My Project

Generated by Doxygen 1.8.13

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Namespace Documentation

3.1 chemistry2quant.chem2quant_analysis Namespace Reference

Classes

- class rdkitProcessDf
- class rdkitPsi4DataGenerator

Functions

- def **neuron_layer** (X, n_neurons, name, activation=None)
- def sdfToMol (sdf)
- def **substructure_search** (substruct, struct_array)

Variables

- string **directory** = "/home/noh/Desktop/CURRENT_WORK_IN_PROGRESS/Chemiinformatics/RDK↩ IT/rdkit/Docs/Book/data"
- string **sdf_file** = 'bzr.sdf'
- **process** = rdkit_processdf(directory, sdf_file)
- **molList** = process.returnMol()
- **molSmiles** = process.MoltoSmiles()
- list **mol2VecList** = [mol2alt_sentence(x,1) for x in molList]
- int **n_hidden1** = 300
- int **n_hidden2** = 100
- int **n_hidden3** = 100
- **feature_colummns** = tf.contrib.learn_infer_real_valued_columns_from_input(X_train)
- def **hidden1** = neuron_layer(X, n_hidden1, "hidden1", activation = "relu")
- def **hidden2** = neuron_layer(hidden1, n_hidden2, "hidden2", activation = "relu")
- def **logits** = neuron_layer(hidden2, n_outputs, "outputs")
- **init** = tf.global_variables_initalizer()
- **saver** = tf.train.Saver()

3.1.1 Detailed Description

```
@package docstring
Documentation for this module

More details here
```

3.1.2 Function Documentation

3.1.2.1 sdfToMol()

```
def chemistry2quant.chem2quant_analysis.sdfToMol (
              sdf )
```

```
Returns array of mols from sdf
```

3.2 chemistry2quant.chem2quant_chembl2pandas Namespace Reference

Classes

- class chemblConnect

Variables

- **cur** = conn.cursor()
- **database** = cur.fetchall()

3.2.1 Detailed Description

```
Python module which returns the chembl database as a pandas table, depending on the
specifications you would like in the end.

Make sure that the permissions settings on the table has been granted to the USER.
```

3.3 chemistry2quant.chem2quant_lipophilicity Namespace Reference

Classes

- class rdkit_lipophilicity

Variables

- **zipFileDir** = zipfile.ZipFile('../zip/lipophilicity.zip')
- **Pd_df** = pd.read_csv(zipFileDir.open('Lipophilicity.csv'))
- **batch_x**
- **batch_y**
- **train_op**
- **feed_dict**
- **loss**
- **acc**

3.3.1 Detailed Description

3.4 chemistry2quant.chem2quant_mol2vec_ChEMBL Namespace Reference

Functions

- def **emolecule_command** (sql_command, user, password)
- def **chembl24_command** (sql_command, user, password)

Variables

- def **table_names** = chembl24_command("SELECT table_name FROM information_schema.tables WHERE table_schema='public'", "sang", "silver!!")
- list **data_list** = [ ]
- string **sql_command** = "SELECT ∗ FROM {} LIMIT 100;".format(str(title))

3.4.1 Detailed Description

```
Module for downloading ChemBL data
```

3.5 chemistry2quant.chem2quant_NN Namespace Reference

Functions

- def next_batch (num, data, labels)
- def **mol2arr** (mol)

Variables

- int **n_hidden1** = 300
- int **n_hidden2** = 300
- int **n_hidden3** = 300
- float **learning_rate** = 0.01
- int **n_outputs** = 3
- string **datadir** = 'data'
- list **train_mol** = [mol for mol in Chem.SDMolSupplier(os.path.join(datadir,'solubility.train.sdf')) if mol != None]
- list **test_mol** = [mol for mol in Chem.SDMolSupplier(os.path.join(datadir,'solubility.test.sdf')) if mol != None]
- **cls_mol** = list(set([mol.GetProp('SOL_classification') for mol in train_mol]))
- dictionary **cls_dic** = {}
- **train_X** = np.array([mol2arr(mol) for mol in train_mol])
- **train_y** = np.array([cls_dic[mol.GetProp('SOL_classification')] for mol in train_mol])
- **test_X** = np.array([mol2arr(mol) for mol in test_mol])
- **test_y** = np.array([cls_dic[mol.GetProp('SOL_classification')] for mol in test_mol])
- **X** = tf.placeholder(tf.float32, shape = (None, np.shape(train_X[0])[0]), name = "X")
- **y** = tf.placeholder(tf.int64, shape=(None), name="y")
- **he_init** = tf.contrib.layers.variance_scaling_initializer()
- **hidden1** = fully_connected(X, n_hidden1, weights_initializer = he_init, scope = "hidden1")
- **hidden2** = fully_connected(hidden1, n_hidden2, weights_initializer = he_init, scope = "hidden2")
- **logits** = fully_connected(hidden2, n_outputs, weights_initializer = he_init, scope = "outputs", activation_fn = None)
- **xentropy** = tf.nn.sparse_softmax_cross_entropy_with_logits(labels = y, logits = logits)
- **loss** = tf.reduce_mean(xentropy, name = "loss")
- **correct** = tf.nn.in_top_k(logits, y ,1)
- **accuracy** = tf.reduce_mean(tf.cast(correct, tf.float32))
- **optimizer** = tf.train.GradientDescentOptimizer(learning_rate)
- **training_op** = optimizer.minimize(loss)
- **init** = tf.global_variables_initializer()
- **saver** = tf.train.Saver()
- int **n_epochs** = 20
- int **batch_size** = 100
- **X_batch**
- **y_batch**
- **feed_dict**
- **acc_train** = accuracy.eval(feed_dict={X: X_batch, y: y_batch})
- **acc_test** = accuracy.eval(feed_dict={X: test_X, y: test_y})
- **save_path** = saver.save(sess, "./my_model_final.ckpt")

3.5.1 Detailed Description

```
Tensorflow implementation of
```

3.5.2 Function Documentation

3.5.2.1 next_batch()

```
def chemistry2quant.chem2quant_NN.next_batch (
              num,
              data,
              labels )
```

```
Return a total of ‘num‘ random samples and labels.
```

Chapter 4

Class Documentation

4.1 chemistry2quant.chem2quant_gen_psi4_input.chem2quant_psi4 Class Reference

Public Member Functions

- def **mol2psi4** (mol)

4.1.1 Detailed Description

```
Generate coordinate file from smiles for calculation with
psi4
```

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Chemiinformatics_work/Chemistry2quant/src/chemistry2quant/chem2quant↩
_gen_psi4_input.py

4.2 chemistry2quant.chem2quant_chembl2pandas.chemblConnect Class Reference

Public Member Functions

- def **__init__** (self, database, user, host, password)
- def **issue_command** (self)

Public Attributes

- **database**
- **user**
- **host**
- **password**
- **string**

Static Public Attributes

- **conn** = psycopg2.connect("dbname='emolecules' user='sang' host='localhost' password='Blad1bl@1234'")

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Chemiinformatics_work/Chemistry2quant/src/chemistry2quant/chem2quant↩ _chembl2pandas.py

4.3 chemistry2quant.chem2quant_lipophilicity.rdkit_lipophilicity Class Reference

Public Member Functions

- def **__init__** (zipFile, name, featurizer, reload=True, move_mean=True)
- def **load_lipo** (featurizer='ECFP', split='index', reload=True, move_mean=True)
- def process_data_column ()
- def mols2feat ()
- def **load_tensorflow** (learning_rate, training_epochs, batch_size)
- def **weights** ()
- def **neural_net** (x)
- def **fingerprints** (mols)
- def **categories** (fingerprints, index)

Public Attributes

- **rdkit_zip**
- **rdkit_csv**
- **lipophilicity_csv**
- **initial_score**
- **initial_smiles**
- **Mol**
- **MFingerprints**
- **np_fps**
- **X**
- **Y**
- **batch_size**
- **learning_rate**
- **training_epochs**
- **weights**
- **biases**

4.3.1 Detailed Description

```
TODO
```

4.3.2 Member Function Documentation

4.3.2.1 mols2feat()

```
def chemistry2quant.chem2quant_lipophilicity.rdkit_lipophilicity.mols2feat ( )
```

```
Converting fingerprint data into vector data
```

4.3.2.2 process_data_column()

```
def chemistry2quant.chem2quant_lipophilicity.rdkit_lipophilicity.process_data_column ( )
```

```
Processing data columns in csv to pandas, then to Morgan Fingerprints ('')
```

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Chemiinformatics_work/Chemistry2quant/src/chemistry2quant/chem2quant↩
_lipophilicity.py

4.4 chemistry2quant.chem2quant_analysis.rdkitProcessDf Class Reference

Public Member Functions

- def **__init__** (self, directory, sdf_file_name)
- def **returnMol** (self)
- def **MoltoSmiles** (self)
- def **MACCSfingerprintList** (self)
- def **torsionalfingerprintList** (self)

Public Attributes

- **rdkit_directory**
- **lig_data**
- **dataMol**
- **ms_smiles**
- **MACCSlist**
- **Pairslist**

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Chemiinformatics_work/Chemistry2quant/src/chemistry2quant/chem2quant↩
_analysis.py

4.5 chemistry2quant.chem2quant_analysis.rdkitPsi4DataGenerator Class Reference

Public Member Functions

- def **__init__** (molfile)
- def **molToPsi4** (self)
- def **forEachSimilarity** (ref, array)
- def **storeMolecule** ()

Public Attributes

- **molfile**

4.5.1 Detailed Description

```
Here, we want to translate the smilestoMol file into a psi4 file and run DFT calculations for each.
Based on the code seen in "https://iwatobipen.wordpress.com/2018/08/24/calculate-homo-and-lumo-with-psi4-rdkit
```

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Chemiinformatics_work/Chemistry2quant/src/chemistry2quant/chem2quant↩
_analysis.py

4.6 chemistry2quant.chem2quant_screening.two_step_screen Class Reference

Public Member Functions

- def **__init__** (self, smilesList, molecule)
- def first_screen ()
- def second_screen ()

Public Attributes

- **smilesList**
- **molecule**
- smileMol

4.6.1 Detailed Description

```
Initial screening of a database with a substructure - The first screen removes 99% of all the options in the t
For example, the emoleucles or the chembl database. The second screen studies the work using a number of algir
we use the standard rdkit substructure search
```

4.6.2 Member Function Documentation

4.6.2.1 first_screen()

```
def chemistry2quant.chem2quant_screening.two_step_screen.first_screen ( )
```

```
Remove 99% of the structural database which doesn't match the general structure
at all
```

4.6.2.2 second_screen()

```
def chemistry2quant.chem2quant_screening.two_step_screen.second_screen ( )
```

```
Building from a simple screening
```

4.6.3 Member Data Documentation

4.6.3.1 smileMol

```
chemistry2quant.chem2quant_screening.two_step_screen.smileMol
```

```
Remove 99% of the structural database which doesn't match the general structure
at all
```

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Chemiinformatics_work/Chemistry2quant/src/chemistry2quant/chem2quant↩_screening.py

Index