Spec2Class Manual

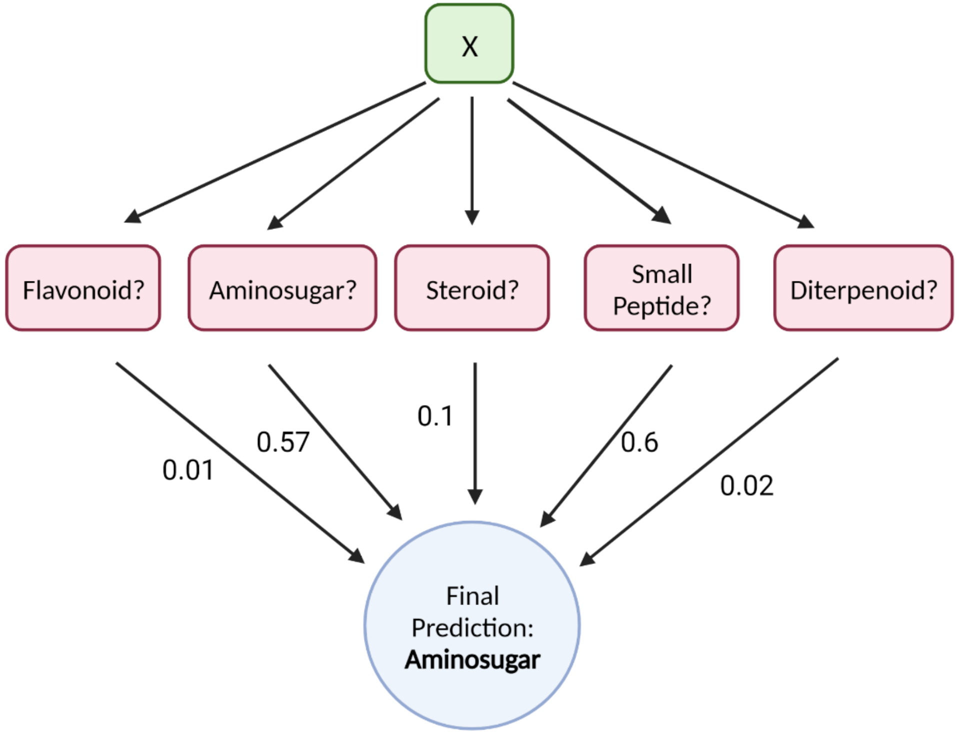
Victoria Poltorak, April 23

Introduction:
Spec2Class is an ensemble model

It contains two steps major steps of prediction

1. Binary prediction by 43 binary models. Each binary model predicts if a spectrum belongs to a certain class or not

2. Final multiclass prediction done based on the output of the previous step vis SVM model



Introduction: Binary classification

SteroidXtract,
Xing et al. *Analytical Chemistry* 2021

A preprocessing step is required for the use of Spec2Class.

At the moment it accepts only pickle file (see later full description of the input)

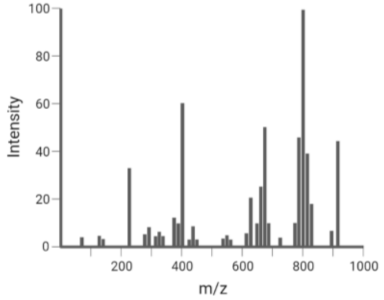
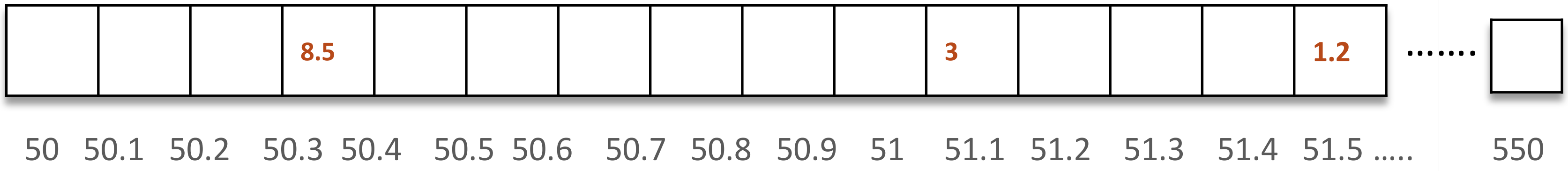
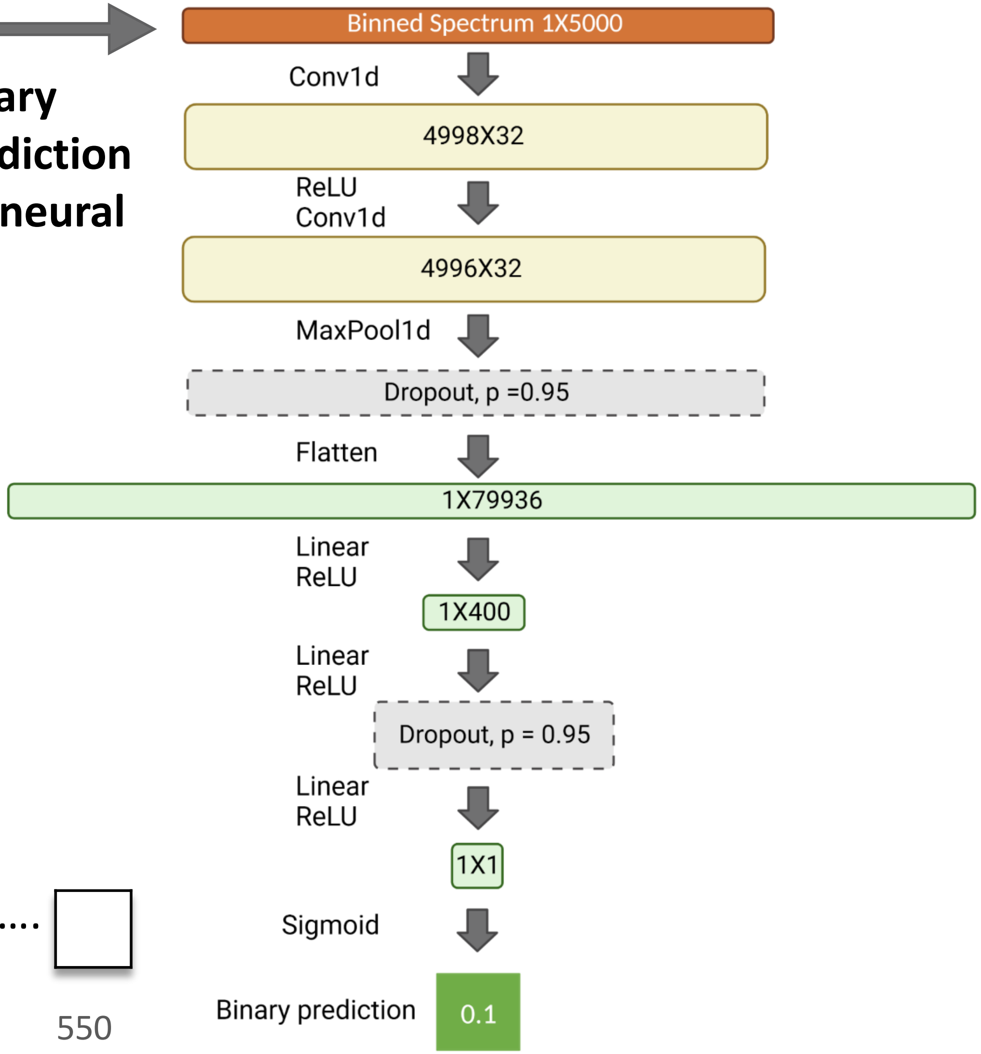


Table with mz and intensity arrays per each spectrum
Saved as a pickle file

Binning

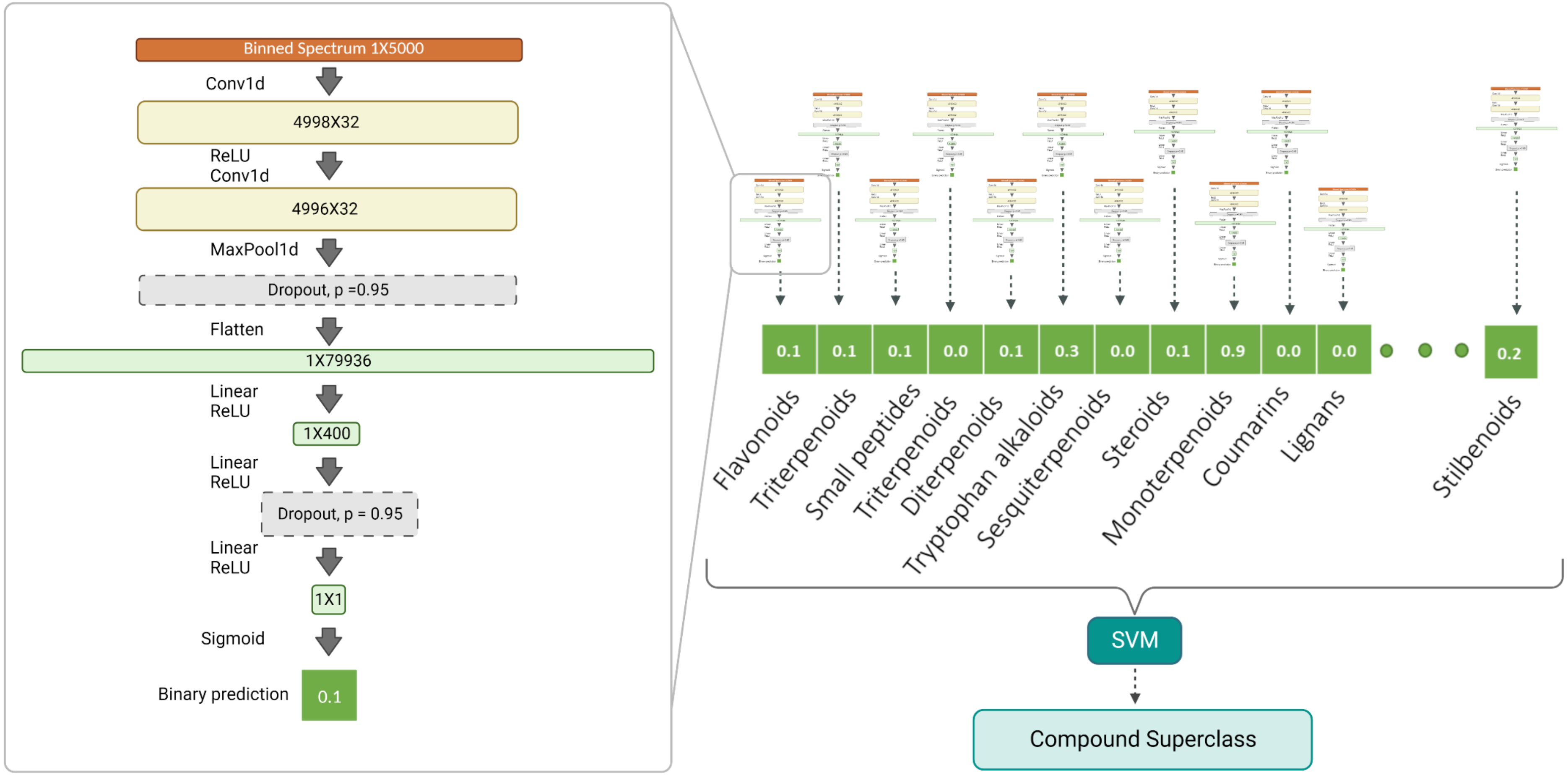


Binary prediction via neural net



Introduction: Spec2Class structure

Each binary model is a trained convolutional neural net



Binned Spectrum 1X5000

Conv1d

4998X32

ReLU
Conv1d

4996X32

MaxPool1d

Dropout, p =0.95

Flatten

1X79936

Linear
ReLU

1X400

Linear
ReLU

Dropout, p = 0.95

Linear
ReLU

1X1

Sigmoid

Binary prediction 0.1

0.1 0.1 0.1 0.0 0.1 0.3 0.0 0.1 0.9 0.0 0.0 ● ● ● 0.2

Flavonoids

Triterpenoids

Small peptides

Triterpenoids

Diterpenoids

Tryptophan alkaloids

Sesquiterpenoids

Steroids

Monoterpenoids

Coumarins

Lignans

Stilbenoids

SVM

Compound Superclass

Instructions

1. Download the models

Hugging Face

1.1 Download the binary models from:

- The binary models are saved in the huggingface hub.

https://huggingface.co/VickiPol/binary_models

- It is possible to download them from the website, but is easier to do it with

```
# Make sure you have git-lfs installed (https://git-lfs.github.com)
git lfs install
git clone https://huggingface.co/VickiPol/binary_models
```

- If you want to clone it to certain directory and not the current one:

```
git clone <git_repo_url> <your_custom_directory_name>
```

1. Download the models

1.2 Download the SVM model from:

https://huggingface.co/VickiPol/SVM_model/tree/main

Download from the website or clone the repository (as described in 1.1)

2. Set the environment

- Make sure Anaconda is installed
- Use **anaconda prompt** and create an environment to run Spec2Class with spec2class.yml file

```
conda env create -f spec2class_env.yml
```

- Activate the environment

```
conda activate spec2class_env
```

3. Go to the project's git repository and download it's content

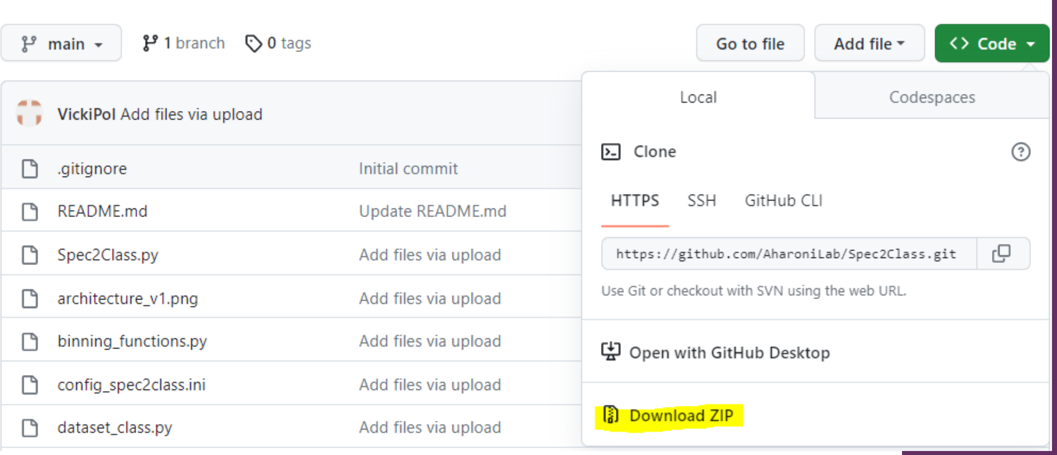
- If you would like to use git:

```
git clone https://github.com/AharoniLab/Spec2Class.git <your_custom_directory_name>
```

- Otherwise, you can download the scripts in a zip file:

Enter https://github.com/AharoniLab/Spec2Class.git

Press the code button and choose 'Download ZIP'



VickiPol Add files via upload

.gitignore	Initial commit
README.md	Update README.md
Spec2Class.py	Add files via upload
architecture_v1.png	Add files via upload
binning_functions.py	Add files via upload
config_spec2class.ini	Add files via upload
dataset_class.py	Add files via upload

https://github.com/AharoniLab/Spec2Class.git

Use Git or checkout with SVN using the web URL.

What is in the repository?

1. **spec2Class.py** – the main script that runs the model
2. **config_spec2class.ini** – a configuration file that holds the paths and the constant parameters
3. **dataset_class.py** – this is a class that builds a dataloader object, which loads the data in batches in a fixed size (128). The dataloader significantly accelerates the model's inference when GPU is used. It is also helps when not enough memory is available in CPU.
4. **neural_net.py** – A file that contains the neural net binary model class
5. **binning_functions.py** – a file with all the functions that are related to the spectrum binning
6. **prediction_vectors_functions.py** – a file with the functions that are related to the generation of the binary prediction vectors
7. **multiclass_prediction_functions.py** – a file with functions that are related to the multiclass prediction step which is done via SVM model
8. **utility_functions.py** – a file with all the other functions that are used
9. **mona_100_spec.pkl** – an input example with 100 spectra from Mass Bank of North America
10. **mona_100_spec_output.pkl** – an output example for the given input
11. **Spec2class_env.yml** - spec2class conda environment

4. Edit the config file

Update the paths for the models in the file ‘config_multiclass_prediction.ini’

For example:

[paths]

#The path to the trained SVM model

- svm_model_path = Spec2Class\spec2class_trained_svm.sav

#The path to the directory where the 43 binary models are saved

- binary_models_dir = Spec2Class\binary_models

#The path to the Neural Net class

- net_path = Spec2Class**new_model_b550.py**

#The path to the directory Neural Net class

- net_dir = \Spec2Class

5. Model's Input

- The model's input should be pandas dataframe saved in a .pkl (pickle) format.
- Each row in the input dataframe should represent one ms/ms spectrum.

- The dataframe should contain the following mandatory columns:

1. **'mz'** – list or array of m/z values for each row
2. **'Intensity'** – list or array of corresponding relative intensities
3. **'DB.'** – spectrum identifier

6. Model's Output

- The output is a tabular file in three formats: .pkl,.tsv,.csv
- The output will contain the following columns:

1. **DB.** – spectrum identifier
2. **final pred** – chemical class prediction
3. **estimated top2 pred** – chemical class prediction with the 2nd highest probability
4. **estimated top3 pred** – chemical class prediction with the 3rd highest
5. **probabilities** – array of the top 3 probability values

7. Test the model with the given input example

To make sure that everything is working, try to run the script with the attached small test of 100 spectra from MONA (these spectra were included in the training set)

Run via anaconda prompt or with any other IDE or terminal:

```
python Spec2Class.py <config_file_path>  <input_path> <output_dir> <output_name>
```

Compare the results with the given output example

8. Using GPU

- The model runs much quicker on GPU.
- If ‘cuda’ is available the script will run automatically on GPU, otherwise – on CPU