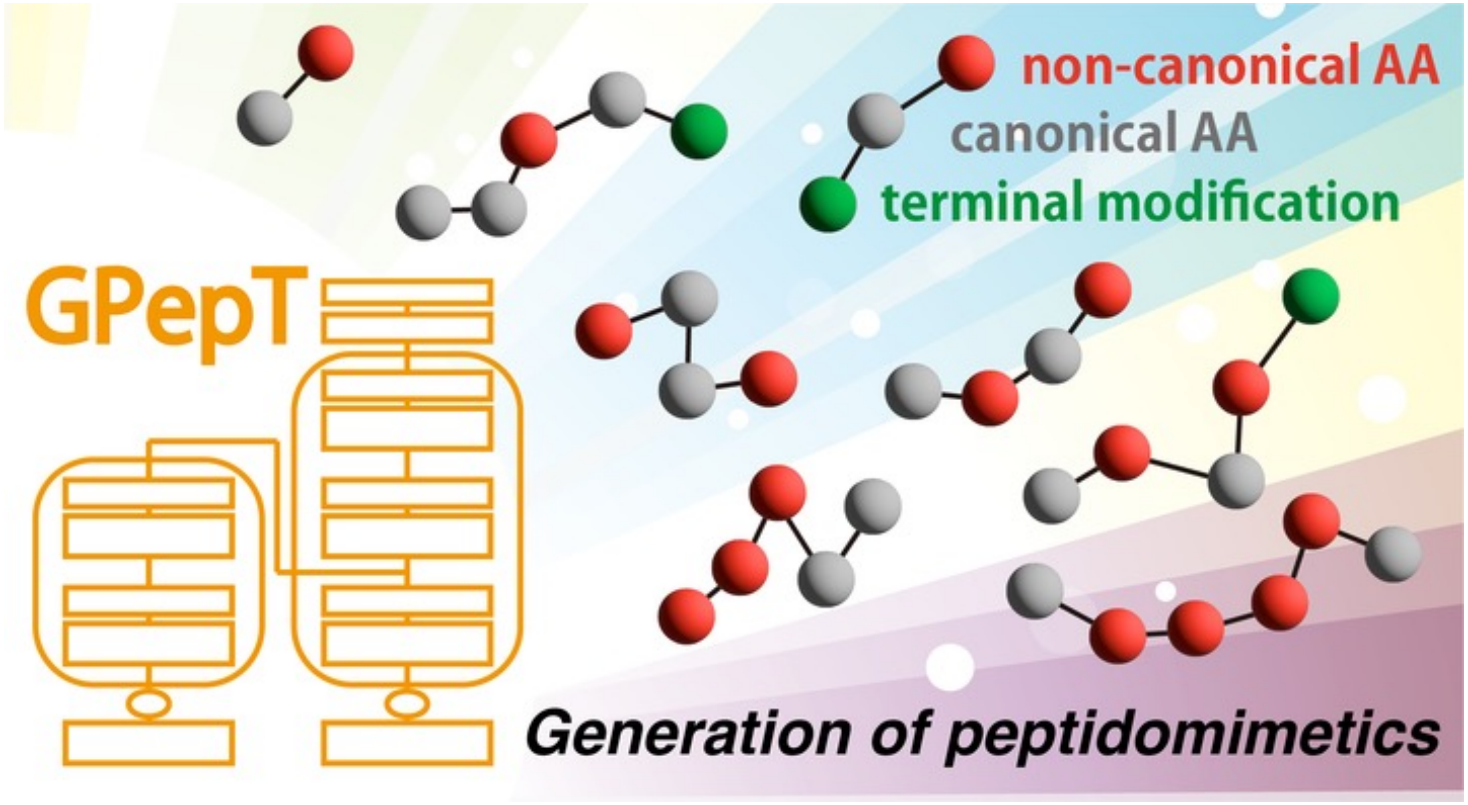
GPepT: A Foundation Language Model for Peptidomimetics Incorporating Noncanonical Amino Acids



Yuna Oikawa (D1), Tsuda Lab, UTokyo

Why LLM (and not other AI models) for bio ?

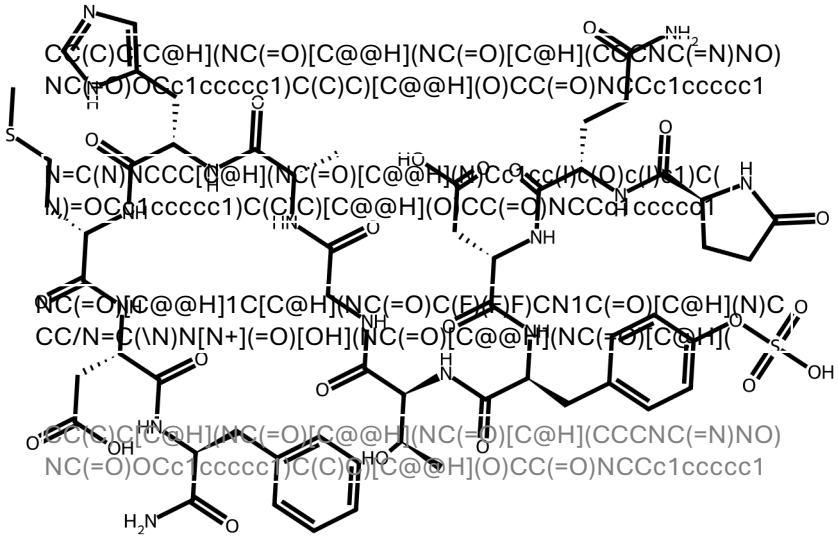
—Easy to use.

—Capability to connect with any data.

What should be done now?

—Feed LLMs with the “right” data.

Inconsistent data: SMILES



CC(C)C[C@H](NC(=O)[C@@H](NC(=O)[C@H](CCCNC(=N)NO)
NC(=O)OCc1ccccc1)C(C)C)[C@@H](O)CC(=O)NCCc1ccccc1

N=C(N)NCCC[C@H](NC(=O)[C@@H](N)Cc1cc(I)c(O)c(I)c1)C(
N)=OCc1ccccc1)C(C)C)[C@@H](O)CC(=O)NCCc1ccccc1

NC(=O)[C@@H]1C[C@H](NC(=O)C(F)(F)F)CN1C(=O)[C@H](N)C
CC/N=C(\N)N[N+](=O)[OH](NC(=O)[C@@H](NC(=O)[C@H](

CC(C)C[C@H](NC(=O)[C@@H](NC(=O)[C@H](CCCNC(=N)NO)
NC(=O)OCc1ccccc1)C(C)C)[C@@H](O)CC(=O)NCCc1ccccc1

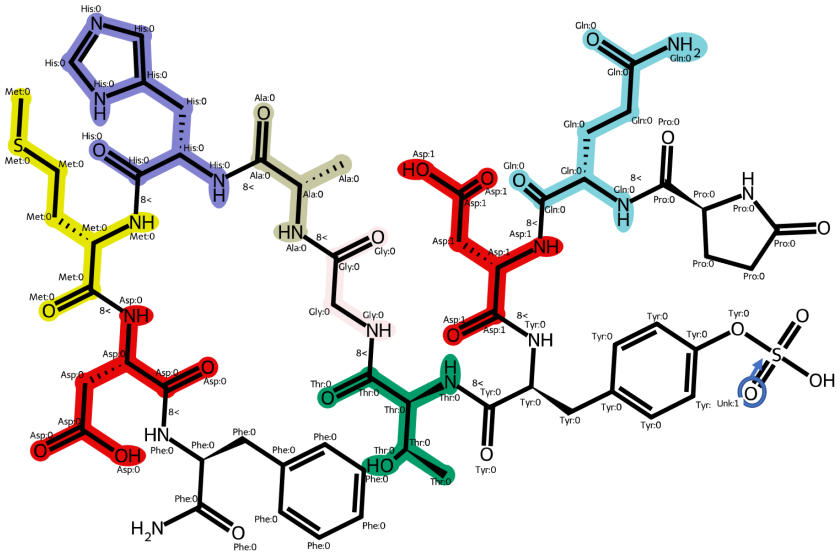
N=C(N)NCCC[C@H](NC(=O)[C@@H](N)Cc1cc(I)c(O)c(I)c1)C(
N)=OCc1ccccc1)C(C)C)[C@@H](O)CC(=O)NCCc1ccccc1

. . .

Consistent data: Amino acid sequence

PQDPTGAHMDP

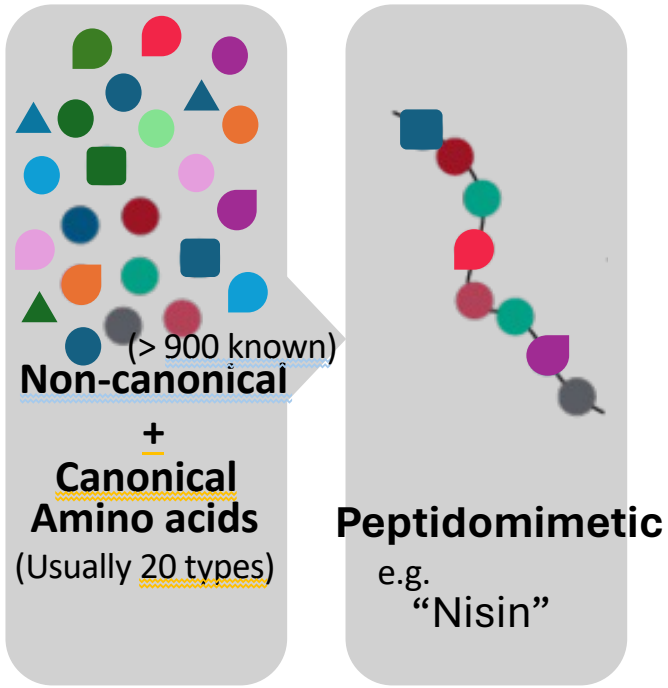
Inconsistent data:
SMILES



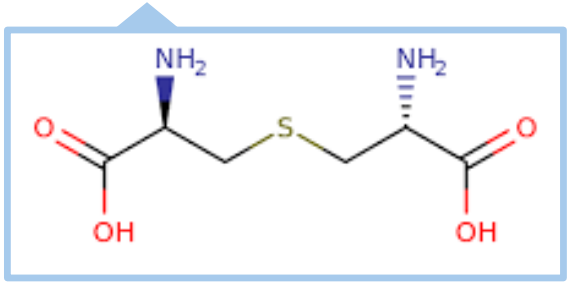
Consistent data:
Amino acid sequence

?QD?TGAHMD?

Non-canonical amino acids

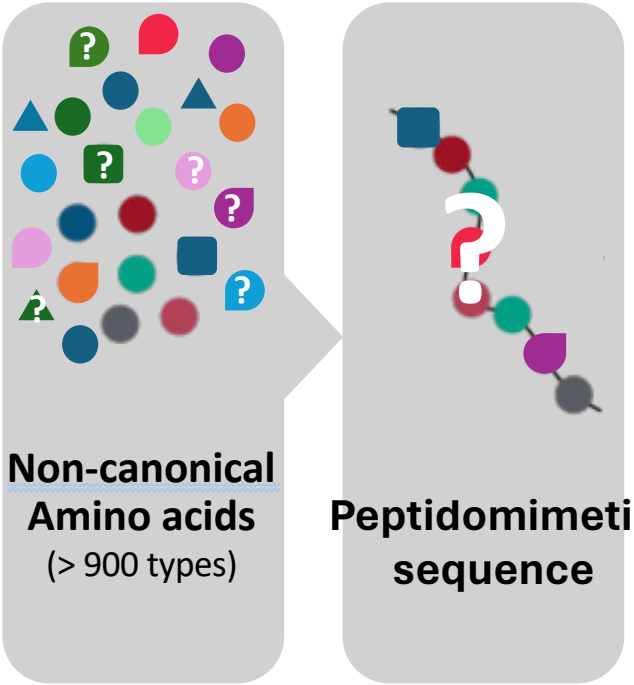


- e.g. Nisin :
  - Natural antimicrobial peptide
  - Uses lanthionine etc ➡ unique structure



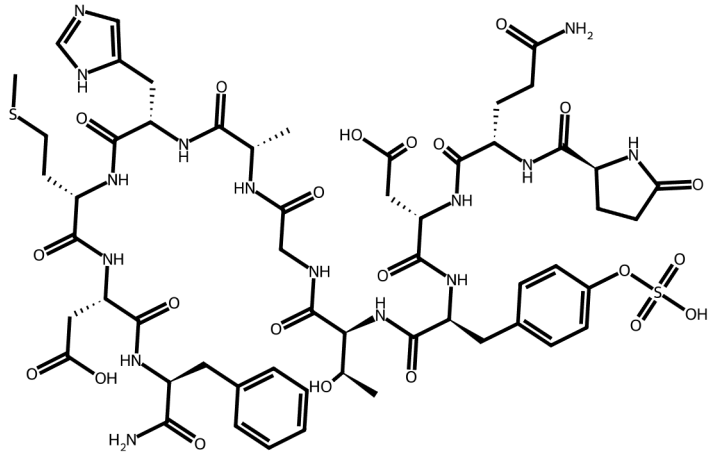
  - Very few resistance reported

Peptidomimetics

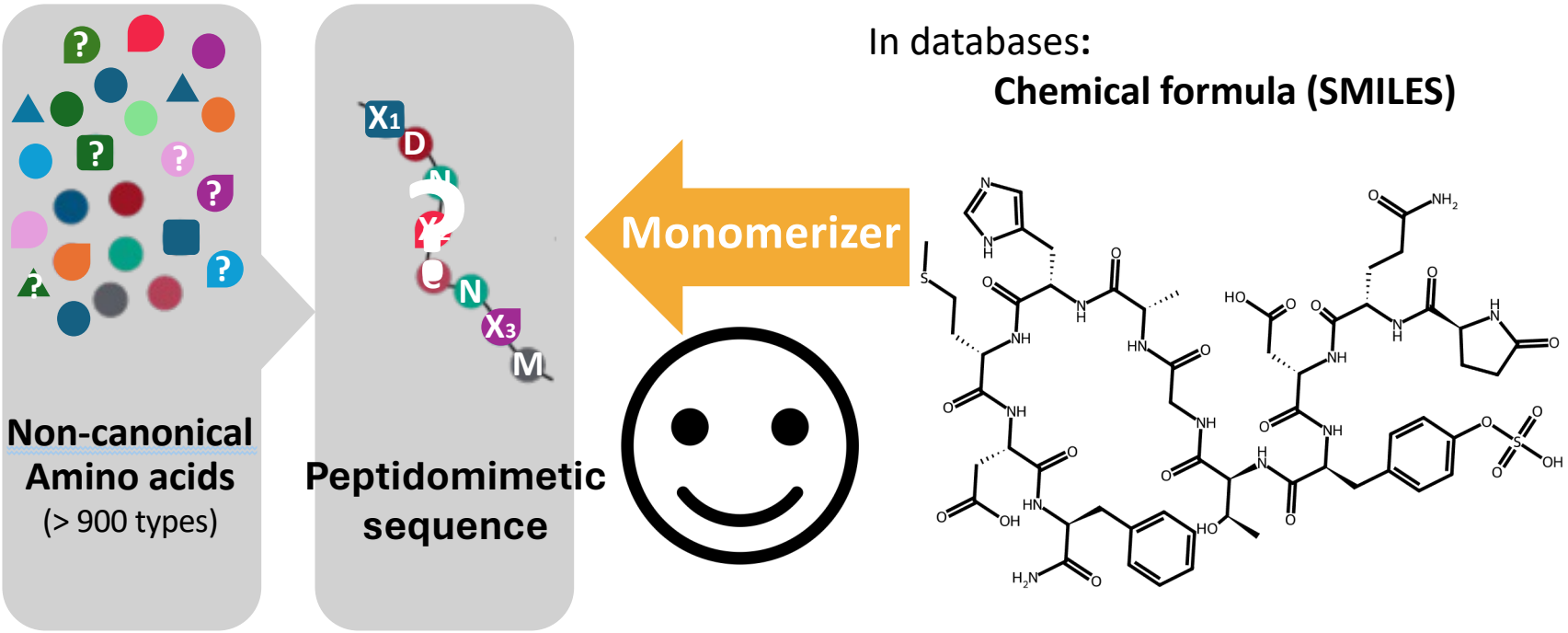


In databases:

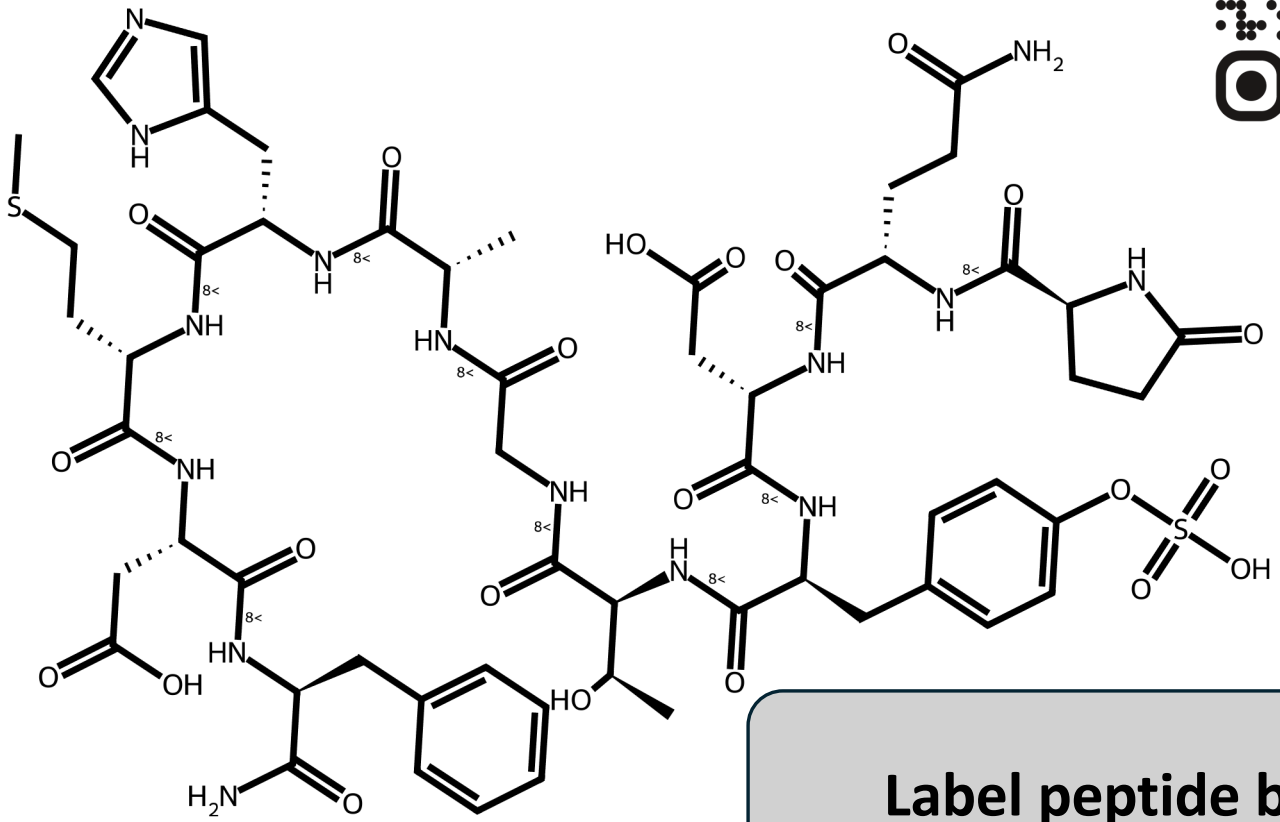
Chemical formula (SMILES)



Peptidomimetics

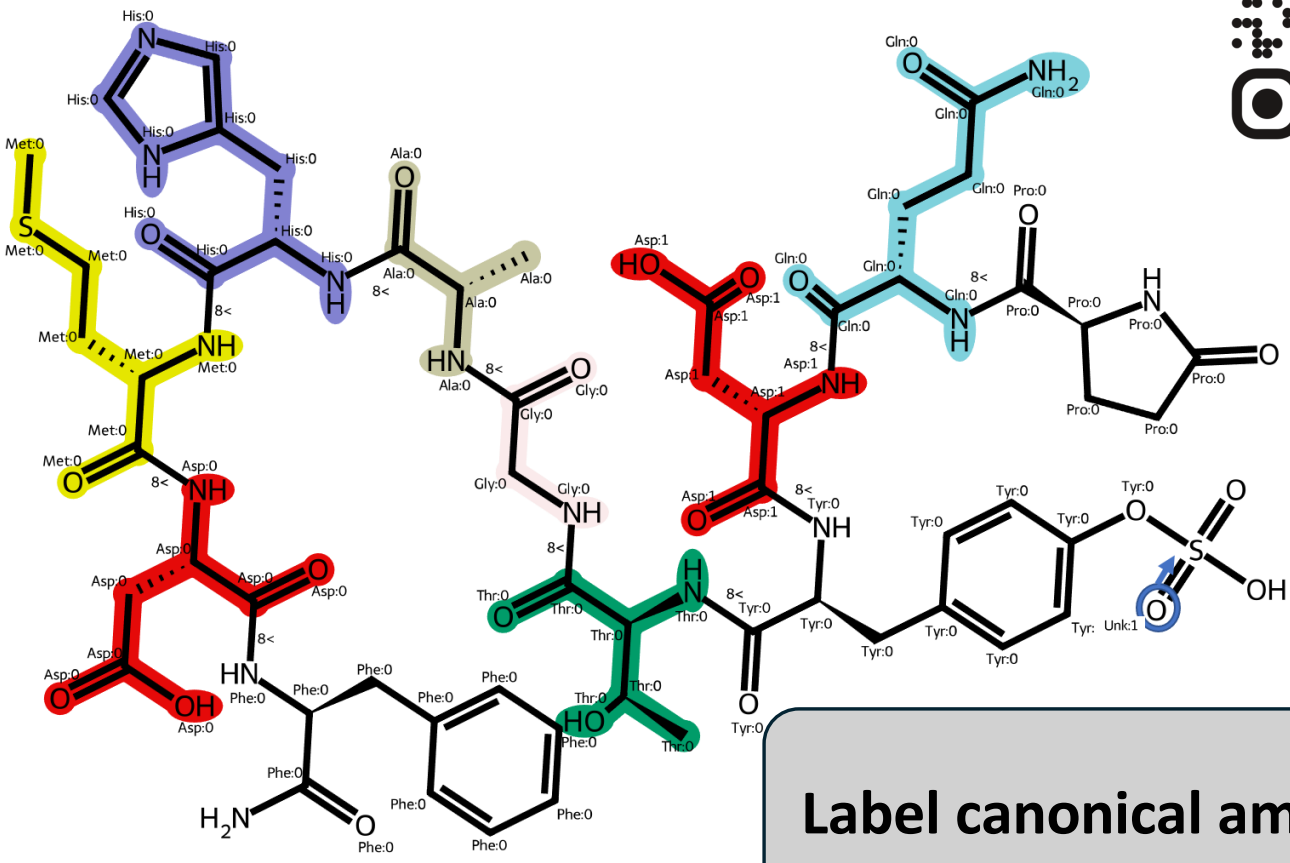


Monomerizer



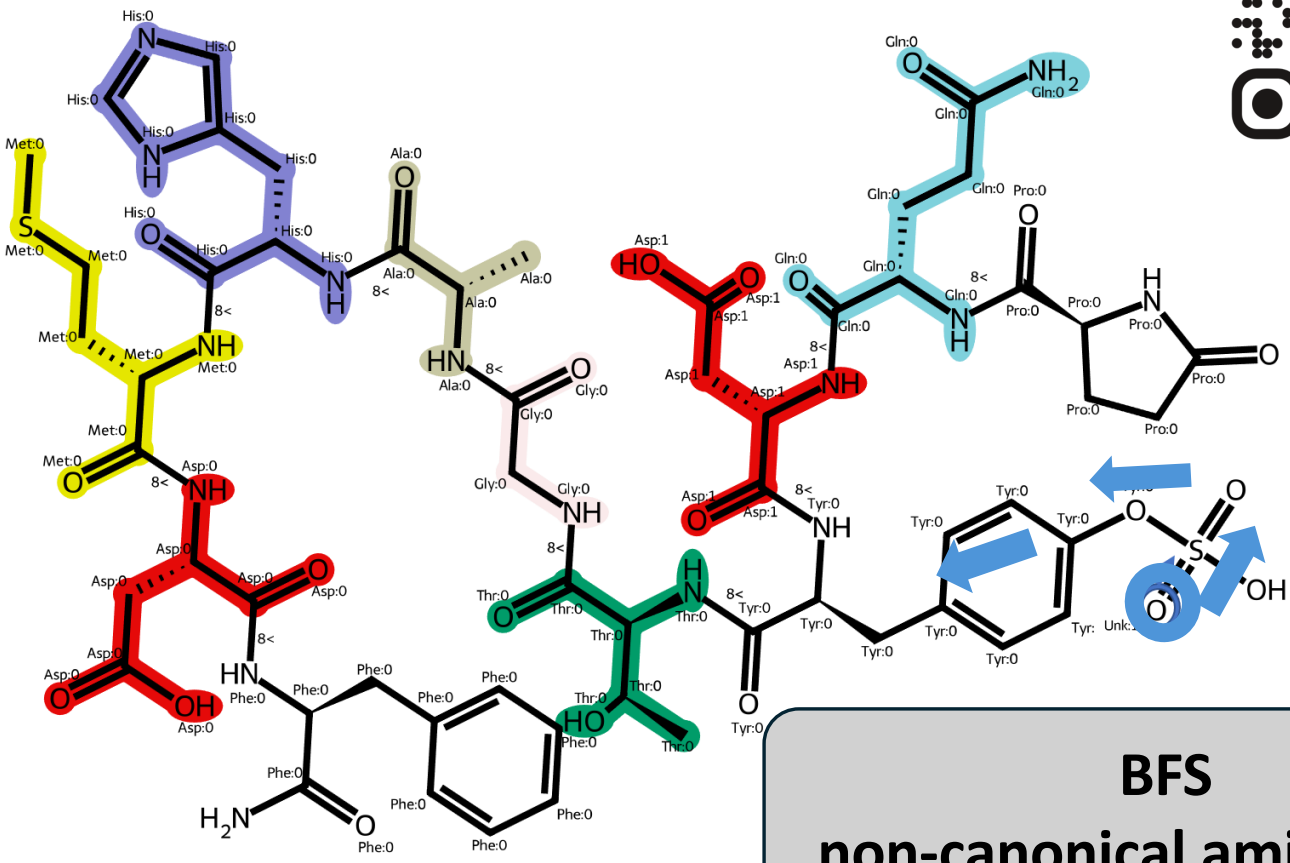
Label peptide bonds

Monomerizer



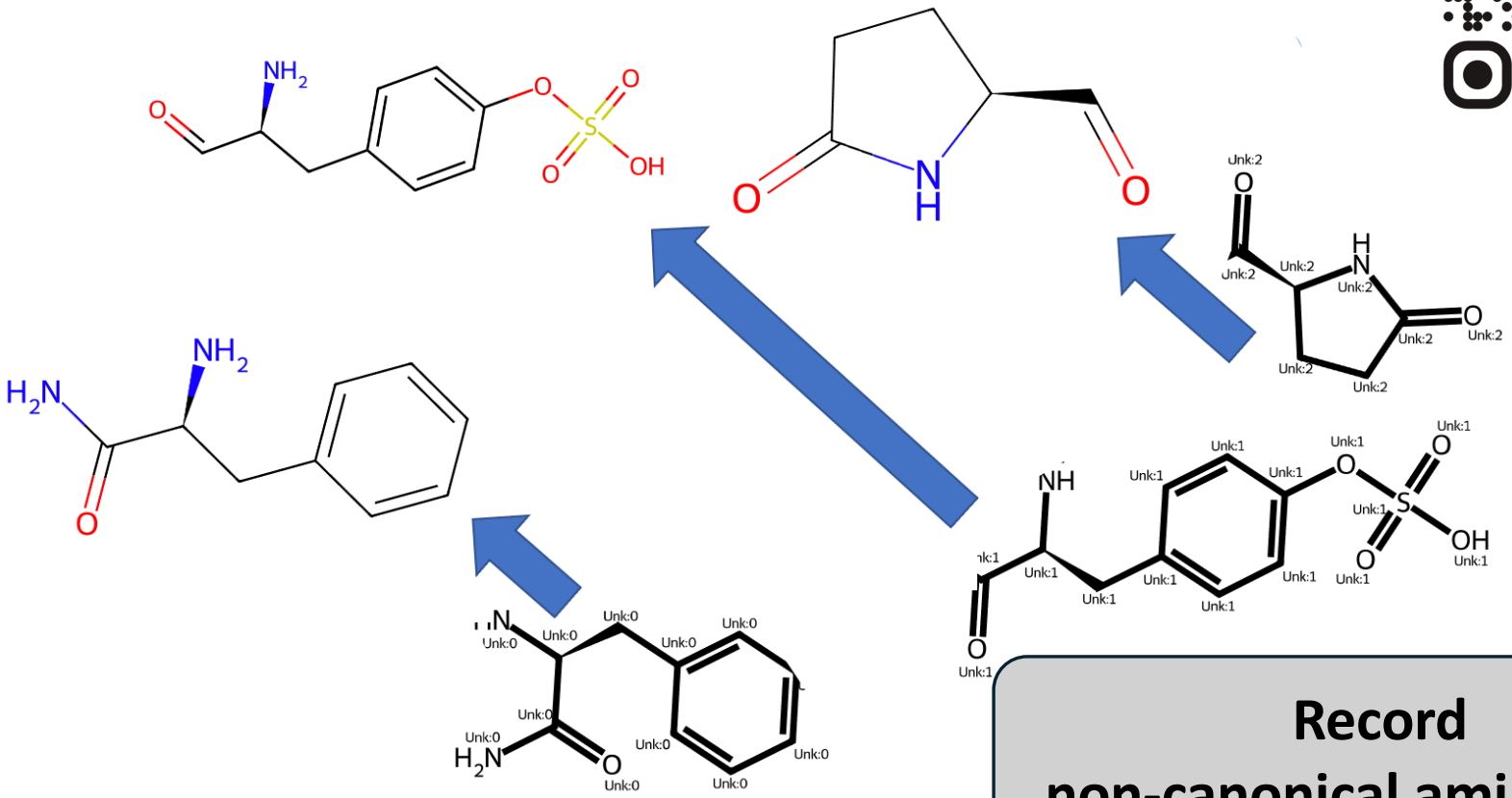
Label canonical amino acids

Monomerizer



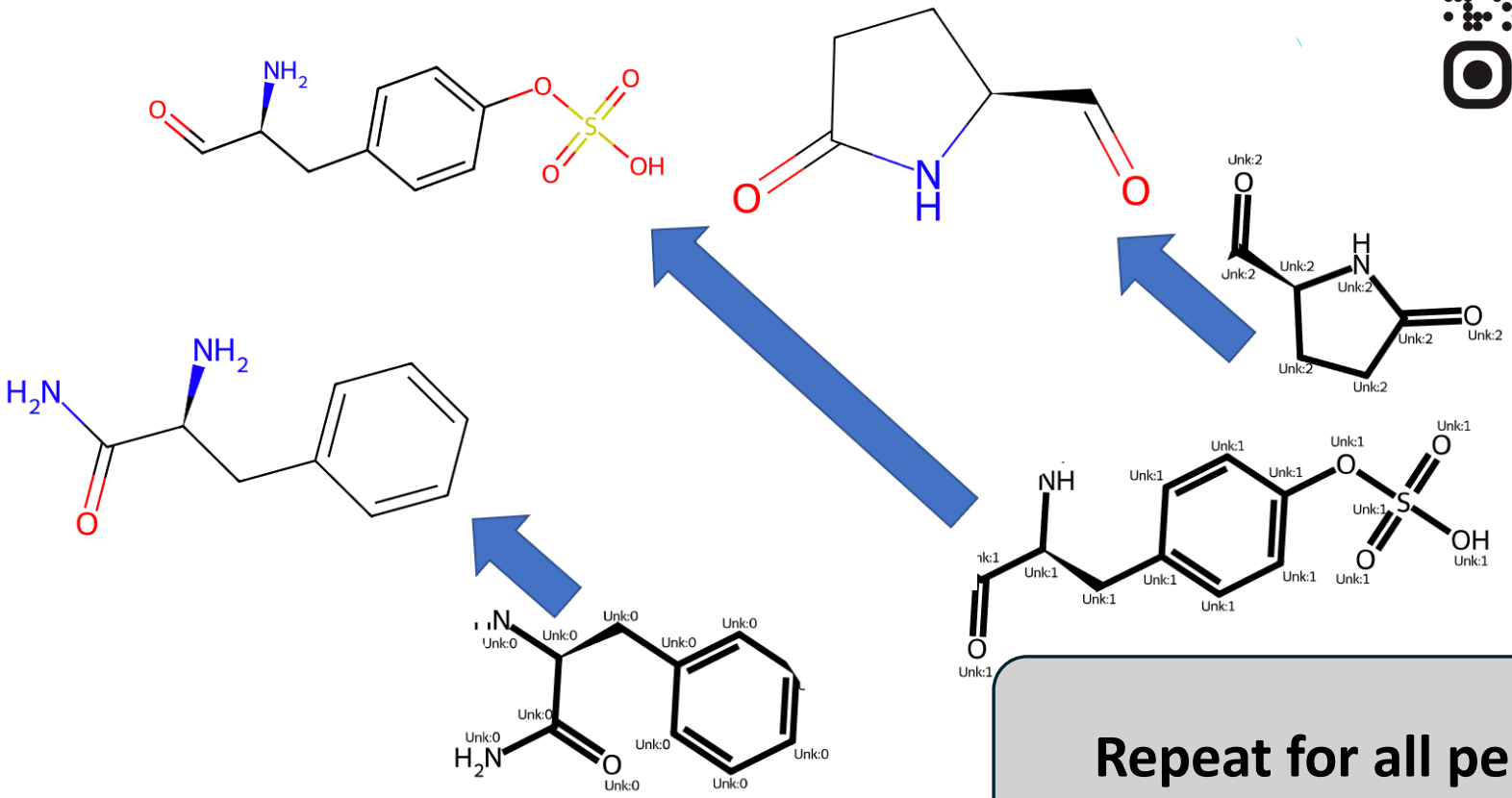
BFS
non-canonical amino acids

Monomerizer



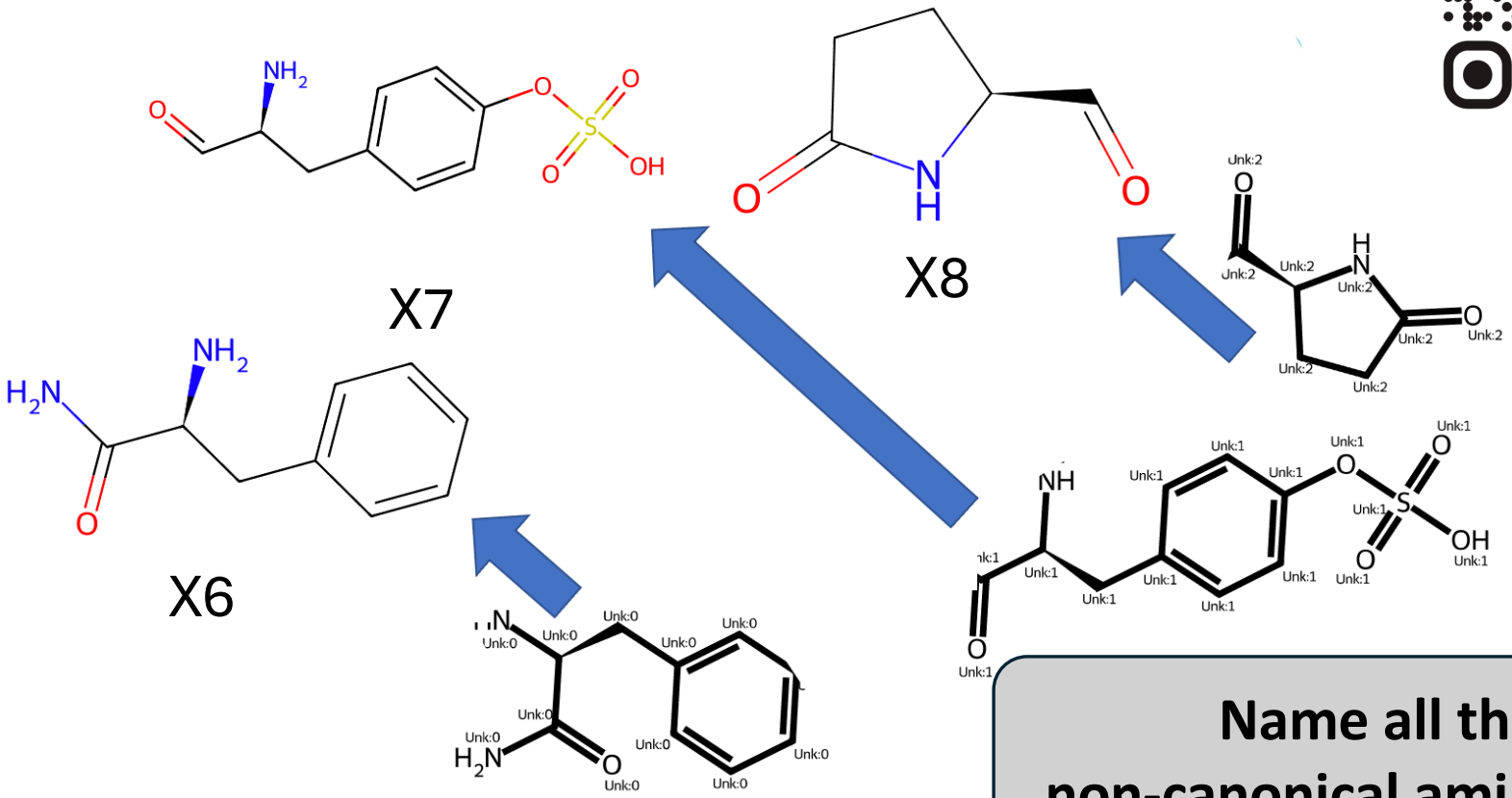
Record
non-canonical amino acids

Monomerizer



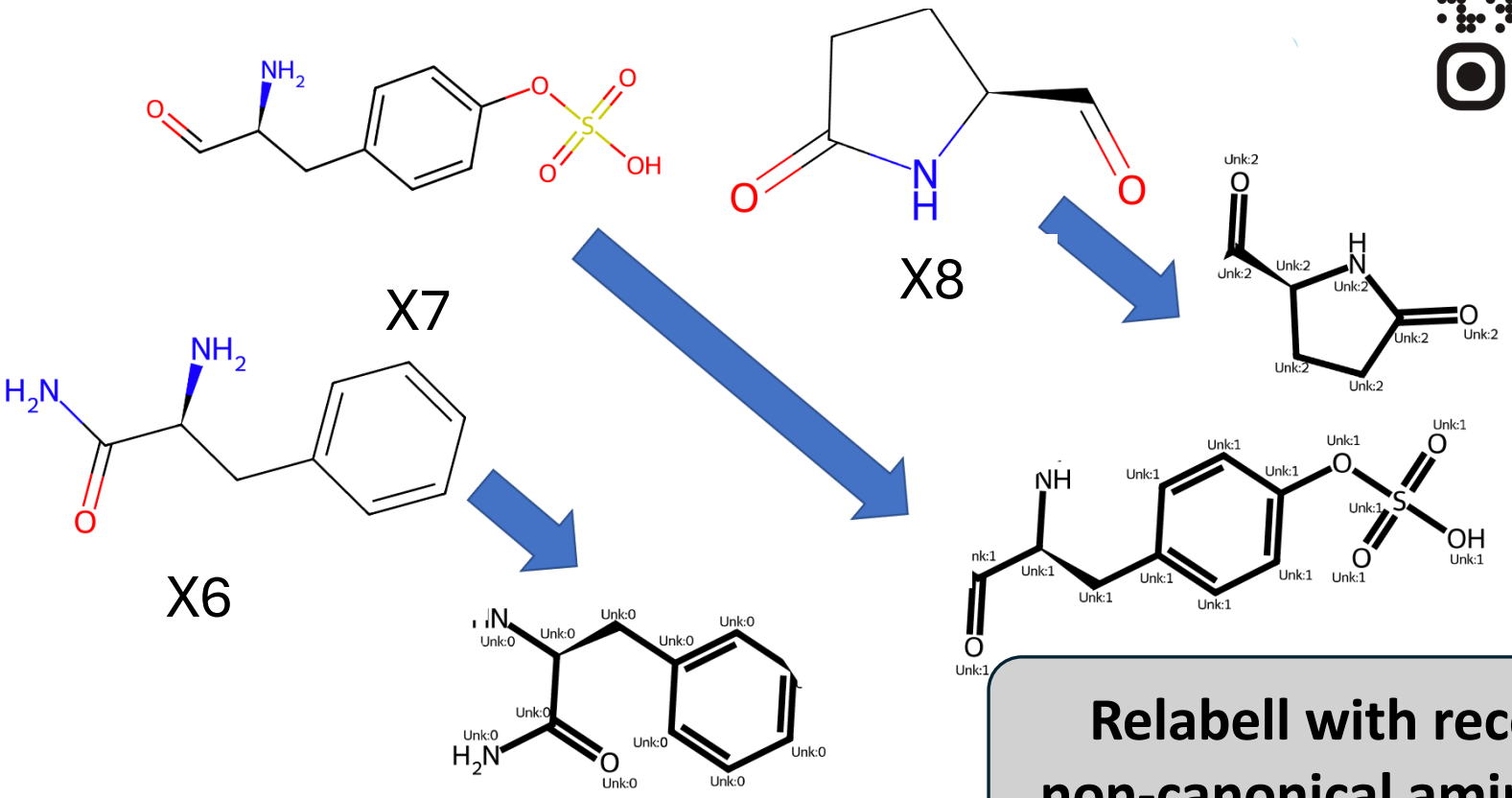
Repeat for all peptides

Monomerizer

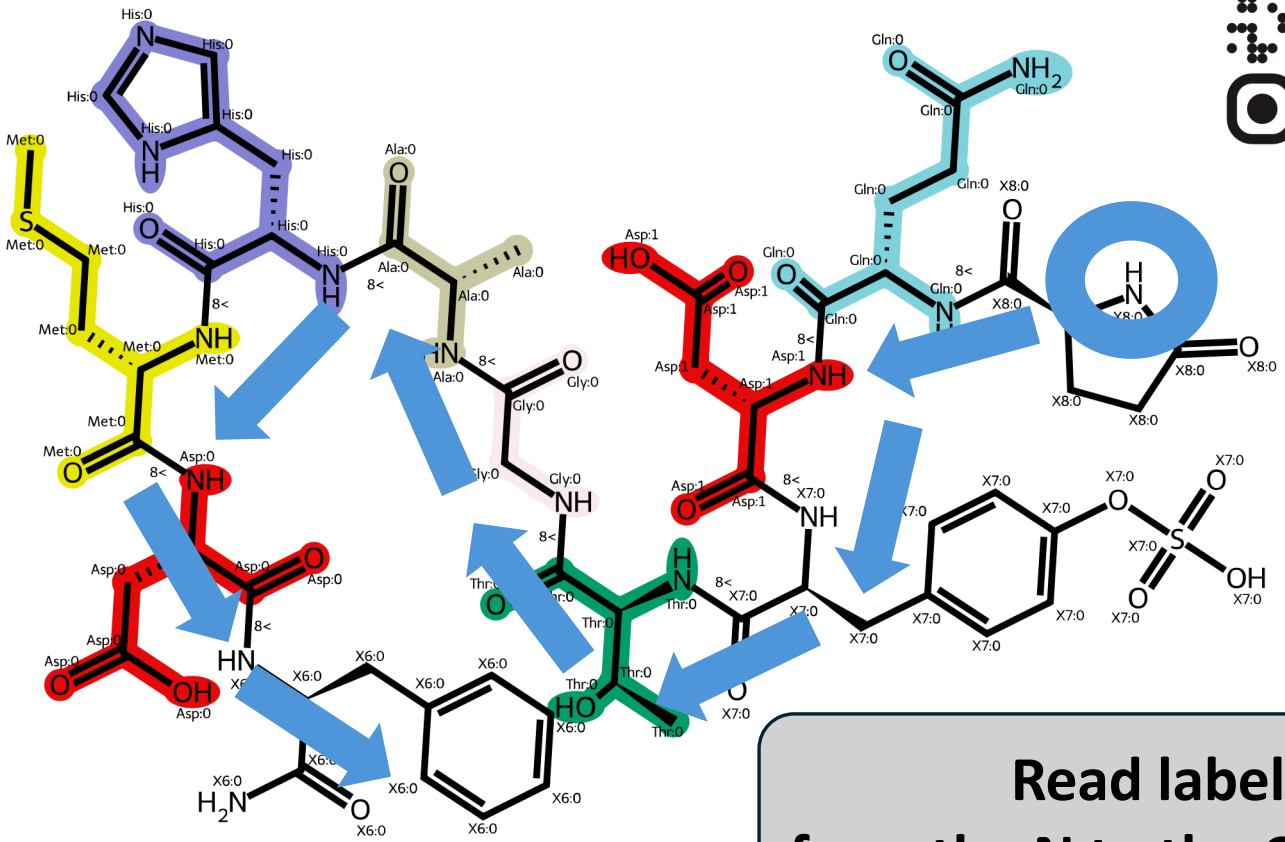


Name all the non-canonical amino acids

Monomerizer

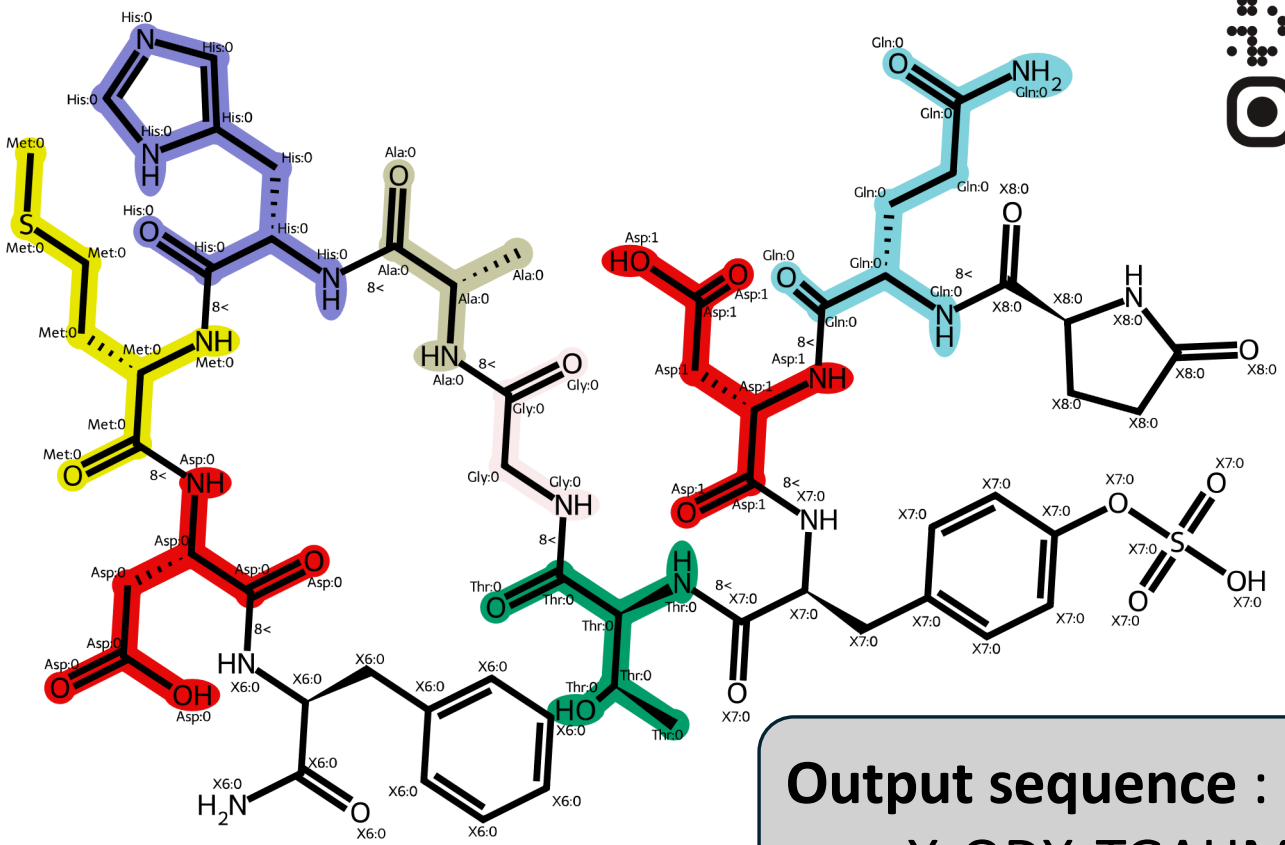


Monomerizer



Read labels
from the N to the C terminal

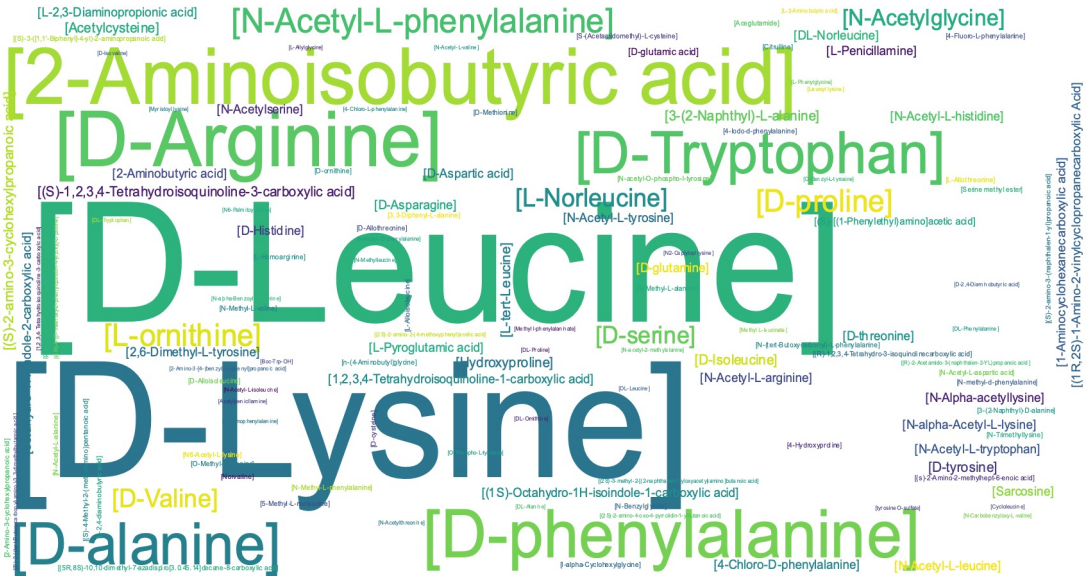
Monomerizer



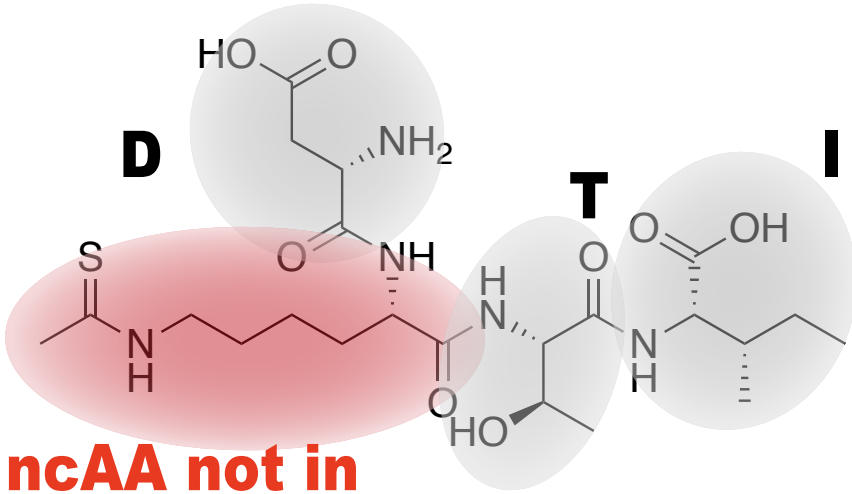
Output sequence :

$X_8QDX_7TGAHMDX_6$

> 17,000 non-canonical amino acids



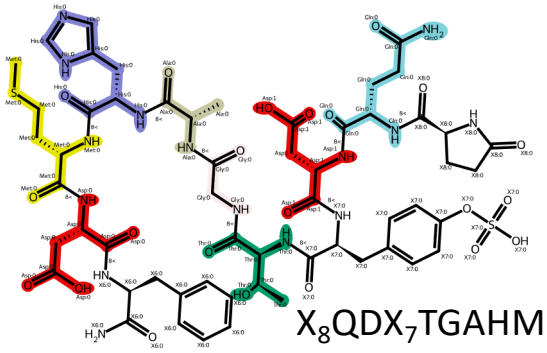
CHEMBL493160



ncAA not in Chembl/Pubchem

Overview

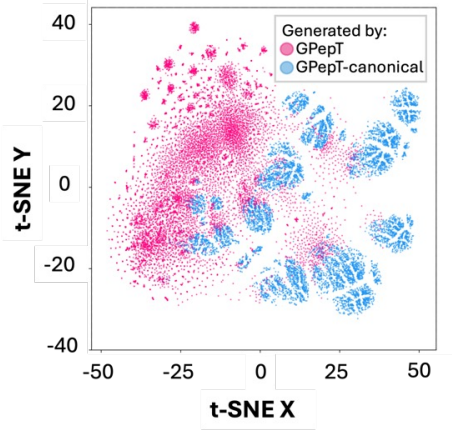
ChEMBL

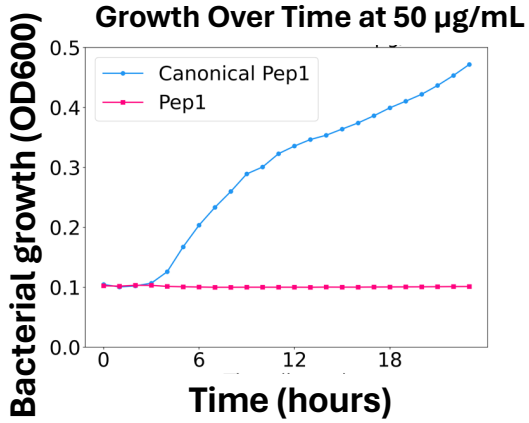


$X_8QDX_7TGAHMDX_6$

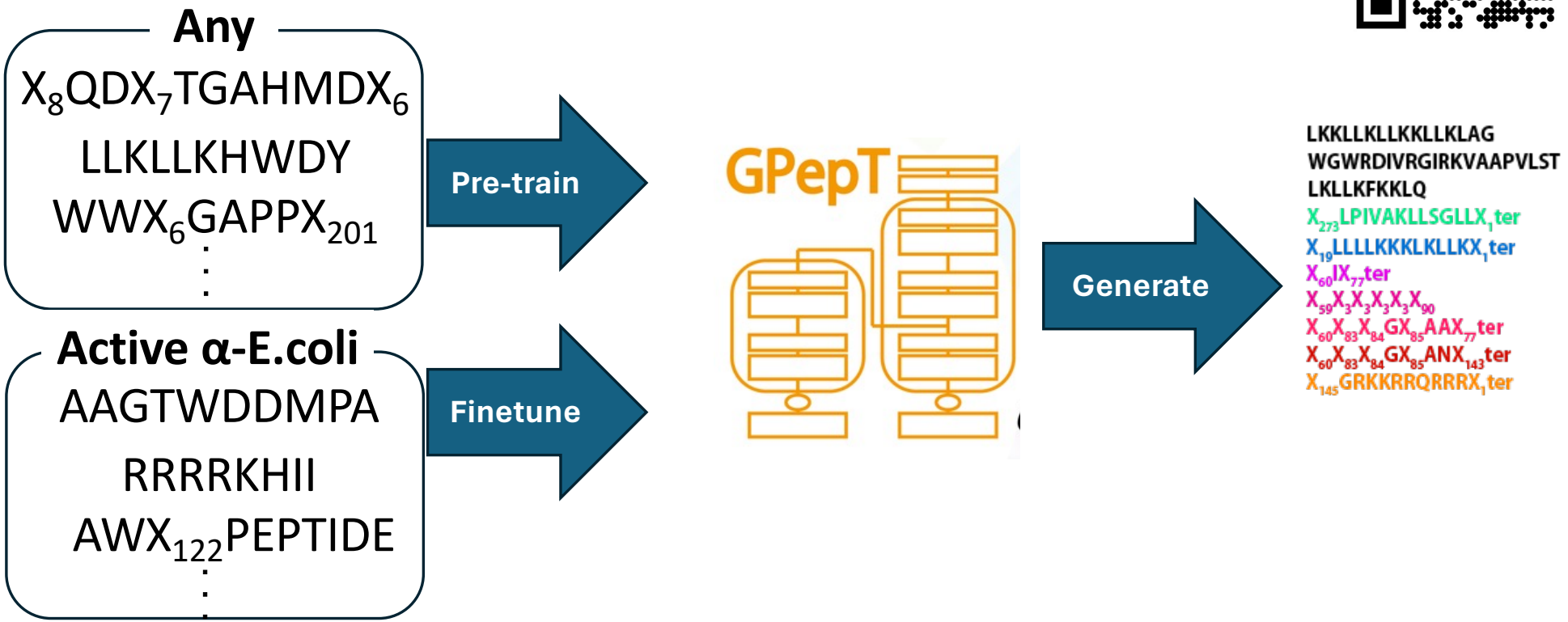
- 38,138 peptidomimetics
- 4,605 peptides

GPepT

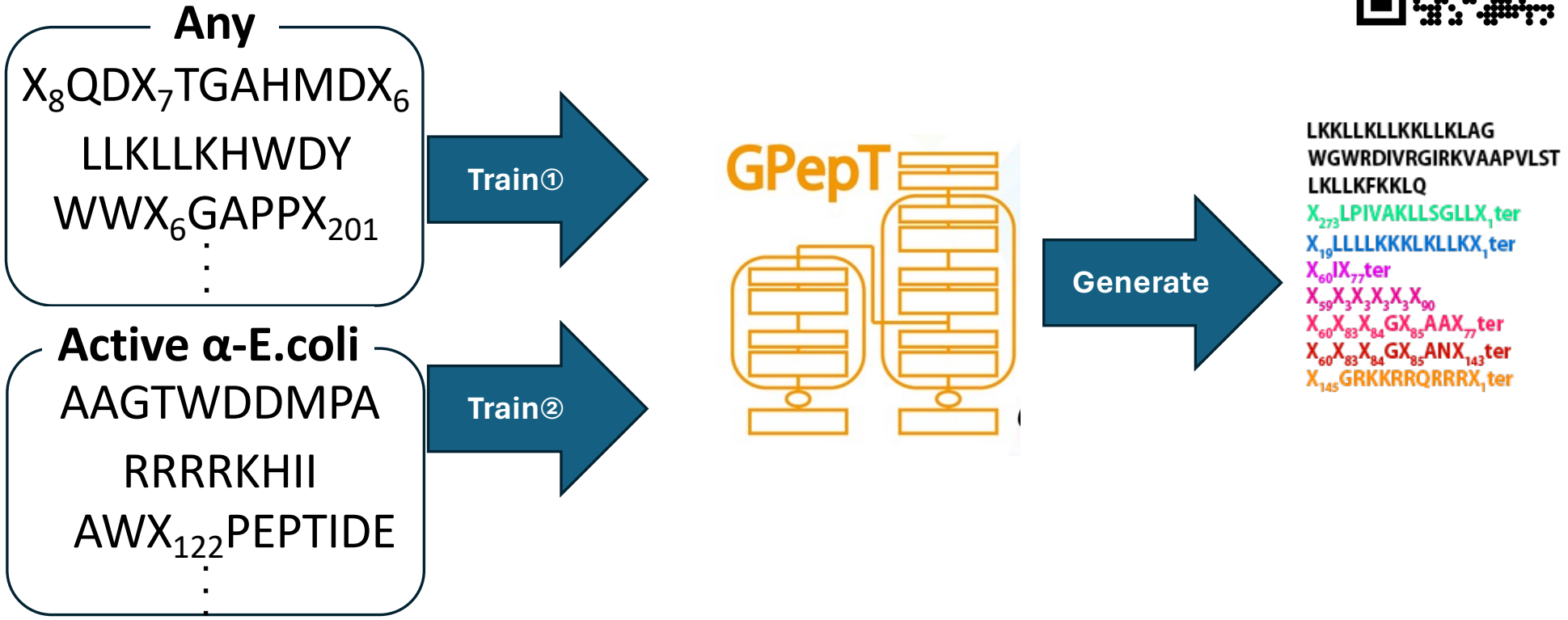




How to Make a Language Model Speak the Language of Peptides



How to Make a Language Model Speak the Language of Peptides



How to Make a Language Model Speak the Language of Peptides

1. Load the tokenizer, model, and config.

```
# Define the model path (e.g., "username/model-name")
model_repo = "openai-community/gpt2"

# Load the tokenizer, model, and config
tokenizer = AutoTokenizer.from_pretrained(model_repo)
model = AutoModelForCausalLM.from_pretrained(model_repo)
config = AutoConfig.from_pretrained(model_repo)
```

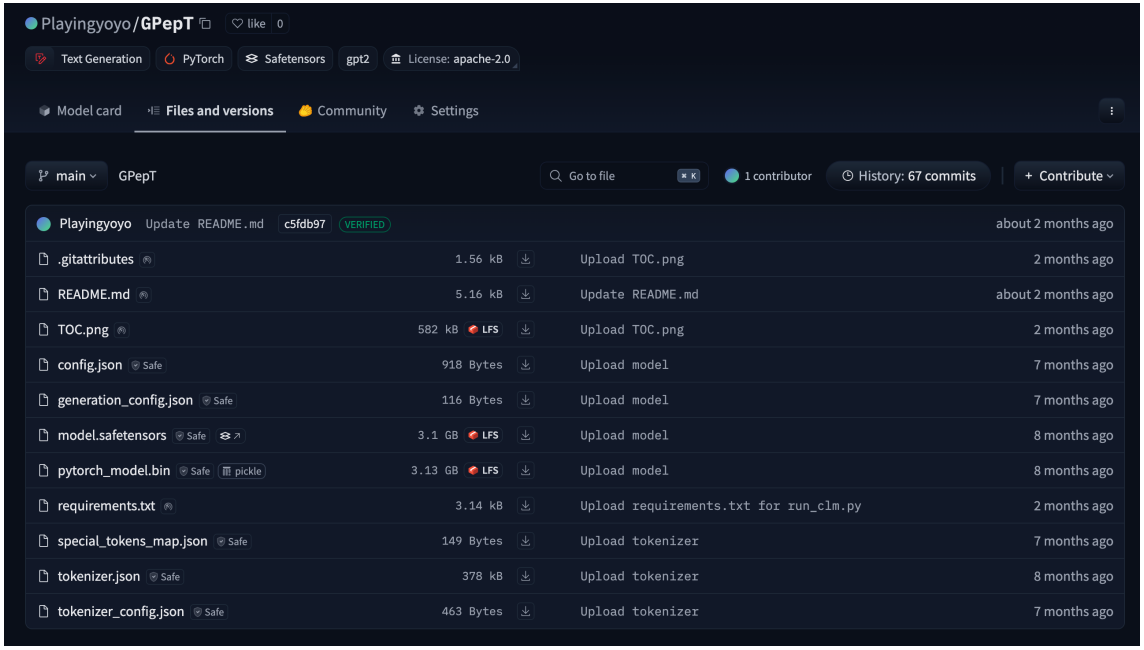
How to Make a Language Model Speak the Language of Peptides

2. Initialize model weights

```
model = AutoModelForCausalLM.from_config(config)
```

How to Make a Language Model Speak the Language of Peptides

3. Modify tokenizer




```
{
  "version": "1.0",
  "truncation": null,
  "padding": null,
  "added_tokens": [
    {
      "id": 0,
      "content": "<|endoftext|>",
      "single_word": false,
      "lstrip": false,
      "rstrip": false,
      "normalized": false,
      "special": true
    }
  ],
  "normalizer": null,
  "pre_tokenizer": {
    "type": "Split",
    "pattern": {
      "Regex": "(?=[A-Z])|(?=<\\|endoftext\\|>)"
    },
    "behavior": "Isolated",
    "invert": false
  },
  "post_processor": null,
  "decoder": null,
  "model": {
    "type": "WordLevel",
    "vocab": {
      "<|endoftext|>": 0,
      "[UNK]": 1,
      "X10000": 2,
      "X10001": 3,
      "X10002": 4,
      "X10003": 5,
```

How to Make a Language Model Speak the Language of Peptides

4. Adjust vocab size

```
# Load the tokenizer
tokenizer = AutoTokenizer.from_pretrained("Playingyoyo/GPepT")

# Resize the model's token embeddings to accommodate the new tokens
model.resize_token_embeddings(len(tokenizer))
```

How to Make a Language Model Speak the Language of Peptides

5. Pre-train

```
python run_clm.py --model_name_or_path Playingyoyo/GPepT \
                  --train_file path_to_train90.txt \
                  --validation_file path_to_val10.txt \
                  --tokenizer_name Playingyoyo/GPepT \
                  --do_train \
                  --do_eval \
                  --output_dir ./output \
                  --learning_rate 1e-5
```

No GPU required

How to Make a Language Model Speak the Language of Peptides

6. Finetune

```
python run_clm.py --model_name_or_path Playingyoyo/GPepT \
                  --train_file path_to_train90.txt \
                  --validation_file path_to_val10.txt \
                  --tokenizer_name Playingyoyo/GPepT \
                  --do_train \
                  --do_eval \
                  --output_dir ./output \
                  --learning_rate 1e-5
```

No GPU required

How to Make a Language Model Speak the Language of Peptides

7. Generate

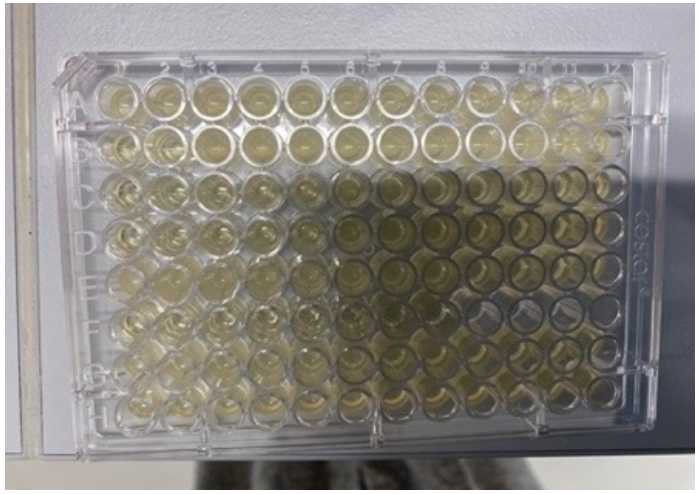
```
# Generate sequences (expressed in tokens, average ~4 amino acids per token)
sequences = GPepT("<|endoftext|>",
                  max_length=25,
                  do_sample=True,
                  top_k=950,
                  repetition_penalty=1.5,
                  num_return_sequences=5,
                  eos_token_id=0)

# Print generated sequences
for seq in sequences:
    print(seq['generated_text'])
```

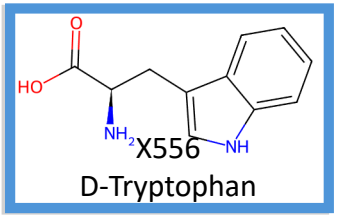
No GPU required

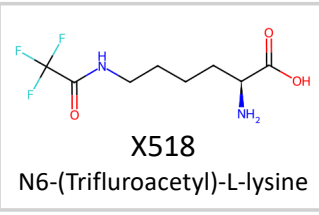
Experimental validation

MIC=50 μg/mL

Name	Sequence	Monomers
Pep1	X556WX556WWKZ0	X556=D-Tryptophan, Z0=Ammonia
Pep2	X556WWWWWWWWWWWWW	X556=D-Tryptophan
Pep3	LQKYYRRVRGGRX518F	X518=N6-(Trifluoroacetyl)-L-lysine
Pep4	QGRKQGRRX518	X518=N6-(Trifluoroacetyl)-L-lysine
Pep5	X3449WWWWWR	X3449=Citrulline

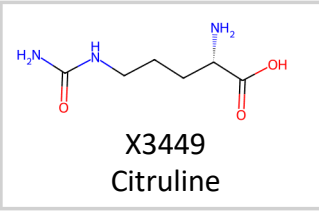




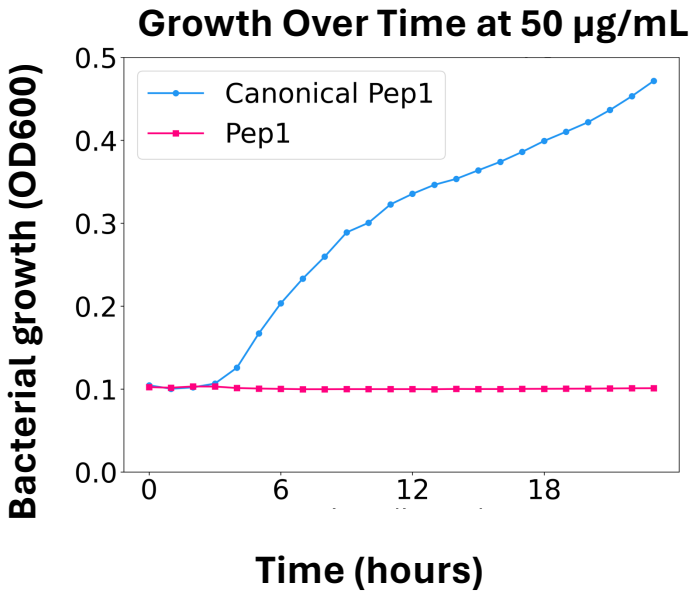


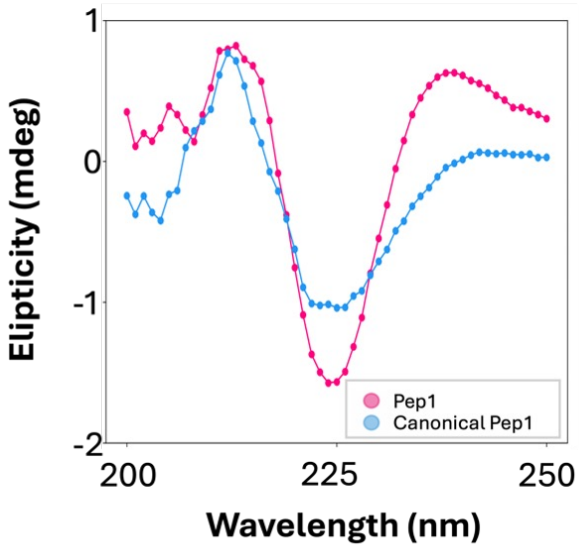
NH_3

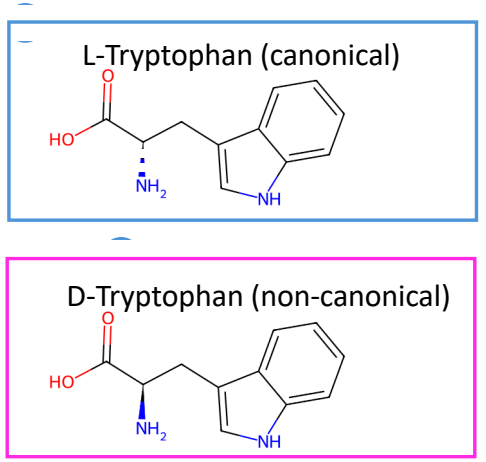
Z0
Ammonia



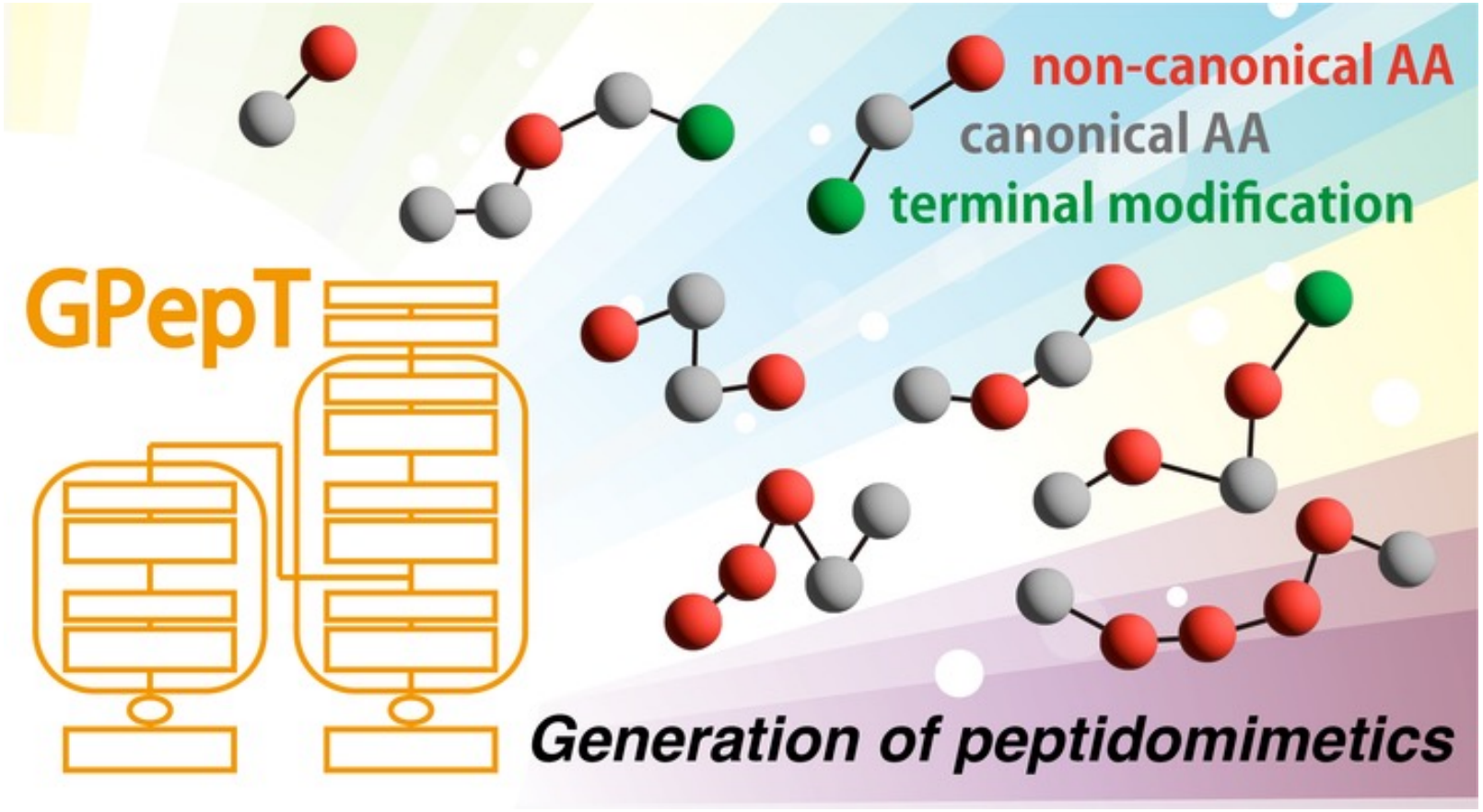
Experimental validation





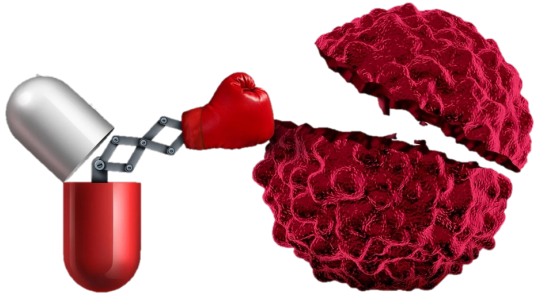


First Peptidomimetic Language Model

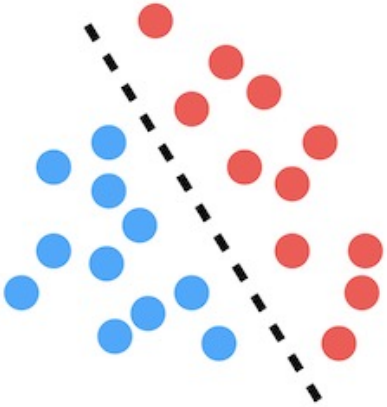


Future Prospects①

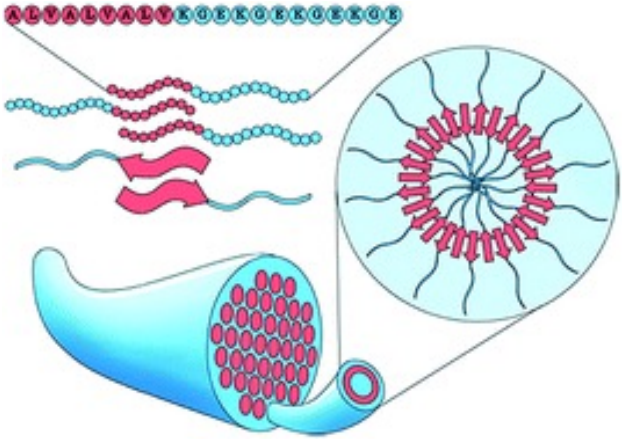
Anti-cancer peptidomimetics generation



Toxicity classifier



Fiber development



Future Prospects ②

ChatGPT Share

when
X7681 = Morpholine,
Z81 = Ethyl carbazate,
Z0 = Amide,
X1132 = 3,5-Diiodo-L-tyrosine,
X415 = 3,4-Diethoxyaniline and
X2326 = Glycine, N-(3-iodobenzoyl)-,

which of the peptidomimetics sequences below are most likely to be active against E. coli ?

X7681VZ81
X1132RZ0
X415X2326Z0

1. X7681VZ81

- **X7681 = Morpholine**: A small, polar, heterocyclic compound. Not inherently antimicrobial, but found in drug scaffolds.
- **V = Valine**: Hydrophobic amino acid; may aid membrane interaction.
- **Z81 = Ethyl carbazate**: Can form hydrogen bonds; may enhance stability and interaction.
  → Moderate polarity with potential hydrogen bonding and slight hydrophobicity. May cross membranes, but lacks strong antimicrobial motifs.
  ⚖️ **Verdict**: **Possibly weak activity**.

✅ Conclusion:

Most likely to be active against *E. coli*:
→ **X1132RZ0** (Sequence #2)
This contains a **halogenated aromatic** + **Arginine**, a classic combination for antimicrobial activity via membrane disruption.

First Peptidomimetic Language Model