DeepCDR (2019)

Monday, December 1, 2025 9:45 PM

Multimodal deep learning baseline.

Αποτελείται από:

(1) Drug Encoder → Uniform Graph Convolutional Network (UGCN)

- Παίρνει το molecule (drug) ως **molecular graph**.
- Κάθε άτομο = node.
- Κάθε χημικός δεσμός = edge.
- Χρησιμοποιεί graph convolution layers
- Βγάζει **drug embedding** περίπου 128–256 dimensions.

drug encoding

Προβλημα !

Τα drugs έχουν **διαφορετικό αριθμό atoms**, αλλά τα Graph Convolutional Networkθέλουν σταθερό μέγεθος.

Λύση DeepCDR: **Uniform Graph**

- Δημιουργούν complementary graph για κάθε drug
- Στόχος = όλα τα drug graphs να έχουν 100 atoms (N = 100)
- Τα missing atoms → "dummy nodes"

Άρα κάθε drug έχει:

- 0–96 πραγματικά atoms
- remaining μέχρι 100 → dummy atoms

Καθε **ατομο** εχει:
75 dimensional feature vector
που περιεχει atom type, degree, hybridization κλπ.

Έτσι προκύπτουν:

- adjacency matrix: 100×100
- feature matrix: 100×75

Για κάθε drug.

---> UGCN propagation

Εφαρμόζουν GCN convolution :

Κάθε atom μαθαίνει πληροφορία από τα "γειτονικά" atoms του.
Εχουμε ενα G = (V,E)
V = nodes (entitites)
E = edges (ενωσεις των ατομων)
Η βασική εξισωση ειναι $H^{(l+1)}=\sigma(A\sim H^{(l)}W^{(l)})$

- **H^(l)** = features των nodes στο layer l
- **W^(l)** = learnable weight matrix
- **σ** = activation (ReLU)
- **Â = A + I** = adjacency με self-connections
- **D̂** = degree matrix

Εστω drug με 4 atoms.

- **Drug graph**

```
C — C — O
    \
     N
```

- **$H^{(0)}$**

Κάθε atom έχει vector 75 features:

```
H^(0) =
A1: [75-dim features]
A2: [75-dim features]
A3: [75-dim features]
A4: [75-dim features]
```

```
(A1) C — (A2) C — (A3 )O
            \
          (A4) N
```

- **Adjacency matrix (A)**

```
    A1 A2 A3 A4
A1   1  1  0  0
A2   1  1  1  1
A3   0  1  1  0
A4   0  0  1  1
```

(1 = bonded atoms, plus diagonal 1s for self-loops)

(1) Aggregation

[A1 features]

$H^{(1)}$

Global Max Pooling → DRUG EMBEDDING

A4 0 0 1 1

(1) Aggregation

[A1 features]
\
\
[A3] → (aggregate) → new A2
/
/
[A4 features]

(2) Linear Transformation → εφαρμόζει W
(aggregated_features) × W (75×100)

(3) Nonlinearity (ReLU)
H^(1) = ReLU(A_norm H^(0) W)

$H^{(1)}$

Κάθε atom έχει τώρα **νέο** vector 100 dims.
Node features (after GCN layer 1):

A1 → [h11 h12 ... h1,100]
A2 → [h21 h22 ... h2,100]
A3 → [h31 h32 ... h3,100]
A4 → [h41 h42 ... h4,100]

Global Max Pooling → DRUG EMBEDDING

A1: [2, 1, 7]
A2: [3, 2, 5]
H = A3: [0, 4, 6]
A4: [1, 3, 4]

Global Max →
[max(2,3,0,1), max(1,2,4,3), max(7,5,6,4)]
= [3, 4, 7]

Άρα το **τελικό embedding** είναι:

drug embedding = 100-dimensional vector

Τελικό αποτέλεσμα όλης της διαδικασίας

Μετά τα 3 GCN layers + global max pooling, παίρνουμε: drug_embedding = ένα vector με 100 αριθμούς
[0.23, -0.11, 0.98, 1.21, ..., 0.44] (100 διαστάσεις)

Αυτό είναι:

- **σταθερού μήκους** (πάντα 100 τιμές)
- **εκπαιδευμένο** να περιγράφει τη χημική δομή
- **μάθει μοτίβα** όπως rings, aromaticity, functional groups
- **μάθει τι κάνουν οι γείτονες ατόμων**
- **ενσωματώνει όλο το γράφημα του molecule**

Γιατί είναι χρήσιμο;

Γιατί δεν μπορείς να ταΐσεις ένα νευρωνικό με MOL files ή με atoms.
Το μοντέλο για IC50 prediction χρειάζεται input:

[drug_embedding ⊕ cell_embedding]
Άρα:

- Το DeepCDR βγάζει **ένα συνεχές, μαθημένο vector** για το drug.
- Μετά αυτό "ενώνεται" με omics της cell line.
- Και αυτό πάει στο τελικό CNN/MLP για να προβλέψει IC50.

(2) Omics-specific subnetworks

Το DeepCDR έχει **τρία διαφορετικά είδη δεδομένων για κάθε cell line**:

1) **Genomic mutations** (binary vector: 34,673 χαρακτηριστικά)

Για κάθε mutation position:
0 = δεν υπάρχει μετάλλαξη **Έξοδος** → ένα vector 100 διαστάσεων (embedding)
1 = υπάρχει μετάλλαξη

2) **Gene expression** (697 χαρακτηριστικά)

Normalized, real-valued vector με συνεχείς τιμές
Εδώ η πληροφορία **δεν είναι positional**, δεν υπάρχει σειρά. **Έξοδος** → ένα embedding 100 διαστάσεων
Γι' αυτό χρησιμοποιούν: Fully-connected network (MLP)

3) **DNA methylation** (808 χαρακτηριστικά)

continuous vector
Δεν έχει σειριακή φύση που να απαιτεί CNN. **Έξοδος** → 100-dimensional embedding
Fully-connected network (MLP)

Γιατί CNN σε ένα binary vector;
Οι θέσεις είναι ταξινομημένες κατά chromosome position.
(A) Οι μεταλλάξεις είναι σειριακές στο γονιδίωμα
Κάθε feature στο mutation vector έχει *βιολογική σειρά*:

Chr1 position 2134 → index 0
Chr1 position 3322 → index 1
...
Chr2 ...
...
Chr22 ...

Αυτό είναι όπως μια 1D εικόνα / sequence, όχι απλά μια λίστα.
(B) Στις μεταλλάξεις υπάρχουν τοπικά μοτίβα
Cell line A:

[0, 1, 1, 0, 0, 1, ...]
Cell line B:

[1, 1, 0, 1, 0, 0, ...]
Τα clusters των μεταλλάξεων:

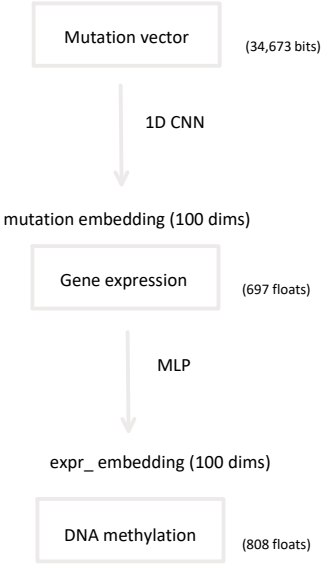
- γύρω από hotspots
- μέσα στο ίδιο γονίδιο
- στο ίδιο exon
- σε regulatory regions

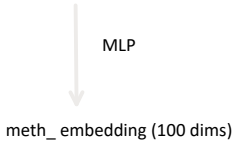
Έχουν σημασία.
Και *CNN is extremely good* στο να βρίσκει τέτοια patterns.
(C) Το CNN "σκανάρει" γειτονικές θέσεις όπως στο NLP
Παράδειγμα kernel 5:

[1,0,1,1,0] → pattern A
[0,1,1,0,1] → pattern B

Το convolutional filter μαθαίνει patterns όπως:

- "mutation cluster"
- "consecutive 1s"
- "hotspot signature"
- "tumor suppressor region hit"





Τελικα εχουμε :
mutation_emb → 100 dims
expression_emb → 100 dims
methyl_emb → 100 dims

concatenate

cell_ embedding = [mut_ emb ⊕ expr_emb ⊕ meth_emb] = 300 dimensions

Πλεονεκτηματα εναντι One-hot cell line:

One-hot cell line:
Cell_ID → [0,0,0,0,...,1,...0] (958 dims)

Deep CDR 3 subnetworks:
πραγματικό biological content της cell line, όχι απλά ID

(3) Τι έχουμε μέχρι τώρα (4 embeddings)