[C@]34C)[C@@H]1CC[C@]2(O)C(=O)CO</chem>

4. Δημιουργία μορίων

4.1 Περί της διαδικασίας

Ο σκοπός του κεφαλαίου αυτού είναι η δημιουργία ενός μοντέλου που θα παράγει druglike μόρια. Επειδή το σύνολο των πιθανών μορίων είναι αστρονομικό, για να μη ξεκινήσουμε τη δημιουργία του μοντέλου μας χωρίς κάποιο καλό σημείο αναφοράς θα εκτελέσουμε προεκπαίδευση.

Δηλαδή θα δημιουργήσουμε ένα αρχικό μοντέλο που θα παράγει μόρια που θα “μοιάζουν” με αυτά που υπάρχουν σε ένα dataset που περιέχει 250 χιλιάδες μόρια που είναι ήδη υποψήφια για φάρμακα. Έπειτα, θα πάρουμε αυτό το αρχικό μοντέλο και θα το εκπαιδεύσουμε περαιτέρω έτσι ώστε να έχει ιδιότητες φαρμάκου (drug-like). Το αποτέλεσμα αυτής της εκπαίδευσης του αρχικού μας μοντέλου θα είναι ο τελικός μας αλγόριθμος παραγωγής πιθανών φαρμάκων.

4.2 Το dataset της προεκπαίδευσης

Η δημιουργία γράφων μορίων είναι μια βασική διεργασία στην ανακάλυψη φαρμάκων. Είναι δύσκολη όμως λόγω του ότι τα πιθανά μόρια που μπορεί να έχουν φαρμακευτική χρήση (druglike chemical space) είναι πάρα πολλά, περίπου 10^{60} (Lipinski., Lombardo, Dominy., & Paul J. Feeney, 1997) και πρέπει να ικανοποιούν διάφορες προϋποθέσεις που θα δούμε αργότερα.

Για να διευκολύνουμε τη δημιουργία νέων μορίων θα προεκπαιδεύσουμε το μοντέλο μας στο dataset [ZINC250k](#) το οποίο είναι υποσύνολο του dataset [ZINC](#). Το [ZINC250k](#) περιέχει 250 χιλιάδες έτοιμα μόρια που είναι γνωστός ο τρόπος παρασκευής τους μαζί με την αναπαράσταση SMILES τους καθώς και 3 χημικές ιδιότητες (logP, QED και SAS). Σημειώνουμε ότι η διαδικασία λήψης του dataset και η κατασκευή των γράφων των μορίων από την αναπαράσταση SMILES τους θα χρειαστεί πάνω από 7 λεπτά.

```
18:26:07 Downloading https://raw.githubusercontent.com/aspuru-guzik-group/chemical\_vae/master/models/zinc\_properties/250k\_rndm\_zinc\_drugs\_clean\_3.csv
to /root/molecule-datasets/250k_rndm_zinc_drugs_clean_3.csv

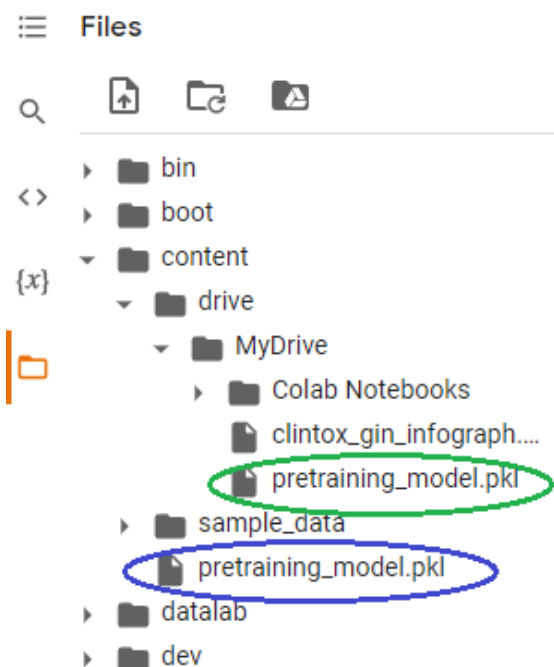
Loading /root/molecule-datasets/250k_rndm_zinc_drugs_clean_3.csv:
 50%|██████| 249456/498911 [00:04<00:04, 56225.76it/s]
Constructing molecules from SMILES:
100%|██████████| 249455/249455 [08:52<00:00, 468.20it/s]
```

Εικόνα 9: προετοιμασία του dataset

Θα χρησιμοποιήσουμε αυτά τα έτοιμα μόρια από το ZINC250k dataset για να δημιουργήσουμε ένα αρχικό μοντέλο που θα παράγει νέα μόρια που κατά κάποιον τρόπο θα τους “μοιάζουν”. Έπειτα αυτό το μοντέλο που δημιουργήσαμε σε βάση μπορούμε να το προσαρμόσουμε περαιτέρω εκπαιδεύοντάς το να ικανοποιεί συγκεκριμένες επιθυμητές ιδιότητες που θα του ορίσουμε.

4.3 Εκτέλεση της προεκπαίδευσης

Το μοντέλο που θα χρησιμοποιήσουμε για την προεκπαίδευση θα είναι το RGCN (Relational Graph Convolutional Networks) που προτάθηκε το 2017 (Schlichtkrull, et al., 2017). Το νευρωνικό μας δίκτυο θα έχει 4 hidden layers με 256 neurons το καθένα, το κριτήριο επιλογής είναι το “nll” δηλαδή το negative log-likelihood. Το task που θα αναθέσουμε στο μοντέλο μας θα είναι το GCPN (Graph Convolutional Policy Network) Generation που είναι μια μορφή δημιουργίας γράφου μέσω reinforcement learning.



Εικόνα 11: το file system του Google Colab

Στην παραπάνω φωτογραφία εικονίζεται το file system του Google Colab, όπως καταλαβαίνουμε από τα αρχεία πρόκειται για ένα λειτουργικό σύστημα τύπου Linux. Η τοποθεσία /content είναι ο τόπος που αποθηκεύονται τα αρχεία μας ενώ το Google Drive μας βρίσκεται στο /content/drive/MyDrive. Έτσι με μια απλή εντολή bash μπορούμε να μεταφέρουμε αρχεία που δημιουργήθηκαν στη συνεδρία μας στο Google Drive για χρήση αργότερα (χρησιμοποιώντας τον χαρακτήρα “!” στην αρχή κάθε γραμμής μπορούμε να έχουμε άμεση πρόσβαση στο shell) καθώς και ανάκτηση οποιουδήποτε αρχείου βρίσκεται στο Google Drive.

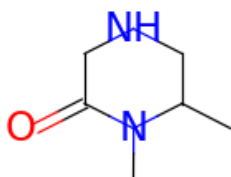
Χρησιμοποιώντας το μοντέλο προεκπαίδευσης δημιουργήσαμε 3 ενδεικτικά μόρια. Παραθέτουμε την αναπαράσταση SMILES τους καθώς και την οπτικοποίησή τους.

Molecule 0: CC1CNCC(=O)N1C

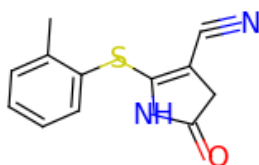
Molecule 1: CC1=CC=CC=C1SC1=C(C#N)CC(=O)N1

Molecule 2: CCNC(C)C1=NC(C)=CN=C1C1=CC(C)=CC=C1F

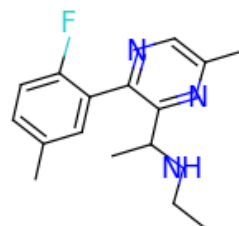
Molecule 0



Molecule 1



Molecule 2



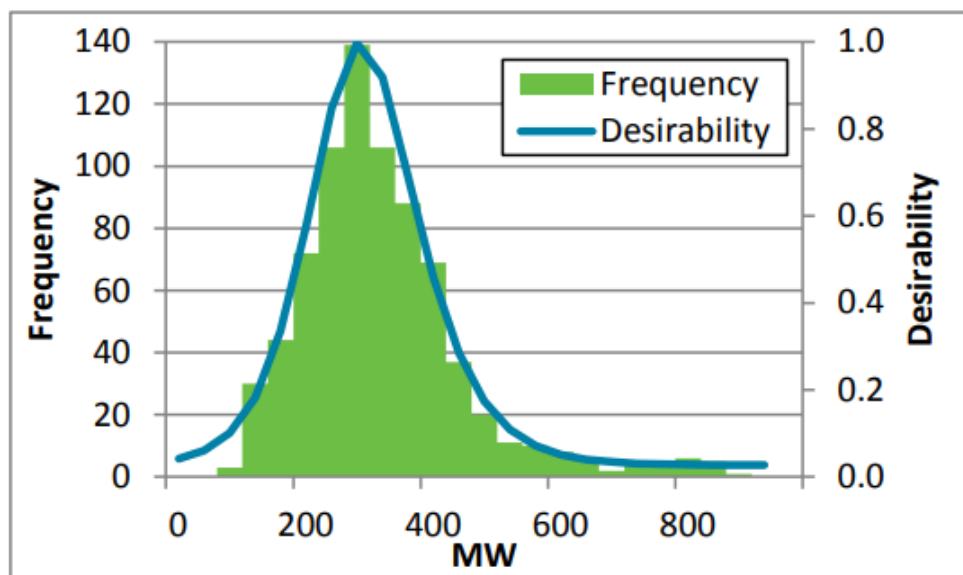
Εικόνα 11: μόρια που δημιουργήθηκαν από το μοντέλο προεκπαίδευσης

4.4 QED score ενός μορίου

Σε αυτό το σημείο θα κάνουμε μια παράκαμψη για να παρουσιάσουμε την έννοια του QED score ενός μορίου. Το 2012 μια ομάδα ερευνητών πρότειναν τη χρήση ενός μέτρου που ονόμασαν QED (Bickerton, Paolini, Besnard, Muresan, & Hopkins, 2012) το οποίο είναι ένας αριθμός μεταξύ του 0 και 1 που αντιστοιχίζεται σε κάθε χημική ουσία. Όσο πιο κοντά στο 1 είναι αυτό το μέτρο, τόσο πιο πολύ η ουσία έχει τα χαρακτηριστικά φαρμάκου. Το QED είναι τα αρχικά του “Quantitative Estimate of Druglikeness” που σε ελεύθερη μετάφραση σημαίνει “ποσοτική εκτίμηση του κατά πόσο ένα μόριο έχει χαρακτηριστικά φαρμάκου”.

Η διαδικασία με την οποία οι ερευνητές κατέληξαν σε αυτό το μέτρο είναι η εξής. Μελέτησαν 8 μοριακές ιδιότητες 771 στοματικά χορηγούμενων φαρμάκων που κυκλοφορούν στην αγορά και δημιούργησαν 8 συναρτήσεις που προσεγγίζουν την κατανομή των τιμών αυτών των ιδιοτήτων (desirability functions).

- Μοριακό βάρος
- Λιποφιλικότητα (logP)
- Αριθμός δοτών δεσμών υδρογόνου (HBD)
- Αριθμός δεκτών δεσμών υδρογόνου (HBA)
- Πολική έκταση επιφάνειας (PSA)
- Αριθμός δεσμών που μπορούν να περιστραφούν (ROTB)
- Αριθμός αρωματικών δακτυλίων (AROM)
- Αριθμός ενδείξεων για πιθανές μη επιθυμητές δομές (ALERTS)



Εικόνα 12: η desirability function του μοριακού βάρους (MW)

Στην παραπάνω εικόνα απεικονίζεται το γράφημα της desirability function για το μοριακό βάρος (Optibrium, n.d.). Όπως παρατηρούμε, όσο πιο πολλά φάρμακα είχαν μια συγκεκριμένη τιμή μοριακού βάρους, τόσο υψηλότερη είναι και η τιμή της

αντίστοιχης desirability function. Δηλαδή η desirability function είναι “fitted” στην κατανομή της πραγματικής τιμής.

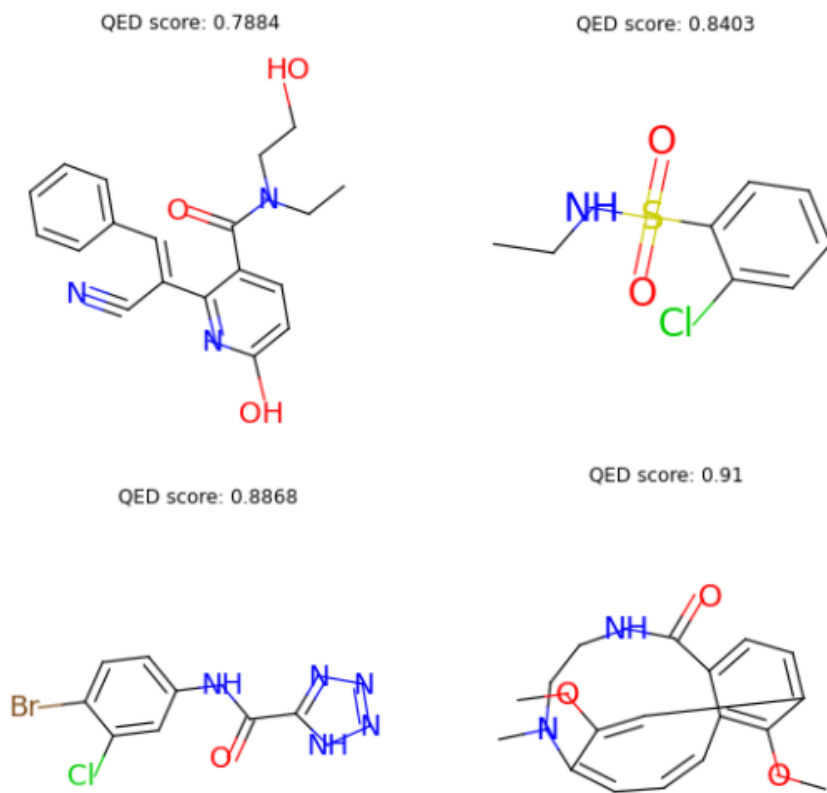
Έπειτα, οι τιμές των 8 desirability functions συνθέτονται στον παρακάτω μαθηματικό τύπο και παράγουν μια τιμή μεταξύ του 0 και 1, όπου d_i είναι η τιμή της i -οστής desirability function .

$$QED = \exp \left(\frac{1}{n} \sum_{i=1}^n \ln d_i \right)$$

4.5 Δημιουργία του τελικού μοντέλου

Για το κύριο μέρος της εκπαίδευσης, θα χρησιμοποιήσουμε σαν είσοδο το αρχικό μοντέλο της προεκπαίδευσης και θα το βελτιστοποιήσουμε έτσι ώστε τα μόρια που παράγει να έχουν όσο το δυνατόν υψηλότερο QED score. Το κριτήριο επιλογής ήταν το ίδιο που χρησιμοποιήσαμε και για την προεκπαίδευση, δηλαδή το “nll”.

Όπως βλέπουμε στην παρακάτω εικόνα, για την εκπαίδευση του μοντέλου για μόνο μια επανάληψη (epoch) χρειάστηκε αρκετός χρόνος, 1h 16m 48s. Στο συνοδευτικό Google Colab notebook παραθέτουμε το μοντέλο που δημιουργήσαμε για λήψη προς αποφυγή αυτής της χρονοβόρας διαδικασίας.



Εικόνα 14: μόρια που παρήγαγε το τελικό μας μοντέλο με υψηλό QED score

5. Βιοιατρικές αναπαραστάσεις γνώσης

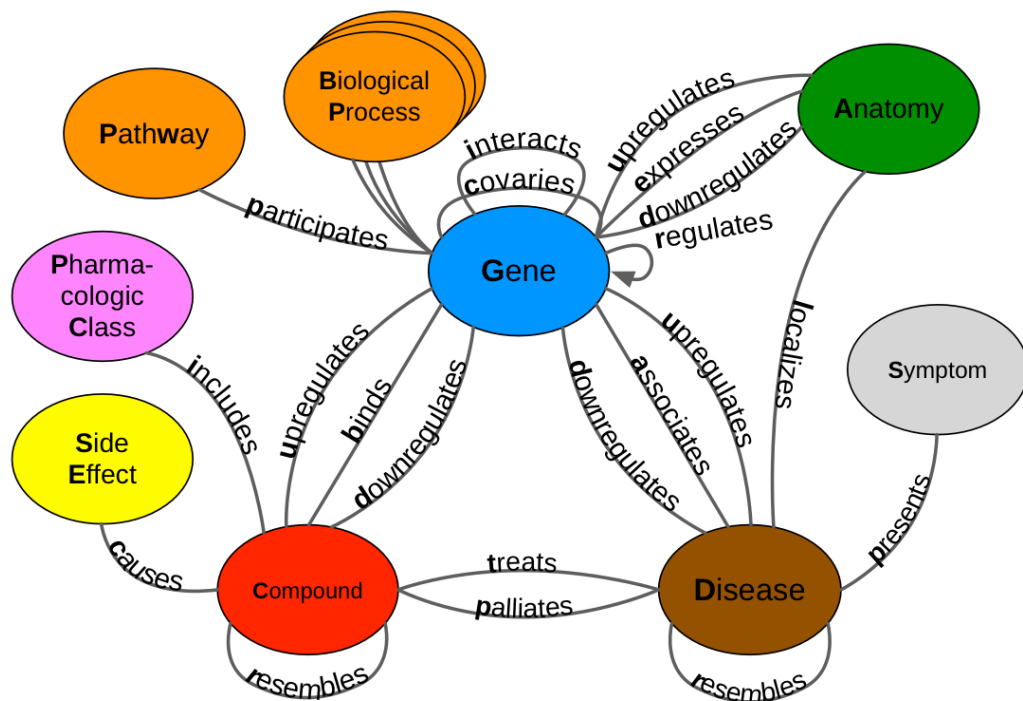
5.1 Σημασιολογικά δίκτυα

Όπως αναφέραμε στο δεύτερο κεφάλαιο, η βασική μορφή δεδομένων που χρησιμοποιεί το TorchDrug είναι οι γράφοι. Εκτός από την αναπαράσταση μορίων,

μια από τις χρήσεις των γράφων μπορεί να είναι η αναπαράσταση οντοτήτων του πραγματικού κόσμου καθώς και των σχέσεων μεταξύ τους, αυτό ορίζεται ως γράφος γνώσης ή αλλιώς σημασιολογικό δίκτυο (IBM, n.d.).

Έτσι λοιπόν, εκτός από εργασίες που σχετίζονται με χημικές ουσίες, το TorchDrug μας παρέχει τη δυνατότητα χειρισμού σχεσιακών γράφων γνώσεων για εξαγωγή συμπερασμάτων σχετικά με τα δεδομένα τους. Ένα από τα datasets που παρέχονται και θα μελετηθεί στο παρόν κεφάλαιο είναι το [Hetionet](#).

5.2 Η βάση δεδομένων Hetionet



Εικόνα 15: η μορφή του Hetionet. Τα οβάλ σχήματα είναι οι οντότητες και οι ακμές είναι οι σχέσεις μεταξύ τους.

Σύμφωνα με τη βιβλιογραφία, για παράδειγμα (Grey & Thompson, 2010), η διαδικασία ανακάλυψης φαρμάκων (drug discovery pipeline) αρχίζει με την αναγνώριση του στόχου που θέλουμε το φάρμακό μας να δράσει. Το Hetionet μπορεί να μας βοηθήσει σε αυτό καθώς και σε άλλα βήματα της pipeline.

Αρχικά θα περιγράψουμε τη δομή του dataset. Το Hetionet είναι ένας γράφος με 47031 κόμβους (οντότητες) 11 τύπων και 2250197 ακμές (σχέσεις) 24 τύπων. Στον παρακάτω πίνακα παραθέτουμε τους παραθέτουμε αναλυτικά.

Πίνακας 3: οι τύποι συσχετίσεων του Hetionet.

Edge Type	Abbreviation	Edge Type	Abbreviation
[Anatomy downregulates Gene]	AdG	[Disease localizes Anatomy]	DIA
[Anatomy expresses Gene]	AeG	[Disease presents Symptom]	DpS
[Anatomy upregulates Gene]	AuG	[Disease resembles Disease]	DrD
[Compound binds Gene]	CbG	[Disease upregulates Gene]	DuG
[Compound causes Side Effect]	CcSE	[Gene covaries Gene]	GcG
[Compound downregulates Gene]	CdG	[Gene interacts Gene]	GiG
[Compound palliates Disease]	CpD	[Gene participates Biological Process]	GpBP
[Compound resembles Compound]	CrC	[Gene participates Cellular Component]	GpCC
[Compound treats Disease]	CtD	[Gene participates Molecular Function]	GpMF

[Compound upregulates Gene]	CuG	[Gene participates Pathway]	GpPW
[Disease associates Gene]	DaG	[Gene regulates Gene]	Gr>G
[Disease downregulates Gene]	DdG	[Pharmacologic Class includes Compound]	PCiC

Πίνακας 4: οι τύποι οντοτήτων του Hetionet

Node type	Abbreviation	Node type	Abbreviation
Anatomy	A	Molecular Function	MF
Biological Process	BP	Pathway	PW
Cellular Component	CC	Pharmacologic Class	PC
Compound	CC	Side Effect	SE
Disease	D	Symptom	SE
Gene	G		

Σημειώνουμε ότι εκτός από το Gene regulates Gene, όλοι οι άλλοι τύποι ακμών είναι μη κατευθυνόμενοι. Για να καταλάβουμε καλύτερα τη μορφή του Hetionet θα δείξουμε το πρώτο του στοιχείο.

Gene::11344 Gr>G Gene::7485

Όπως βλέπουμε, τα στοιχεία του Hetionet είναι τριάδες. Το πρώτο του στοιχείο (τριάδα) μας λέει ότι το γονίδιο με αριθμό 11344 ρυθμίζει (regulates) το γονίδιο 7485. Η ερμηνεία έγινε βάσει των πινάκων που παραθέσαμε ανωτέρω. Επίσης αν θέλουμε να ταυτοποιήσουμε τα γονίδια μπορούμε να τα αναζητήσουμε σε μια βάση δεδομένων όπως π.χ. η GenBank.

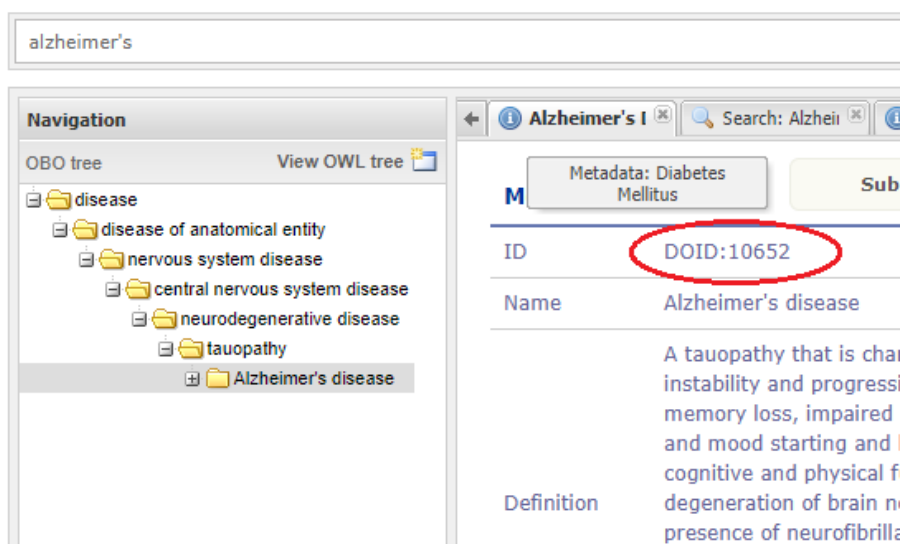
5.3 Παράδειγμα χρήσης

Επειδή το Hetionet περιλαμβάνει μια πληθώρα οντοτήτων και σχέσεων, μας επιτρέπει να εκτελέσουμε ένα μεγάλο αριθμό εργασιών που εμπίπτουν στη διαδικασία της ανακάλυψης φαρμάκων. Στην παρούσα εργασία θα χρησιμοποιήσουμε το Hetionet μέσω του TorchDrug για να ανακαλύψουμε ποια γονίδια συνδέονται με τη νόσο του Alzheimer (AD). Αυτό θα βοηθούσε στη σχεδίαση φαρμάκων για την καταπολέμησή της διότι όπως αναφέραμε στην αρχή του κεφαλαίου, η αναγνώριση του στόχου είναι το πρώτο στάδιο στην drug discovery pipeline.

Η διαδικασία που θα ακολουθήσουμε είναι η εξής. Θα τρέξουμε ένα βρόχο πάνω στις 2250195 συσχετίσεις (τριάδες) του Hetionet με λέξη κλειδί την εξής.

```
Disease::DOID:10652 DaG
```

Το 10652 είναι η disease ID της AD η οποία ανακτήθηκε από το disease-ontology.org και το “DaG” σημαίνει “Disease associates Gene” (νόσος σχετίζεται με γονίδιο) που ανακτήθηκε από τον πίνακα 3.



Εικόνα 16: εύρεση της disease ID της AD

Στην παρακάτω συντετμημένη εικόνα παραθέτουμε τα αποτελέσματα της αναζήτησης ποιων γονιδίων εμπλέκονται με την AD. Βρέθηκαν 196 εμπλεκόμενα γονίδια και η διαδικασία αναζήτησης ολοκληρώθηκε σε 28s παρόλο που τρέξαμε το βρόχο για πάνω από 2 εκατομμύρια επαναλήψεις. Το TorchDrug λοιπόν μας επιτρέπει να εκτελούμε εργασίες που με παραδοσιακές μεθόδους θα συναντούσαμε σημαντικά προβλήματα.

Για παράδειγμα το αρχείο της βάσης δεδομένων είναι σε μορφή .txt, συμβατικά προγράμματα που χρησιμοποιούνται για την επεξεργασία τέτοιας μορφής όπως το Notepad δεν έχουν τη δυνατότητα χειρισμού αρχείων ούτε καν της τάξης μεγέθους των χιλιάδων γραμμών.

Gene::3479	Gene::3375	Gene::596	Gene::7133
Gene::3162	Gene::4524	Gene::4023	Gene::207
Gene::3416	Gene::4535	Gene::6647	Gene::1234
Gene::3569	Gene::4846	Gene::3480	Gene::836
Gene::1636	Gene::348	Gene::5468	Gene::4968
Gene::5743	Gene::4843	Gene::10891	Gene::3952
Gene::3481	Gene::177	Gene::7124	Gene::7422
Gene::5444	Gene::6347	Gene::169792	Gene::6648
Gene::5176	Gene::7040	Gene::801	Gene::3630

Number of common genes: 36

Εικόνα 18: τα κοινά γονίδια που συνδέονται με την AD και τον T2D

Όπως βλέπουμε, από τα 196 γονίδια που συνδέονται με την AD και τα 275 με τον T2D, 36 είναι κοινά. Η γνώση αυτή μπορεί να βοηθήσει στη σχεδίαση φαρμάκων για αυτές τις ασθένειες με διάφορους τρόπους. Για παράδειγμα, ερευνώντας αυτά τα γονίδια μπορούν να ανακαλυφθούν μηχανισμοί και παράγοντες που τις προκαλούν που δεν έχουν βρεθεί μέχρι τώρα. Έτσι μπορούν να εντοπιστούν νέοι και πιθανώς αποτελεσματικότεροι targets για φαρμακευτική αντιμετώπιση.

Η σύνδεση αυτή μεταξύ των 2 ασθενειών είναι ιδιαίτερα σημαντική για την AD δοθέντος ότι μέχρι σήμερα είναι ανίατη. Γνωρίζοντας ότι συνδέεται με τον T2D μας δίνει προοπτικές για έρευνα πάνω στο drug repurposing, δηλαδή την έρευνα για το αν φάρμακα αρχικά φτιαγμένα για τη θεραπεία του T2D μπορούν να αντιμετωπίσουν την AD. Για παράδειγμα το 2021 ανακαλύφθηκε ότι το φάρμακο liraglutide, αρχικά σχεδιασμένο για τη θεραπεία του T2D, μπορεί να μειώσει την παθολογία της AD, μεταξύ άλλων μειώνοντας την παραγωγή βλαβερών αμυλοειδών πλακών στον εγκέφαλο (Vargas-Soria, Carranza-Naval, & Marco, 2021).

6. Συμπεράσματα

Στην παρούσα εργασία είδαμε πως η νεοσύστατη πλατφόρμα TorchDrug μπορεί να εκτελέσει διάφορες εργασίες που εμπίπτουν στη διεξαγωγή ανακάλυψης φαρμάκων. Παρόλο που είναι λιγότερο από 6 μηνών προσφέρει λειτουργίες που ανταγωνίζονται ακόμα και το DeepChem, τη μεγαλύτερη αντίστοιχη πλατφόρμα ανακάλυψης φαρμάκων που βρίσκεται σε λειτουργία πάνω από 4 χρόνια.

torchdrug		deepchem
Project		
3	@ Mentions	2
753	☆ Stars	3,424
26.0%	↗ Growth	2.5%
7.6	⚡ Activity	9.8
9 days ago	Latest Commit	6 days ago
Python	<> Language	Python
Apache License 2.0	License	MIT License
The number of mentions indicates the total number of mentions that we've tracked plus the number of user suggested alternatives. Stars - the number of stars that a project has on GitHub. Growth - month over month growth in stars. Activity is a relative number indicating how actively a project is being developed. Recent commits have higher weight than older ones. For example, an activity of 9.0 indicates that a project is amongst the top 10% of the most actively developed projects that we are tracking.		

Εικόνα 19: σύγκριση του TorchDrug με το DeepChem που ανακτήθηκε από το [LibHunt](#)

Συγκεκριμένα δημιουργήσαμε έναν ταξινομητή που παίρνει σαν είσοδο μια χημική ουσία και δίνει ως έξοδο το αν διαπερνά τον αιματοεγκεφαλικό φραγμό ή όχι. Αφού τον δοκιμάσαμε σε μερικά φάρμακα που κυκλοφορούν στην αγορά διαπιστώσαμε ότι είχε ικανοποιητική αποτελεσματικότητα με μέτρα AUPRC και AUROC μεγαλύτερα από 0.8.

Έπειτα δείξαμε πως μπορεί το TorchDrug να επιτελέσει την πιο κεντρική διεργασία της ανακάλυψης φαρμάκων, την παραγωγή μορίων. Με τη βοήθειά του χτίσαμε ένα μοντέλο που παράγει μόρια που είναι υποψήφια για φαρμακευτική χρήση. Αυτό το πετύχαμε θέτοντας ως κριτήριο τα μόρια να έχουν υψηλό QED score, ένα μέτρο που υποδεικνύει πόσο druglike είναι μια ουσία. Το μοντέλο ήταν σε θέση να παράγει μοντέλα με QED score>0.9, κάτι που υποδεικνύει υψηλή πιθανότητα για druglikeness.

Τέλος είδαμε πως το TorchDrug μπορεί εκτός από διεργασίες που σχετίζονται με μόρια και χημικές ουσίες να χειριστεί και σημασιολογικά δίκτυα, επιτρέποντάς μας να εξάγουμε γνώση από γνωσιακούς γράφους μεγάλου μεγέθους που με παραδοσιακές τεχνικές θα ήταν δύσκολη η πρόσβασή τους. Χειριζόμενοι τη βάση δεδομένων Hetionet παρατηρήσαμε ότι η νόσος του Alzheimer και ο διαβήτης τύπου 2 έχουν 36 κοινά εμπλεκόμενα γονίδια, πληροφορία που μπορεί να βοηθήσει στην αντιμετώπισή ειδικότερα της AD που μέχρι σήμερα είναι ανίατη.

Αξιοσημείωτο είναι ότι για την ολοκλήρωση των παραπάνω διεργασιών δε χρειάστηκαν εξειδικευμένες γνώσεις οργανικής χημείας, βιολογίας ή φαρμακολογίας πέραν κάποιων βασικών εννοιών. Η χρήση του TorchDrug κυρίως απαιτεί εξοικείωση με το παραδοσιακό machine learning pipeline χωρίς ιδιαίτερο domain knowledge των τομέων που εμπλέκονται στην ανακάλυψη φαρμάκων.

Επίσης ένα μεγάλο προτέρημα του TorchDrug είναι ότι μπορεί να χρησιμοποιηθεί μέσω της cloud computing πλατφόρμας Google Colaboratory. Οι υπολογισμοί του γίνονται μέσω νευρωνικών δικτύων που απαιτούν τη χρήση GPUs αν θέλουμε να πραγματοποιηθούν σε εύλογο χρονικό διάστημα. Αυτές όμως κοστίζουν εκατοντάδες έως δεκάδες χιλιάδων ευρώ, έτσι αν κάποιος δεν έχει πρόσβαση σε τέτοιου είδους hardware μπορεί να χρησιμοποιήσει τη δωρεάν υπολογιστή ισχύ που παρέχεται από τη Google.

Συνοψίζοντας, τονίζουμε ότι η διαδικασία ανακάλυψης φαρμάκων είναι ένας τομέας που έχει πολύ υψηλά εμπόδια εισόδου, κυρίως επειδή απαιτεί διεπιστημονική γνώση και πρόσβαση σε ακριβό εξοπλισμό και όπως είδαμε, το TorchDrug μπορεί να βοηθήσει τον μέσο επιστήμονα δεδομένων να υπερκεράσει αυτά τα εμπόδια. Έτσι λοιπόν δανειζόμενοι το motto του μεγαλύτερου αντιπάλου του, δηλαδή του DeepChem, μπορούμε να πούμε ότι όντως το TorchDrug “δημοκρατικοποιεί την ανακάλυψη φαρμάκων”.

Παράρτημα

Google Colab notebooks

[Google Colab notebook για το κεφάλαιο «Πρόβλεψη ιδιοτήτων»](#)

[Google Colab notebook για το κεφάλαιο «Δημιουργία μορίων»](#)

[Google Colab notebook για το κεφάλαιο «Βιοιατρικές αναπαραστάσεις γνώσης»](#)

Κώδικας του κεφαλαίου “Πρόβλεψη ιδιοτήτων”

Εγκατάσταση

Ρύθμιση του περιβάλλοντος εργασίας μας και εντολές εγκατάστασης των απαιτούμενων βιβλιοθηκών.

```
import os
import torch
os.environ["TORCH_VERSION"] = torch.__version__
```

```
!pip install torch-scatter -f https://pytorch-geometric.com/whl/torch- $\text{TORCH\_VERSION}$ .html
!pip install torchdrug
```

Dataset

Εδώ θα κάνουμε import τη βιβλιοθήκη του torchdrug. Το dataset που θα χρησιμοποιήσουμε είναι το BBBP το οποίο μπορεί να ληφθεί [εδώ](#), τα στοιχεία του έχουν τιμή 1 αν ένα μόριο διαπερνά τον αιματοεγκεφαλικό φραγμό (BBB) και 0 αλλιώς. Θα χρησιμοποιήσουμε την εντολή *random_split* του PyTorch για να διαχωρίσουμε το dataset σε training, validation και testing σύνολα σε αναλογία 80%/10%/10% αντίστοιχα.

```
import torch
from torchdrug import data, datasets

dataset = datasets.BBBP("~/molecule-datasets/")
lengths = [int(0.8 * len(dataset)), int(0.1 * len(dataset)
)]
lengths += [len(dataset) - sum(lengths)]
train_set, valid_set, test_set = torch.utils.data.random_
split(dataset, lengths)
```

Διερεύνηση του dataset. Θα δούμε το μήκος του και τι μορφή έχουν τα στοιχεία του. Όπως θα διαπιστώσουμε, το dataset είναι μια συλλογή από dictionaries.

```
print("Το BBBP dataset έχει "+str(len(dataset))+" αν  
τι κείμενα.")
print("Αυτό είναι το πρώτο στοιχείο του BB  
BP:")
print(dataset[0])
```

Τώρα θα οπτικοποιήσουμε μερικά τυχαία μόρια από το dataset BBBP που η ποσότητά τους θα επιλέγεται από εμάς. Πάνω από το γράφο του κάθε μορίου υπάρχει η ετικέτα "p_nr" (pass or not pass) που υποδηλώνει αν το μόριο διαπερνά τον BBB ή όχι με 1 ή 0 αντίστοιχα.

```
%matplotlib inline
import random

n = int(input("Πόσα τυχαία μόρια θέλετε να ο  
πτικοποιήσετε?\n"))
a = ""
b = ""
```

```

labels = []
graphs = []

#Create labels
for i in range(n):
    rand = random.randint(0, len(dataset))
    for k in dataset[rand]:
        if k != 'graph':
            a = k + ": " + str(dataset[rand][k])
            if b == "":
                b = a
            else:
                b = b + ", " + a
            a = ""
    labels.append(b)
    b = ""

#Create array with molecules
for i in range(n):
    rand = random.randint(0, len(dataset))
    for k in dataset[rand]:
        if k == 'graph':
            graphs.append(dataset[rand][k])

#Visualize the molecules
graph = data.Molecule.pack(graphs)
graph.visualize(labels, num_row=2 )

```

Μοντέλο

Θα κατασκευάσουμε ένα μοντέλο που θα προβλέπει αν ένα μόριο μπορεί να διαπεράσει τον BBB ή όχι. Θα είναι ένα νευρωνικό δίκτυο γράφων, συγκεκριμένα το Graph Isomorphism Network που προτάθηκε στο paper [How Powerful are Neural Networks?](#)

Το κριτήριο σειράς κατάταξης θα είναι το bce (binary cross entropy) και τα μέτρα που θα χρησιμοποιηθούν για την αξιολόγηση θα είναι τα auroc και auROC.

```

from torchdrug import core, models, tasks, utils

model = models.GIN(input_dim=dataset.node_feature_dim,
                    hidden_dims=[256, 256, 256, 256],
                    short_cut=True, batch_norm=True, concat_hidden=True)
task = tasks.PropertyPrediction(model, task=dataset.tasks,
                                criterion="bce", metric=(
"auprc", "auroc"))

```

Εκπαίδευση και αξιολόγηση του μοντέλου

Τώρα θα εκπαιδεύσουμε το μοντέλο μας για 100 επαναλήψεις (epochs) έχοντας σαν κριτήριο το bce και θα το αξιολογήσουμε χρησιμοποιώντας ως μέτρα τα auprc και auroc όπως αναφέραμε ανωτέρω.

Ο αλγόριθμος βελτιστοποίησης των παραμέτρων που θα χρησιμοποιηθεί μέσω του PyTorch θα είναι ο Adam που είναι ένας αλγόριθμος τύπου stochastic gradient descent.

Χρησιμοποιήσαμε την εντολή `gpus=[0]` διότι θέλουμε να χρησιμοποιήσουμε την GPU που μας παρέχεται από το Google Colab. Χωρίς τη χρησιμοποίηση GPU η ίδια διαδικασία χρειάζεται περίπου τον επταπλάσιο χρόνο στο συγκεκριμένο παράδειγμα. Όπως είναι γνωστό, οι υπολογισμοί νευρωνικών δικτύων επιταχύνονται σημαντικά με την χρήση GPUs λόγω του ότι στην ουσία πρόκειται για πράξεις πινάκων.

Ο λόγος είναι ο εξής: οι GPUs είναι στην ουσία πολλές processing units που λειτουργούν παράλληλα και οι πιο αποτελεσματικοί αλγόριθμοι πολλαπλασιασμού πινάκων επωφελούνται δραστικά από το parallelization.

Στο τέλος του output window βλέπουμε τα αποτελέσματα εκπαίδευσης που τυπικά είναι μεταξύ 0.8 και 0.9 στο παράδειγμά μας.

```

optimizer = torch.optim.Adam(task.parameters(), lr=1e-3)
solver = core.Engine(task, train_set, valid_set, test_set,
optimizer,

```

```

        gpus=[0], batch_size=1024)
solver.train(num_epoch=100)
solver.evaluate("valid")

```

Παράδειγμα

Σε αυτό το σημείο θα δούμε πως λειτουργεί το μοντέλο μας στην πράξη. Η activation function που χρησιμοποιήθηκε μέσω του PyTorch είναι η sigmoid αλλά υπάρχει η δυνατότητα επιλογής και άλλων δημοφιλών συναρτήσεων όπως η ReLu.

Ανοίγουμε λοιπόν το [BBBP dataset](#) και εισάγουμε την αναπαράσταση SMILES για το μόριο που θέλουμε να προβλέψουμε αν διαπερνά τον BBB ή όχι (την κάνουμε copy and paste στο input window που θα εμφανιστεί όταν τρέξουμε το παρακάτω κελί). Έπειτα μπορούμε να δούμε πόσο καλή ήταν η πρόβλεψη του μοντέλου της συγκρίνοντάς την με την πραγματική τιμή που βρίσκεται στο BBBP dataset δίπλα στο όνομα της ουσίας.

```

inpt = input("Εισάγετε την αναπαράσταση SMILES του μορίου:\n")
smiles_list = [inpt]
m = data.PackedMolecule.from_smiles(smiles_list)
mol = {'graph': m}

task = utils.cuda(task)
mol = utils.cuda(mol)

prediction = torch.sigmoid((task.predict(mol)))
print("Η πρόβλεψη του μοντέλου είναι: ")
print(prediction)

```

Κώδικας του κεφαλαίου “Δημιουργία μορίων”

Εγκατάσταση

Ρύθμιση του περιβάλλοντος εργασίας μας και εντολές εγκατάστασης των απαιτούμενων βιβλιοθηκών.

```
In []:
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
In [ ]:
```

```
import os
import torch
os.environ["TORCH_VERSION"] = torch.__version__
```

```
!pip install torch-scatter -f https://pytorch-geometric.c
om/whl/torch-$TORCH_VERSION.html
!pip install torchdrug
```

Dataset

Εδώ θα κάνουμε import τη βιβλιοθήκη του torchdrug. Το dataset που θα χρησιμοποιήσουμε είναι το ZINC250k το οποίο είναι υποσύνολο της ZINC database η οποία είναι μια βάση δεδομένων που περιέχει πάνω από 230 εκατομμύρια μόρια που είναι υποψήφια για πιθανά φάρμακα. Σημείωση: η διαδικασία ενδέχεται να χρειαστεί πάνω από 5 λεπτά.

```
In [ ]:
```

```
import torch
from torchdrug import datasets, core, models, tasks, util
s
from torch import optim
```

```
dataset = datasets.ZINC250k("~/molecule-datasets/", kekul
ize=True,
                                node_feature="symbol")
```

Προεπαίδευση

Καθορισμός του μοντέλου

Το μοντέλο που θα χρησιμοποιήσουμε θα είναι το RGCN με την GCPN task.

```
In [ ]:
```

```
dataset.transform = None
```

```
model = models.RGCN(input_dim=dataset.node_feature_dim,
```

```

num_relation=dataset.num_bond_type,
hidden_dims=[256, 256, 256, 256], bat
ch_norm=False)

task = tasks.GCPNGeneration(model, dataset.atom_types, ma
x_edge_unroll=12,
max_node=38, criterion="nll")

```

Διαδικασία προεκπαίδευσης

Θα εκπαιδεύσουμε το μοντέλο μας για μια επανάληψη (epoch) και θα αποθηκεύσουμε το προεκπαιδευμένο μοντέλο. Σημειώνουμε ότι η διαδικασία θα χρειαστεί αρκετό χρόνο ακόμα και με τη χρήση GPU.

```

In [ ]:

optimizer = optim.Adam(task.parameters(), lr = 1e-3)
solver = core.Engine(task, dataset, None, None, optimizer
,
                    gpus=(0,), batch_size=128, log_inter
val=10)

```

Για να μη χρειαστεί να ξανακάνουμε αυτή τη διαδικασία, με το παρακάτω κελί μπορούμε να αποθηκεύσουμε το μοντέλο που δημιουργήσαμε στο Google Drive μας. Πρέπει να σιγουρευτούμε ότι τον έχουμε κάνει mount πρώτα, η σχετική εντολή βρίσκεται στην αρχή του notebook.

```

In [ ]:
solver.train(num_epoch=1)
solver.save("/content/drive/MyDrive/pretraining_model.pkl")

```

Προς αποφυγή της μεγάλης σε διάρκεια εκπαίδευσης υπάρχει η δυνατότητα λήψης του μοντέλου που ετοιμάσαμε [από αυτόν τον σύνδεσμό](#). Το αρχείο πρέπει να τοποθετηθεί στο root folder του Google Drive μας. Μπορούμε να το φορτώσουμε με το παρακάτω κελί.

```

In [ ]:

```

```
solver.load("/content/drive/MyDrive/pretraining_model.pkl")
```

Αφού φορτώσαμε το μοντέλο θα παράγουμε μερικά μόρια χρησιμοποιώντας το. Θα εκτυπώσουμε την αναπαράσταση SMILES τους καθώς και θα τα οπτικοποιήσουμε

```
In [ ]:
%matplotlib inline
results = task.generate(num_sample=8, max_resample=5)
results.visualize(num_row=4, num_col=4)
```

Εκπαίδευση

Αυτό είναι το κύριο μέρος της εργασίας. Θα φορτώσουμε το μοντέλο προεκπαίδευσης που δημιουργήσαμε προηγουμένως και θα το βελτιστοποιήσουμε έτσι ώστε τα μόρια που παράγει να έχουν όσο το δυνατόν υψηλότερο QED score (ένας αριθμός μεταξύ του 0 και 1, όσο πιο κοντά στο 1 τόσο πιο πολύ μια ουσία έχει χαρακτηριστικά φαρμάκου).

```
In [ ]:
dataset.transform = None
model = models.RGCN(input_dim=dataset.node_feature_dim,
                    num_relation=dataset.num_bond_type,
                    hidden_dims=[256, 256, 256, 256], batch_norm=True,
                    ch_norm=False)
task = tasks.GCPNGeneration(model, dataset.atom_types,
                             max_edge_unroll=12, max_node_degree=
38,
                             task="qed", criterion=("nll"))
,
                             reward_temperature=1,
                             agent_update_interval=3, gamma=0.9)
task = utils.cuda(task)
optimizer = optim.Adam(task.parameters(), lr = 1e-5)
solver = core.Engine(task, dataset, None, None, optimizer,
,
                    gpus=(0,), batch_size=4, log_interval=1000)
```

```
l=10)
solver.load("/content/drive/MyDrive/pretraining_model.pkl", load_optimizer=False)
```

```
In [ ]:
solver.train(num_epoch=1)
solver.save("/content/drive/MyDrive/qed_model.pkl")
```

Προς αποφυγή της μεγάλης σε διάρκεια εκπαίδευσης υπάρχει η δυνατότητα λήψης του μοντέλου που ετοιμάσαμε [από αυτόν τον σύνδεσμό](#). Το αρχείο πρέπει να τοποθετηθεί στο root folder του Google Drive μας. Μπορούμε να το φορτώσουμε με το παρακάτω κελί.

```
In [ ]:
solver.load("/content/drive/MyDrive/qed_model.pkl")
```

Αφού φορτώσαμε το μοντέλο θα παράγουμε μερικά μόρια χρησιμοποιώντας το. Θα τα οπτικοποιήσουμε και θα τοποθετήσουμε στο καθένα σαν ετικέτα το αντίστοιχο QED score του.

```
In [ ]:

%matplotlib inline
from torchdrug import metrics
from torchdrug import data

task = utils.cuda(task)
results = task.generate(num_sample=8, max_resample=5)

pretrained_sample = results.to_smiles()

graph = data.Molecule.pack(results)

labels = metrics.QED(graph)
labels1 = []
for i in labels:
    labels1.append(i.item())
```

```
labels1 = [round(x,4) for x in labels1]
labels2 = ["QED score: " + str(x) for x in labels1]

graph.visualize(figure_size=(4,4), titles = labels2, num_row=2 )
```

Κώδικας του κεφαλαίου “Βιοιατρικές αναπαραστάσεις γνώσης”

Εγκατάσταση

Ρύθμιση του περιβάλλοντος εργασίας μας και εντολές εγκατάστασης των απαιτούμενων βιβλιοθηκών.

```
In [ ]:
import os
import torch
os.environ["TORCH_VERSION"] = torch.__version__

!pip install torch-scatter -f https://pytorch-geometric.com/whl/torch-$TORCH_VERSION.html
!pip install torchdrug
```

Dataset

Εδώ θα κάνουμε import το Hetionet dataset.

```
In [ ]:
from torchdrug import datasets
hetionet = datasets.Hetionet("~/knowledge-graph-datasets")
```

Εύρεση γονιδίων που εμπλέκονται με την Alzheimer's disease (AD)

Σε αυτό το σημείο θα εισάγουμε το disease ID της AD (το οποίο θα ανακτηθεί από το <https://disease-ontology.org/>) και θα βρούμε με ποια γονίδια συσχετίζεται.

```
In [ ]:
h_graph = hetionet.graph
```

```

i = 0
match_count = 0
for i in range(2250196):

    h, t, r = h_graph.edge_list[i]
    a = str(hetionet.entity_vocab[h]) + " " + str(hetionet.
relation_vocab[r]) + " " + str(hetionet.entity_vocab[t])
    if "Disease::DOID:10652 DaG" in a:
        print("Triplet in position " + str(i) + ": " + a)
        match_count += 1
    i += 1
print("\nTotal number of matches: " + str(match_count))

```

Συσχέτιση AD με διαβήτη τύπου 2 (T2D)

Θα αποθηκεύσουμε τα γονίδια που συσχετίζονται με αυτές τις 2 ασθένειες σε 2 lists και έπειτα θα βρούμε τα κοινά τους στοιχεία, δηλαδή τα κοινά γονίδια που εμπλέκονται μεταξύ της AD και του T2D.

Γονίδια AD

```

In [ ]:
#alzheimer's
h_graph = hetionet.graph

i = 0
match_count = 0
genes_alzheimer = []
for i in range(2250196):

    h, t, r = h_graph.edge_list[i]
    a = str(hetionet.entity_vocab[h]) + " " + str(hetionet.
relation_vocab[r]) + " " + str(hetionet.entity_vocab[t])
    if "Disease::DOID:10652 DaG" in a:
        genes_alzheimer.append(hetionet.entity_vocab[t])
for i in genes_alzheimer:
    print(i)

```

Γονίδια T2D

```

In [ ]:
#diabetes
h_graph = hetionet.graph

i = 0
match_count = 0
genes_diabetes = []
for i in range(2250196):

    h, t, r = h_graph.edge_list[i]
    a = str(hetionet.entity_vocab[h]) + " " + str(hetionet.
relation_vocab[r]) + " " + str(hetionet.entity_vocab[t])
    if "Disease::DOID:9352 DaG" in a:
        genes_diabetes.append(hetionet.entity_vocab[t])
for i in genes_diabetes:
    print(i)

```

Εύρεση κοινών γονιδίων και αποθήκευσή τους σε list

```

In [ ]:
genes_alzheimer_as_set = set(genes_alzheimer)
intersection = genes_alzheimer_as_set.intersection(genes_
diabetes)

intersection_as_list = list(intersection)

for i in intersection_as_list:
    print(i, end='\n')
print("\nNumber of common genes: " + str(len(intersection_as_lis
t)))

```

Βιβλιογραφία

AboYoussef, A. M., Khalaf, M. M., Malak, M. N., & Hamzawy, M. A. (2021).
 Repurposing of sildenafil as antitumour; induction of cyclic guanosine
 monophosphate/protein kinase G pathway, caspase-dependent apoptosis

- and pivotal reduction of Nuclear factor kappa light chain enhancer of activated B cells in lung cancer. *J Pharm Pharmacol*.
- Baker, H. (2021, 07 10). *How many atoms are in the observable universe?* Ανάκτηση από Live Science: <https://www.livescience.com/how-many-atoms-in-universe.html>
- Behjati, S., & Tarpey, P. (2013). What is next generation sequencing? *Arch Dis Child Educ Pract Ed.*, 236-238.
- Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S., & Hopkins, A. L. (2012). Quantifying the chemical beauty of drugs. *Nat Chem*.
- Blattner, F. (1997). The complete genome sequence of Escherichia coli K-12. *Science*, 1453-74.
- Caulfield, B. (2009). *What's the Difference Between a CPU and a GPU?* Ανάκτηση από NVIDIA Blog: <https://blogs.nvidia.com/blog/2009/12/16/whats-the-difference-between-a-cpu-and-a-gpu/>
- Ghofrani, H. A., Osterloh, I. H., & Grimminger, F. (2006). Sildenafil: from angina to erectile dysfunction to pulmonary hypertension and beyond. *Nat Rev Drug Discov*.
- Grey, J., & Thompson, D. H. (2010). Challenges and Opportunities for New Protein Crystallization Strategies in Structure-Based Drug Design. *Expert Opinion on Drug Discovery*.
- Hosmer, D., & Lemeshow, S. (2000). Applied Logistic Regression, 2nd. Στο D. Hosmer, & S. Lemeshow, *Applied Logistic Regression, 2nd* (σσ. 160-164). NY: John Wiley and Sons.
- Lipinski, C. A., Lombardo, F., Dominy, B. W., & Paul J. Feeney. (1997). Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced Drug Delivery Reviews*, 3-25.
- Milmanll, B., & Zhurkovich, K. (2020). Big Data in Modern Chemical Analysis. *Journal of Analytical Chemistry*, 444.
- Moses, H., Dorsey, E., Matheson, D., & Tier, S. (2005). Financial anatomy of biomedical research. *JAMA*, 1333-42.
- Myers, S., & Baker, A. (2001). Drug discovery - an operating model for a new era. 727-30.
- Nguyen, T. T., Ta, Q. T., Nguyen, T. K., Nguyen, T. T., & Giau, V. V. (2020). Type 3 Diabetes and Its Role Implications in Alzheimer's Disease. *Int J Mol Sci*.

Optibrium. (χ.χ.). *Quantitative Estimate of Drug-likeness in StarDrop*. Ανάκτηση από Optibrium:

<https://www.optibrium.com/downloads/QED%20for%20StarDrop.pdf>

Pardridge, W. M. (2012). Drug transport across the blood–brain barrier. *Journal of Cerebral Blood Flow & Metabolism*, 1959-1972.

Reymond, J., & Awale, M. (2012). Exploring chemical space for drug discovery using the chemical universe database. *ACS Chem Neurosci*.

Ryabukhin, D. M., Moroz, Y. S., Savych, O., Chuprina, A., Horvath, D., Zabolotna, Y., . . . Judd, D. B. (2019). Evolution of commercially available compounds for HTS. *Drug Discov Today*.

S Hollander, M. G. (1990). The plateau in cytotoxic cancer drug discovery. *J Med*.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Berg, R. v., Titov, I., & Welling, M. (2017, 03 17). *Modeling Relational Data with Graph Convolutional Networks*.

Ανάκτηση από arXiv.org: <https://arxiv.org/pdf/1703.06103>

Tetko, I. (2016). *Big Data in Chemistry*. Ανάκτηση από BigChem:

http://bigchem.eu/sites/default/files/School1_Tetko.pdf

U.S. Environmental Protection Agency. (2016). *SMILES Tutorial*. Ανάκτηση από U.S. Environmental Protection Agency:

https://archive.epa.gov/med/med_archive_03/web/html/smiles.html

Vargas-Soria, M., Carranza-Naval, M. J., & Marco, A. d. (2021). Role of liraglutide in Alzheimer's disease pathology. *Alzheimer's Research & Therapy*.

Wallach, O. (2021, November 1). *Ranked: The Most Prescribed Drugs in the U.S.*

Ανάκτηση από Visual Capitalist: <https://www.visualcapitalist.com/ranked-the-most-prescribed-drugs-in-the-u-s/>

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). *How Powerful are Graph Neural Networks?* . Ανάκτηση από arXiv: <http://arxiv.org/abs/1810.00826>

Zhu, Z., Liu, S., Shi, C., Chen, J., Zhang, Z., Qu, M., . . . Tang, J. (2021). *TorchDrug: A powerful and flexible machine learning platform for drug discovery [Computer software]*. Ανάκτηση από GitHub:

<https://github.com/DeepGraphLearning/torchdrug>

IBM. (χ.χ.). *What is a knowledge graph?* Ανάκτηση από IBM:

<https://www.wordreference.com/engr/semantic>