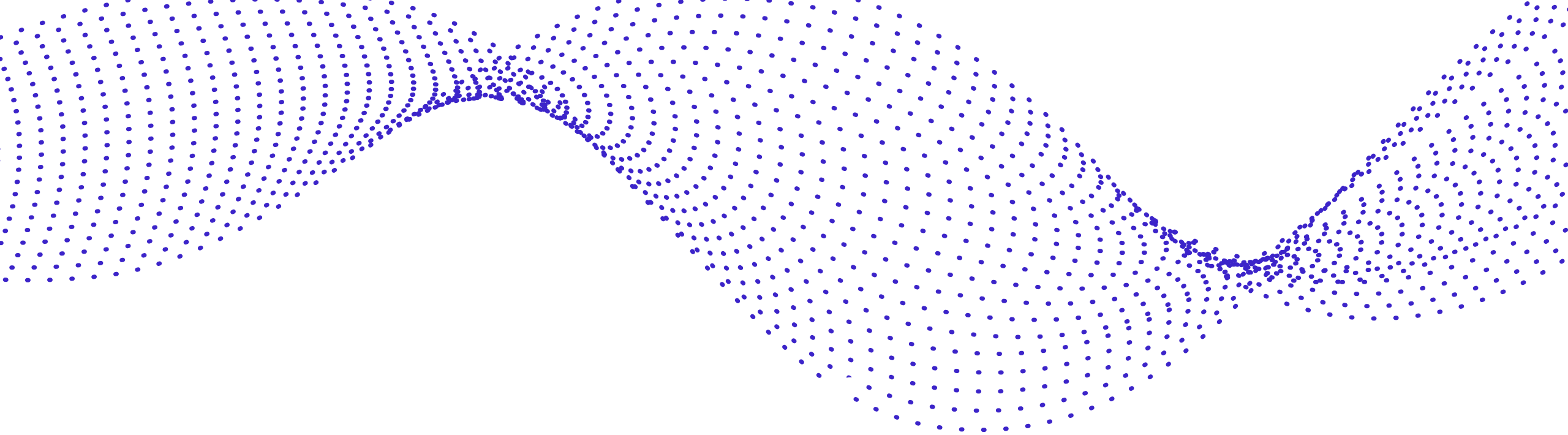
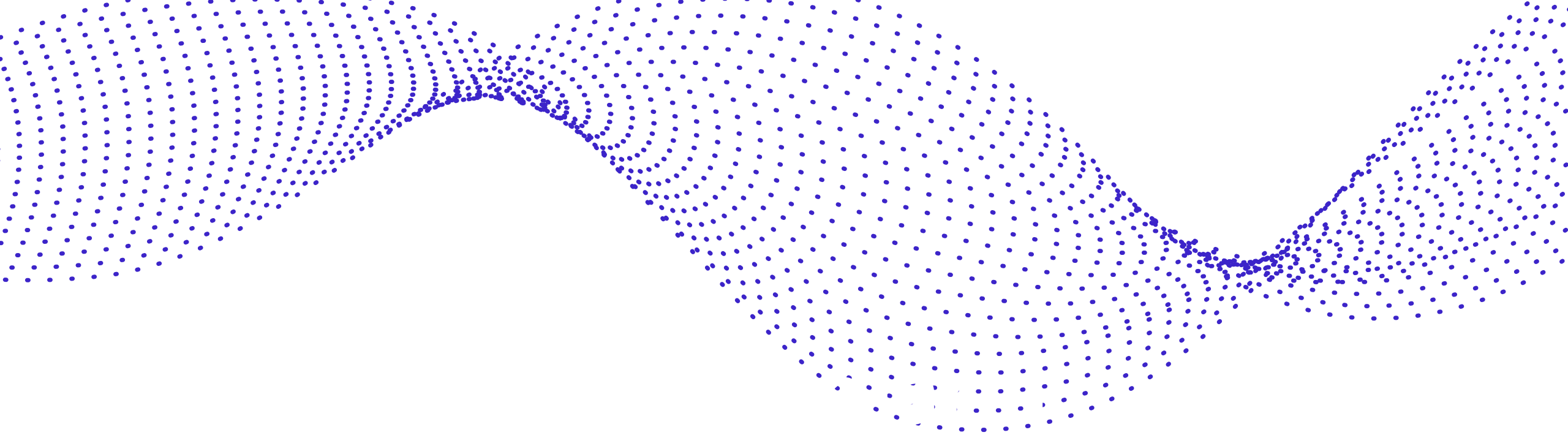


Moscow Institute of Physics and Technology

Egor Marin, PhD student

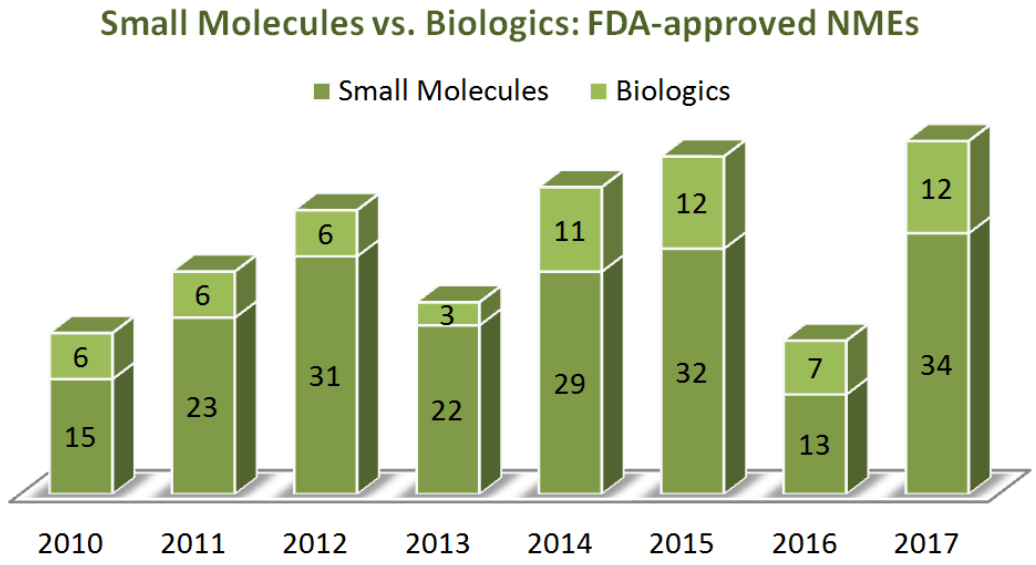


Moscow Institute of Physics and Technology

Egor Marin, PhD student

# Background

- Small molecules – still most popular drugs
- Small-molecule : protein interaction is hard to model
- Drug-discovery problem: for a given protein target, predict, which molecules will bind to it
- Side-effects problem: for a given approved drug, predict, which off-targets it will engage



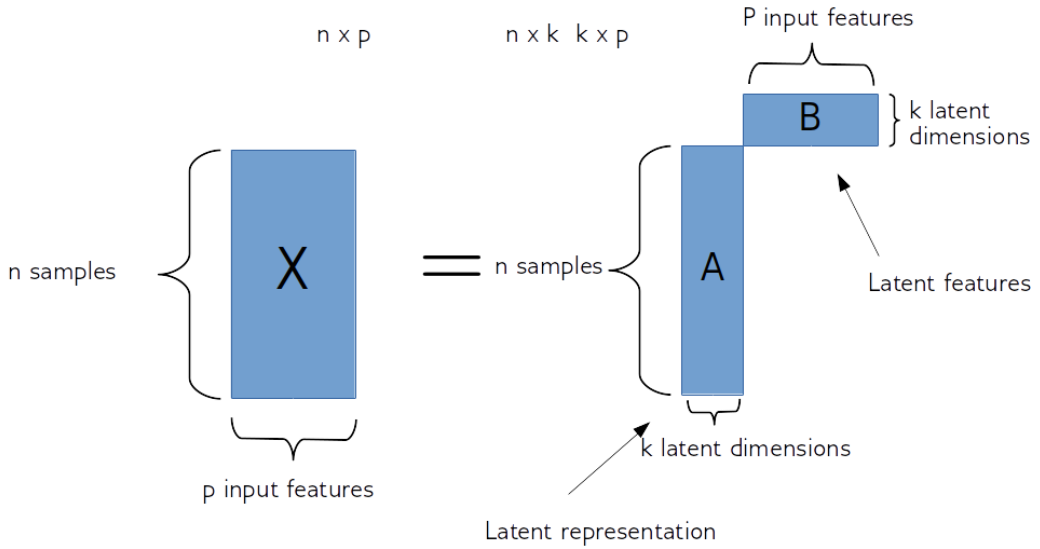

https://www.biopharmatrend.com/post/67-will-small-molecules-sustain-pharmaceutical-race-with-biologics/

# Challenge: description

- explore protein-ligand interaction matrix
- extract the protein and ligand fingerprints using matrix decomposition
- predict novel drug-target interactions, i.e. fill in the missing values in the matrix

$$X = AB$$

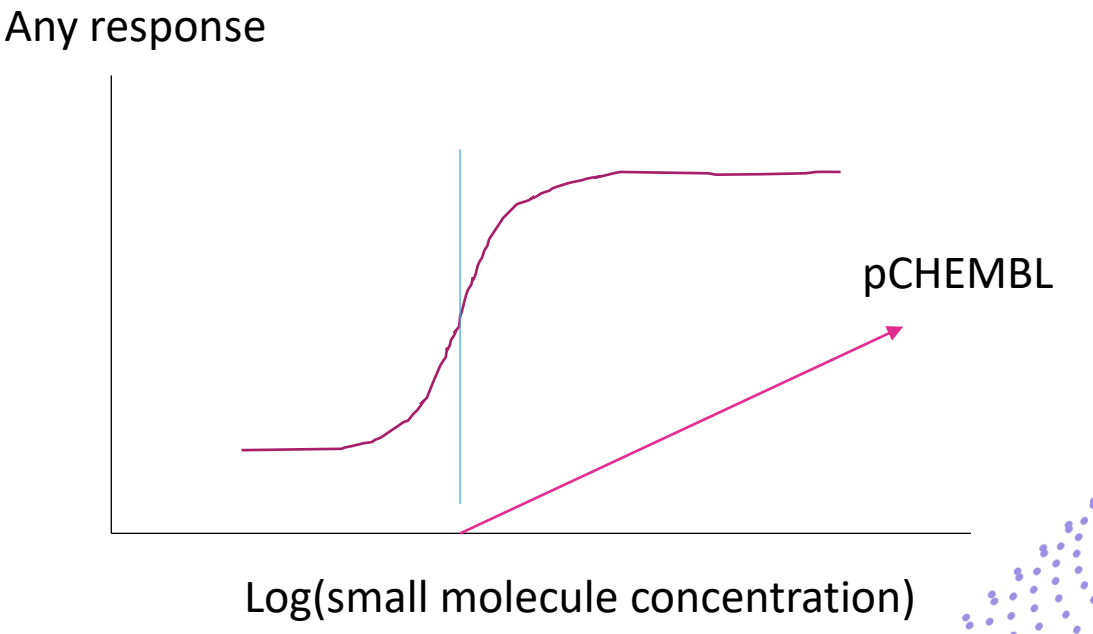
n x p  n x k  k x p




(protein, ligand) ≃ (protein, feature) x (feature, ligand)

# Data: physical meaning

- What does “interaction” mean?
  - Enzyme modifies a small-molecule
  - Transporter moves molecule across the membrane
  - Receptor recognizes the molecule
  - …
- Most of the interactions – sigma-curve response
- We’ll use pCHEMBL value: “pChEMBL is defined as: -Log(molar IC50, XC50, EC50, AC50, Ki, Kd or Potency)” (source)



# Data: physical meaning

- What does “interaction” mean?
  - Enzyme modifies a small-molecule
  - Transporter moves molecule across the membrane
  - Receptor recognizes the molecule
  - …
- Most of the interactions – sigma-curve response
- We’ll use pCHEMBL value: “pChEMBL is defined as: -Log(molar IC50, XC50, EC50, AC50, Ki, Kd or Potency)” (source)

```
:
%%time

import numpy as np
from collections import defaultdict

cols = ['compound_chembl_id','target_chembl_id','pchembl_value']
d = defaultdict(lambda: [])
N = 2_000_000

for _, cmpd, target, pchembl in df[cols].sample(n=N).itertuples():
    if not np.isnan(pchembl):
        d[(cmpd, target)].append(pchembl)

CPU times: user 4.16 s, sys: 40.7 ms, total: 4.2 s
Wall time: 4.2 s
```

# Data: source

- drug-target interactions: Chembl
  - There are also GOSTAR, Pubchem, Reaxys, Pharos, Binding DB – we'll stick to Chembl
- Database schema: https://www.ebi.ac.uk/chembl/db_schema
- Examples by Chembl: https://chembl.gitbook.io/chembl-interface-documentation/frequently-asked-questions/schema-questions-and-sql-examples
- Simple script that writes a smaller csv: https://github.com/marinegor/agnostic_fp/blob/main/get_bioactivities.sql

# Data: proteins & ligands

Matrix is super large & super sparse. Ideas how to deal with it?

- Protein-wise:
  - Limit yourself to human proteins
  - Limit yourself to Chembl's selection of targets
    - here: https://jcheminf.biomedcentral.com/track/pdf/10.1186/s13321-018-0325-4.pdf, Data availability and materials – 353 targets in xlsx table
    - "Retrieve target ChEMBL_ID, target_name, target_type, protein accessions and sequences for all protein targets" – in chembl interface documentation
- Ligand-wise:
  - Limit molecules to only approved drugs (could be too few – ~20,000)
  - Limit molecules to only those that have `more than threshold` number of measured interactions
  - Limit molecules to rule-of-5 ("most likely do be drugs") – compound_properties → num_lipninski_ro5_violations in schema

# Decomposition: things to try?

- SVD/TruncatedSVD
  - pros
    - Implemented in sklearn
    - Fast
  - cons
    - non-physical – can have negative values, whereas iteraction / interaction constant is probably positive
    - (could be) unstable
- NMF (non-negative matrix factorization)
  - Pros
    - Only positive values → better interpretability
    - Implemented in sklearn
  - Cons
    - Slower than SVD/TruncatedSVD
- <a non-linear method>
  - Pros
    - Can have better performance
    - Can be faster if trained on GPU
  - Cons
    - Probably not implemented in sklearn
    - Can be unable to work with sparse matrices

# Challenge: roadmap

- Decide which data you'll use
  - Which proteins?
  - Which ligands?
  - Binarize/not binarize interaction constants?
- Decompositions: look at the timings for your data and decide, which algorithm family you'll use
  - Can start them running & look at the other problems in the meantime
- Benchmarking
  - For your data, build class-imbalance-aware random baseline
  - For your decompositions, see whether a many-features decomposition is better than 1-feature

# Challenge: roadmap-if-you're-done

- Whether a metrics on sequences, imposed by them, is similar to alignment-based sequence similarity itself? Same for ligands.
- (protein, ligand) ≃ (protein, feature) x (feature, ligand)
  - "feature" from protein point of view: sensitivities to chemotypes
  - "feature" from ligand point of view: presence of a chemotype in ligand
  - Can agnostic fingerprints "see" particular target types?
    - GPCRs, kinases, …
    - Lipid-sensing molecules vs. soluble molecules
    - …
- If you cluster proteins by their fingerprints, how can you interpret the clusters?
- After you've built a interaction prediction model, can you add confidence level to it?
  - pyMCA, pyro – you name it

# Communication

- Slack: Egor Marin -->
- Unavailable starting 10 pm
- Definitely available: Saturday before 11 am & Saturday 1-3 pm
- Unavailable: 11-12 am & 3-5 pm Saturday

Rest of the time – can be slow to respond