yield_from_HTEppt

July 15, 2020

0.0.1 Get reaction yield from HTE PPT and Excel files

on going: extract reactants, stoichiometric ratio and products information to build a knowledge base

[1]:
```
# import libraries
import re
import glob
import numpy as np
import pandas as pd
from pptx import Presentation
```

[2]:
```
# get the list of pptx and xlsx files in current directory
ppt_lst = glob.glob('*.pptx')
xlsx_lst = glob.glob('*.xlsx')

print(f'\033[1mPPT files:\033[0m \n\t{ppt_lst}. \n\n\033[1mExcel files:\033[0m␣
 ↪\n\t{xlsx_lst}.')
```

```
PPT files:
        ['2019_10_15 - LPAR - MikeHawkins - Suzuki.pptx',
'2019_10_16_jdiccian_5_LPAR - Suzuki.pptx', '2019_11_05 - jcompto4_2 - DGAT2 -
Tudge Photoredox Minisci .pptx', '2020_07_01_emercad2_692_SIK_Pd
CN_report.pptx'].

Excel files:
        ['2019_10_15 - LPAR - MikeHawkins - Suzuki.xlsx',
'2019_10_16_jdiccian_5_LPAR - Suzuki.xlsx', '2019_11_05-jcompto4_2 - DGAT2-Tudge
photoredox Minisci .xlsx', '2020_07_01_emercad2_692_SIK_Pd C-N.xlsx'].
```

[3]:
```
# define a funciton to extract yield from pptx files

def get_yield(ppt):
    """
    Get yield from HTE pptx files

    INPUT: pptx filepath, or filename if in the same directory

    """
```

```
    # define a list to hold all text from the pptx
    exp_text = []

    # initiate the Presentaiton function
    prs = Presentation(ppt)

    # loop through all slides and look for texts, append to exp_text list
    for slide in prs.slides:
        for shape in slide.shapes:
            if hasattr(shape, 'text'):
                exp_text.append(shape.text)

    # slice the strings containing 'yield'
    yield_text = [text for text in exp_text if 'yield' in text.lower()]

    try:
        # find the yield value which ends with '%'
        yield_value = re.findall(r'\d+%', ' '.join(yield_text))
    except:
        # assign yield = 0 if not available
        yield_value = 0

    return yield_value
```

[4]:
```
# define a yield dictionary to hold pptxfile:yield pairs
yield_dict = {}

# loop through all HTE pptx files in current directory
for ppt in ppt_lst:

    # get the yield
    yield_value = get_yield(ppt)

    # add to the dictionary to correlate with pptx filename
    yield_dict[ppt] = yield_value

# print the yield dictionary
yield_dict
```

[4]:
```
{'2019_10_15 - LPAR - MikeHawkins - Suzuki.pptx': ['80%'],
 '2019_10_16_jdiccian_5_LPAR - Suzuki.pptx': ['87%'],
 '2019_11_05 - jcompto4_2 - DGAT2 - Tudge Photoredox Minisci .pptx': ['35%'],
 '2020_07_01_emercad2_692_SIK_Pd CN_report.pptx': ['30%']}
```

[5]:
```
def get_reaction_df(xlsx):
    """
```

```
    Get xlsx dataframe from HTE xlsx files

    INPUT: xlsx filepath, or filename if in the same directory

    OUTPUT: xlsx dataframe

    """
    df = pd.read_excel(xlsx, sheet_name = 'Start')

    return df
```

[6]:
```
# define a funciton to extract ELN from the dataframe from xlsx files
def get_eln(df_xlsx):

    """
    Get ELN from HTE xlsx dataframe

    INPUT: xlsx dataframe

    """
    # import 'Start' sheet from the file
    df = df_xlsx

    # found inconsistent locations across different files, some in D3, some in␣
 ↪E3
    # get bot D3 and E3 and get the one other than 'Experiment Notebook Number'
    eln = [df.iloc[1,3], df.iloc[1,4]]
    eln = [x for x in eln if not str(x).startswith('Experiment')]

    # if there is no ELN, name it 'TBD'
    if eln == [np.nan]:
        eln = 'TBD'

    else:
        eln = eln[0]

    return eln
```

[7]:
```
# define an ELN dictionary to hold ELN: xlsxfile pairs
eln_dict = {}

# loop through all HTE xlsx files in current directory
for xlsx in xlsx_lst:

    df_xlsx = get_reaction_df(xlsx)

    # get the ELN
```

```
    eln = get_eln(df_xlsx)

    # add to the dictionary to correlate with xlsx filename
    eln_dict[eln] = xlsx

# print the ELN dictionary
eln_dict
```

[7]:
```
{'TBD': '2019_10_15 - LPAR - MikeHawkins - Suzuki.xlsx',
 'jdiccian_5': '2019_10_16_jdiccian_5_LPAR - Suzuki.xlsx',
 'jcompto4_2': '2019_11_05-jcompto4_2 - DGAT2-Tudge photoredox Minisci .xlsx',
 'emercad2_692': '2020_07_01_emercad2_692_SIK_Pd C-N.xlsx'}
```

[8]:
```
# define a funciton to combine ELN and yield together

def get_eln_yield(eln_dict, yield_dict):
    """
    Get ELN: yield pair from eln_dict and yield_dict

    INPUT:
    eln_dict:         ELN: xlsxfile dictionary
    yield_dict:       pptxfile:yield dictionary

    """
    # initiate an empty dictionary
    eln_yield = {}

    # loop through the ELN dictionary
    for eln, xlsx in eln_dict.items():

        # loop through the yield dictionary keys
        for ppt in yield_dict.keys():

            # if ELN is part of the ppt filename
            if eln in ppt:

                # let ELN be the key of the new ELN-yield dictionary
                # let yield be the value of the new ELN-yield dictionary
                eln_yield[eln] = yield_dict[ppt]

            # if ELN is 'TBD', use the full xlsx filename
            elif eln == 'TBD':

                # get the pptx filename by changing the filetype from .xlsx to .
↪pptx
                pptfile = xlsx.split('.')[0] + '.pptx'
```

```
                # let xlsx filename be the key of the new ELN-yield dictionary
                # let yield be the value of the new ELN-yield dictionary
                eln_yield[xlsx] = yield_dict[pptfile]


    return dict(sorted(eln_yield.items(), key=lambda kv: kv[1], reverse=True))

eln_yield = get_eln_yield(eln_dict, yield_dict)
eln_yield
```

[8]:
```
{'jdiccian_5': ['87%'],
 '2019_10_15 - LPAR - MikeHawkins - Suzuki.xlsx': ['80%'],
 'jcompto4_2': ['35%'],
 'emercad2_692': ['30%']}
```

[9]:
```
df = pd.DataFrame(eln_yield).T
df.reset_index(inplace=True)
df.columns=['ELN','Yield']
df
```

[9]:

	ELN	Yield
0	jdiccian_5	87%
1	2019_10_15 - LPAR - MikeHawkins - Suzuki.xlsx	80%
2	jcompto4_2	35%
3	emercad2_692	30%

[10]:
```
df.sort_values(by='ELN')
```

[10]:

	ELN	Yield
1	2019_10_15 - LPAR - MikeHawkins - Suzuki.xlsx	80%
3	emercad2_692	30%
2	jcompto4_2	35%
0	jdiccian_5	87%

[11]:
```
df.sort_values(by='Yield', ascending = False)
```

[11]:

	ELN	Yield
0	jdiccian_5	87%
1	2019_10_15 - LPAR - MikeHawkins - Suzuki.xlsx	80%
2	jcompto4_2	35%
3	emercad2_692	30%

[ ]:
```

```