Tips and tricks for ligand-protein interaction prediction

Table of Contents

1. Encoding Molecular and Protein Information

Machine learning models can only “understand” numbers — not chemical structures or sequences directly. So, before we can train models, we need to encode molecules and proteins as numerical representations that capture their important properties.

1.1 Encoding Small Molecules

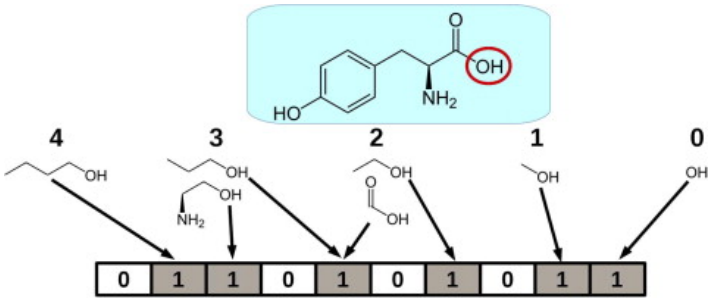
Small molecules (like drugs or ligands) can be described in several ways. Each representation captures different aspects of the molecule — from its raw structure to chemical properties.

a. Fingerprint-based Representations

Fingerprints are compact numerical vectors summarizing which structural patterns exist in a molecule. Examples include ECFP/Morgan fingerprints, MACCS keys, and topological fingerprints.

Common types:

- **ECFP / Morgan fingerprints:**
  Circular patterns around each atom; excellent at capturing local structure.
  (Used in most bioactivity prediction models.)
- **MACCS Keys:**
  A fixed list (166 bits) describing if certain predefined substructures exist.
- **Topological fingerprints:**
  Encode all atom paths up to a certain length.



! You can find a (very extensive, maybe too extensive ;)) tutorial on how to generate fingerprints here: https://github.com/gashawmg/Molecular-fingerprints/blob/main/Calculating%20molecular%20fingerprints%20available%20in%20RDkit%20.ipynb

✅ **Why it matters:**

Fingerprints are compact and well-tested in classical machine learning (e.g., Random Forests, SVMs).

b. Descriptor-based Representations

- Molecular weight
- LogP (lipophilicity)

- Number of hydrogen bond donors/acceptors
- Polar surface area
- Rotatable bonds, aromatic rings, etc.

Tools like RDKit or Mordred can compute hundreds of such descriptors automatically.

✅ **Why it matters:**
Descriptors turn molecules into feature vectors similar to tabular data — ideal for models like Random Forest or XGBoost.

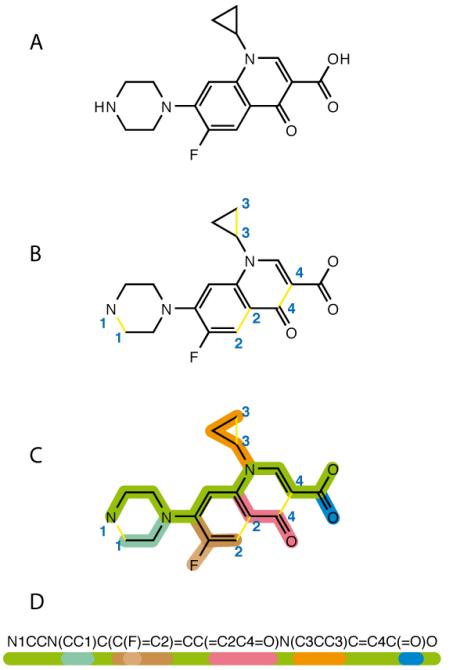
! You can read more about descriptors and fingerprints here:
https://link.springer.com/protocol/10.1007/978-1-4939-7899-1_1

c. String-based Representations

SMILES (Simplified Molecular Input Line Entry System)

- SMILES converts a molecule into a **line of text** describing its atoms and bonds.
- Example:
  Ethanol → "CCO"
  (2 carbon atoms and 1 oxygen connected in a chain)
- Easy to store, fast to process.
- Often used in **language-model-based** approaches (like treating molecules as "sentences" in chemistry).

✅ **Why it matters:** SMILES lets us feed molecules into models that work with text (RNNs, Transformers).



! You can learn more about how we can learn from SMILES (and molecular strings) in this research paper: https://pubs.rsc.org/en/content/articlehtml/2025/dd/d4dd00311j

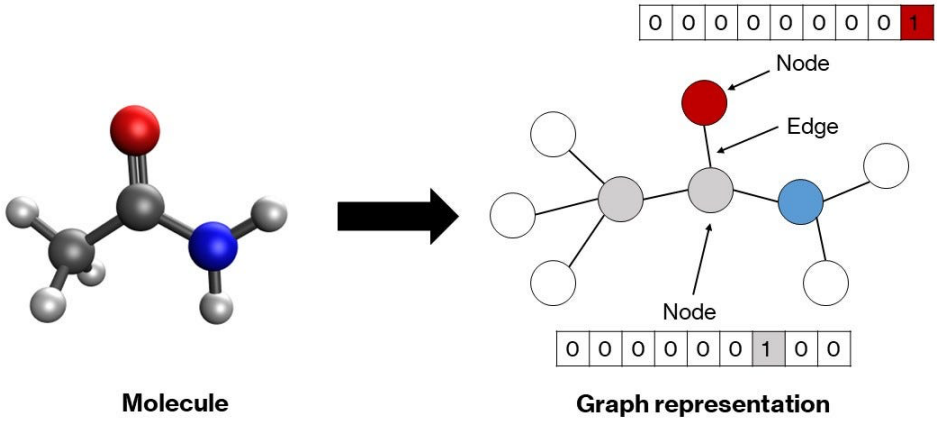
d. Graph-based Representations

A molecule can also be seen as a graph, where:

- Nodes = atoms
- Edges = chemical bonds

We can describe each atom by features (e.g., element type, charge) and each bond by type (single, double, aromatic, etc.).

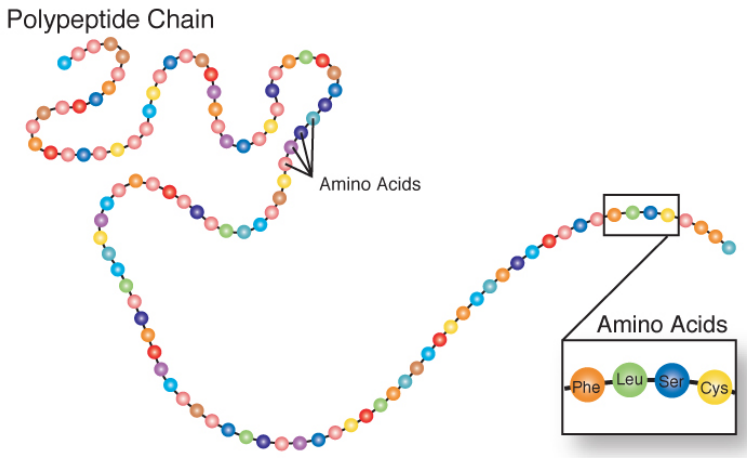
✅ **Why it matters:**
This representation is very natural for chemistry and works well with **Graph Neural Networks (GNNs)** — which learn directly from molecular structure.



! You can read more about how one can learn from graphs using neural networks here: https://link.springer.com/chapter/10.1007/978-3-031-37196-7_2 and here https://medium.com/data-science/building-a-graph-convolutional-network-for-molecular-property-prediction-978b0ae10ec4

1.2 Encoding Protein Sequences

Proteins are **chains of amino acids** (usually 20 different types). Each protein's function and structure depend on this sequence. Like with molecules, we must turn these sequences into numbers that ML models can understand.



a. One-hot Encoding

- Each amino acid becomes a vector of length 20 (one entry per amino acid type).
- Example:
  "A" (Alanine) → [1, 0, 0, 0, …]
- Combine all amino acids in a sequence into a 2D matrix.

✅ **Why it matters:**
Simple and intuitive — but ignores similarities between amino acids.

b. Physicochemical Property Encoding

Each amino acid can be replaced by **numerical values describing its properties**, like:

- Hydrophobicity (water-loving or not)
- Charge (positive/negative/neutral)
- Molecular size or mass

✅ **Why it matters:**
Adds biochemical meaning — the model can "feel" chemical similarities between residues.

c. Substitution Matrix Embeddings (e.g., BLOSUM62, PAM250)

These matrices tell how likely one amino acid is to substitute another during evolution. Each amino acid can thus be represented by its substitution score vector.

✅ **Why it matters:**
Captures **evolutionary similarity** — important for identifying functionally similar proteins.

d. Pretrained Protein Language Model Embeddings

Modern deep learning models (like transformers used in natural language processing efforts, e.g., ChatGPT) can learn from millions of protein sequences.

Examples:

- ESM (Evolutionary Scale Modeling) — by Meta AI
- ProtBERT, ProtT5 — by HuggingFace / Rostlab
- UniRep, SeqVec, etc.

These models generate embeddings (i.e., vectors of features):

- Per residue (each amino acid)
- Or per protein (pooled embedding)

✅ **Why it matters:**
These embeddings can capture **biological meaning**, such as structure, binding sites, or function — even without explicit labels.

! Protein embeddings can be expensive to compute. Here you can find a repository of protein embeddings, with an explanation on how to use them
https://www.uniprot.org/help/embeddings

e. Structure-based Encodings

If 3D structure is available (from **PDB** or **AlphaFold**):

- **Distance maps:** Pairwise distances between residues.
- **Contact maps:** Whether residues are close in space.
- **Graph representations:** Residues as nodes, 3D distances as edges.

✅ **Why it matters:**
These encodings reflect the **true 3D shape** — essential for modeling binding or interactions.

Here you can find a tutorial on how to extract features from AlphaFold.
https://www.youtube.com/watch?v=QBOIAw7W_ss

2. Combining Different Representations

In drug–target interaction (DTI) prediction or related tasks, we often have:

- A **molecule representation**
- A **protein representation**

We then need to **combine** (or *fuse*) them to predict whether the molecule binds to the protein.

Level	How to Combine	Explanation	Example/Use Case
1. Feature Concatenation	Just join both feature vectors end-to-end.	Simplest approach; works with classical ML.	[Molecule_features + Protein_features] → RandomForest
2. Late Fusion (Model-level)	Train separate models for molecule and protein, then combine predictions.	Combine results (average, weighted sum, etc.)	Ensemble predictions of two independent models
3. Mid-level Fusion (Representation-level)	Join learned embeddings before the final prediction layer in a neural network.	Allows some interaction between molecule and protein features.	CNN for protein + GNN for molecule → merged in MLP
4. Cross-attention / Co-embedding Models	The model explicitly learns how parts of the molecule "attend to" parts of the protein.	Mimics molecular docking logic.	Transformers with cross-attention (e.g., MolTrans)
5. End-to-end Multimodal Learning	Learn directly from raw data (SMILES + amino acid sequence) together.	Deep joint representation learning.	DeepDTA, GraphDTA, etc.

✅ **Why it matters:**

Choosing how to combine representations affects model performance, complexity and interpretability.

3. Useful Packages

Below are recommended Python packages — categorized by their use.

3.1. For Small Molecules

Package	What It Does	Links
RDKit (recommended)	Core cheminformatics library. Compute fingerprints, descriptors, molecular drawings, conversions.	https://www.rdkit.org/
DeepChem	Simplifies ML on molecular data; integrates TensorFlow and PyTorch.	https://deepchem.io/
Mordred	Computes thousands of molecular descriptors.	https://mordred-descriptor.github.io/

3.2. For Protein Representation

Package	What It Does	Notes / Example
Peptidy (recommended)	Protein encoding for ML applications (Global descriptors, Amino acid descriptors, BLOSUM62 encoding, One-hot encoding, label encoding)	https://github.com/molML/peptidy
Biopython	Basic bioinformatics (FASTA reading, sequence manipulation, structural data).	https://biopython.org/
ESM (Meta AI)	Pretrained transformer models for protein embeddings.	https://github.com/facebookresearch/esm
ProtTrans	Provides ProtBERT, ProtT5, and more protein LMs.	https://github.com/agemagician/ProtTrans
BioEmbeddings	Easy access to multiple embedding models (ESM, ProtBERT, SeqVec).	https://docs.bioembeddings.com/

3.3. For Machine Learning

Type	Packages	Notes
Classical ML	scikit-learn, XGBoost, LightGBM	For regression, classification, baseline models.
Deep Learning	PyTorch, TensorFlow, Keras	Core frameworks for neural networks.
Graph Learning	PyTorch Geometric, DGL	Specialized for graph neural networks.
Data & Visualization	pandas, numpy, matplotlib, seaborn	For data cleaning, feature exploration, and visualization.