IDA-ML II: Project Presentation

Project: 3D GCN for Serotonergic Binding Affinity Prediction
Examinee: Paul Utsch
Date: March 18, 2025

Table of Contents

1. Problem Setting & Data

- **Predict serotonergic binding affinity** of molecules towards 5-HT2A receptor
  - Crucial for neuropsychiatric drug discovery
- 2D molecular descriptors are frequently used for this task
  - -> Compare predictive power: precomputed 2D molecular descriptors vs. 3D molecular graphs

1. Problem Setting & Data

- **Supervised Regression Problem**
  - Input Space: $X = \{x_i \mid i = 1, \ldots, n\}$
    - Graph Representations: $x_i = (V_i, E_i)$
      - V_i: set of nodes (atoms), E_i: set of edges (bonds)
    - 2D Descriptors: $x_i \in \mathbb{R}^d$
  - Output Space: $Y = \{y_i \in \mathbb{R}\}$
  - Hypothesis space: $H = \{h \mid h : X \to Y\}$
  - Objective: $h^* = \arg\min_{h \in H} \mathbb{E}_{(x,y)\sim D}[L(h(x), y)]$
- **Criterion**: MSE

1. Problem Setting & Data

- **Data set** manually curated from ChEMBL 35 database
- **14058 samples** (activity records) in total, **2353 of target receptor** (5-HT2A), rest for pre-training
  - **9 target types** in total
- Each data point consists of SMILES identifier, target name, target index, and pKi value

data > serotonin_receptors_subset.csv > data

```
ChEMBL ID;Name;UniProt Accessions;Type;Organism;Compounds;Activities;Tax ID;Species Group Flag
CHEMBL1898;Serotonin 1b (5-HT1b) receptor;P28222;SINGLE PROTEIN;Homo sapiens;2268;3085;9606;FALSE
CHEMBL214;Serotonin 1a (5-HT1a) receptor;P08908;SINGLE PROTEIN;Homo sapiens;9246;15316;9606;FALSE
CHEMBL3371;Serotonin 6 (5-HT6) receptor;P50406;SINGLE PROTEIN;Homo sapiens;6414;10124;9606;FALSE
CHEMBL225;Serotonin 2c (5-HT2c) receptor;P28335;SINGLE PROTEIN;Homo sapiens;7165;11942;9606;FALSE
CHEMBL1833;Serotonin 2b (5-HT2b) receptor;P41595;SINGLE PROTEIN;Homo sapiens;5395;10739;9606;FALSE
CHEMBL1983;Serotonin 1d (5-HT1d) receptor;P28221;SINGLE PROTEIN;Homo sapiens;2095;3130;9606;FALSE
CHEMBL3155;Serotonin 7 (5-HT7) receptor;P34969;SINGLE PROTEIN;Homo sapiens;4079;5835;9606;FALSE
CHEMBL1875;Serotonin 4 (5-HT4) receptor;Q13639;SINGLE PROTEIN;Homo sapiens;1033;2077;9606;FALSE
CHEMBL224;Serotonin 2a (5-HT2a) receptor;P28223;SINGLE PROTEIN;Homo sapiens;9084;15395;9606;FALSE
```

data > merged_serotonin_data.csv > data

```
smiles,pchembl_value,target_name,target_id
CCN/C(=N\S(=O)(=O)c1ccccc1)N1CC(CC)C=N1,7.47,Serotonin 6 (5-HT6) receptor,7
Cc1c(S(=O)(=O)NCCCN2CCN(c3nsc4ccccc34)CC2)sc2ccc(F)cc12,6.92,Serotonin 6 (5-HT6) receptor,7
O=S(=O)(c1cccc(F)c1)n1cc2c3c(cccc31)N1CCNCC1C2,9.12,Serotonin 6 (5-HT6) receptor,7
O=S(=O)(NCCCCN1CCN(c2nsc3ccccc23)CC1)c1cccc2scnc12,6.92,Serotonin 6 (5-HT6) receptor,7
O=S(=O)(NCCCN1CCN(c2noc3ccccc23)CC1)c1ccc(F)c(Cl)c1,6.48,Serotonin 6 (5-HT6) receptor,7
CCCN/C(=N\S(=O)(=O)c1cccc(Cl)c1)N1CC(CC)C=N1,7.84,Serotonin 6 (5-HT6) receptor,7
CCC1C=NN(/C(N)=N/S(=O)(=O)c2cccc(Cl)c2)C1,7.62,Serotonin 6 (5-HT6) receptor,7
CCN/C(=N\S(=O)(=O)c1cccc(Cl)c1)N1CC2(C=N1)CCNCC2,8.14,Serotonin 6 (5-HT6) receptor,7
CCN/C(=N\S(=O)(=O)c1cccc(Cl)c1)N1CC(CC)C=N1,8.1,Serotonin 6 (5-HT6) receptor,7
CCN/C(=N\S(=O)(=O)c1cccc(Cl)c1)N1CC(C)C=N1,8.1,Serotonin 6 (5-HT6) receptor,7
CCN/C(=N\S(=O)(=O)c1cc(Cl)cc(Cl)c1)N1CC(CC)C=N1,7.66,Serotonin 6 (5-HT6) receptor,7
```

1. Problem Setting & Data

- **Data creation protocol**
  - Selected subset of serotonin receptor targets of homo sapiens (protein targets)
  - Filtered activity records:
    - Must contain pchembl value of Ki ($-\log_{10}(Ki)$)
    - Must not be N/A records (usually below a certain pKi threshold)
    - Must be cell-based records (measured in living cells, not isolated)

2. Data Preprocessing

1. For each molecule, use RDKit to create 3D molecular graph representation from SMILES string
    - Each graph contains:
        - x: vector of atom features
        - pos: vectorial 3D position of atom from conformer, (x, y, z)
        - edge_index: vector of edges of molecule, ([i, j, …], [j, i, …])
        - y: target binding affinity value
2. z-normalize all atom features and target values

2. Data Preprocessing

- x: vector of atom features
  - Atomic mass (mass of the atom – weight influences drug-likeness / binding)
  - Van der Waals Radius (as measure for atomic size – larger atoms tend to fit worse in receptor pockets)
  - Number of valence electrons (influencing bonding capabilities)
  - Charge of the atom (influencing bonding capabilities)
  - Whether part of aromatic system (aromatic rings are common in 5HT ligands)
  - Whether part of a ring (rings are common in psychoactive drugs)
  - Number of neighbors (for identifying functional groups)
  - One-hot hybridization type (influencing shape and bonding behavior of molecule)

2. Data Preprocessing

- 2D molecular descriptors

```
# tried a bunch of descriptor functions from Descriptors._descList
# - these are the ones that did NOT crash the kernel ...
safe_descriptors = [
    "MolWt",
    "MolLogP",
    "MolMR",
    "NumValenceElectrons",
    "NumRadicalElectrons",
    "HeavyAtomCount",
    "NHOHCount",
    "NOCount",
    "RingCount",
    "FractionCSP3",
    "TPSA",
    "NumHDonors",
    "NumHAcceptors",
    "NumRotatableBonds",
    "HallKierAlpha",
    "Kappa1",
    "Kappa2",
    "Kappa3",
    "Chi0",
    "Chi1",
    "fr_Al_COO",
    "fr_Al_OH",
    "fr_Ar_N",
    "fr_C_O",
    "fr_NH1",
    "fr_NH2",
]
```

2. Data Preprocessing

- pos: vectorial 3D position of atom from conformer, (x, y, z)
  - Embed atoms in 3D space such that molecular energy is minimized

2. Data Preprocessing

- Split serotonin data into target / pre-training sets
- Split target set into train / test set (0.9 / 0.1)
- Leave test set untouched

3. ML Methods

- **Naive Baseline**, predicting mean of z-normalized target
- **Random Forest Baseline** trained on 2D molecular descriptors
- **SeroGCN** without pre-training
- **SeroGCN** pre-trained

3. ML Methods

- **SeroGCN**
  1. Two convolutional layers (self-reference included), each:
     - $$h_i' = ReLU(\sum_{j \in N(i) \cup i} \frac{e_{ij}}{\sqrt{d_i \cdot d_j}} \cdot W_l \cdot h_j + b_l)$$
       - e_{ij}: (pos-weighted) adjacency
       - d_i, d_j: number of neighbors of node i / node j (self-reference included)
  2. One global max pooling operation: max feature-wise across latent node embeddings
  3. One final fully-connected linear layer -> prediction
- **Criterion**: (Masked) MSE + L2
- **Optimizer**: Adam

```python
from torch_geometric.nn import GCNConv, global_max_pool
from torch.nn import Linear
import torch.nn.functional as F

n_features = merged_serotonin_data_processed_5ht2a_train[0].x.shape[1]


class SeroGCN(torch.nn.Module):
    def __init__(self, n_hidden, n_out=1):
        super(SeroGCN, self).__init__()

        self.conv1 = GCNConv(n_features, n_hidden)
        self.conv2 = GCNConv(n_hidden, n_hidden)
        self.fc = Linear(n_hidden, n_out)
        self.sigma = 1.0  # distance weighting parameter

    def forward(self, mol_batch) -> torch.Tensor:
        x, pos, edge_index = (
            mol_batch.x,
            mol_batch.pos,
            mol_batch.edge_index,
        )

        row, col = edge_index
        eucl_edge_dist = torch.norm(pos[row] - pos[col], p=2, dim=1)
        weight_distance = torch.exp(
            -(eucl_edge_dist**2) / (2 * self.sigma**2)
        )  # Gaussian distance weighting

        # message passing with diustance weights
        x = self.conv1(x, edge_index, edge_weight=weight_distance)
        x = F.relu(x)
        x = self.conv2(x, edge_index)
        x = F.relu(x)

        # global pooling for graph-level representation
        x = global_max_pool(x, mol_batch.batch)
        x = self.fc(x)

        return x
```

4. Model Selection & Evaluation Protocol

- (k=5)-fold cross-validation protocol
  - For each hyper parameter set, do 5-fold cv and get mean risk as empirical risk estimate
  - pick set with lowest empirical risk estimate
  - fit final model on whole training set

```
hyperparam_grid = {
    "lr": [0.1, 0.01, 0.001],
    "n_hidden": [32, 64, 128],
    "epochs": [10, 20, 30],
}
```

```
# adapted hyperparams based on performance of previous 5-HT2A model
hyperparam_grid = {
    "lr": [0.01, 0.005, 0.001],
    "n_hidden": [64],
    "epochs": [
        30,
        40,
    ],
}
```

```
hyperparam_grid = {
    "lr": [0.1, 0.01, 0.001],
    "n_hidden": [32, 64, 128],
    "epochs": [10, 20, 30],
}
```

```
# adapted hyperparams based on performance of previous 5-HT2A model
hyperparam_grid = {
    "lr": [0.01, 0.005, 0.001],
    "n_hidden": [64],
    "epochs": [
        30,
        40,
    ],
}
```

5. Pre-Training Procedure

1. Pre-train SeroGCN in a multi-target regression task on pre-training data
    - Masked MSE as criterion
    - Same model selection protocol
2. Load these weights into new instance of SeroGCN (excluding FC layer)
3. Get risk estimate of new instance on target task (single-target regression) via 5-fold cv with reduced learning rate
4. Return model trained on full target train set

6. Results

- On validation sets, estimated risks:
  - Naive Baseline: RMSE of 1.01 (holdout)
  - Random Forest Baseline: RMSE of 0.63 (5-fold cv estimate)
  - SeroGCN without pre-training: RMSE of 0.71 (5-fold cv estimate)
  - SeroGCN with pre-training: RMSE of 0.67 (5-fold cv estimate)
- On test set:
  - Random Forest Baseline: RMSE of 0.61
  - SeroGCN with pre-training: RMSE of 0.62

6. Results

- On validation sets, estimated risks:
  - Naive Baseline: RMSE of 1.01 (holdout)
  - Random Forest Baseline: RMSE of 0.63 (5-fold cv estimate)
  - SeroGCN without pre-training: RMSE of 0.71 (5-fold cv estimate)
  - SeroGCN with pre-training: RMSE of 0.67 (5-fold cv estimate)
- On test set:
  - Random Forest Baseline: RMSE of 0.61
  - SeroGCN with pre-training: RMSE of 0.62

6. Results

- Random Forest Baseline: RMSE of 0.63
  - $R^2_{rf} = 1 - (MSE_{rf}/\sigma^2_{target}) \approx 0.60$
- SeroGCN without pre-training: RMSE of 0.71
  - $R^2_{npt} = 1 - (MSE_{npt}/\sigma^2_{target}) \approx 0.50$
- SeroGCN with pre-training: RMSE of 0.67
  - $R^2_{pt} = 1 - (MSE_{pt}/\sigma^2_{target}) \approx 0.55$

6. Results

- 2D molecular descriptors may already carry enough information for prediction of serotonergic binding affinity
- Future implementations may improve SeroGCN performance further

7. Discussion: Future Implementations

- Data
  - Regarding ChEMBL N/A records: use classification model first to avoid bias towards strong binding
  - Inspect noise in ChEMBL data: experiment with single protein records
    - Ki: concentration of ligand to bind 50% of target's active sites in equilibrium
  - Inspect noise in conformer positions
- Hyperparameter tuning
  - Bayesian optimization

7. Discussion: Future Implementations

- Model architecture
  - Expand convolutional operations
    - Work with edge attributes
    - Improve processing of pos (implement positional encoding)
  - Replace global pooling
    - LSTM
    - Transformer-encoder
  - MLP instead of FC Linear