RDKit-Powered Reaction Classification and Yield Prediction using the Differential Reaction Fingerprint DRFP

Disclaimer:

- This work is part of my PhD project (Reymond Group, University of Bern)
- Co-authors: Philippe Schwaller, Jean-Louis Reymond
- Current Position: Research scientist @ IBM Research

Spoilers

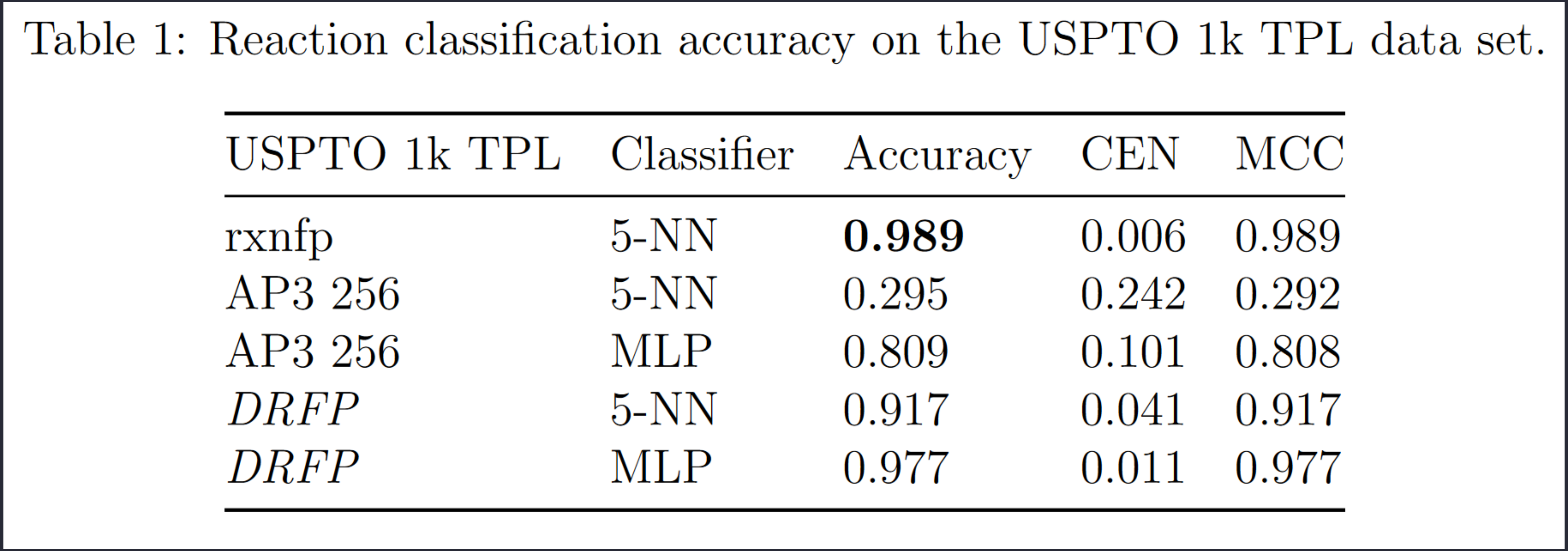


Table 1: Reaction classification accuracy on the USPTO 1k TPL data set.

USPTO 1k TPL	Classifier	Accuracy	CEN	MCC
rxnfp	5-NN	**0.989**	0.006	0.989
AP3 256	5-NN	0.295	0.242	0.292
AP3 256	MLP	0.809	0.101	0.808
DRFP	5-NN	0.917	0.041	0.917
DRFP	MLP	0.977	0.011	0.977

Schwaller, P.; Probst, D.; Vaucher, A. C.; Nair, V. H.; Kreutter, D.; Laino, T.; Reymond, J.-L., Nat Mach Intell 2021

Schneider, N.; Lowe, D. M.; Sayle, R. A.; Landrum, G. A., J. Chem. Inf. Model. 2015

Spoilers

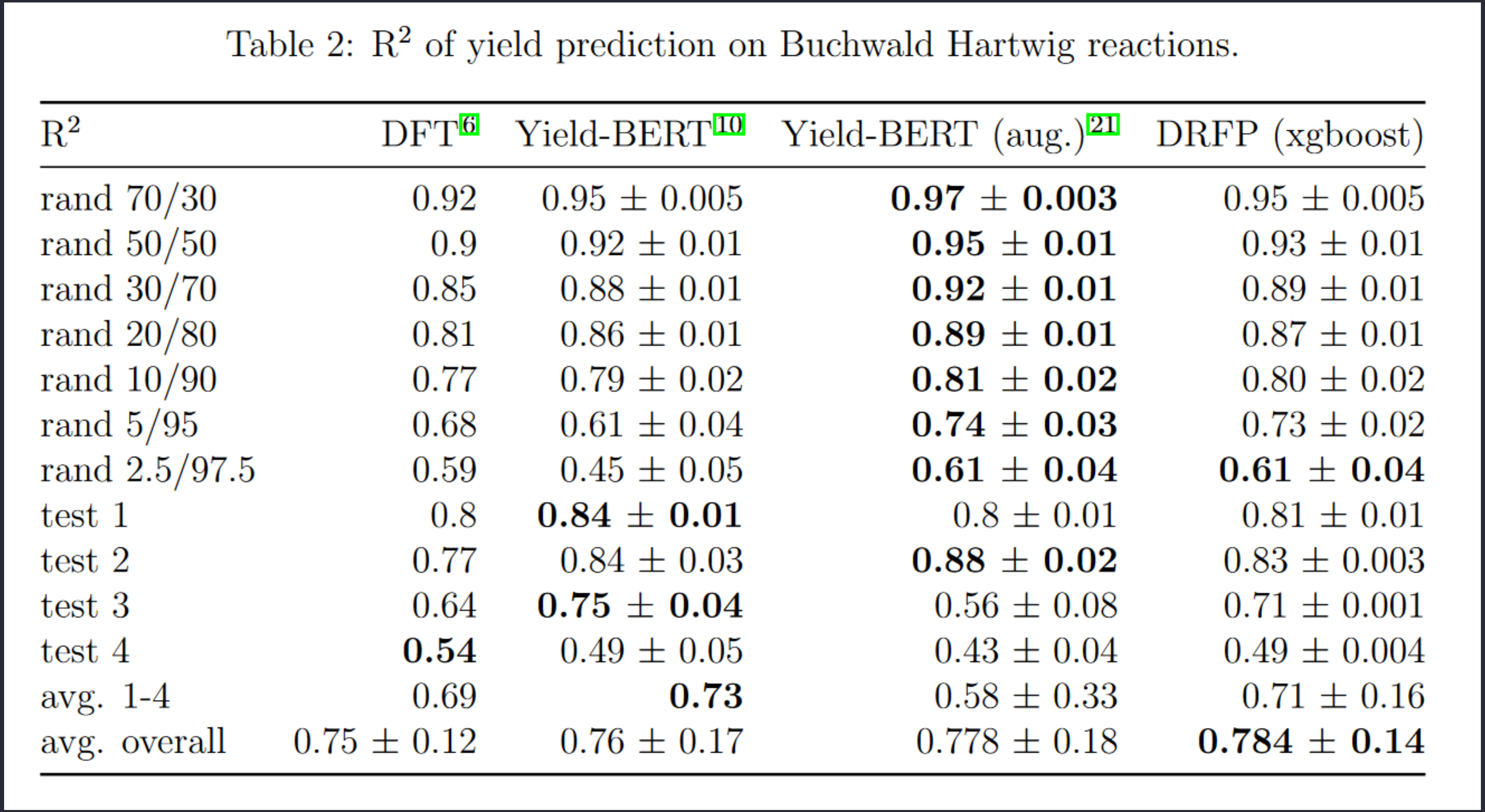


Table 2: R^2 of yield prediction on Buchwald Hartwig reactions.

R^2	DFT[6]	Yield-BERT[10]	Yield-BERT (aug.)[21]	DRFP (xgboost)
rand 70/30	0.92	0.95 ± 0.005	**0.97 ± 0.003**	0.95 ± 0.005
rand 50/50	0.9	0.92 ± 0.01	**0.95 ± 0.01**	0.93 ± 0.01
rand 30/70	0.85	0.88 ± 0.01	**0.92 ± 0.01**	0.89 ± 0.01
rand 20/80	0.81	0.86 ± 0.01	**0.89 ± 0.01**	0.87 ± 0.01
rand 10/90	0.77	0.79 ± 0.02	**0.81 ± 0.02**	0.80 ± 0.02
rand 5/95	0.68	0.61 ± 0.04	**0.74 ± 0.03**	0.73 ± 0.02
rand 2.5/97.5	0.59	0.45 ± 0.05	**0.61 ± 0.04**	**0.61 ± 0.04**
test 1	0.8	**0.84 ± 0.01**	0.8 ± 0.01	0.81 ± 0.01
test 2	0.77	0.84 ± 0.03	**0.88 ± 0.02**	0.83 ± 0.003
test 3	0.64	**0.75 ± 0.04**	0.56 ± 0.08	0.71 ± 0.001
test 4	**0.54**	0.49 ± 0.05	0.43 ± 0.04	0.49 ± 0.004
avg. 1-4	0.69	**0.73**	0.58 ± 0.33	0.71 ± 0.16
avg. overall	0.75 ± 0.12	0.76 ± 0.17	0.778 ± 0.18	**0.784 ± 0.14**

Spoilers

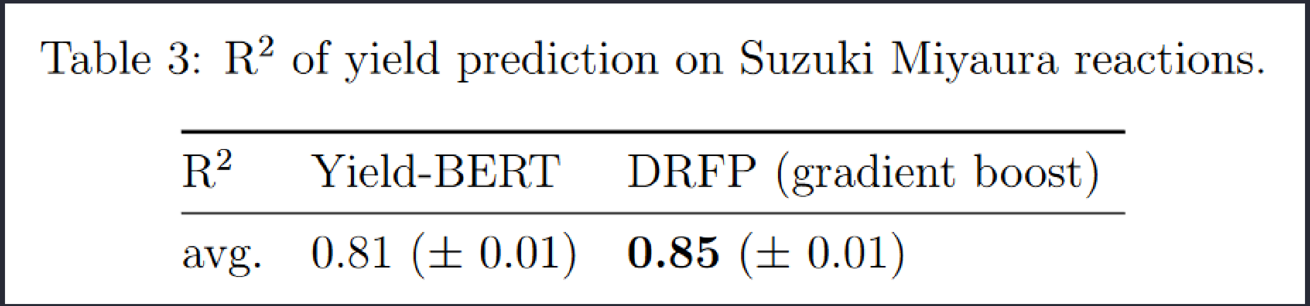


Table 3: R^2 of yield prediction on Suzuki Miyaura reactions.

R^2	Yield-BERT	DRFP (gradient boost)
avg.	0.81 ($\pm$ 0.01)	**0.85** ($\pm$ 0.01)

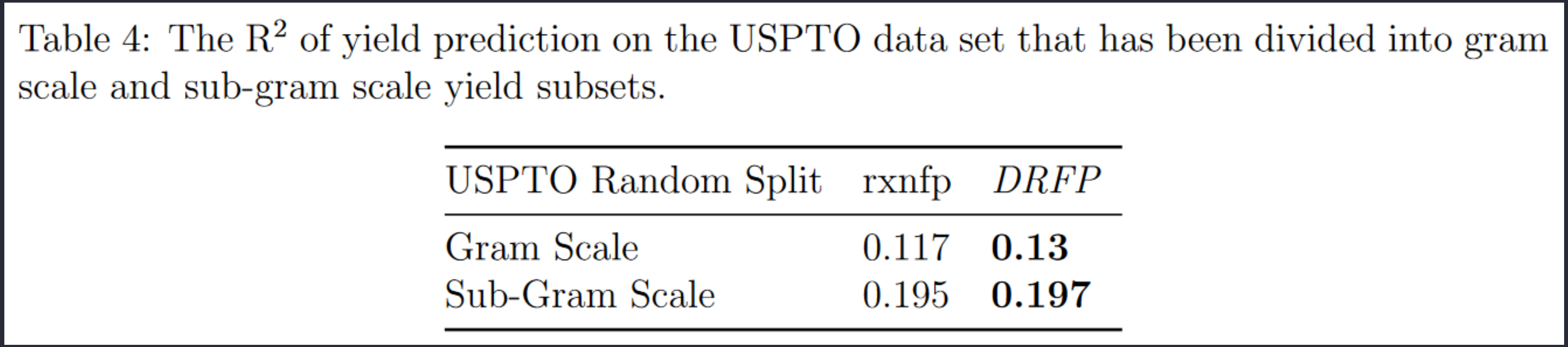


Table 4: The R^2 of yield prediction on the USPTO data set that has been divided into gram scale and sub-gram scale yield subsets.

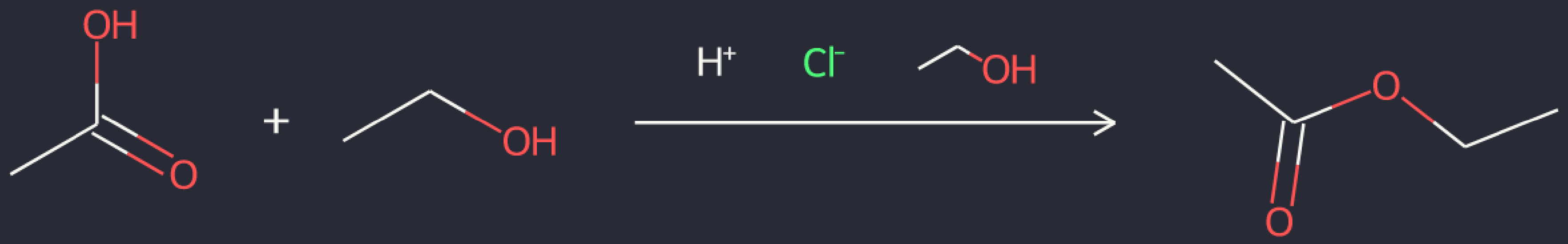
USPTO Random Split	rxnfp	*DRFP*
Gram Scale	0.117	**0.13**
Sub-Gram Scale	0.195	**0.197**

Schwaller, P.; Vaucher, A. C.; Laino, T.; Reymond, J.-L., Machine Learning: Science and Technology 2021, 2, 015016

Ahneman, D. T.; Estrada, J. G.; Lin, S.; Dreher, S. D.; Doyle, A. G., Science 2018

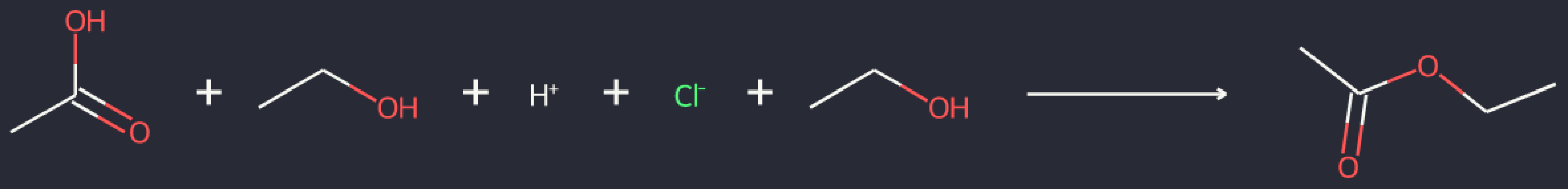
Reaction SMILES

```
rxn_smiles = "CC(=O)O.OCC>[H+].[Cl-].OCC>CC(=O)OCC"
rxn = AllChem.ReactionFromSmarts(rxn_smiles, useSmiles=True) # Thanks @iwatobipen
d2d = dark_mode(MolDraw2DSVG(1024, 300))
d2d.DrawReaction(rxn)
d2d.FinishDrawing()
SVG(d2d.GetDrawingText())
```



Reaction SMILES - Everything is a Reactant[1,2]

```
rxn_smiles = "CC(=O)O.OCC.[H+].[Cl-].OCC>>CC(=O)OCC"
rxn = AllChem.ReactionFromSmarts(rxn_smiles, useSmiles=True) # Thanks @iwatobipen
d2d = dark_mode(MolDraw2DSVG(1024, 300))
d2d.DrawReaction(rxn)
d2d.FinishDrawing()
SVG(d2d.GetDrawingText())
```

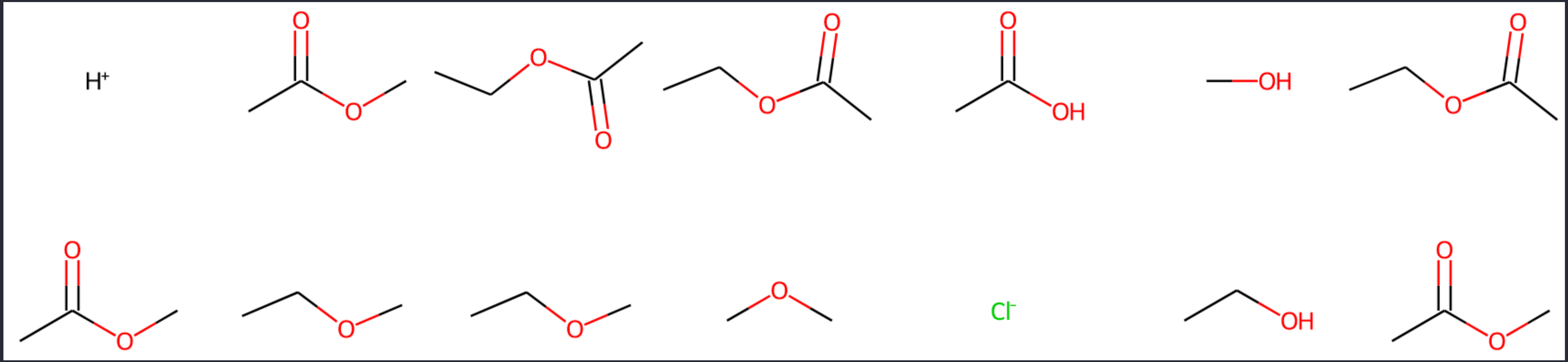


[1]except the product(s), of course

[2]and we don't need atom mappings either

What's in the Box?

```
fps, mapping = DrfpEncoder.encode(rxn_smiles, mapping=True)
mols = [MolFromSmiles(s) for smileses in mapping.values() for s in smileses]
SVG(MolsToGridImage(mols, molsPerRow=7, useSVG=True))
```



Show us the Code!

```
left = sides[0].split(".")
right = sides[2].split(".")

left_shingles = set()
right_shingles = set()
```

...

```
s = right_shingles.symmetric_difference(left_shingles)
```

The Roof, the Roof, ...

```
for ring in AllChem.GetSymmSSSR(in_mol):
```

...

```
for i, atom in enumerate(in_mol.GetAtoms()):
```

...

```
for index, _ in enumerate(in_mol.GetAtoms()):
    for i in range(1, radius + 1):
        p = AllChem.FindAtomEnvironmentOfRadiusN(in_mol, i, index)
```

Hash

```
s = right_shingles.symmetric_difference(left_shingles)
```

...

```
hash_values = []
for t in shingling:
    hash_values.append(int(blake2b(t, digest_size=4).hexdigest(), 16))

return np.array(hash_values, dtype=np.int32)
```

Fold

```
return np.array(hash_values, dtype=np.int32)
```

...

```
folded = np.zeros(length, dtype=np.uint8)
on_bits = hash_values % length
folded[on_bits] = 1

return folded, on_bits
```

That's it... But there's more!

```
fps, mapping = DrfpEncoder.encode(rxn_smiles, mapping=True)
```

...

```
def encode(X: Union[Iterable, str], ...):
    for _, x in enumerate(X):
        if mapping:
            for i, folded_index in enumerate(on_bits):
                result_map[folded_index].add(
                    smiles_diff[i].decode("utf-8")
                )
```

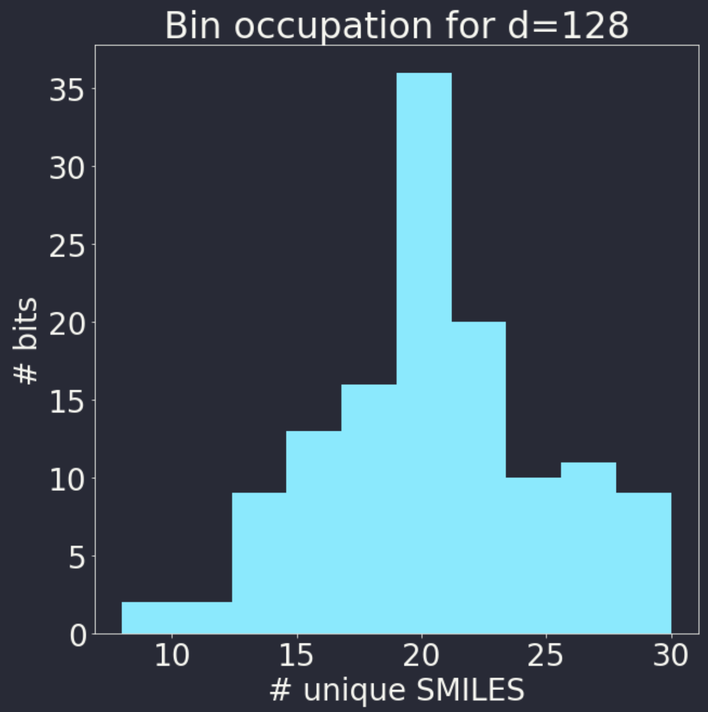
Example

```
smiles = []
url = ("https://raw.githubusercontent.com/reymond-group/"
       "drfp/main/notebooks/reaction_smiles.csv")
with urlopen(url) as f:
    smiles = [line.strip().decode("utf-8") for line in f]

fps, mapping = DrfpEncoder.encode(smiles, mapping=True, n_folded_length=128)

n_grams_per_bin = [len(value) for value in mapping.values()]
print(f"There are {sum(n_grams_per_bin)} unique molecular n-grams.")
```

```
> There are 2610 unique molecular n-grams.
```

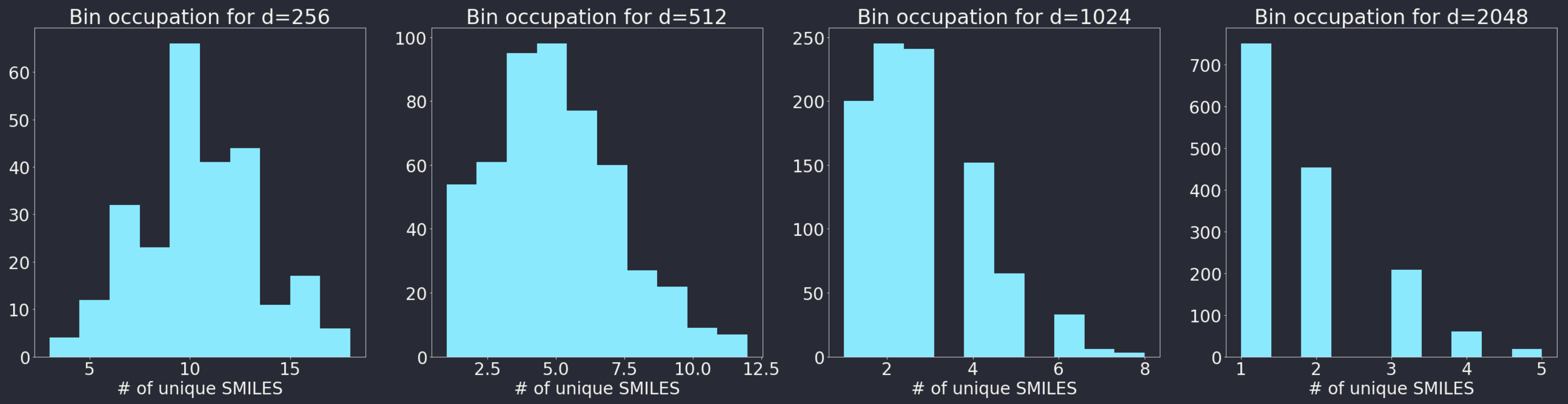
Example

```
plt.hist(n_grams_per_bin)
```



Example

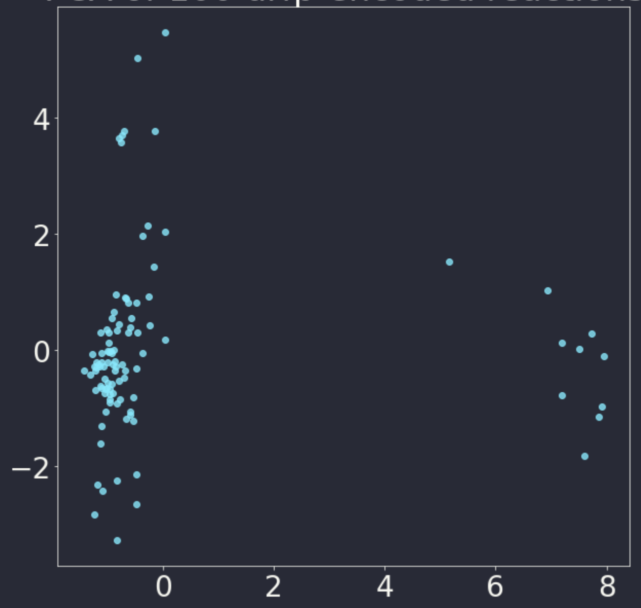
```
for i, d in enumerate([256, 512, 1024, 2048]):
    fps, mapping = DrfpEncoder.encode(smiles, mapping=True, n_folded_length=d)
```



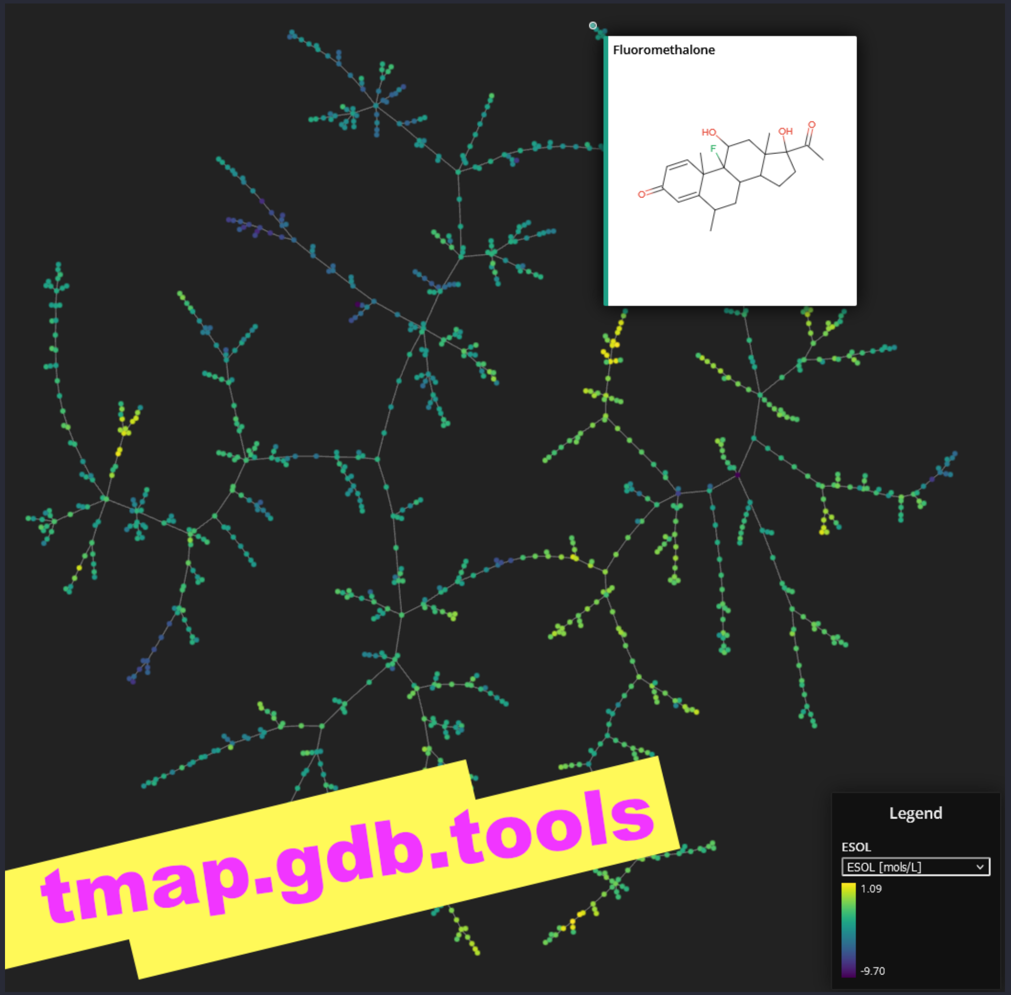
I Should have used TMAP...

```
pca = PCA(n_components=2)
X = pca.fit(fps).transform(fps)
...
```

PCA of 100 drfp-encoded reactions



Really should have...



Encore

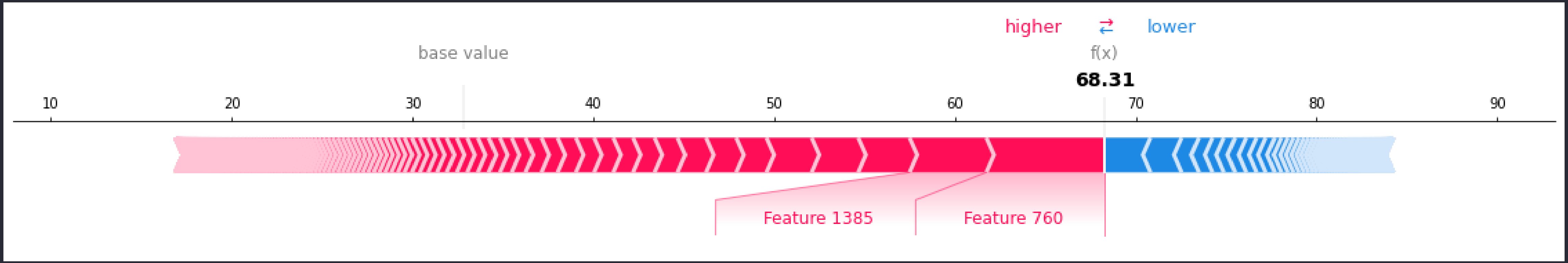
```
X, y = pickle.load(open("buchwald_hartwig.pkl", "rb"))
mapping = pickle.load(open("buchwald_hartwig.map.pkl", "rb"))
model = pickle.load(open("buchwald_hartwig_model.pkl", "rb")) # XGBRegressor
```

...

```
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X[:100])
shap.summary_plot(shap_values, X[:100])
```

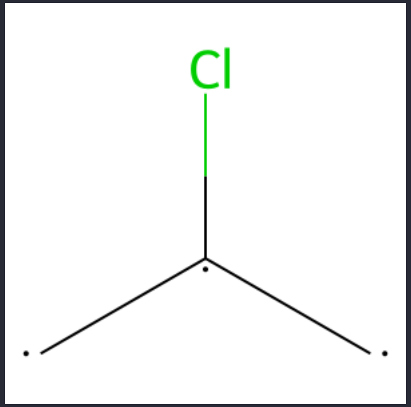
Encore

```
shap.force_plot(explainer.expected_value, shap_values[0,:], matplotlib=True)
```



Encore

```
sub = list(mapping.get(760))[0].replace("c", "C")
MolFromSmiles(sub)
```



Thanks!

Thanks for listening!

Connect with me...
... on the blue bird network: @skepteis
... e-mail: dpr@zurich.ibm.com

Im currently working on...
... retrosynthetic pathway prediction for biocatalysed reactions (https://rxn.res.ibm.com/, preprint: https://bit.ly/BIOCATML)