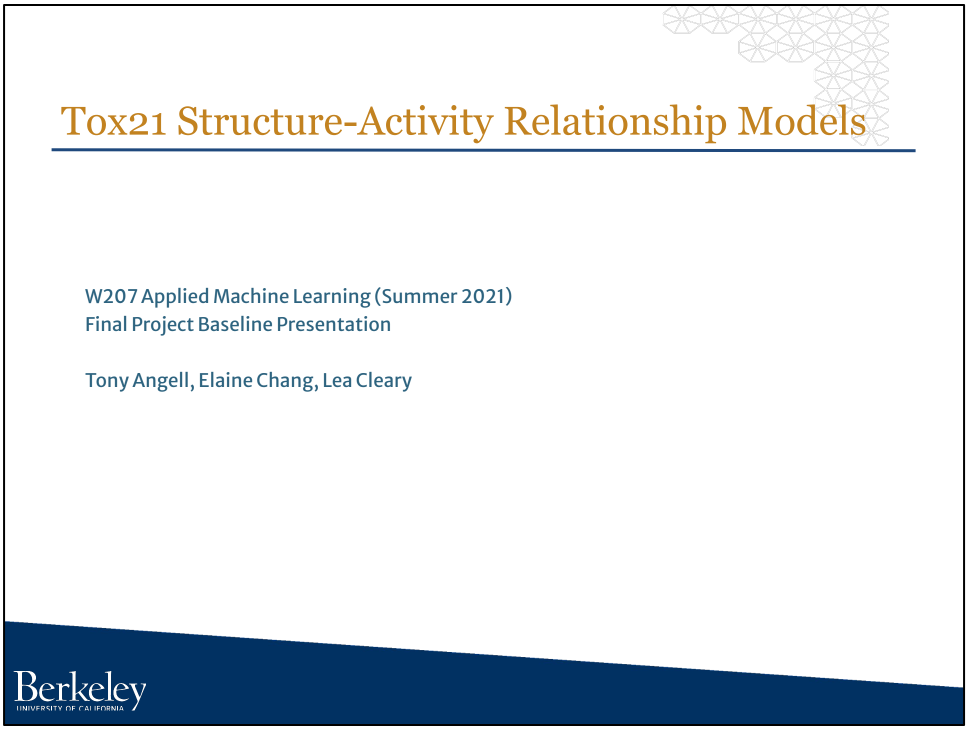


Tox21 Structure-Activity Relationship Models

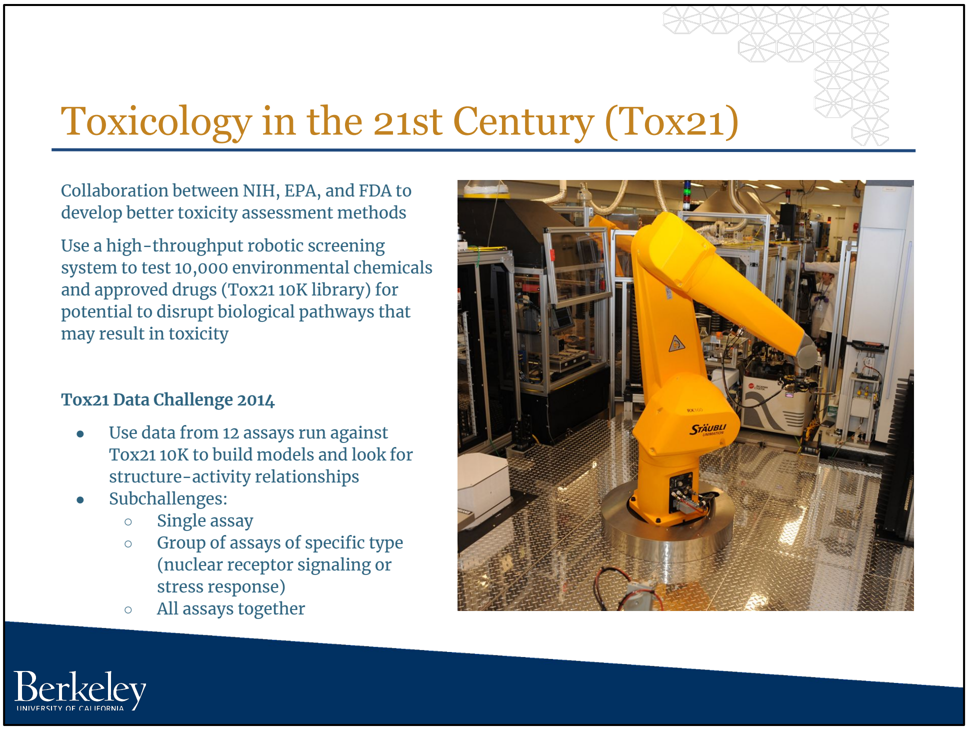
W207 Applied Machine Learning (Summer 2021)
Final Project Baseline Presentation

Tony Angell, Elaine Chang, Lea Cleary

Berkeley
UNIVERSITY OF CALIFORNIA

Our final project aims to develop structure-activity relationship models for toxicology.

Most people are exposed to many different chemicals during their lifetimes through sources including food, household cleaning products, and medicines. To protect humans from potentially harmful effects, these chemicals must pass reliable tests for adverse effects. A compound's effects on human health are assessed by a large number of time-consuming and cost-intensive *in vivo* or *in vitro* experiments. The classic experimental tool of toxicology relies on animal tests, which has some problematic ethical concerns. The aim of the Tox21 initiative is to develop more efficient and less time-consuming approaches to predicting ow chemicals affect human health.



Toxicology in the 21st Century (Tox21)

Collaboration between NIH, EPA, and FDA to develop better toxicity assessment methods

Use a high-throughput robotic screening system to test 10,000 environmental chemicals and approved drugs (Tox21 10K library) for potential to disrupt biological pathways that may result in toxicity

Tox21 Data Challenge 2014

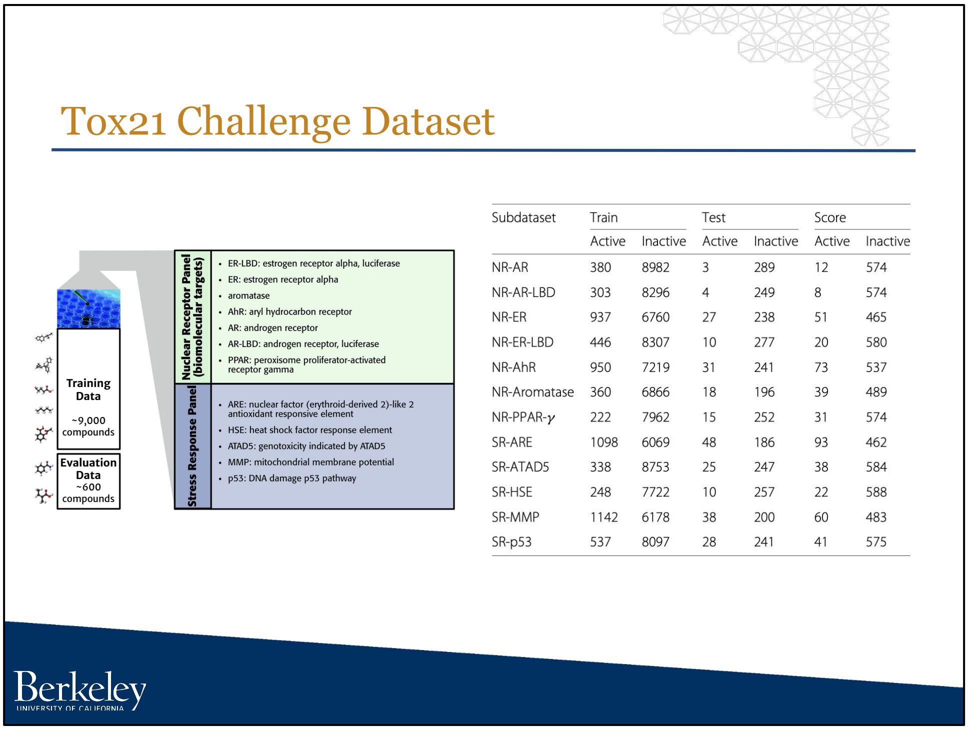
- Use data from 12 assays run against Tox21 10K to build models and look for structure-activity relationships
- Subchallenges:
  - Single assay
  - Group of assays of specific type (nuclear receptor signaling or stress response)
  - All assays together

Berkeley UNIVERSITY OF CALIFORNIA

The Toxicology in the 21st Century (Tox21) program is a federal collaboration between the NIH, EPA, and FDA. Using the their high-throughput robotic screening system, they tested a collection of over 10,000 environmental chemicals and approved drugs for their potential to disrupt processes in the human body that may lead to negative health effects. This collection of data is known as the Tox21 10K library.

Computational models are a potential alternative to *in vivo* and *in vitro* experiments, but they usually suffer from insufficient accuracy and are not as reliable as biological experiments. Thus, in 2014, the collaboration announced the Tox21 Data Challenge to "crowdsource" data analysis by independent researchers to reveal how well they can predict the toxicity of compounds using only chemical structure data.

The challenge uses data from 12 assays run against the Tox 21 10K library to build models. Subchallenges included predicting individual assays, groups of assays of a particular type (either nuclear receptor signaling or stress response), or the grand challenge of all the assays together. Our group will be using this dataset for our project, either taking on a subchallenge of evaluating one assay or one type of assays.



Tox21 Challenge Dataset

Training
Data
~9,000
compounds
Evaluation
Data
~600
compounds

Nuclear Receptor Panel (biomolecular targets)

- ER-LBD: estrogen receptor alpha, luciferase
- ER: estrogen receptor alpha
- aromatase
- AhR: aryl hydrocarbon receptor
- AR: androgen receptor
- AR-LBD: androgen receptor, luciferase
- PPAR: peroxisome proliferator-activated receptor gamma

Stress Response Panel

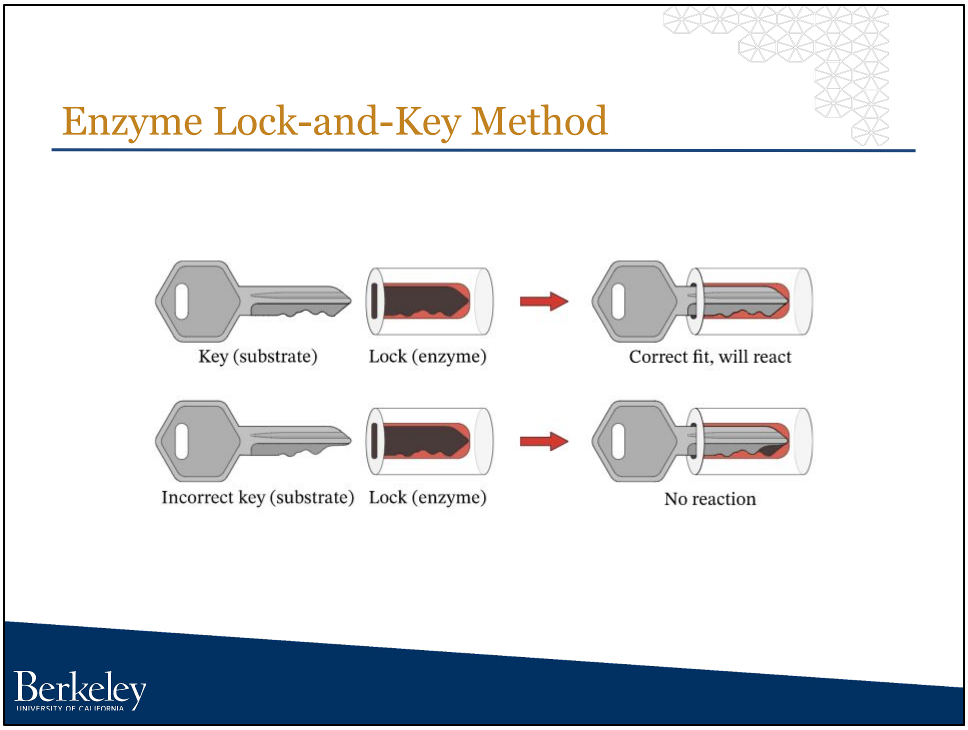
- ARE: nuclear factor (erythroid-derived 2)-like 2 antioxidant responsive element
- HSE: heat shock factor response element
- ATAD5: genotoxicity indicated by ATAD5
- MMP: mitochondrial membrane potential
- p53: DNA damage p53 pathway

Subdataset	Train		Test		Score	
	Active	Inactive	Active	Inactive	Active	Inactive
NR-AR	380	8982	3	289	12	574
NR-AR-LBD	303	8296	4	249	8	574
NR-ER	937	6760	27	238	51	465
NR-ER-LBD	446	8307	10	277	20	580
NR-AhR	950	7219	31	241	73	537
NR-Aromatase	360	6866	18	196	39	489
NR-PPAR-γ	222	7962	15	252	31	574
SR-ARE	1098	6069	48	186	93	462
SR-ATAD5	338	8753	25	247	38	584
SR-HSE	248	7722	10	257	22	588
SR-MMP	1142	6178	38	200	60	483
SR-p53	537	8097	28	241	41	575

Berkeley
UNIVERSITY OF CALIFORNIA

The Tox21 challenge dataset is divided into training and test sets, along with an evaluation set that was used to score the entries for the challenge. The datasets consist of chemical structure data in two different file formats, a chemical compound ID, and an indicator of whether or not the compound was active for the given assay.

So at the heart of this problem is really just a binary classification problem. Since the dataset is highly imbalanced toward inactive compounds, we are considering trying some minority over-sampling techniques to try to improve our classification. However, the real challenge in this dataset is how to create features from the chemical structure data that would represent the molecules as best as possible. In fact, coming up with features can be a whole machine learning problem in itself.

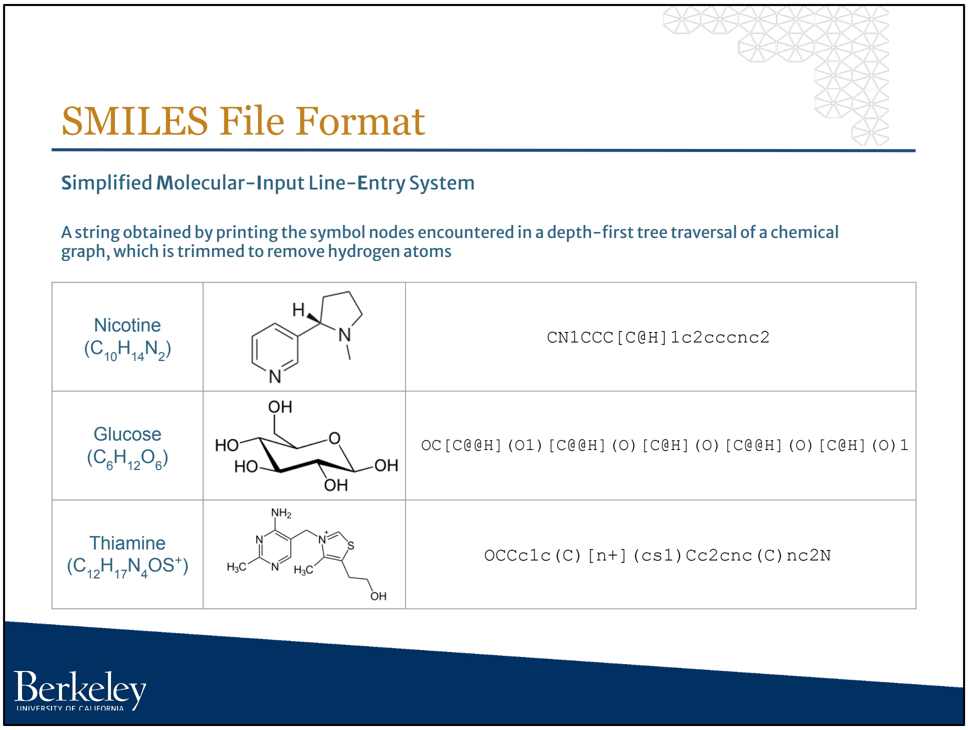


Enzyme Lock-and-Key Method

Before I get any further, let me just go over some enzyme chemistry 101 for the non-biochemists in the room. Biochemical pathways are really a series of reactions mediated by enzymes where the product of one reaction is used as the substrate in the next. Enzymes are high specific proteins that must bind to a specific substrate before they can catalyze a chemical reaction.

One model to explain how enzymes work is the lock-and-key model. Like a key goes into a lock, only the correct size and shape of substrate would fit into the active site of an enzyme. Toxic substances are often structurally similar to substrates, so they can also bind to the active site of the enzyme. However, like the incorrect key in the picture, they don't produce the desired reaction, so they can disrupt biochemical pathways by deactivating certain enzymes.

In vitro assays like those used in Tox21 are, in essence, trying out thousands of chemical "keys" to see which will fit into particular enzyme "locks." If the key fits, then the substance is deemed active. Because of this lock-and-key method, it is very important to pick features that represent the chemical structure of each substance well in order to build an accurate model.



SMILES File Format

Simplified Molecular-Input Line-Entry System

A string obtained by printing the symbol nodes encountered in a depth-first tree traversal of a chemical graph, which is trimmed to remove hydrogen atoms

Nicotine ($C_{10}H_{14}N_2$)		CN1CCC[C@H]1c2cccnc2
Glucose ($C_6H_{12}O_6$)		OC[C@@H](O1)[C@@H](O)[C@H](O)[C@@H](O)[C@H](O)1
Thiamine ($C_{12}H_{17}N_4OS^+$)		OCCc1c(C)[n+](cs1)Cc2cnc(C)nc2N

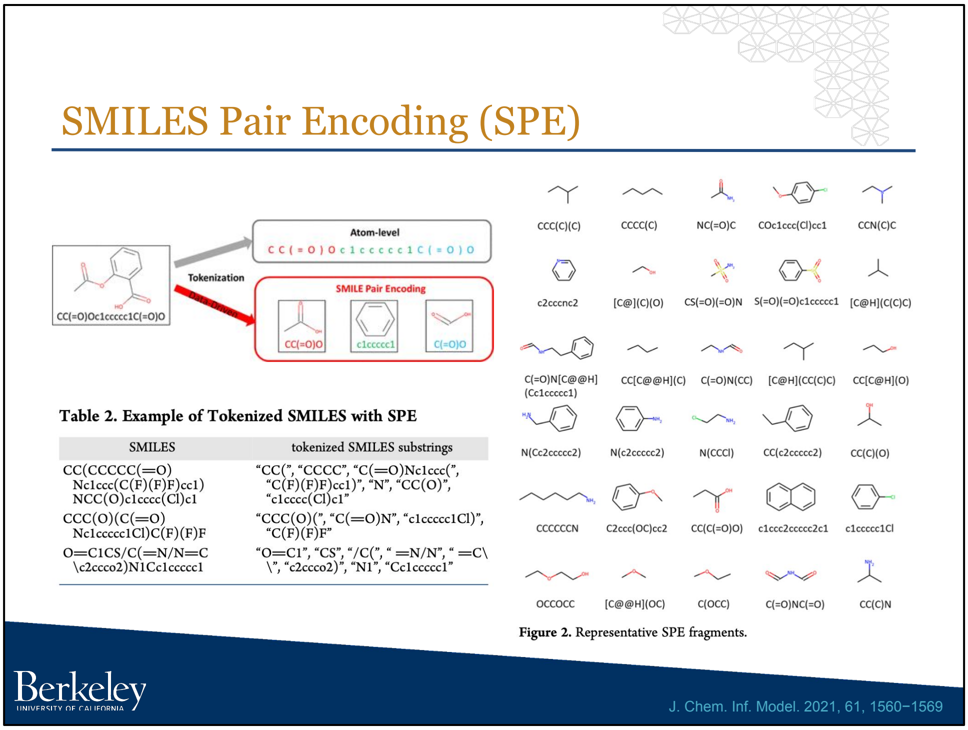
Berkeley
UNIVERSITY OF CALIFORNIA

The Tox21 dataset comes in two different file formats for the chemical structure data. For our project, we will be focusing on SMILES file format, since the other format requires more specialized programs to visualize.

SMILES stands for Simplified Molecular-Input Line-Entry System and is a string obtained by printing the symbol nodes encountered in a depth-first tree traversal of a chemical graph. It is trimmed to remove hydrogen atoms.

Because of its string format, creating features from SMILES strings is akin to vectorizing text documents. In fact, some of the most simplistic feature generating techniques for SMILES strings mirror using CountVectorizer with characters as tokens. In fact the standard approach for SMILES tokenization is to simply break the string character by character. However, this has numerous issues, such as the fact that single atoms can sometimes be represented by multiple characters, since the element they correspond to has a two-character symbol. The next level is atom-level tokenization, but this doesn't really capture how the atoms are connected. K-mers or n-grams can also be used, but this can suffer from the so-called out-of-vocabulary issue that we've seen in text analysis, where models basically ignore k-mers that they have never seen before.

For our project, we plan to explore some more advanced vectorization techniques.



SMILES Pair Encoding (SPE)

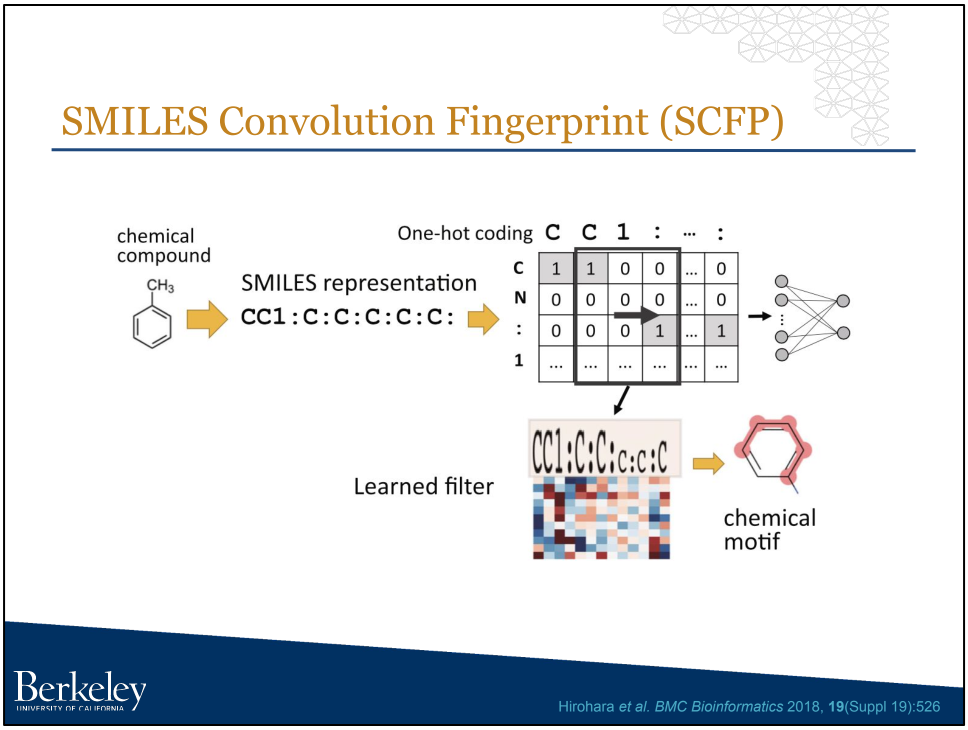
Table 2. Example of Tokenized SMILES with SPE

SMILES	tokenized SMILES substrings
CC(CCCCC(=O)Nc1ccc(C(F)(F)F)cc1)NCC(O)c1cccc(Cl)c1	“CC(”, “CCCC”, “C(=O)Nc1ccc(”, “C(F)(F)F)cc1)”, “N”, “CC(O)”, “c1cccc(Cl)c1”
CCC(O)(C(=O)Nc1ccccc1Cl)C(F)(F)F	“CCC(O)(”, “C(=O)N”, “c1ccccc1Cl)”, “C(F)(F)F”
O=C1CS/C(=N/N=C\c2ccco2)N1Cc1ccccc1	“O=C1”, “CS”, “/C(”, “ =N/N”, “ =C\\”, “c2ccco2)”, “N1”, “Cc1ccccc1”

Figure 2. Representative SPE fragments.

J. Chem. Inf. Model. 2021, 61, 1560−1569

One example of an advanced technique we are looking into is SPE or SMILES Pair Encoding. The idea behind SPE is to use a large existing dataset, like the ChEMBL dataset with millions of compounds in SMILES format, to create a vocabulary of molecular fragments that make sense. This method first tokenizes the ChEMBL dataset at the atom level and initializes the vocabulary with all the unique tokens. Then it will iteratively count the occurrence of all token pairs, merge the most frequently occuring ones, and add the new tokens to the vocabulary. The iteration will stop depending on how two hyperparameters are set: the maximum vocabulary size or when no pairs of tokens affords a frequency larger than a given threshold.



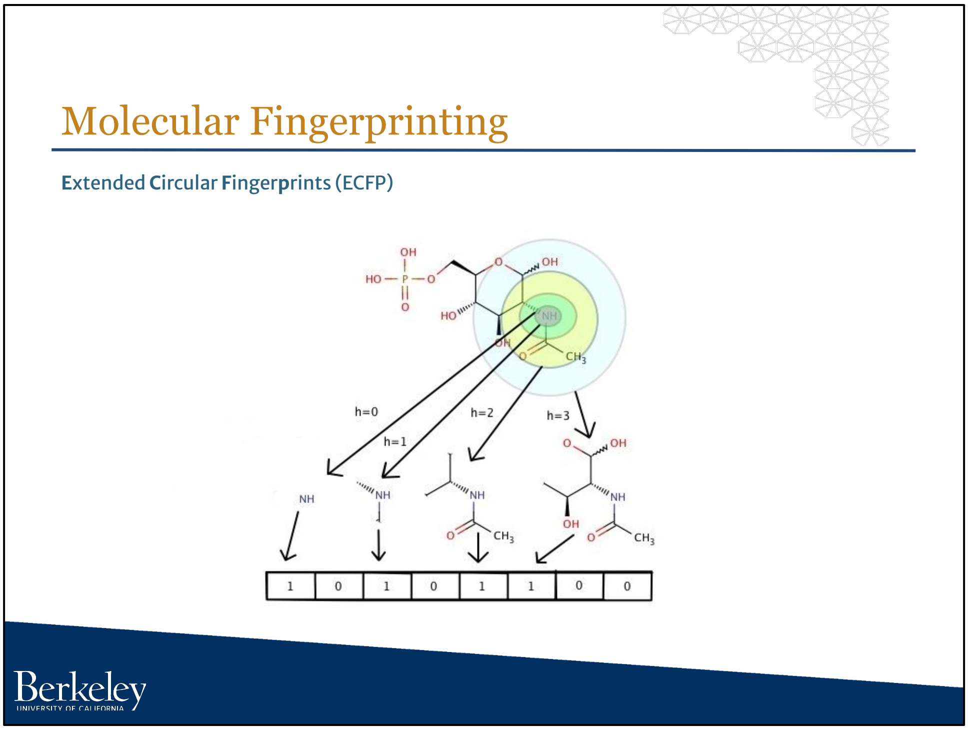
SMILES Convolution Fingerprint (SCFP)

	C	C	1	:	...	:
C	1	1	0	0	...	0
N	0	0	0	0	...	0
:	0	0	0	1	...	1
1	...	...	...	...	...	...

Hirohara *et al. BMC Bioinformatics* 2018, **19**(Suppl 19):526

Another technique we are looking into is running a neural network algorithm to produce logical features. In fact, this is one of the techniques used by the winners of the Tox21 grand challenge. This particular example uses a convolutional neural network or CNN. First, they represent a SMILES string as a distributed feature matrix, then apply a CNN to the matrix in a way that is similar to the application of CNNs to image data. This transforms the SMILES feature matrix into a low-dimensional feature vector they are calling the SMILES convolution finger print (SCFP).

Unfortunately we do not have much to show in terms of code yet, since our project required a lot of paper-reading and identifying packages that can work with chemical structure data. However, our plan is to pick one assay or type of assays and explore different vectorization and classification techniques to determine which ones are most effective. We also plan to visualize and explore our model results to try to explain why certain vectorizations and models are more effective than others.



Molecular Fingerprinting

Extended Circular Fingerprints (ECFP)

1	0	1	0	1	1	0	0

From https://medium.com/atomwise/machine-learning-for-drug-discovery-in-a-nutshell-part-ii-24f90d5963d9:

A common way of mapping variably structures molecules into a fixed-size descriptor vector is “fingerprinting”. Multiple such methods have been developed. Some are based on expert-derived features (e.g., number of specific types of bonds, hydrogen bond donors or acceptors). But circular fingerprints are in more widespread use today. Here, each atom is inspected together with its neighborhood of bonded atoms at distance 1, 2, …; instead of pre-defined chemical concepts, each of such local patterns switches on one bit according to a hash function. A typical size of the bit vector is 1024. One standard implementation are extended circular fingerprints (termed ECFPx, with a number x designating the maximum diameter; e,g, ECFP4 for a radius of 2 bonds). The similarity between two molecules can be estimated using the Tanimoto coefficient (the same metrics is known in other domains as Jaccard index) — the number of bits set to one in both molecules, divided by those in either one. Database similarity searches were the original motivation for circular fingerprints, as they can be implemented very efficiently through bitwise operations. But they later also helped to flexibly extend the number of molecule features for machine learning; the above-mentioned dozens or so of 1-D properties alone obviously go only so far.

Now such an encoding is necessarily non-unique — it is possible that two completely different molecules are hashed to the same fingerprint. This can sometimes be a point of confusion — how can machine learning still work without making gross mistakes?

But there is some guard against this in generally having enough bits set; indeed, hash collisions can even be seen as a welcome measure to prevent overfitting. The argument is exactly analogous to the introduction of feature hashing in the context of machine learning for text applications.