

reg_year_r_random_points_new

February 29, 2024

0.1 Importing

```
[ ]: import xarray as xr
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn import preprocessing

from sklearn.neural_network import MLPRegressor
from sklearn.ensemble import BaggingRegressor

from sklearn.metrics import mean_squared_error as mse

from tqdm.auto import tqdm

import dill
import random

import salishsea_tools.viz_tools as sa_vi
```

0.2 Datasets Preparation

```
[ ]: def datasets_preparation(dataset):

    drivers = np.stack([np.ravel(dataset['Temperature_(0m-15m)']),
                        np.ravel(dataset['Temperature_(15m-100m)']), np.
↪ ravel(dataset['Salinity_(0m-15m)']),
                        np.ravel(dataset['Salinity_(15m-100m)'])])
    indx = np.where(~np.isnan(drivers).any(axis=0))
    drivers = drivers[:,indx[0]]

    diat = np.ravel(dataset['Diatom'])
    diat = diat[indx[0]]
```

```
return(drivers, diat, indx)
```

0.3 Regressor

```
[ ]: def regressor (inputs, targets):  
  
    inputs = inputs.transpose()  
  
    # Regressor  
    scale = preprocessing.StandardScaler()  
    inputs = scale.fit_transform(inputs)  
    X_train, _, y_train, _ = train_test_split(inputs, targets, train_size=0.35)  
  
    drivers = None  
    diat = None  
  
    inputs = None  
    targets = None  
  
    model = MLPRegressor(hidden_layer_sizes=200, alpha=0.002)  
    regr = BaggingRegressor(model, n_estimators=12, n_jobs=4).fit(X_train,   
↪y_train)  
  
    return (regr)
```

0.4 Regressor 2

```
[ ]: def regressor2 (inputs, targets, variable_name):  
  
    inputs = inputs.transpose()  
  
    # Regressor  
    scale = preprocessing.StandardScaler()  
    inputs2 = scale.fit_transform(inputs)  
  
    outputs_test = regr.predict(inputs2)  
  
    m = scatter_plot(targets, outputs_test, variable_name)  
    r = np.round(np.corrcoef(targets, outputs_test)[0][1],3)  
    rms = mse(targets, outputs_test)  
  
    return (r, rms, m)
```

0.5 Regressor 3

```
[ ]: def regressor3 (inputs, targets):

    inputs = inputs.transpose()

    # Regressor
    scale = preprocessing.StandardScaler()
    inputs2 = scale.fit_transform(inputs)

    outputs_test = regr.predict(inputs2)

    # compute slope m and intercept b
    m, b = np.polyfit(targets, outputs_test, deg=1)

    r = np.round(np.corrcoef(targets, outputs_test)[0][1],3)
    rms = mse(targets, outputs_test)

    return (r, rms, m)
```

0.6 Regressor 4

```
[ ]: def regressor4 (inputs, targets, variable_name):

    inputs = inputs.transpose()

    # Regressor
    scale = preprocessing.StandardScaler()
    inputs2 = scale.fit_transform(inputs)

    outputs = regr.predict(inputs2)

    # Post processing
    indx2 = np.full((898*398),np.nan)
    indx2[indx[0]] = outputs
    model = np.reshape(indx2,(898,398))

    m = scatter_plot(targets, outputs, variable_name + str(dates[i].date()))

    # Preparation of the dataarray
    model = xr.DataArray(model,
        coords = {'y': diat_i.y, 'x': diat_i.x},
        dims = ['y','x'],
        attrs=dict( long_name = variable_name + "Concentration",
            units="mmol m-2"),)
```

```
plotting3(targets, model, diat_i, variable_name)
```

1 Printing

```
[ ]: def printing (targets, outputs, m):  
  
    print ('The amount of data points is', outputs.size)  
    print ('The slope of the best fitting line is ', np.round(m,3))  
    print ('The correlation coefficient is:', np.round(np.corrcoef(targets,   
↪outputs)[0][1],3))  
    print (' The mean square error is:', np.round(mse(targets,outputs),5))
```

1.1 Scatter Plot

```
[ ]: def scatter_plot(targets, outputs, variable_name):  
  
    # compute slope m and intercept b  
    m, b = np.polyfit(targets, outputs, deg=1)  
  
    printing(targets, outputs, m)  
  
    fig, ax = plt.subplots(2, figsize=(5,10), layout='constrained')  
  
    ax[0].scatter(targets,outputs, alpha = 0.2, s = 10)  
  
    lims = [np.min([ax[0].get_xlim(), ax[0].get_ylim()]),  
            np.max([ax[0].get_xlim(), ax[0].get_ylim()])]  
  
    # plot fitted y = m*x + b  
    ax[0].axline(xy1=(0, b), slope=m, color='r')  
  
    ax[0].set_xlabel('targets')  
    ax[0].set_ylabel('outputs')  
    ax[0].set_xlim(lims)  
    ax[0].set_ylim(lims)  
    ax[0].set_aspect('equal')  
  
    ax[0].plot(lims, lims,linestyle = '--',color = 'k')  
  
    h = ax[1].hist2d(targets,outputs, bins=100, cmap='jet',  
                    range=[lims,lims], cmin=0.1, norm='log')  
  
    ax[1].plot(lims, lims,linestyle = '--',color = 'k')
```

```

# plot fitted y = m*x + b
ax[1].axline(xy1=(0, b), slope=m, color='r')

ax[1].set_xlabel('targets')
ax[1].set_ylabel('outputs')
ax[1].set_aspect('equal')

fig.colorbar(h[3],ax=ax[1], location='bottom')

fig.suptitle(variable_name)

plt.show()

return (m)

```

1.2 Plotting

```

[ ]: def plotting(variable, name):

    plt.plot(years,variable, marker = '.', linestyle = '')
    plt.legend(['diatom', 'flagellate'])
    plt.xlabel('Years')
    plt.ylabel(name)
    plt.show()

```

1.3 Plotting 2

```

[ ]: def plotting2(variable,title):

    fig, ax = plt.subplots()

    scatter= ax.scatter(dates,variable, marker='.', c=pd.DatetimeIndex(dates).
↪month)

    ax.legend(handles=scatter.legend_elements()[0],
↪labels=['February', 'March', 'April'])
    fig.suptitle('Daily ' + title + ' (15 Feb - 30 Apr)')

    fig.show()

```

1.4 Plotting 3

```
[ ]: def plotting3(targets, model, variable, variable_name):

    fig, ax = plt.subplots(2,2, figsize = (10,15))

    cmap = plt.get_cmap('cubehelix')
    cmap.set_bad('gray')

    variable.plot(ax=ax[0,0], cmap=cmap, vmin = targets.min(), vmax =targets.
    ↪max(), cbar_kwargs={'label': variable_name + ' Concentration [mmol m-2]'})
    model.plot(ax=ax[0,1], cmap=cmap, vmin = targets.min(), vmax = targets.
    ↪max(), cbar_kwargs={'label': variable_name + ' Concentration [mmol m-2]'})
    ((variable-model) / variable * 100).plot(ax=ax[1,0], cmap=cmap,
    ↪cbar_kwargs={'label': variable_name + ' Concentration [percentage]'})

    plt.subplots_adjust(left=0.1,
        bottom=0.1,
        right=0.95,
        top=0.95,
        wspace=0.35,
        hspace=0.35)

    sa_vi.set_aspect(ax[0,0])
    sa_vi.set_aspect(ax[0,1])
    sa_vi.set_aspect(ax[1,0])

    ax[0,0].title.set_text(variable_name + ' (targets)')
    ax[0,1].title.set_text(variable_name + ' (outputs)')
    ax[1,0].title.set_text('targets - outputs')
    ax[1,1].axis('off')

    fig.suptitle(str(dates[i].date()))

    plt.show()
```

1.5 Training (Random Points)

```
[ ]: ds = xr.open_dataset('/data/ibougoudis/MOAD/files/integrated_model_var_old.nc')

# ds = ds.isel(time_counter = (np.arange(0, len(ds.Diatom.time_counter),2)),
#             y=(np.arange(ds.y[0], ds.y[-1], 5)),
#             x=(np.arange(ds.x[0], ds.x[-1], 5)))
```

```

dates = pd.DatetimeIndex(ds['time_counter'].values)

drivers, diat, _ = datasets_preparation(ds)

regr = regressor(drivers, diat)

```

1.6 Other Years (Anually)

```

[ ]: years = range (2007,2024)

r_all = []
rms_all = []
slope_all = []

for year in tqdm(range (2007,2024)):

    dataset = ds.sel(time_counter=slice(str(year)))

    drivers, diat, _ = datasets_preparation(dataset)

    r, rms, m = regressor2(drivers, diat, 'Diatom ' + str(year))

    r_all.append(r)
    rms_all.append(rms)
    slope_all.append(m)

plotting(np.transpose(r_all), 'Correlation Coefficient')
plotting(np.transpose(rms_all), 'Mean Square Error')
plotting (np.transpose(slope_all), 'Slope of the best fitting line')

```

```

0%|          | 0/17 [00:00<?, ?it/s]

```

```

/tmp/ipykernel_28643/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned

```

```

    m, b = np.polyfit(targets, outputs, deg=1)

```

```

The amount of data points is 3485925

```

```

The slope of the best fitting line is  0.497

```

```

The correlation coefficient is: 0.658

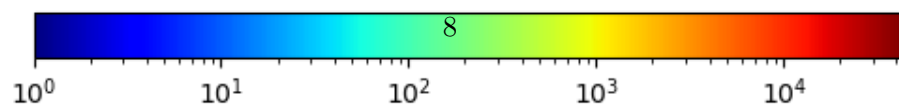
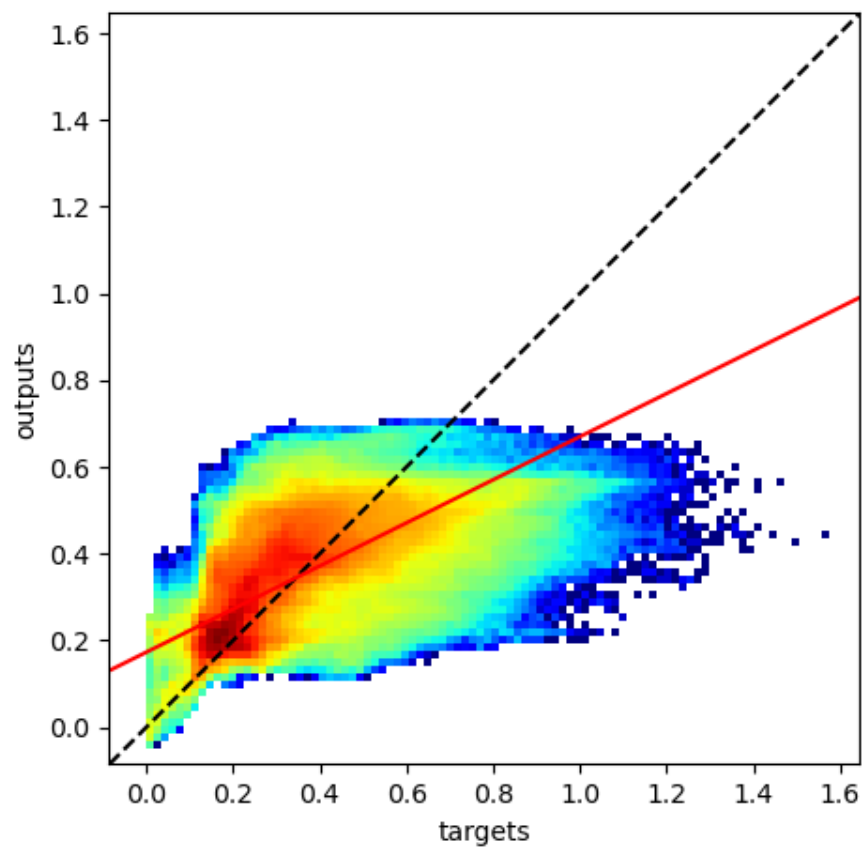
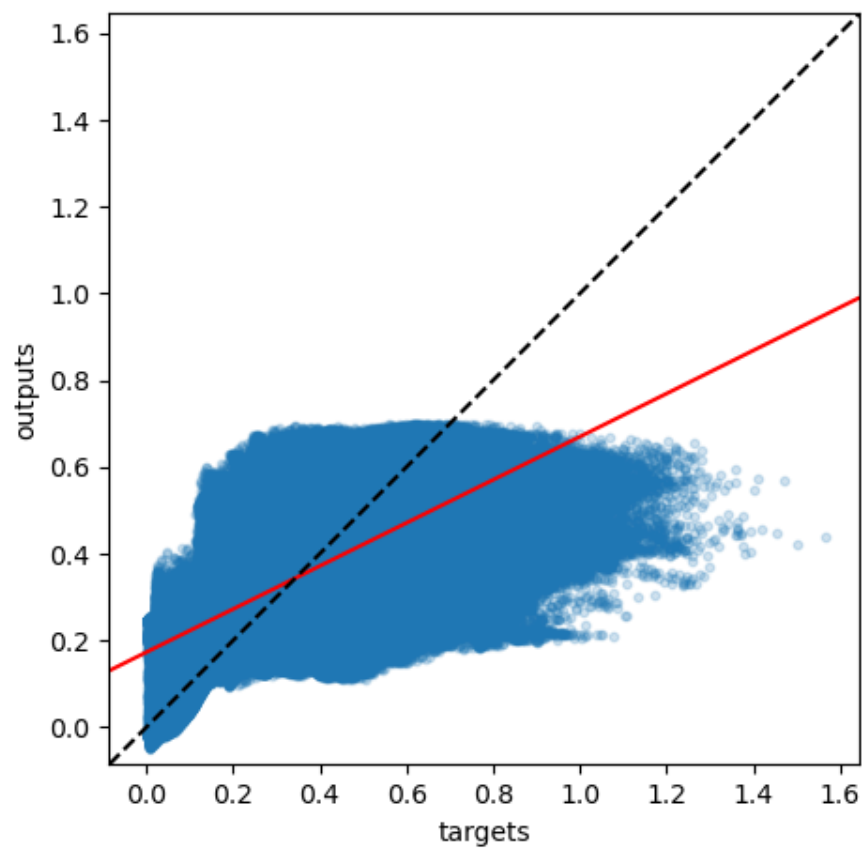
```

```

The mean square error is: 0.01521

```

Diatom 2007




```
/tmp/ipykernel_28643/1980467486.py:4: RankWarning: Polyfit may be poorly conditioned
```

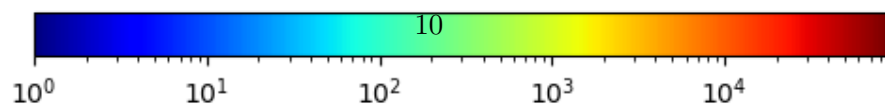
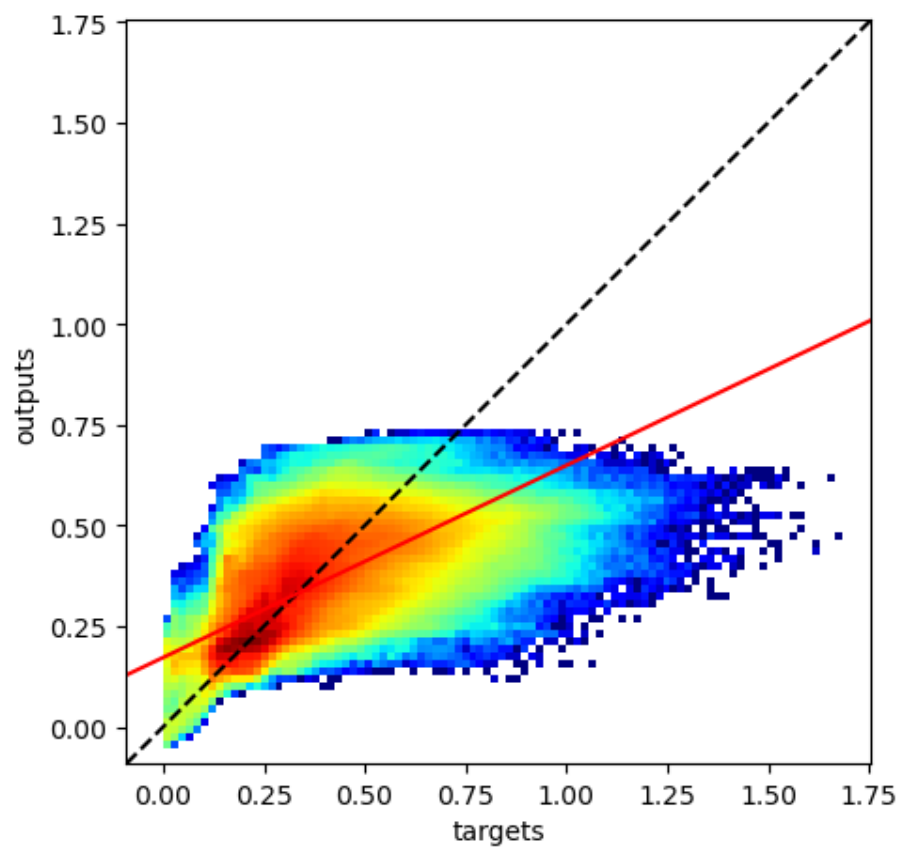
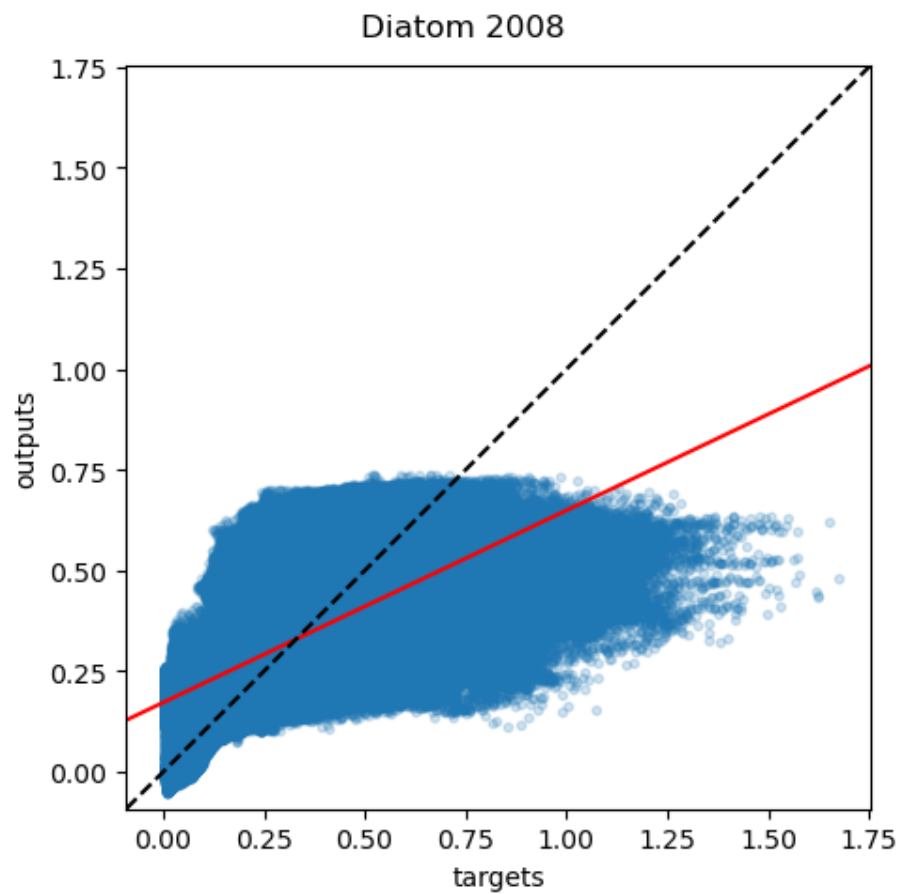
```
    m, b = np.polyfit(targets, outputs, deg=1)
```

```
The amount of data points is 7018329
```

```
The slope of the best fitting line is 0.477
```

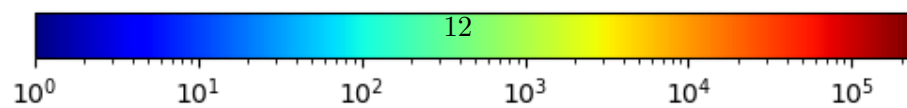
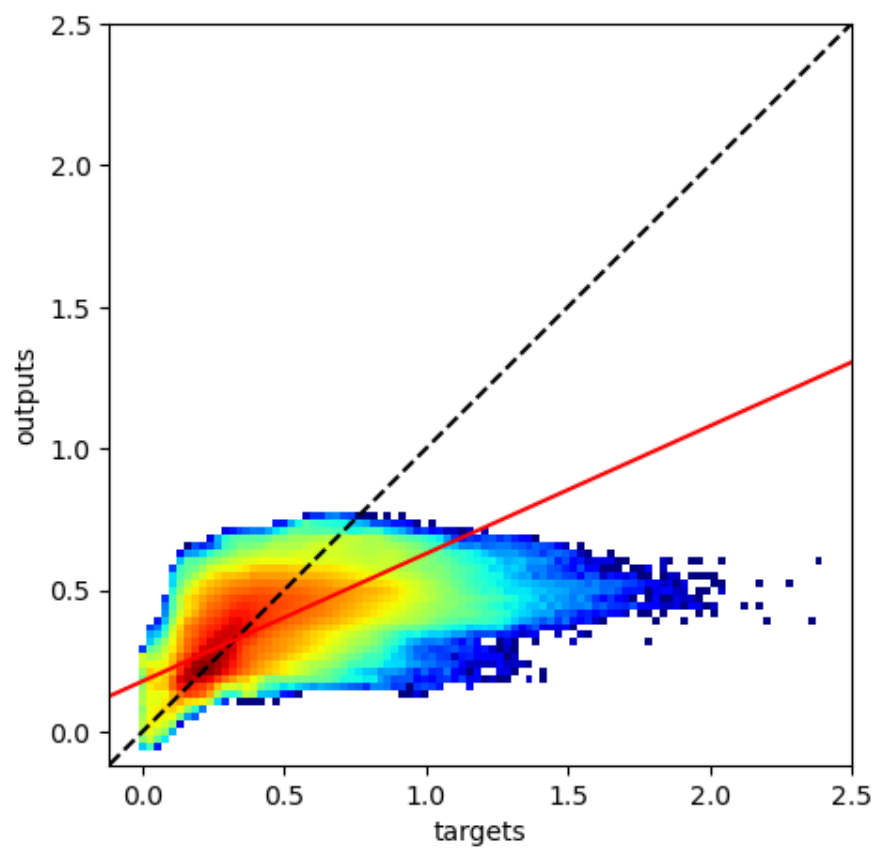
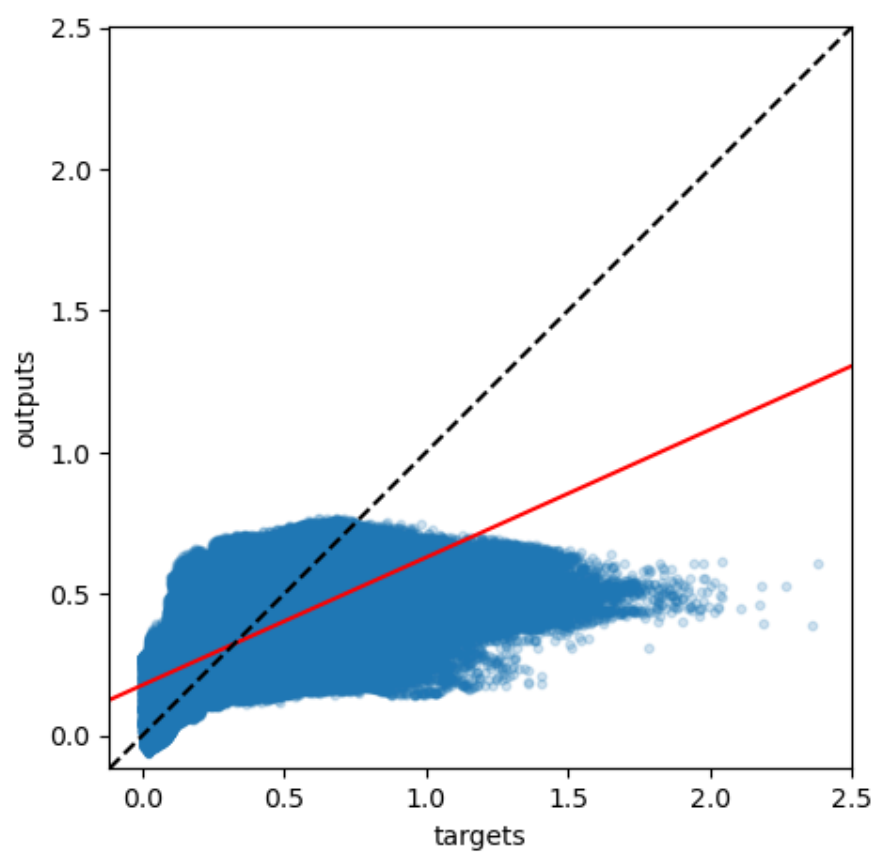
```
The correlation coefficient is: 0.671
```

```
The mean square error is: 0.01319
```



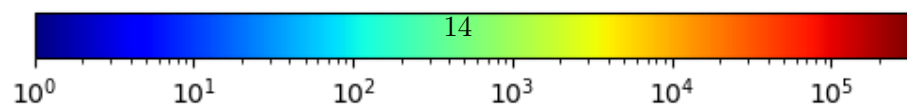
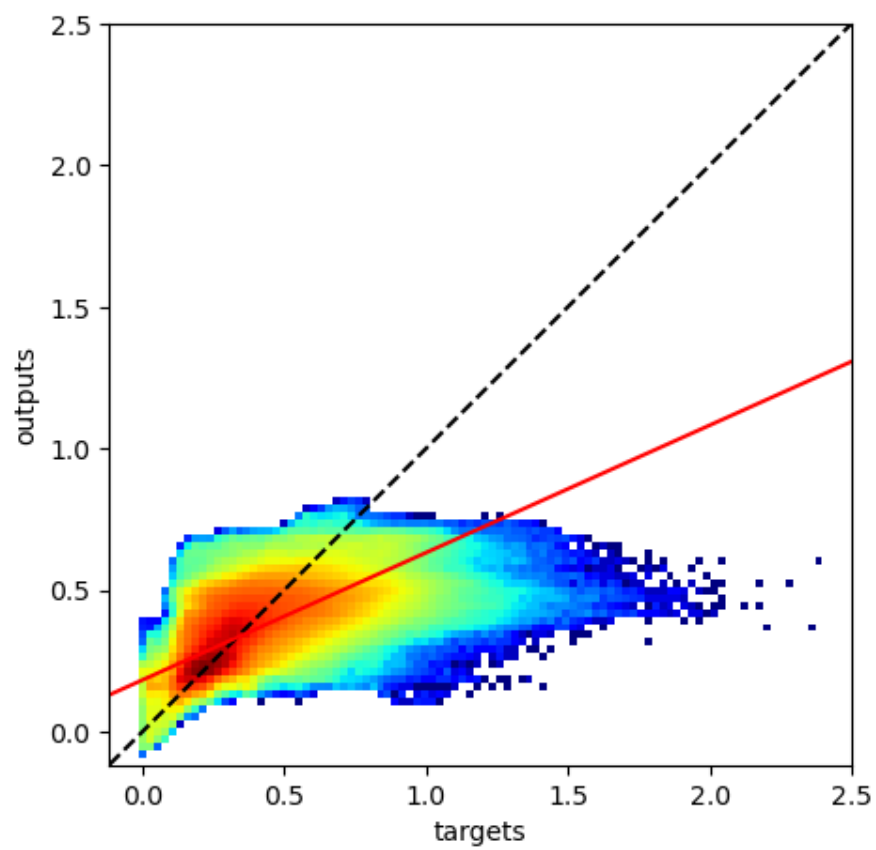
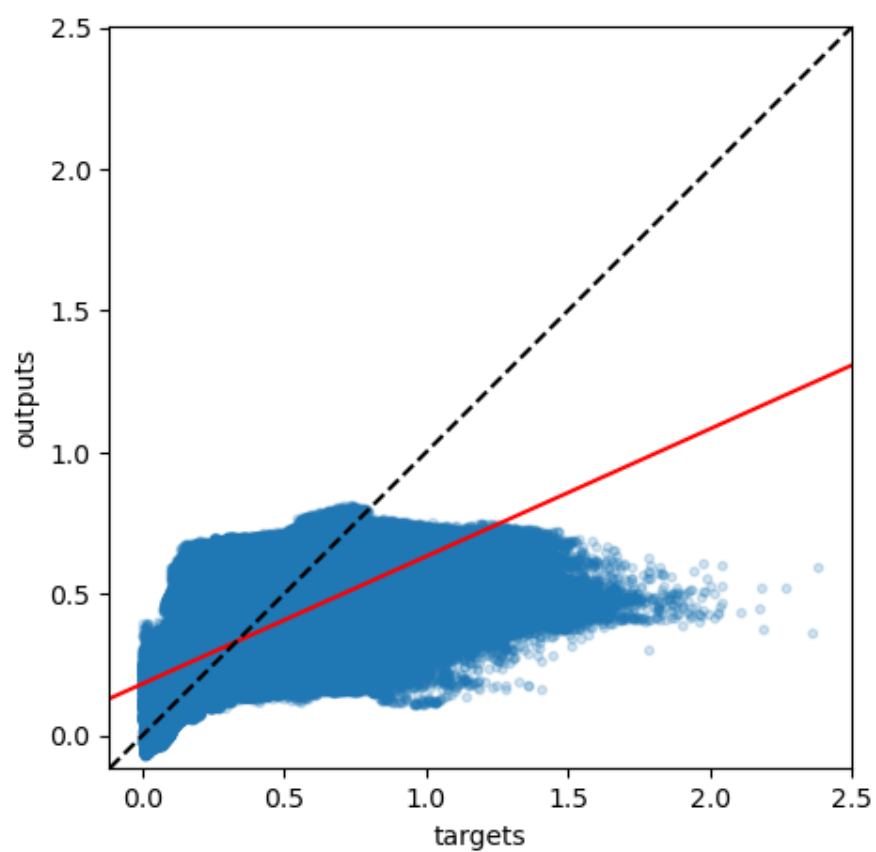
The amount of data points is 10504254
The slope of the best fitting line is 0.451
The correlation coefficient is: 0.666
The mean square error is: 0.01606

Diatom 2009



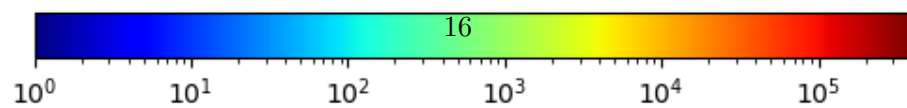
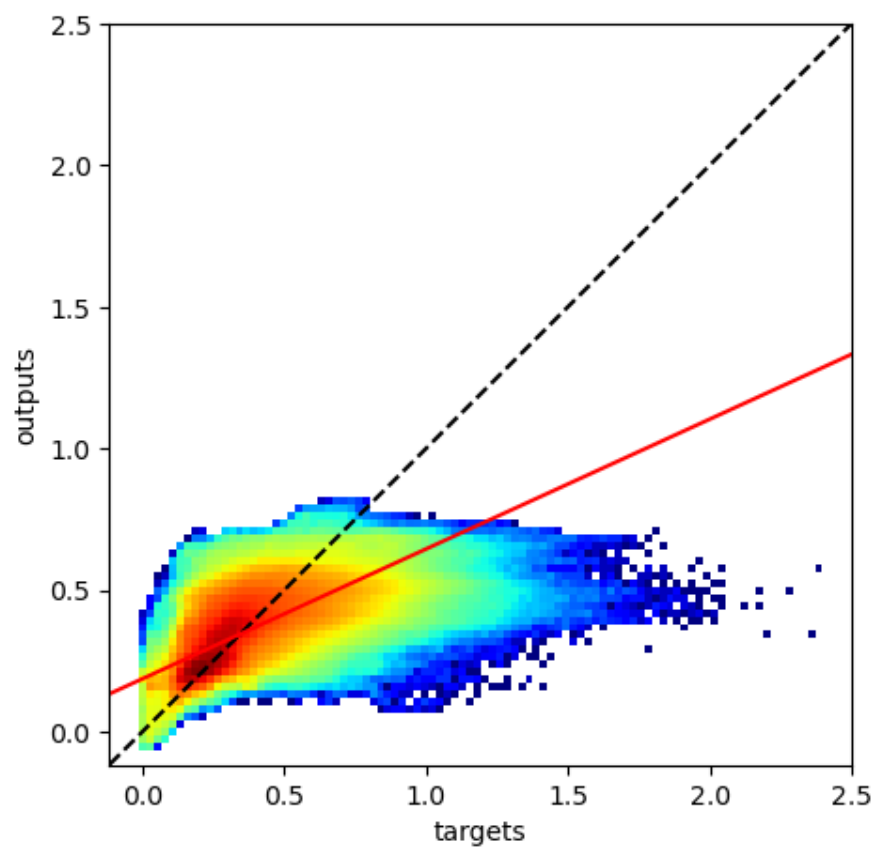
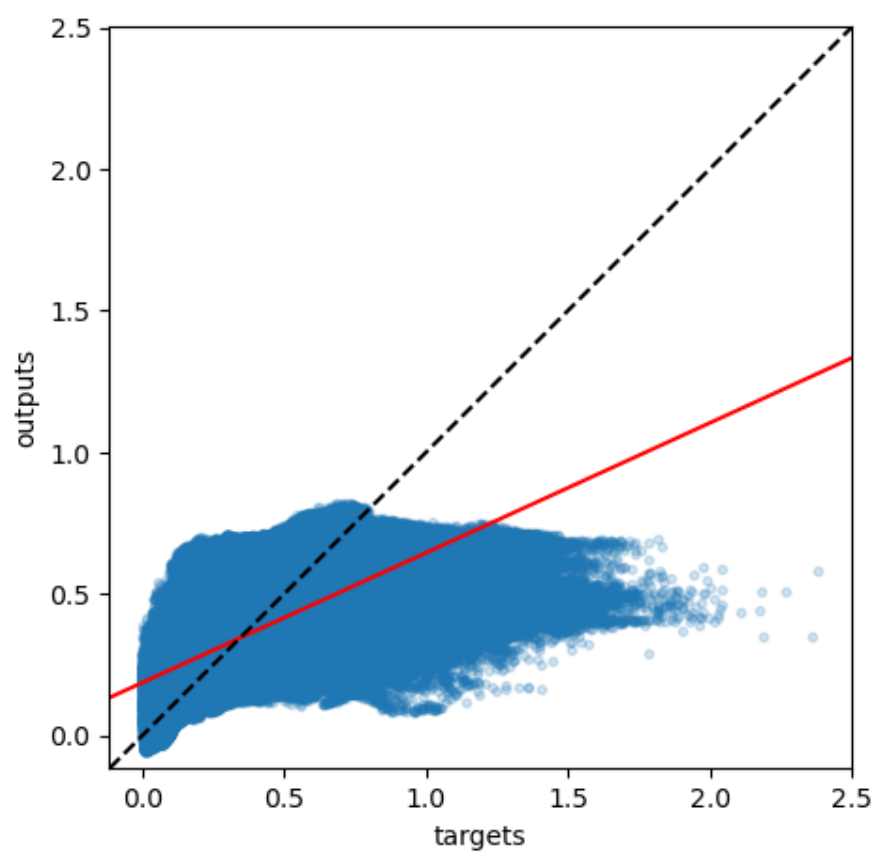
The amount of data points is 13990179
The slope of the best fitting line is 0.45
The correlation coefficient is: 0.67
The mean square error is: 0.01487

Diatom 2010



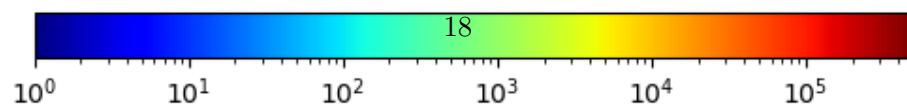
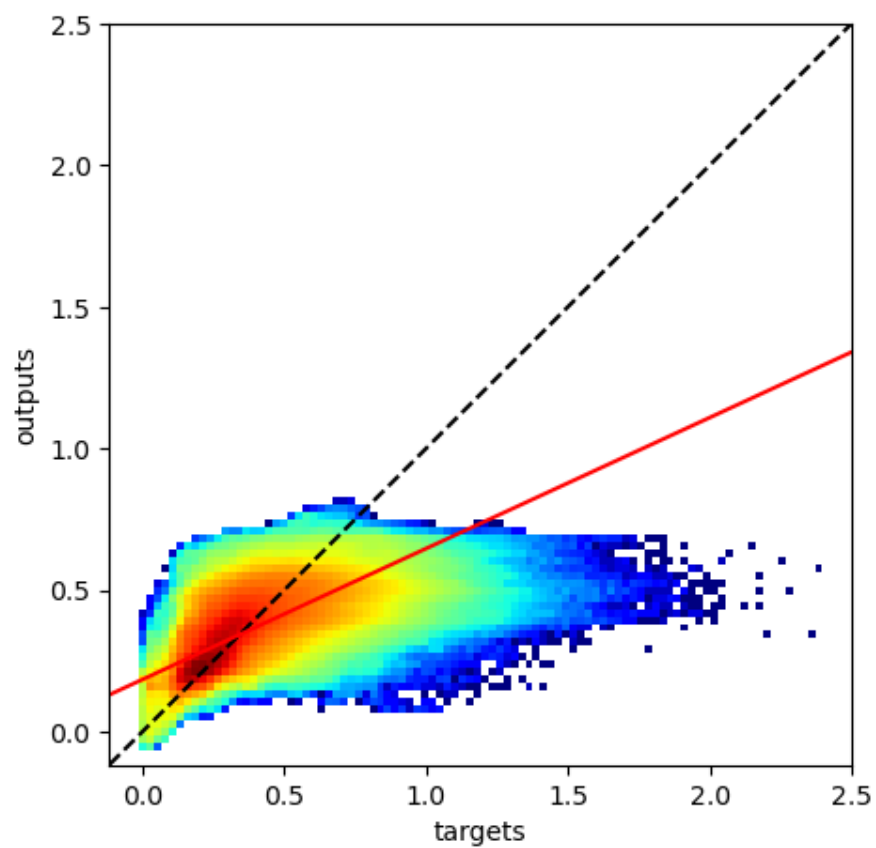
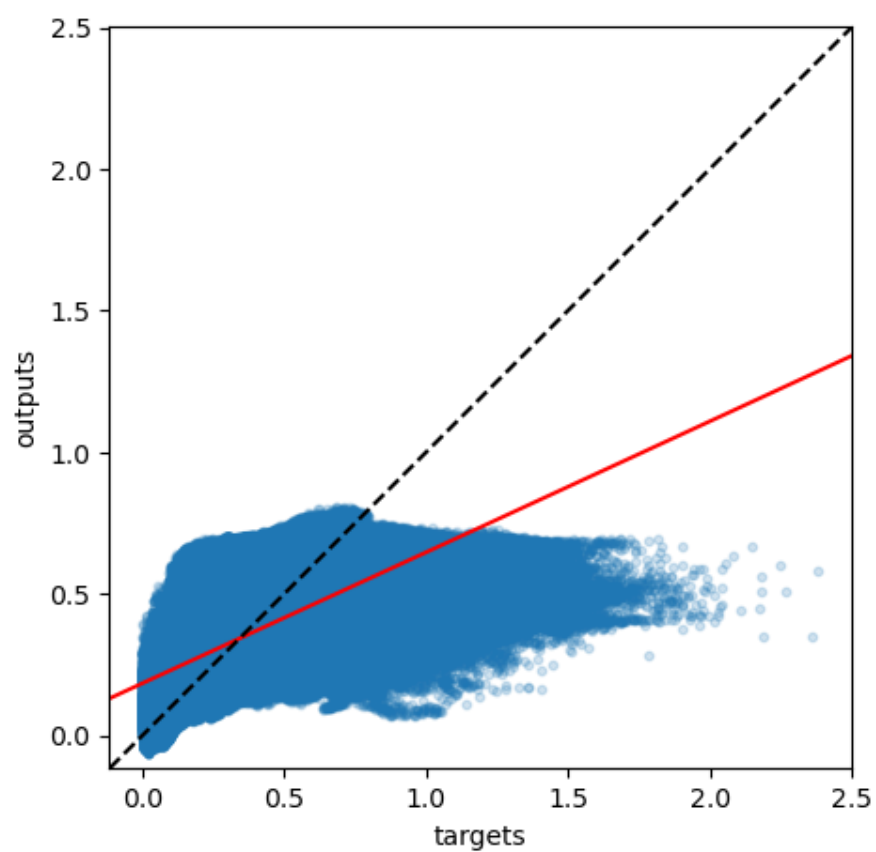
The amount of data points is 17476104
The slope of the best fitting line is 0.458
The correlation coefficient is: 0.672
The mean square error is: 0.01471

Diatom 2011



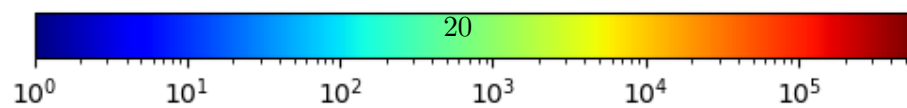
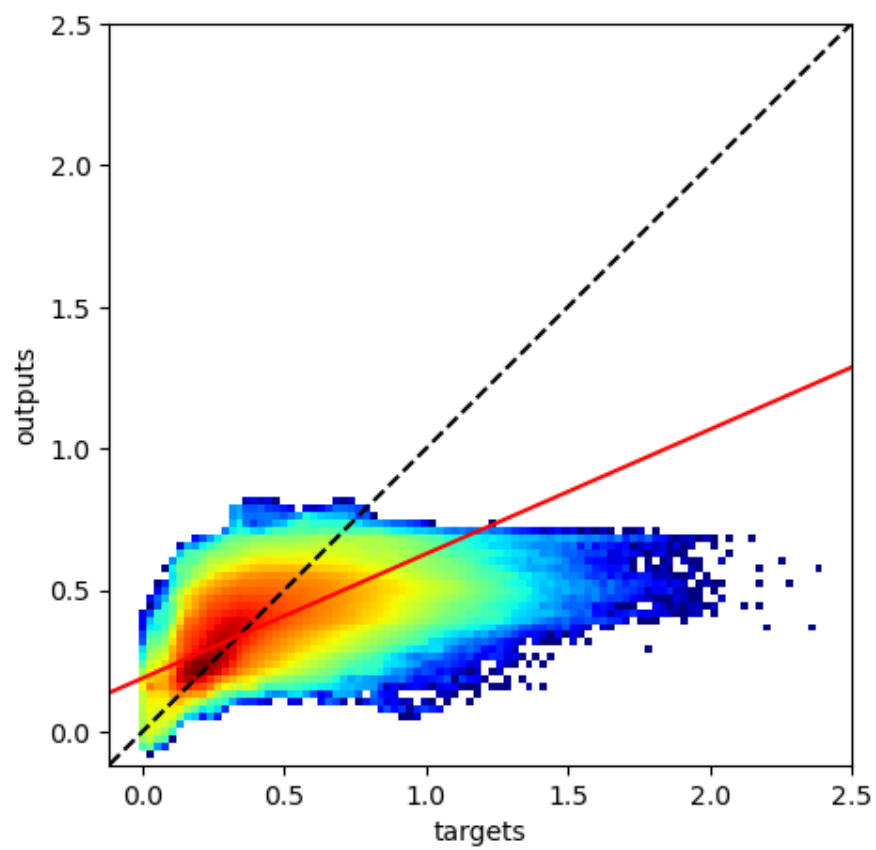
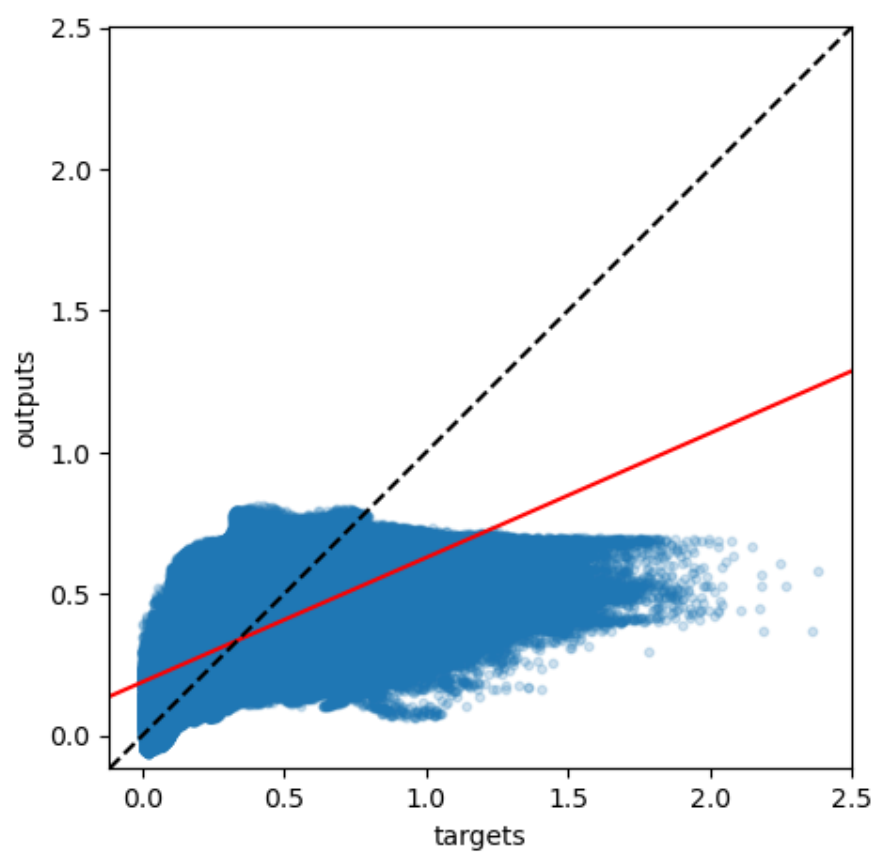
The amount of data points is 21008508
The slope of the best fitting line is 0.462
The correlation coefficient is: 0.677
The mean square error is: 0.01441

Diatom 2012



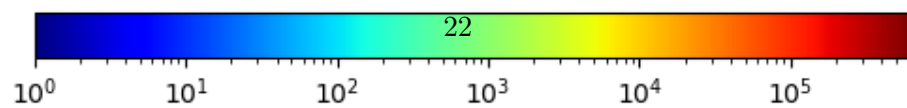
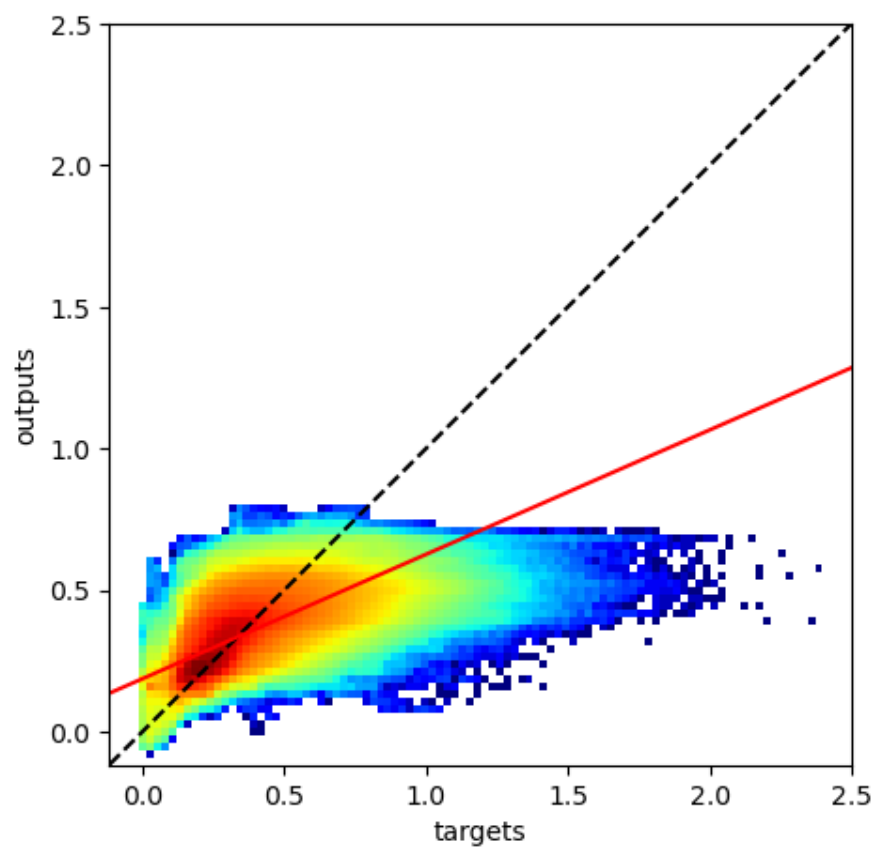
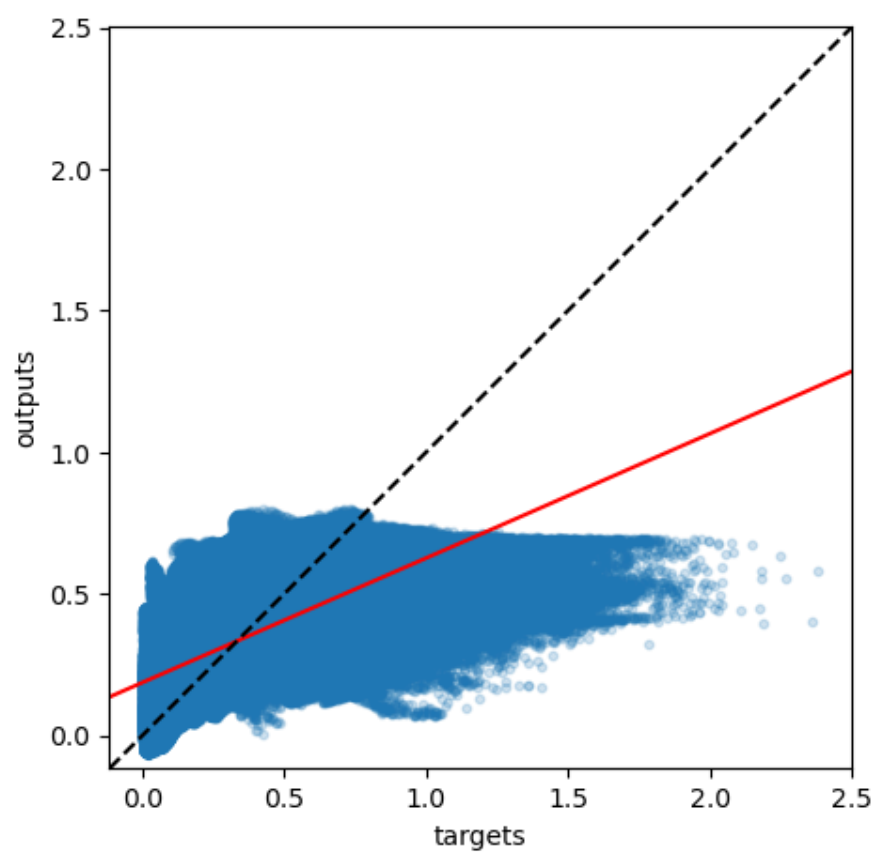
The amount of data points is 24494433
The slope of the best fitting line is 0.439
The correlation coefficient is: 0.666
The mean square error is: 0.01517

Diatom 2013



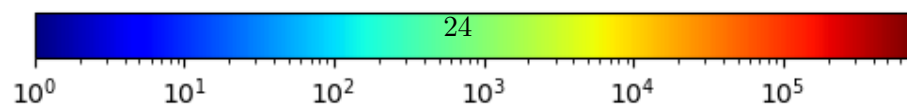
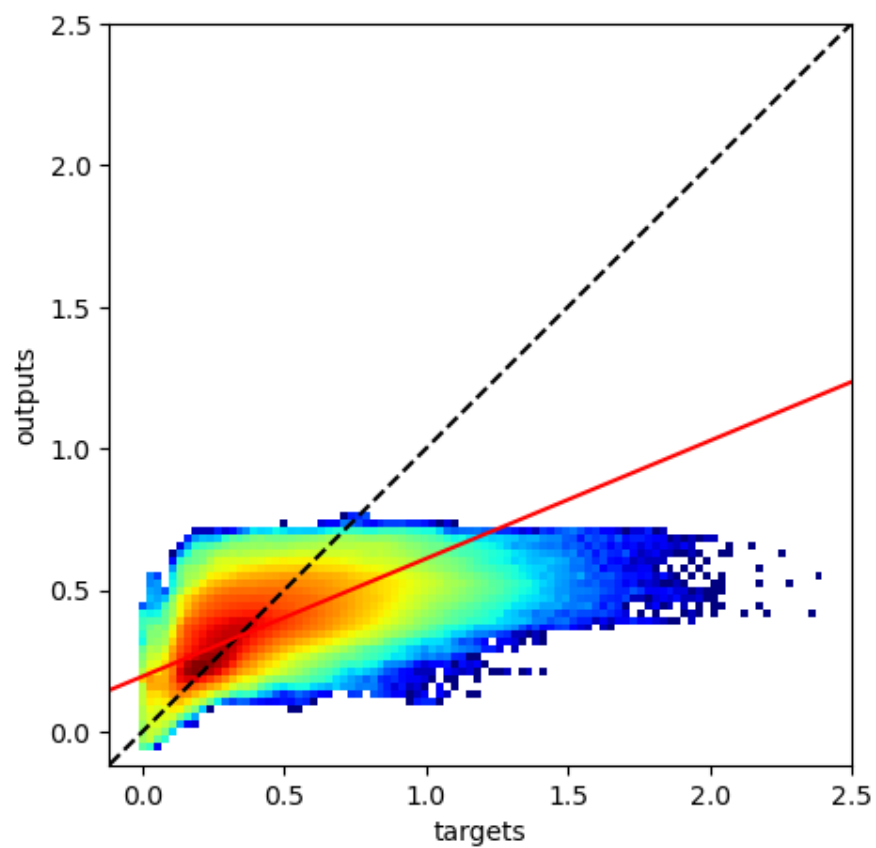
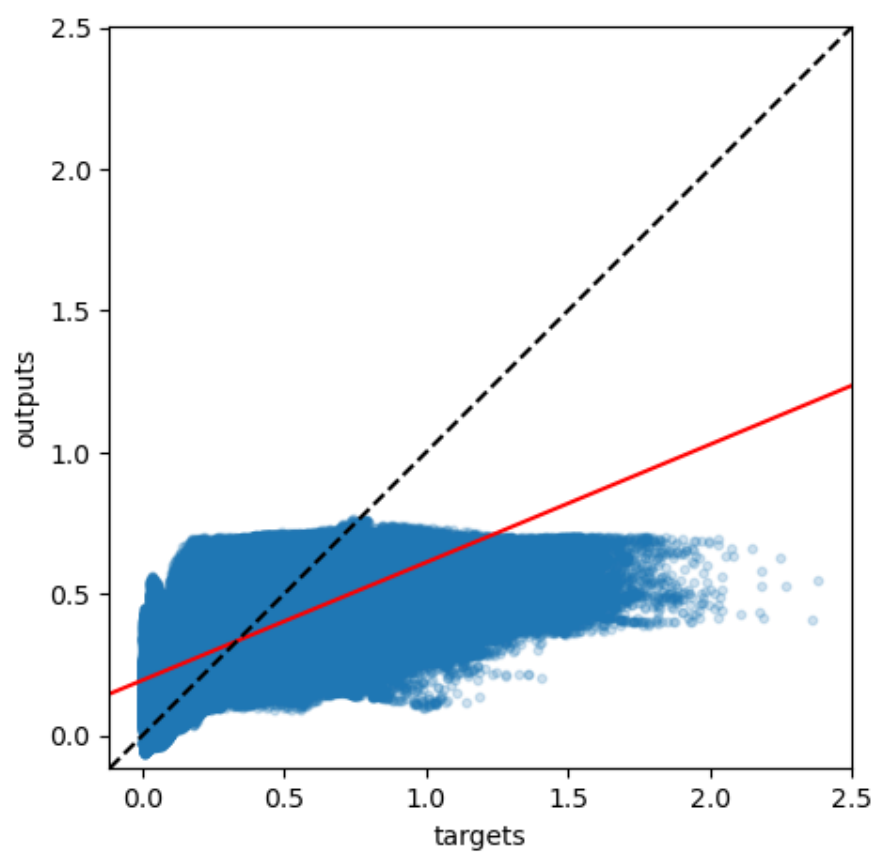
The amount of data points is 27980358
The slope of the best fitting line is 0.439
The correlation coefficient is: 0.662
The mean square error is: 0.01483

Diatom 2014



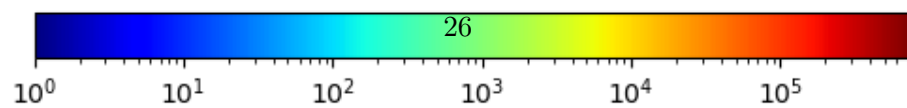
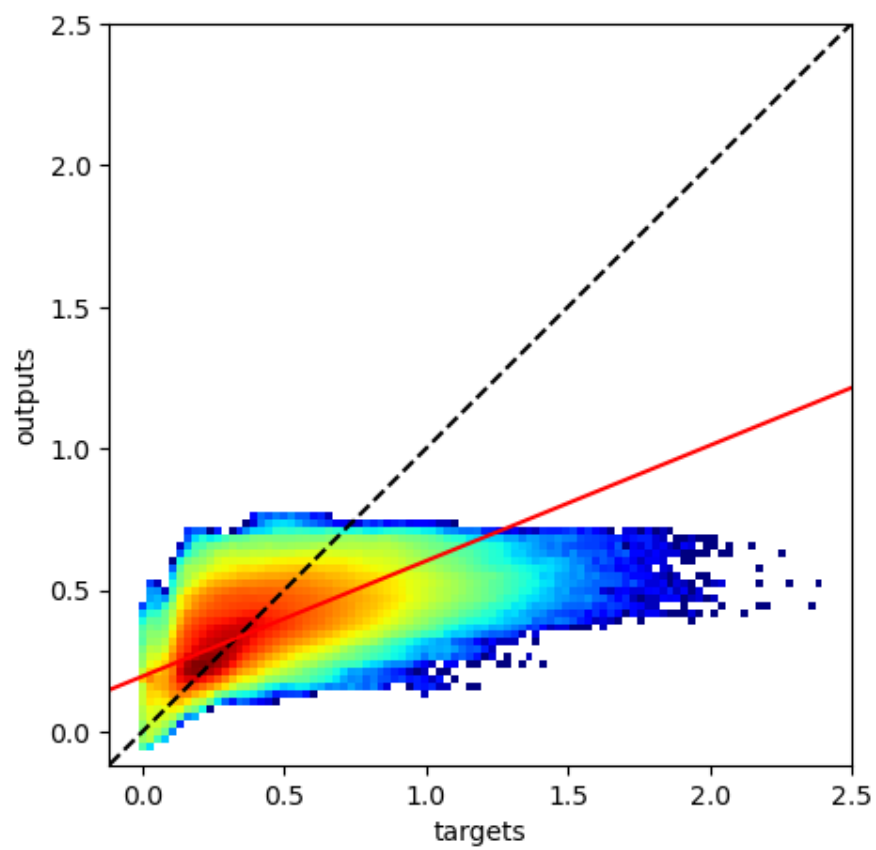
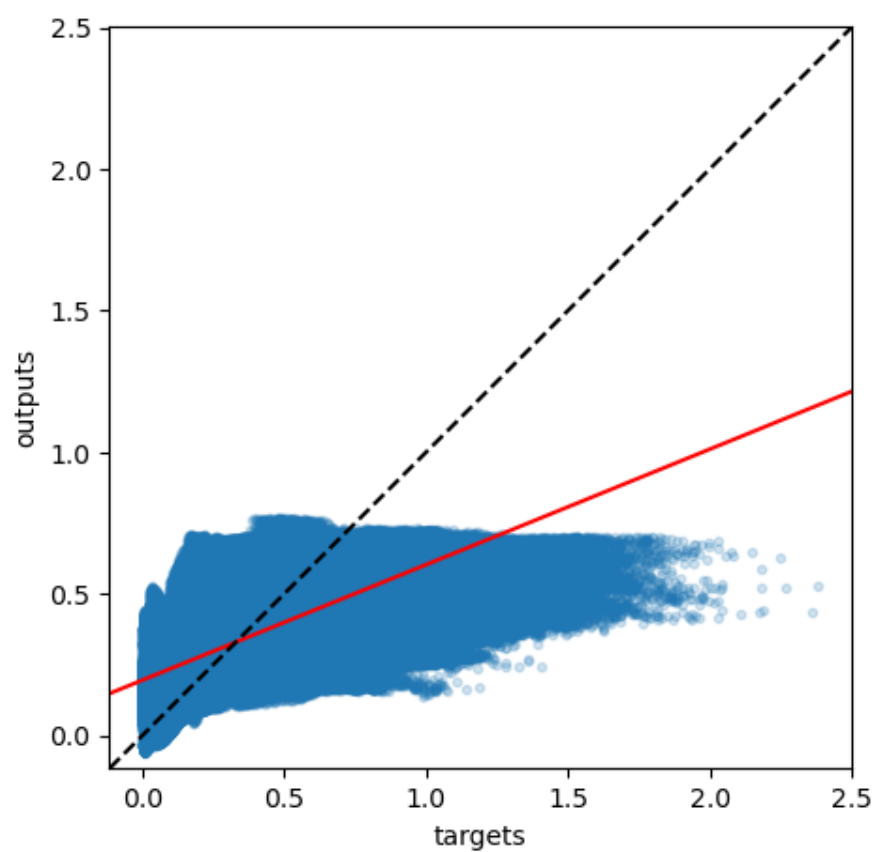
The amount of data points is 31466283
The slope of the best fitting line is 0.416
The correlation coefficient is: 0.664
The mean square error is: 0.01463

Diatom 2015



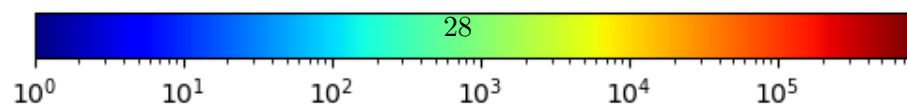
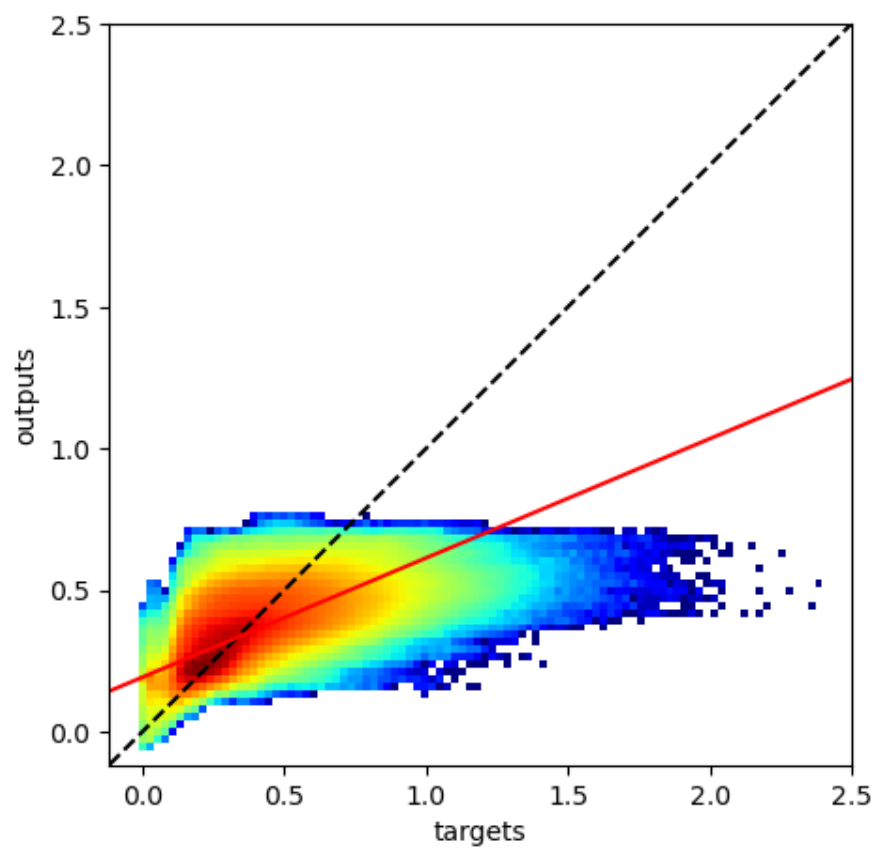
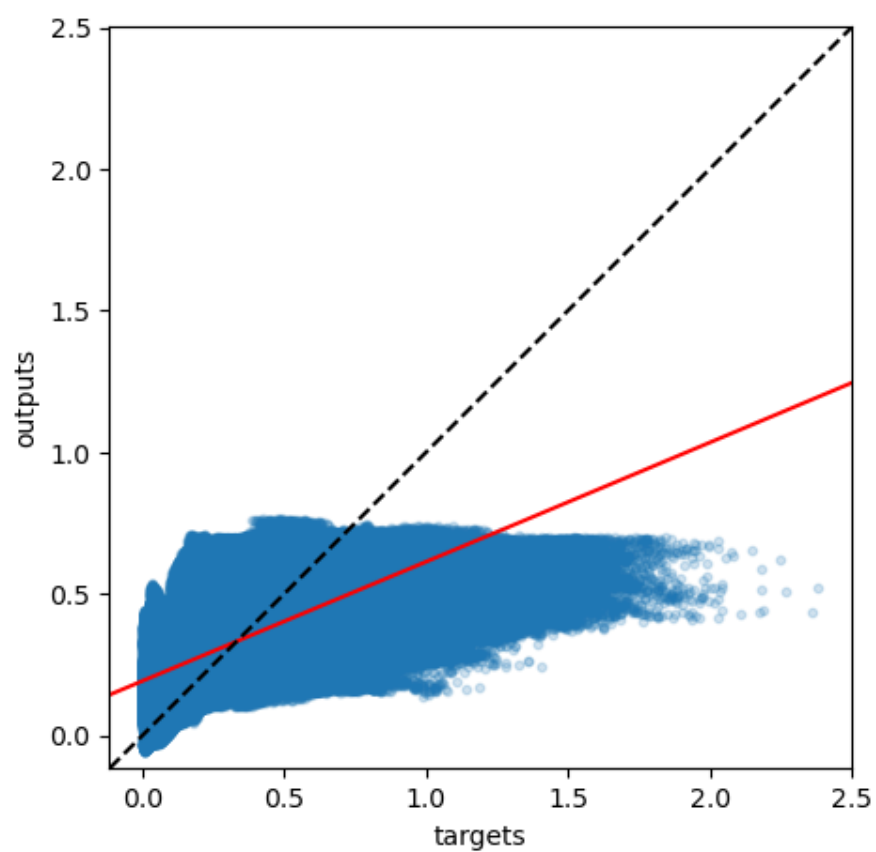
The amount of data points is 34998687
The slope of the best fitting line is 0.408
The correlation coefficient is: 0.659
The mean square error is: 0.01494

Diatom 2016



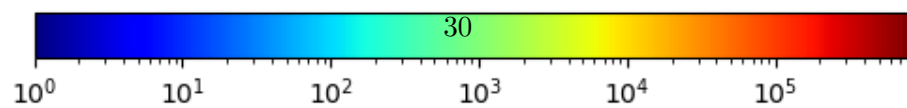
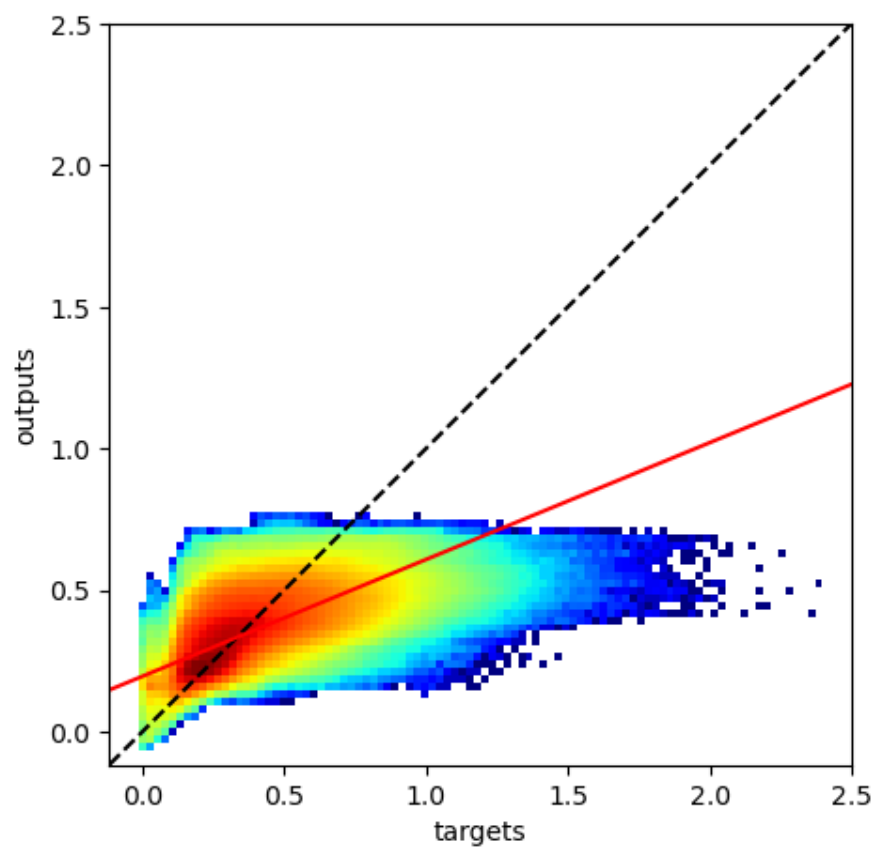
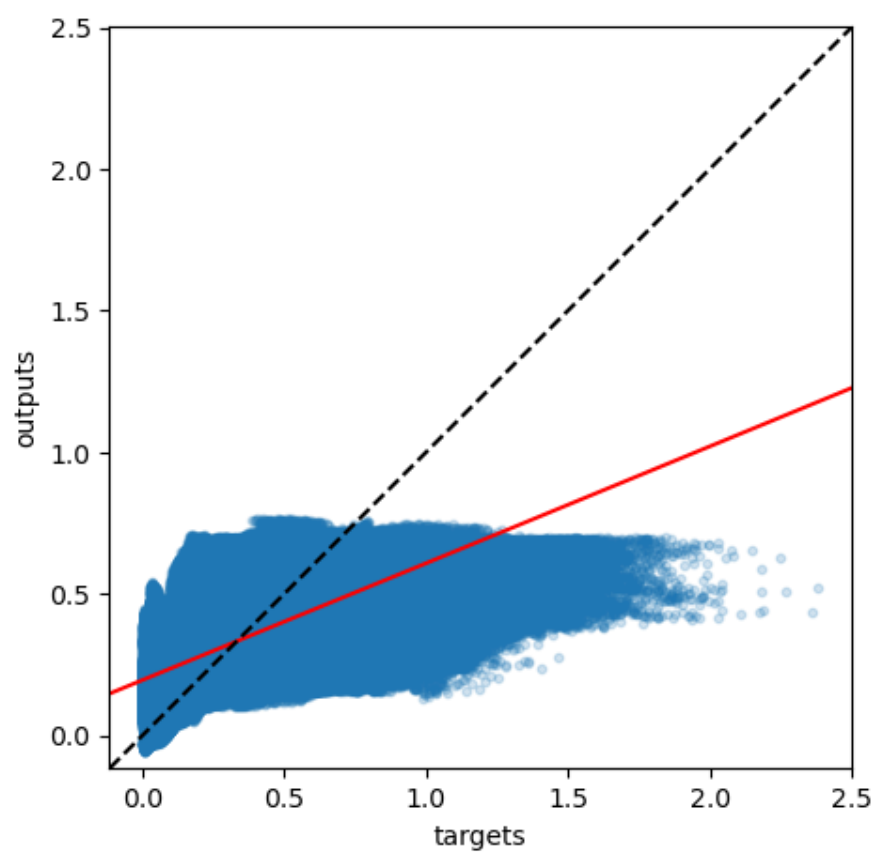
The amount of data points is 38484612
The slope of the best fitting line is 0.421
The correlation coefficient is: 0.66
The mean square error is: 0.01447

Diatom 2017



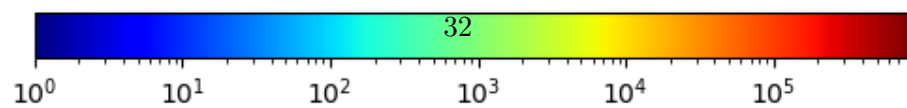
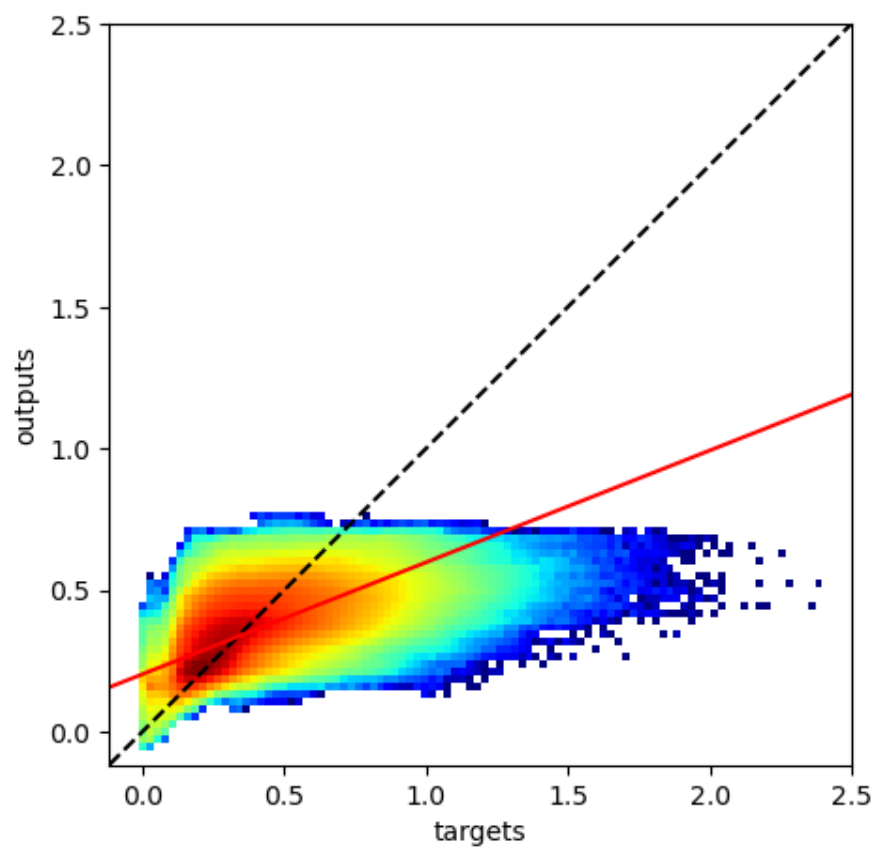
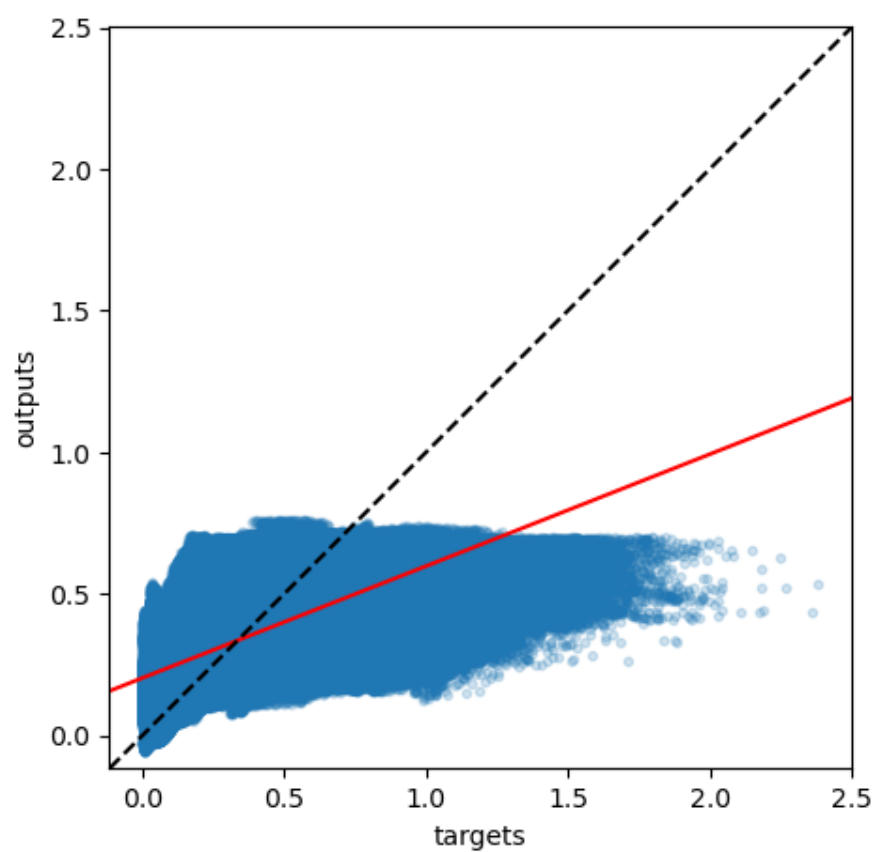
The amount of data points is 41970537
The slope of the best fitting line is 0.413
The correlation coefficient is: 0.648
The mean square error is: 0.01484

Diatom 2018



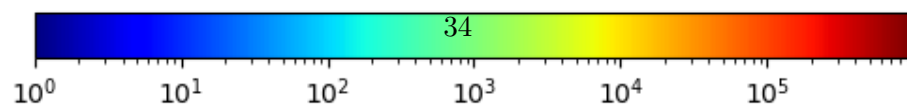
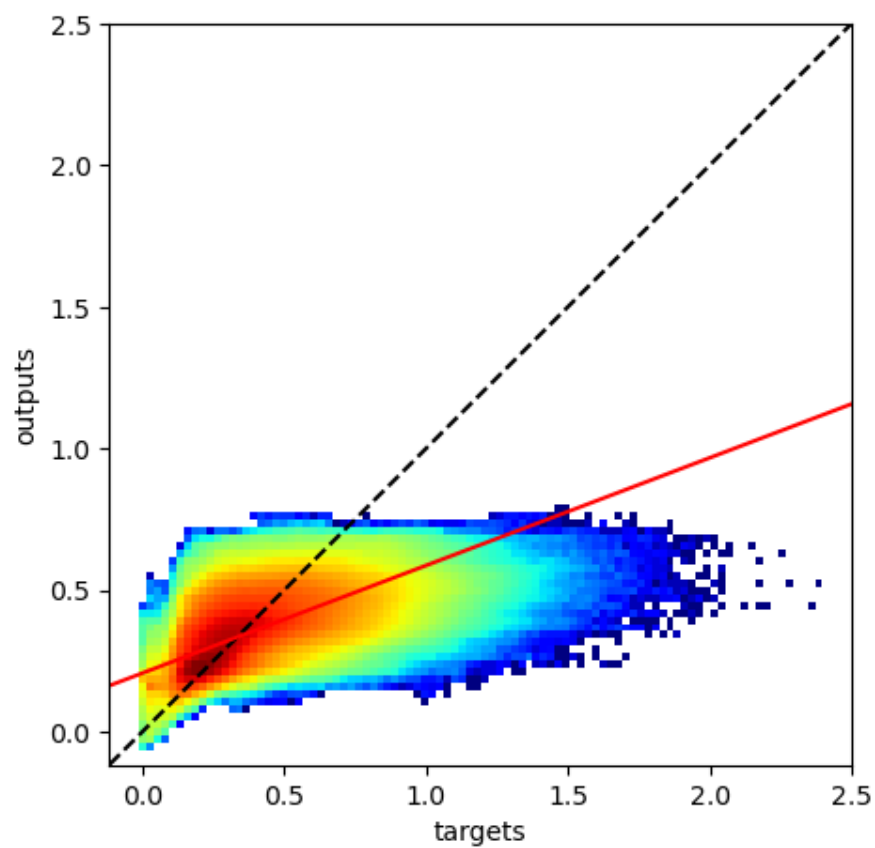
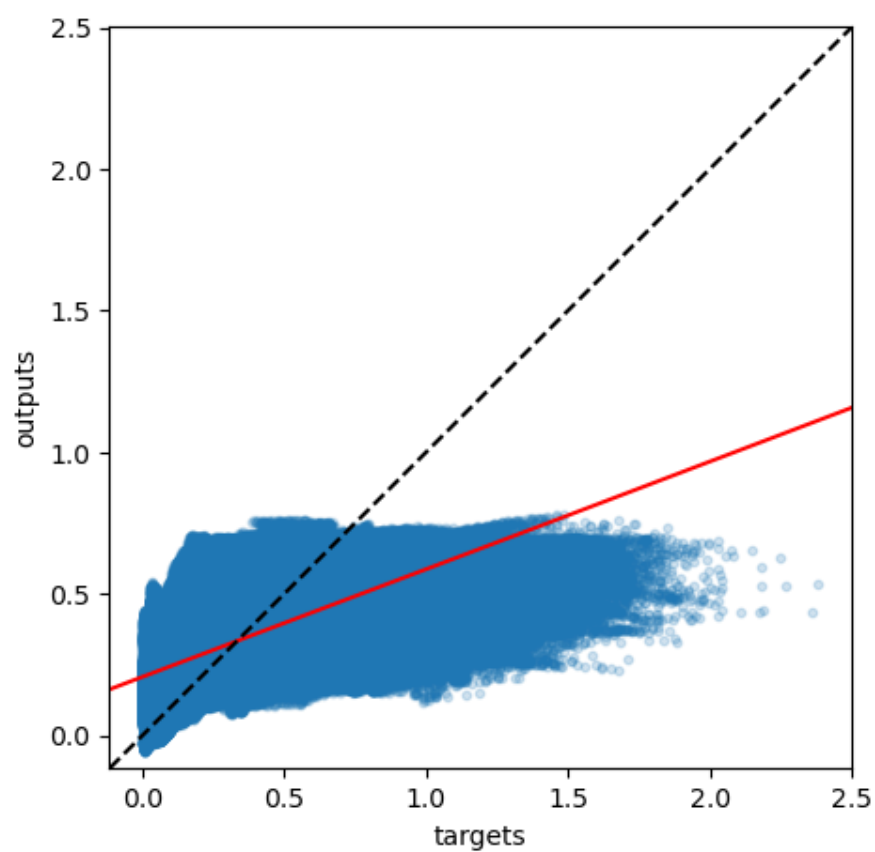
The amount of data points is 45456462
The slope of the best fitting line is 0.395
The correlation coefficient is: 0.63
The mean square error is: 0.01573

Diatom 2019



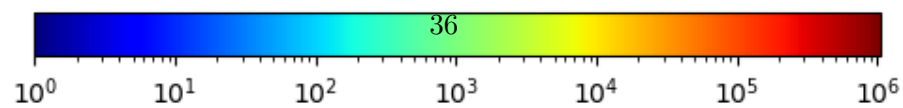
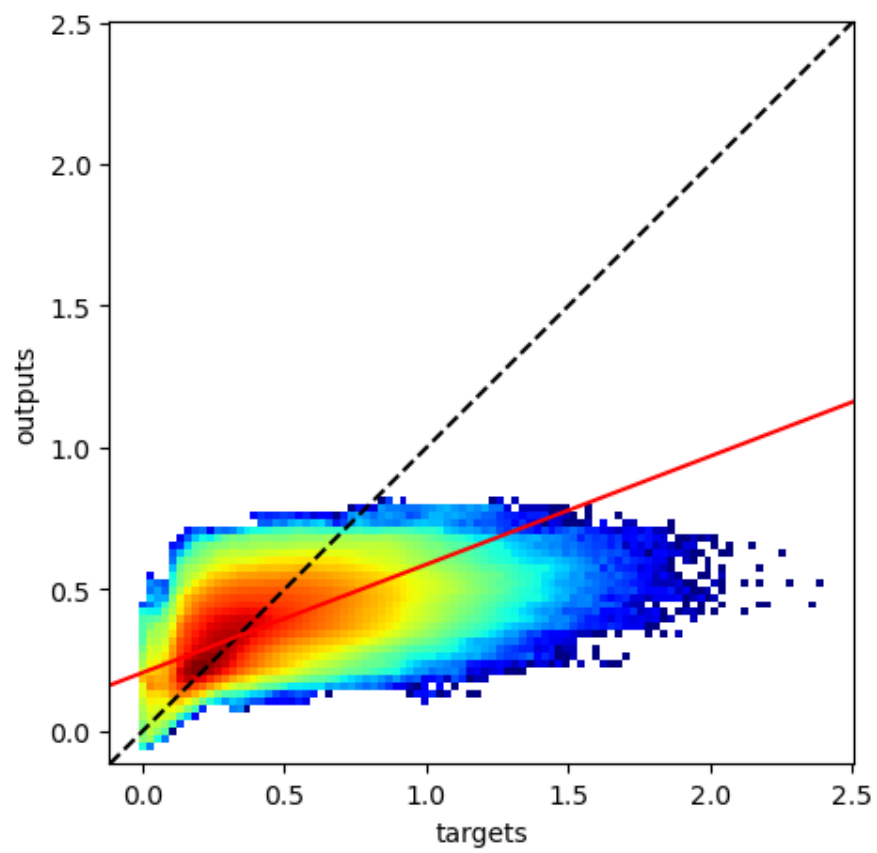
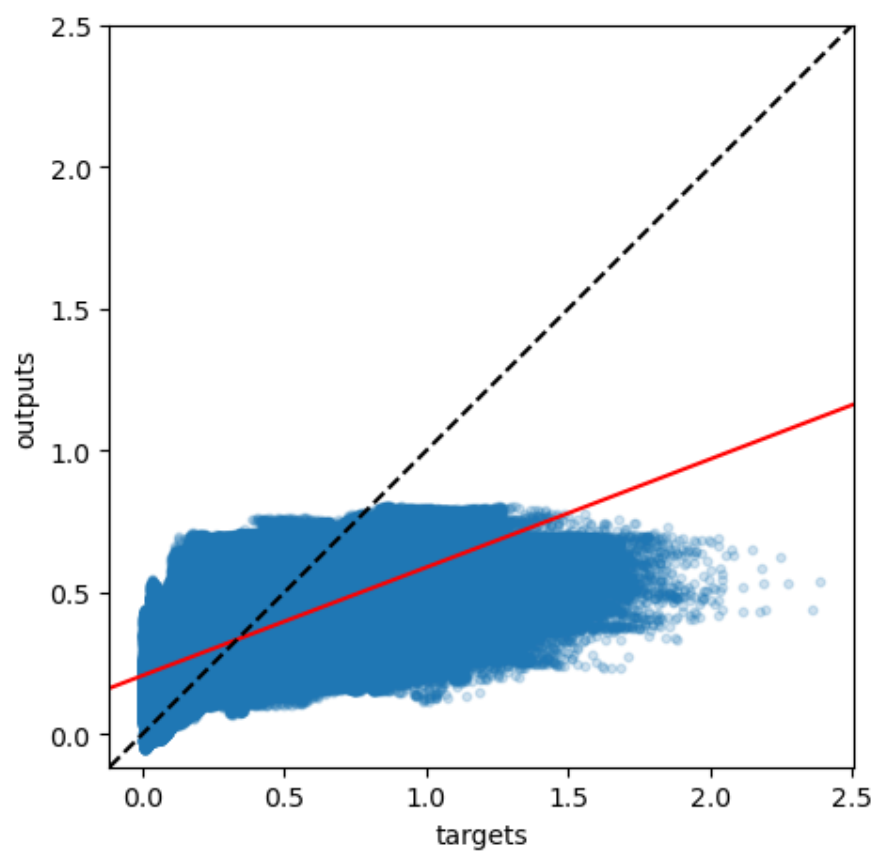
The amount of data points is 48988866
The slope of the best fitting line is 0.38
The correlation coefficient is: 0.624
The mean square error is: 0.01662

Diatom 2020



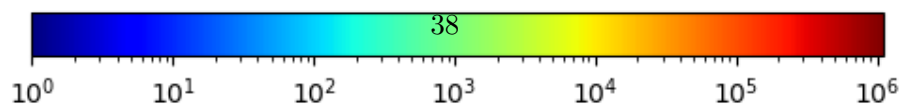
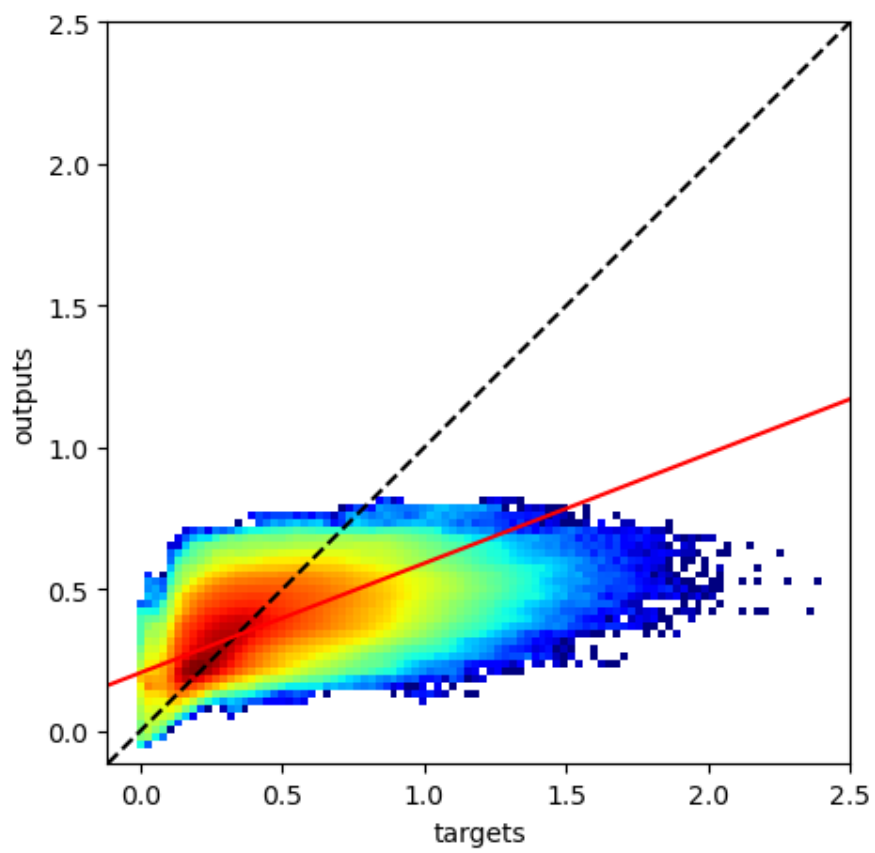
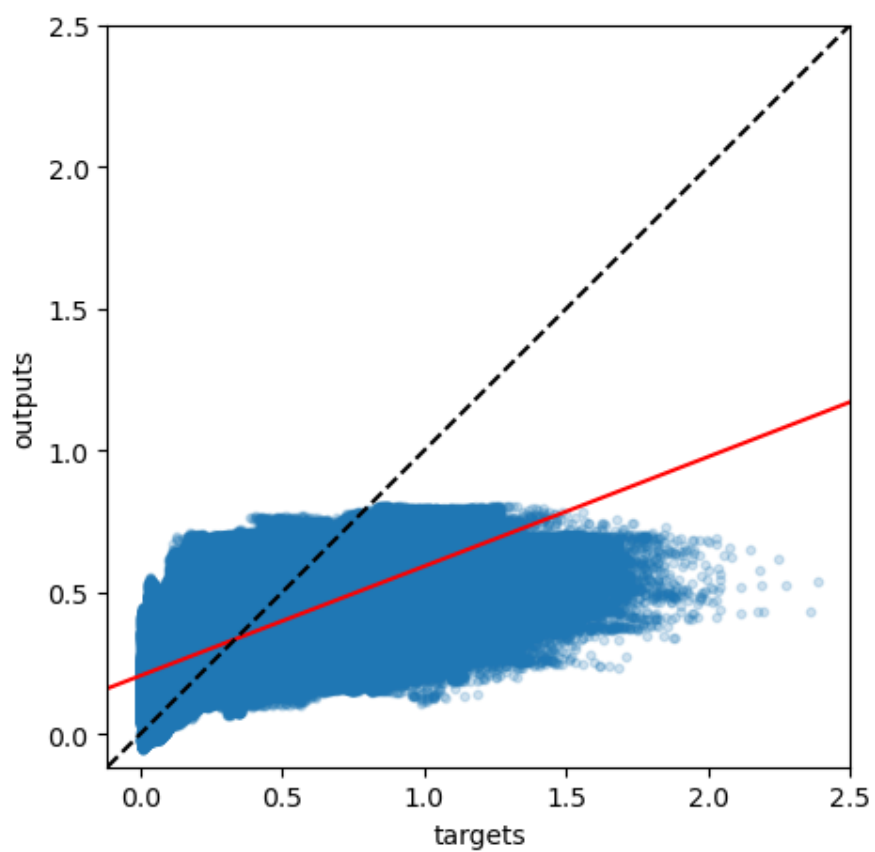
The amount of data points is 52474791
The slope of the best fitting line is 0.382
The correlation coefficient is: 0.632
The mean square error is: 0.01653

Diatom 2021



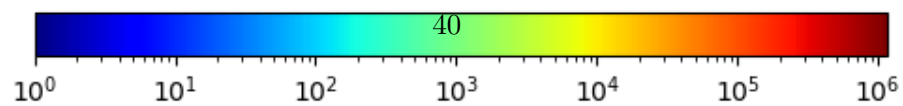
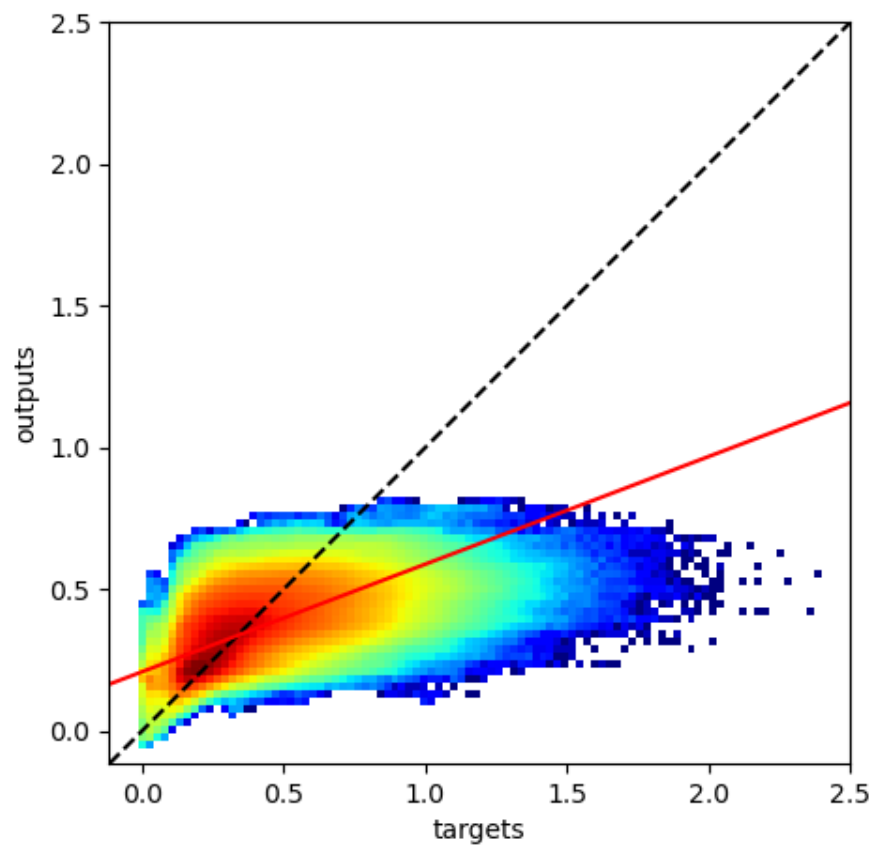
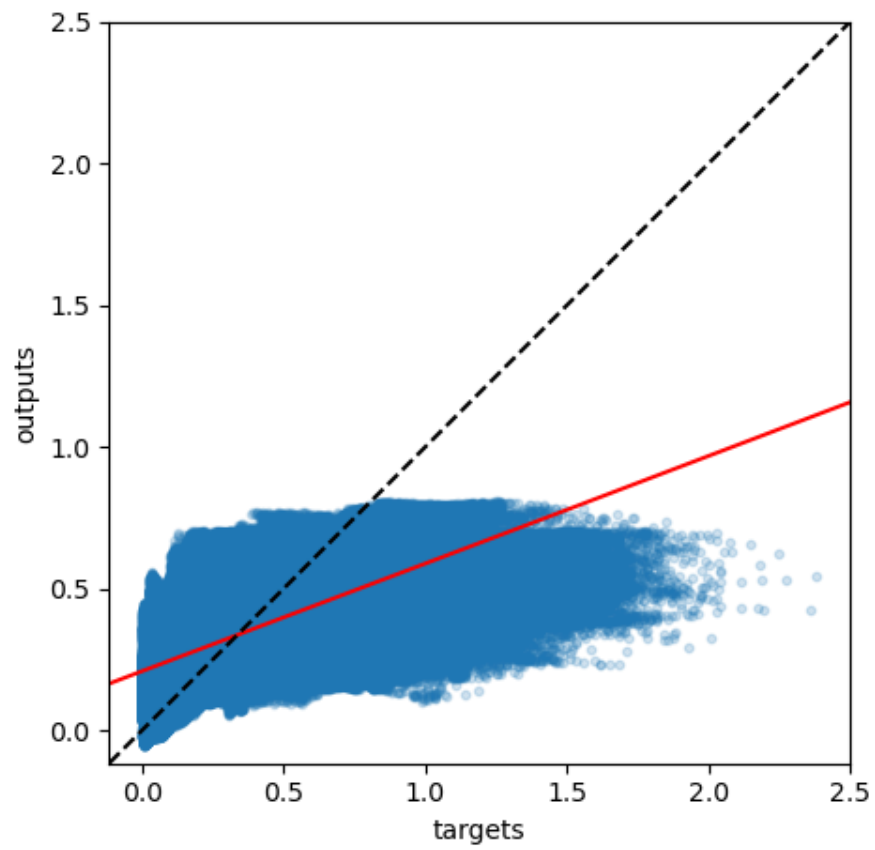
The amount of data points is 55960716
The slope of the best fitting line is 0.386
The correlation coefficient is: 0.634
The mean square error is: 0.01624

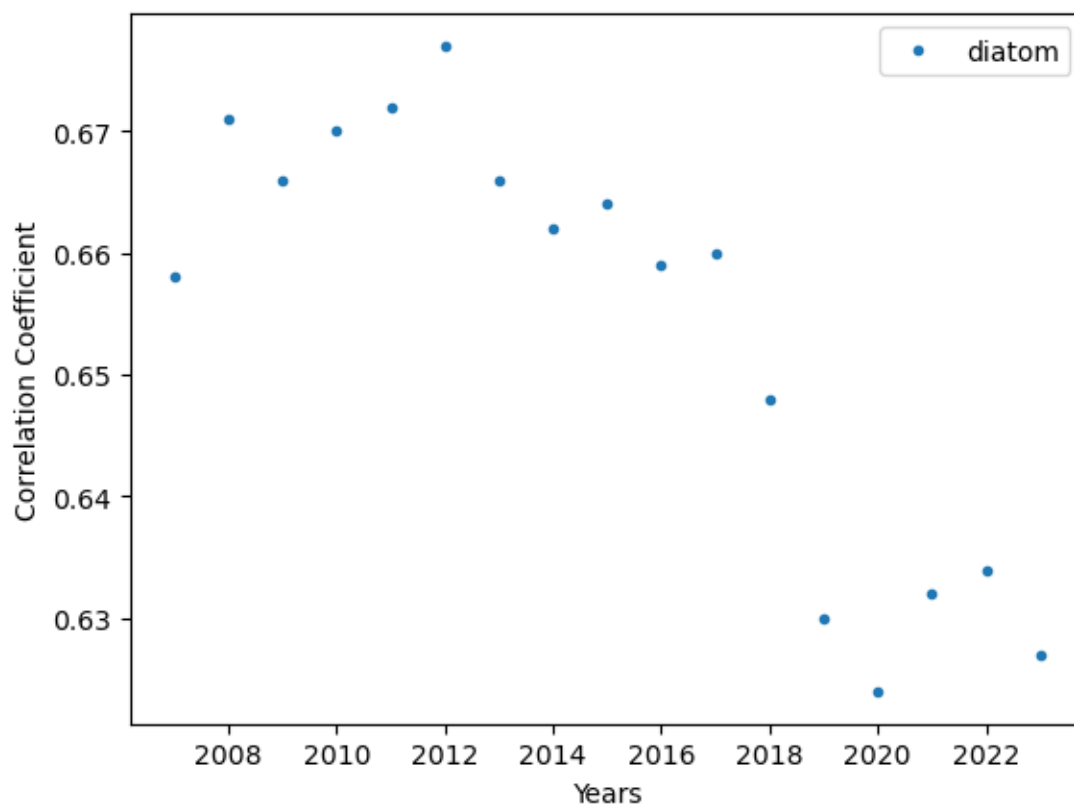
Diatom 2022

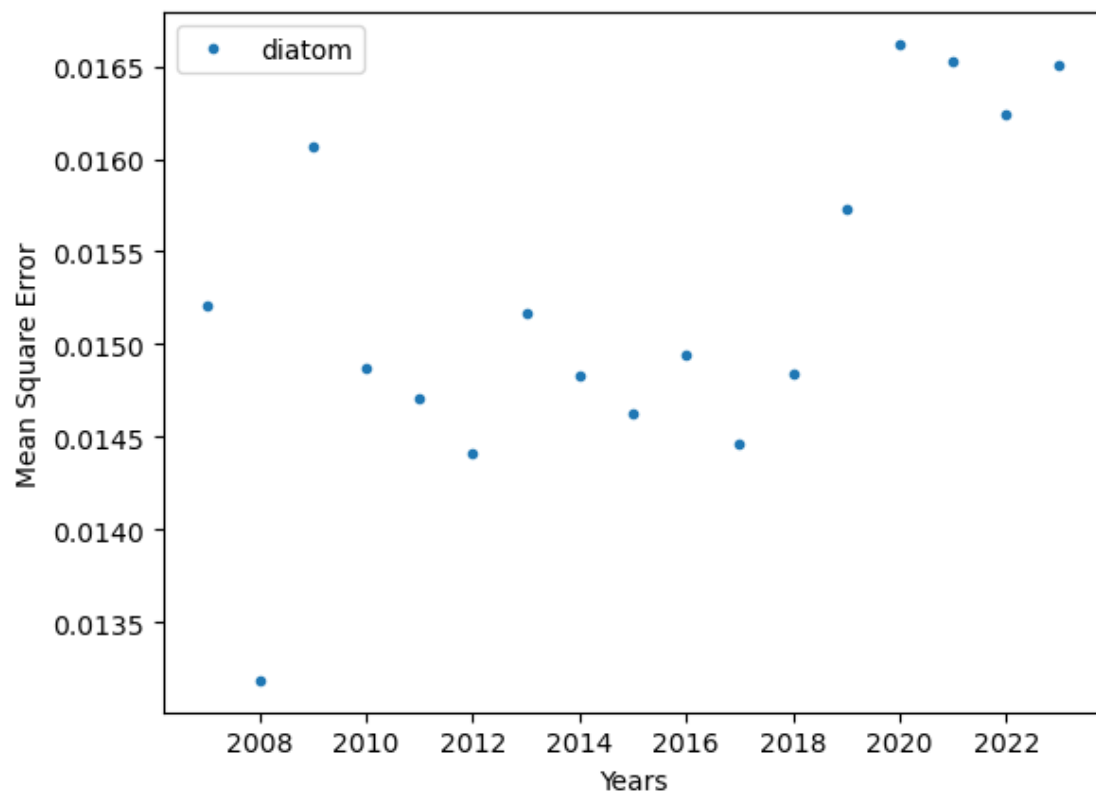


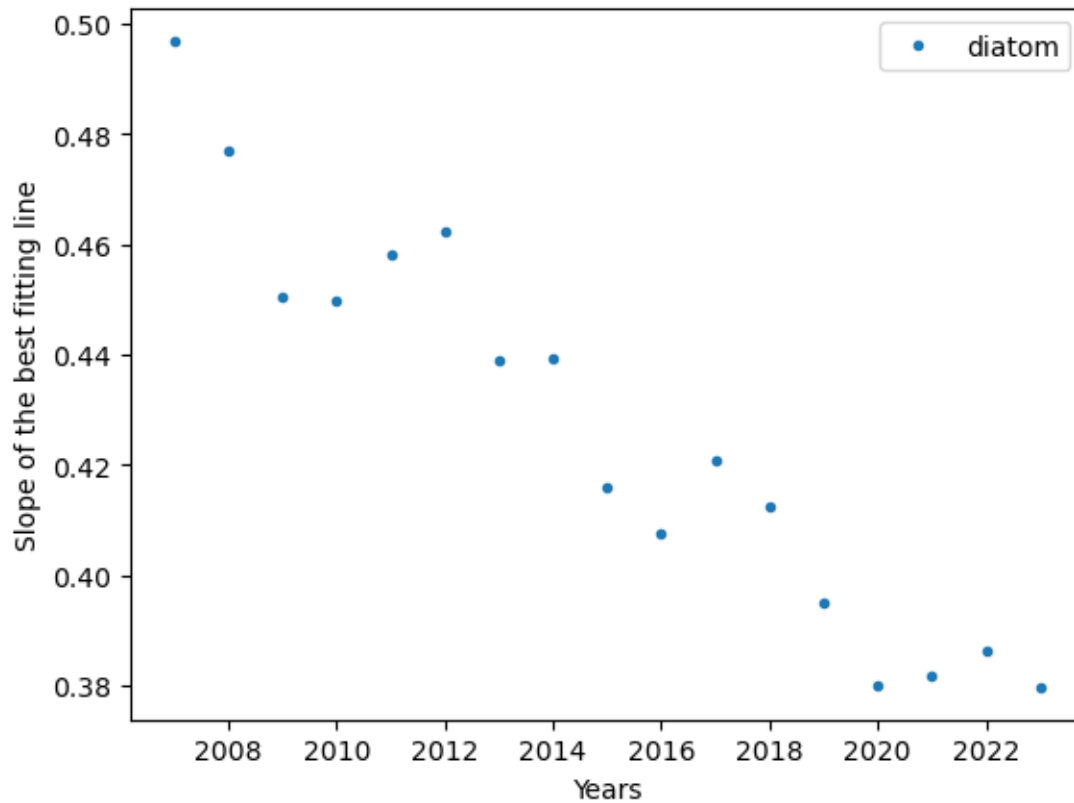
The amount of data points is 59446641
The slope of the best fitting line is 0.38
The correlation coefficient is: 0.627
The mean square error is: 0.0165

Diatom 2023









1.7 Other Years (Daily)

```
[ ]: r_all2 = np.array([])
rms_all2 = np.array([])
slope_all2 = np.array([])

for i in tqdm(range(0, len(ds.time_counter))):

    dataset = ds.isel(time_counter=i)

    drivers, diat, _ = datasets_preparation(dataset)

    r, rms, m = regressor3(drivers, diat)

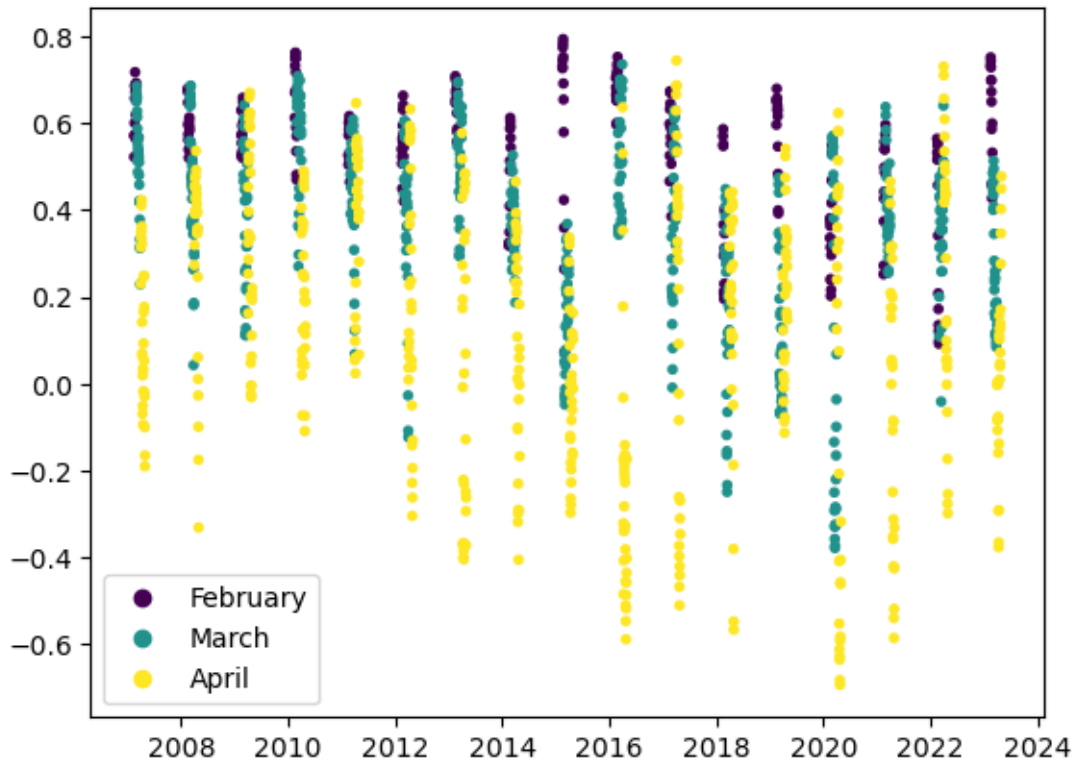
    r_all2 = np.append(r_all2, r)
    rms_all2 = np.append(rms_all2, rms)
    slope_all2 = np.append(slope_all2, m)

plotting2(r_all2, 'Correlation Coefficients')
plotting2(rms_all2, 'Mean Square Errors')
```

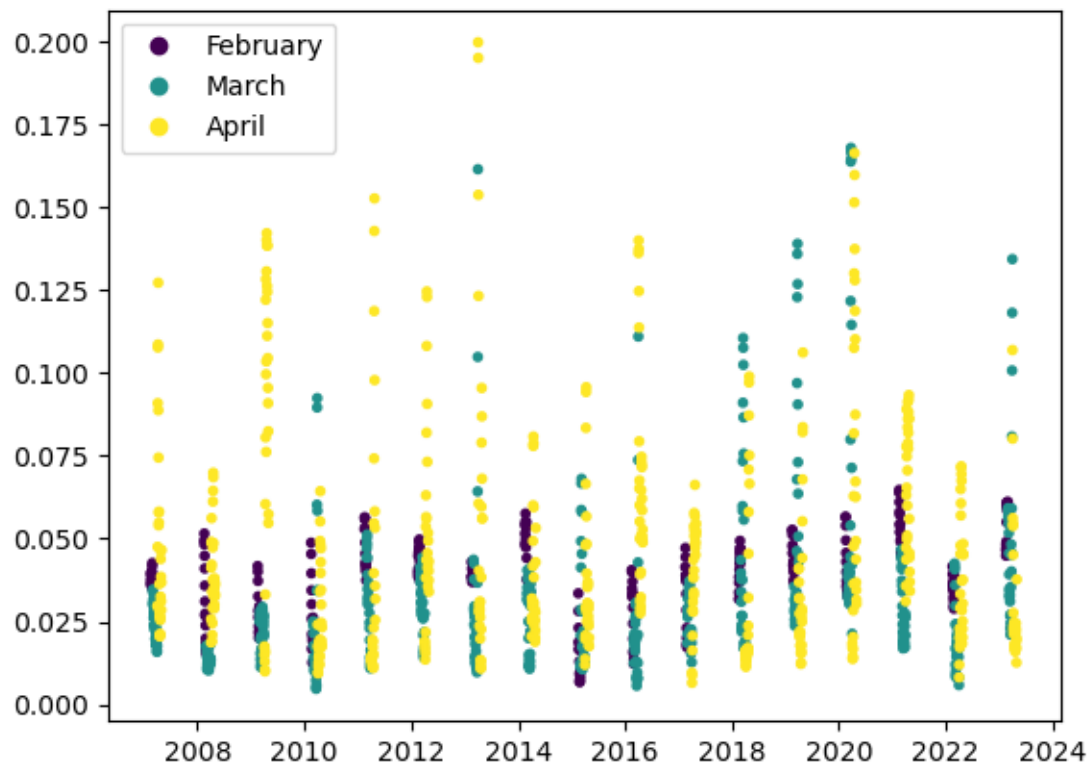
```
plotting2(slope_all2, 'Slope of the best fitting line')
```

0% | 0/1279 [00:00<?, ?it/s]

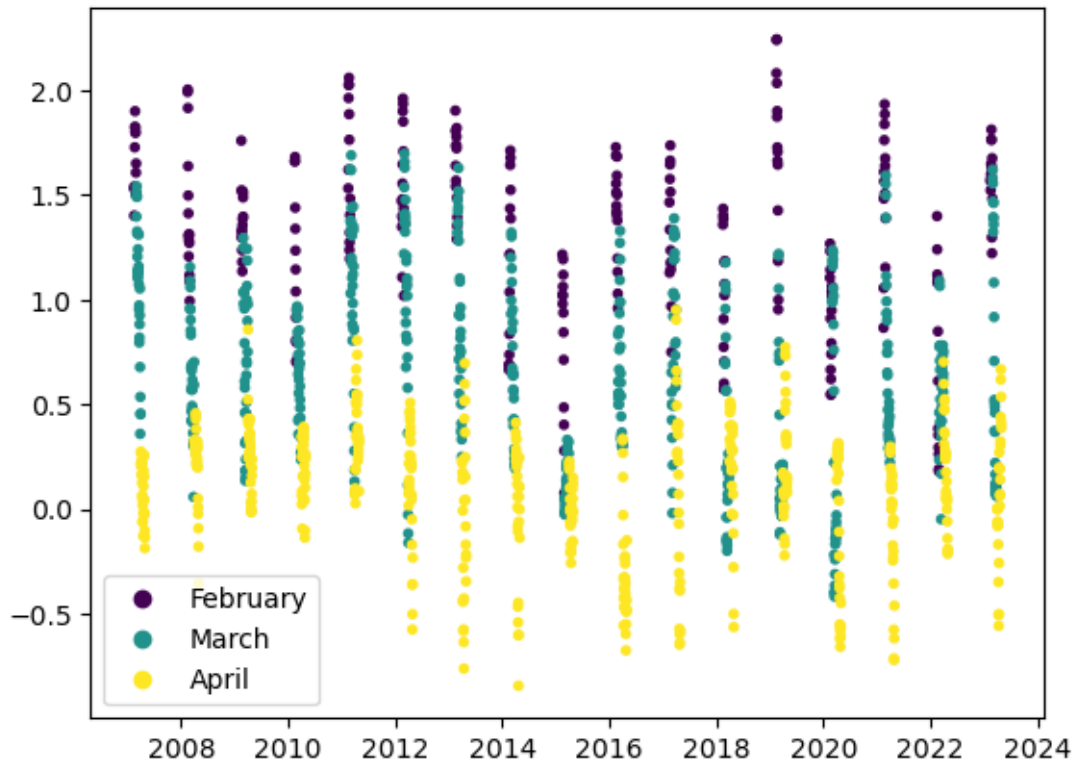
Daily Correlation Coefficients (15 Feb - 30 Apr)



Daily Mean Square Errors (15 Feb - 30 Apr)



Daily Slope of the best fitting line (15 Feb - 30 Apr)



2 Daily Maps

```
[ ]: maps = random.sample(range(0,100),10)

for i in tqdm(maps):

    dataset = ds.isel(time_counter=i)
    drivers, diat, indx = datasets_preparation(dataset)

    diat_i = dataset['Diatom']

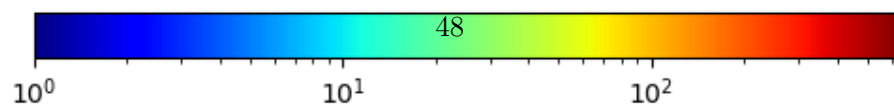
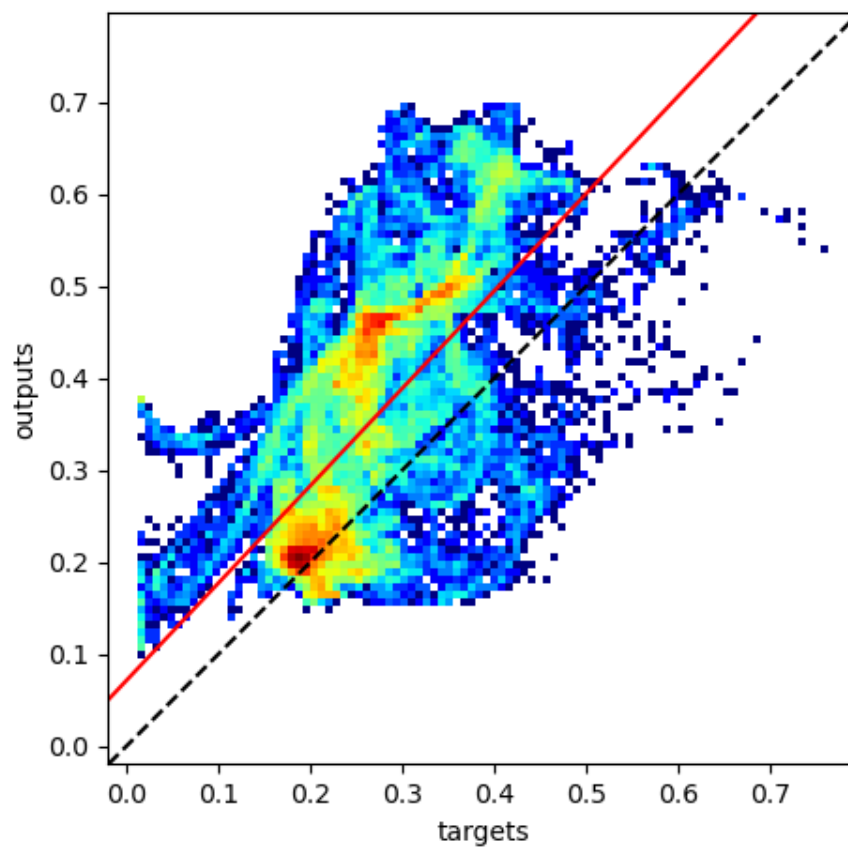
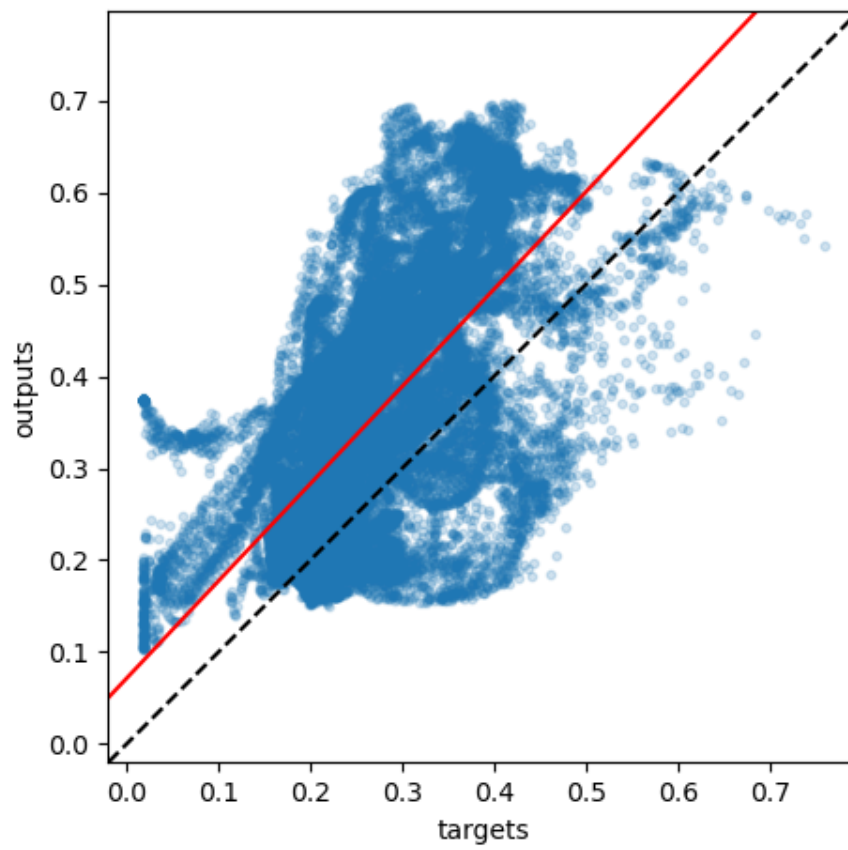
    regressor4(drivers, diat, 'Diatom ')
```

0%| | 0/10 [00:00<?, ?it/s]

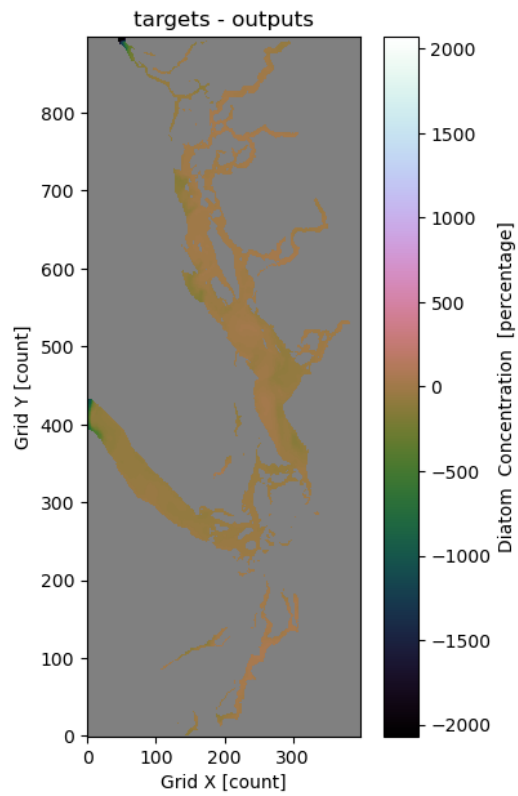
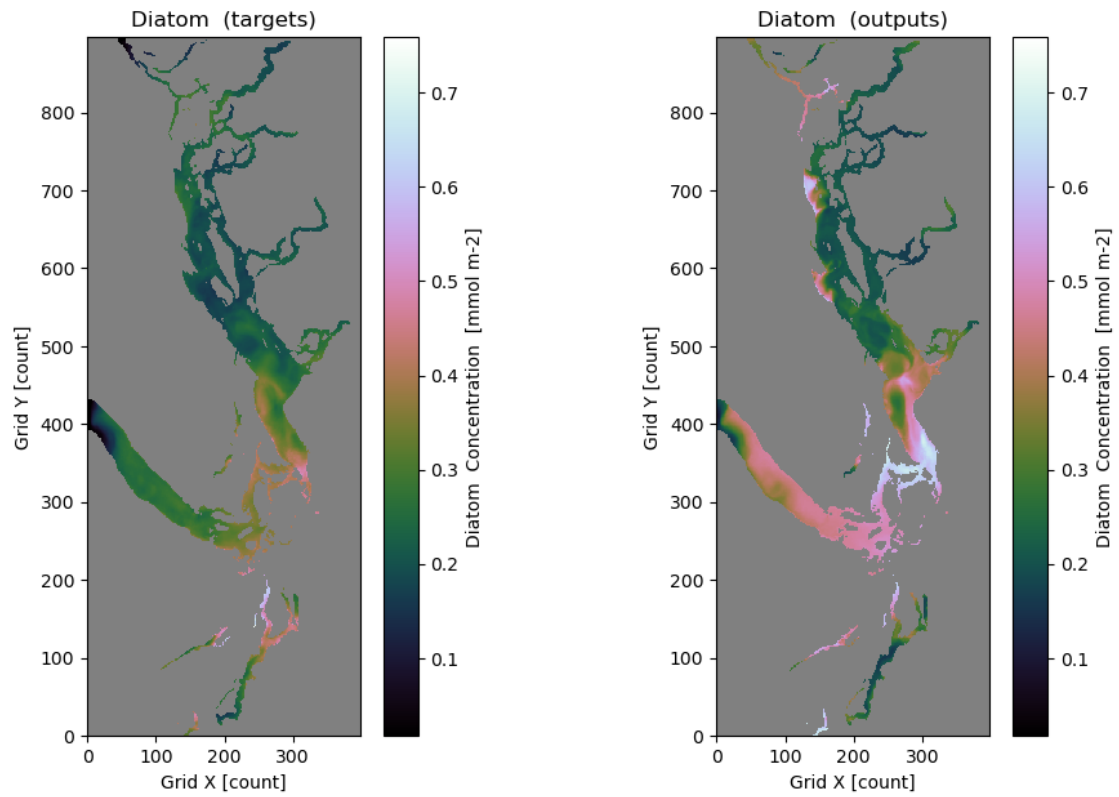
The amount of data points is 46479
 The slope of the best fitting line is 1.059
 The correlation coefficient is: 0.654

The mean square error is: 0.01776

Diatom 2008-03-04

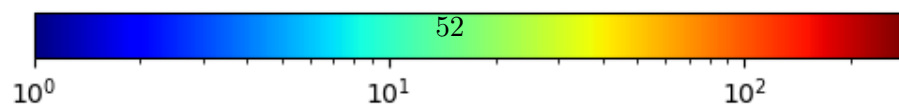
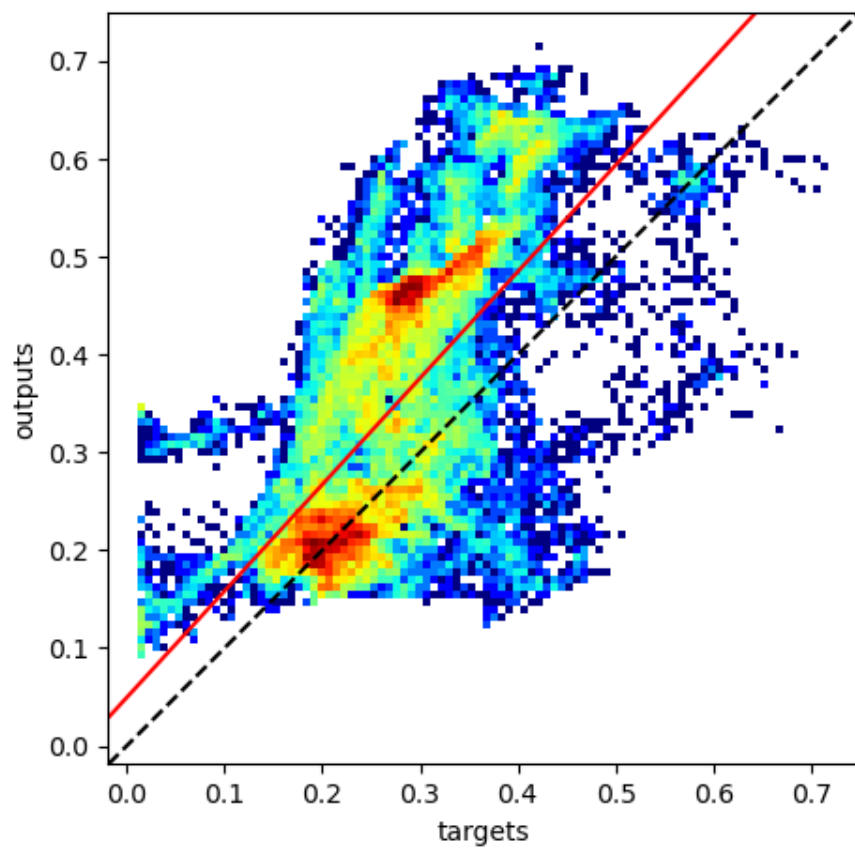
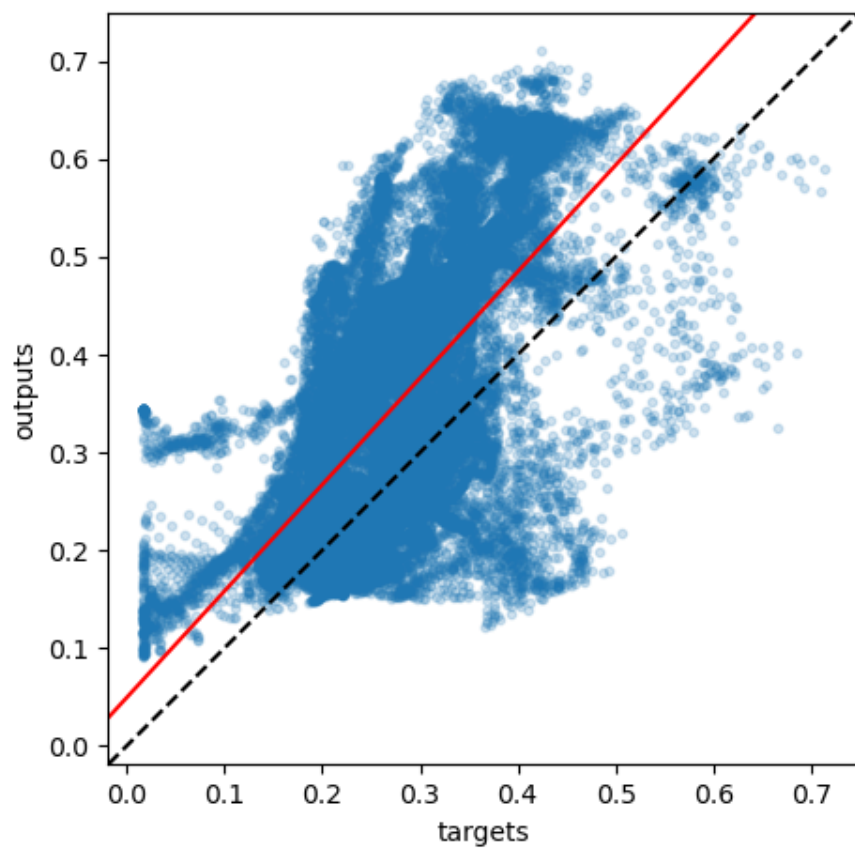


2008-03-04

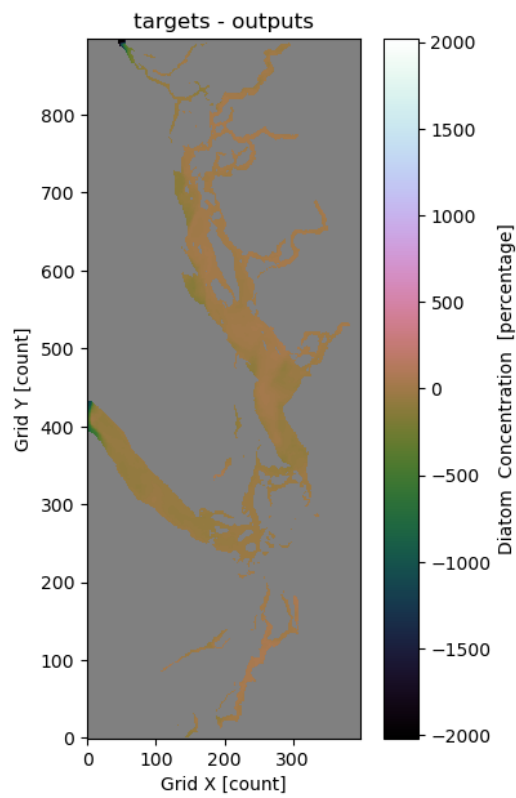
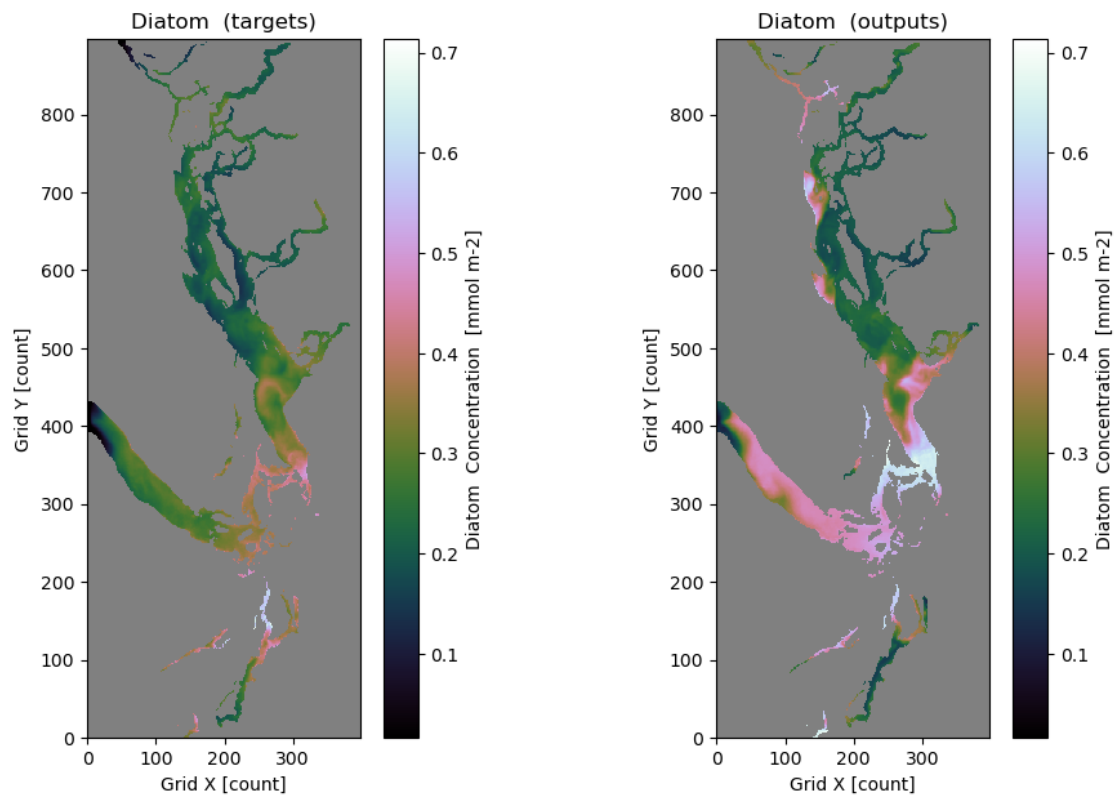


The amount of data points is 46479
The slope of the best fitting line is 1.088
The correlation coefficient is: 0.661
The mean square error is: 0.01589

Diatom 2008-03-03

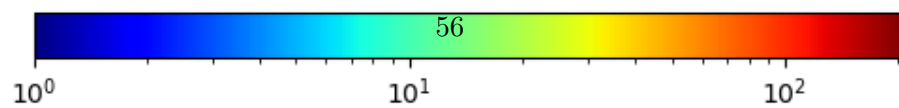
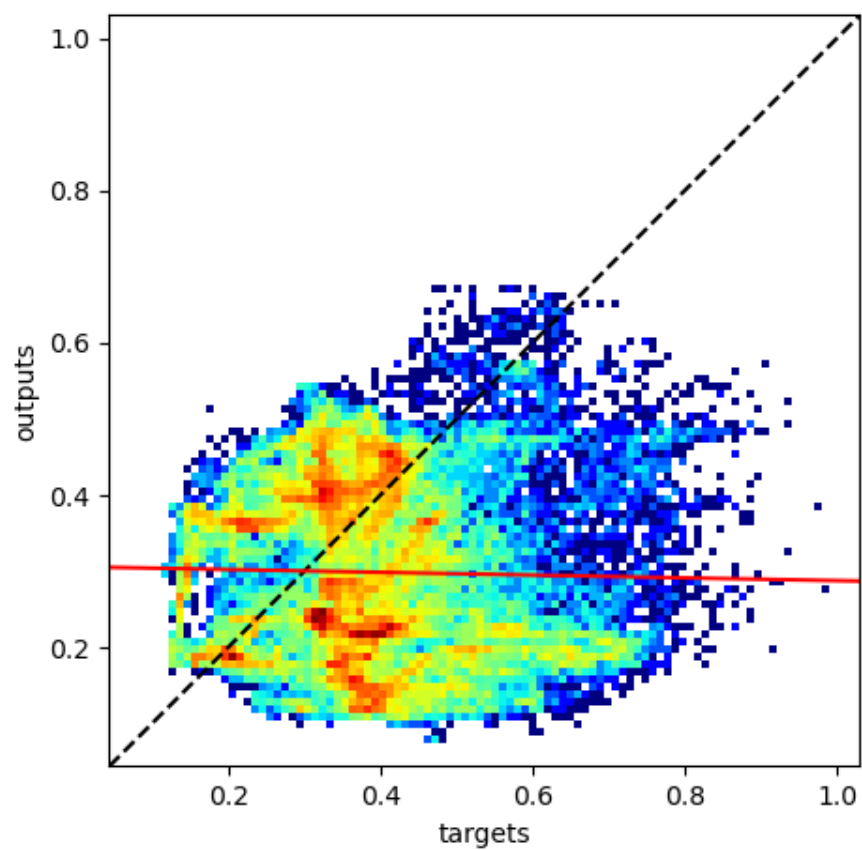
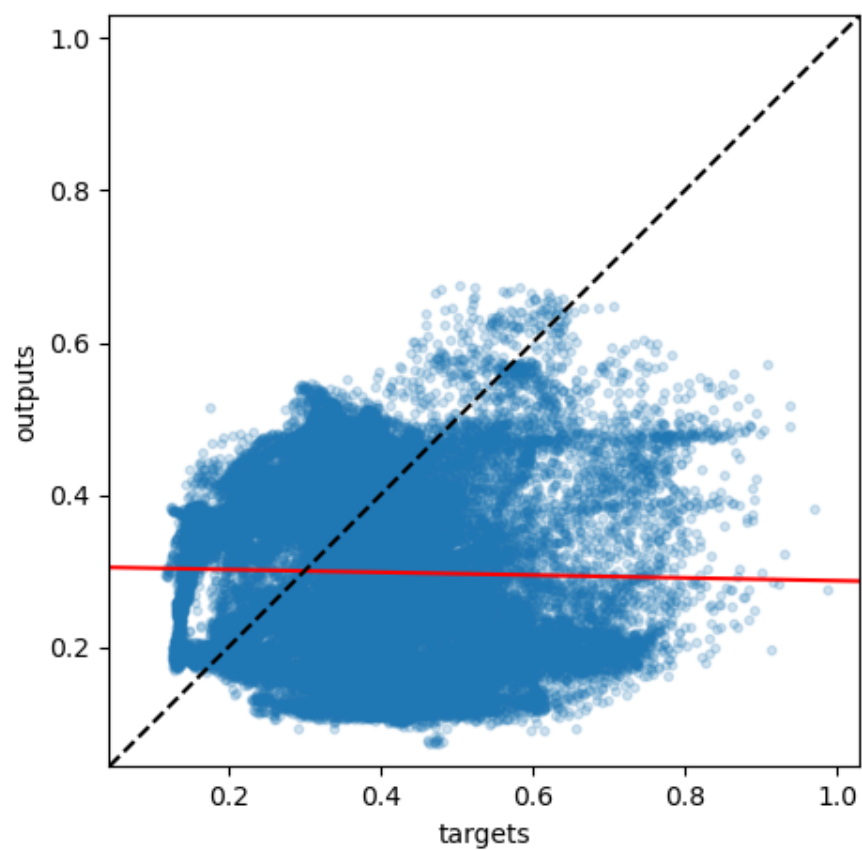


2008-03-03

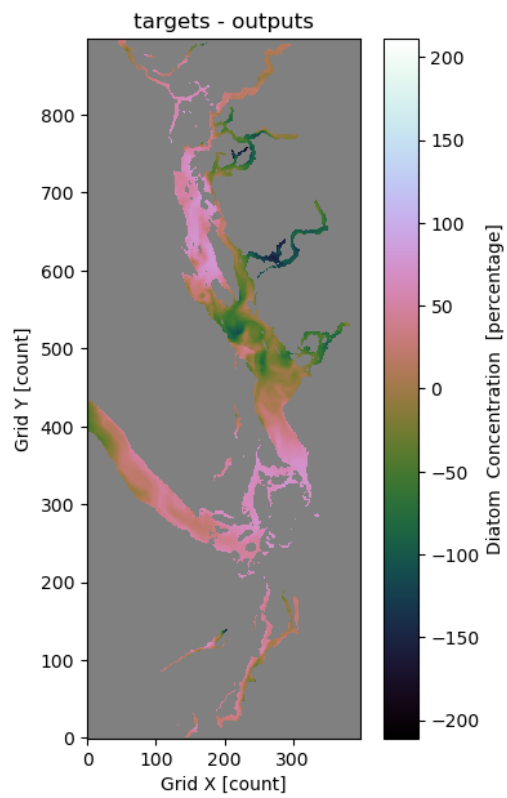
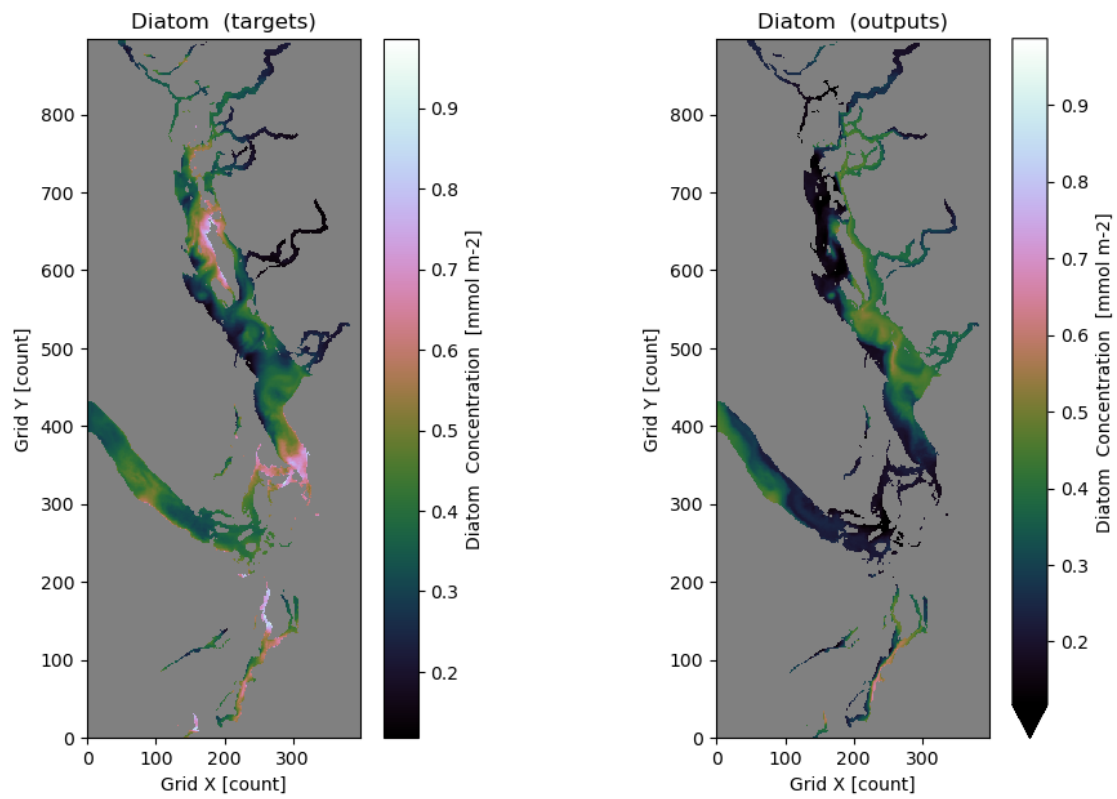


The amount of data points is 46479
The slope of the best fitting line is -0.018
The correlation coefficient is: -0.021
The mean square error is: 0.03852

Diatom 2007-04-26

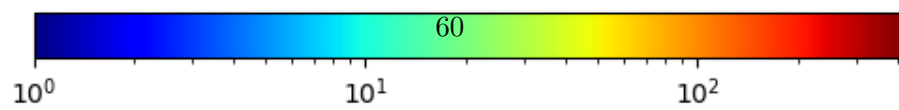
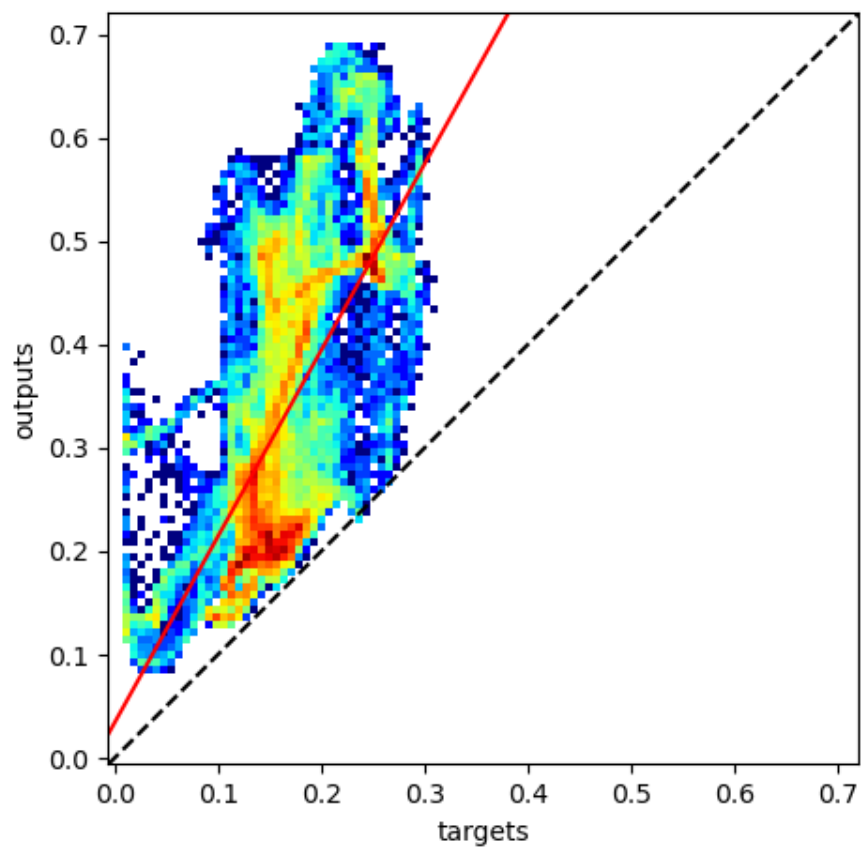
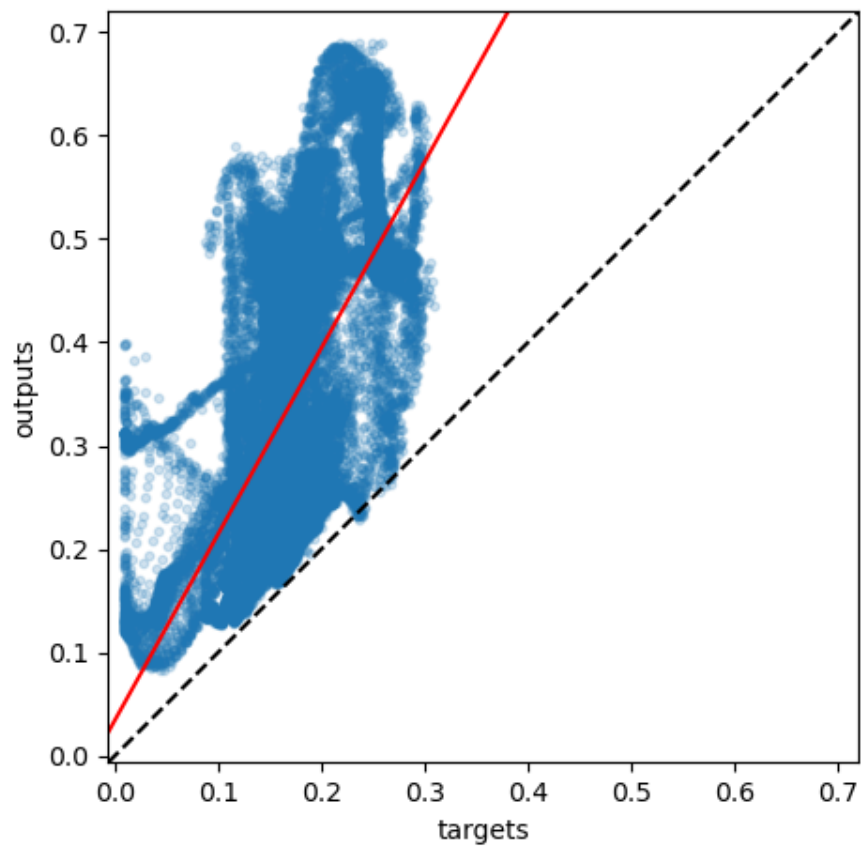


2007-04-26

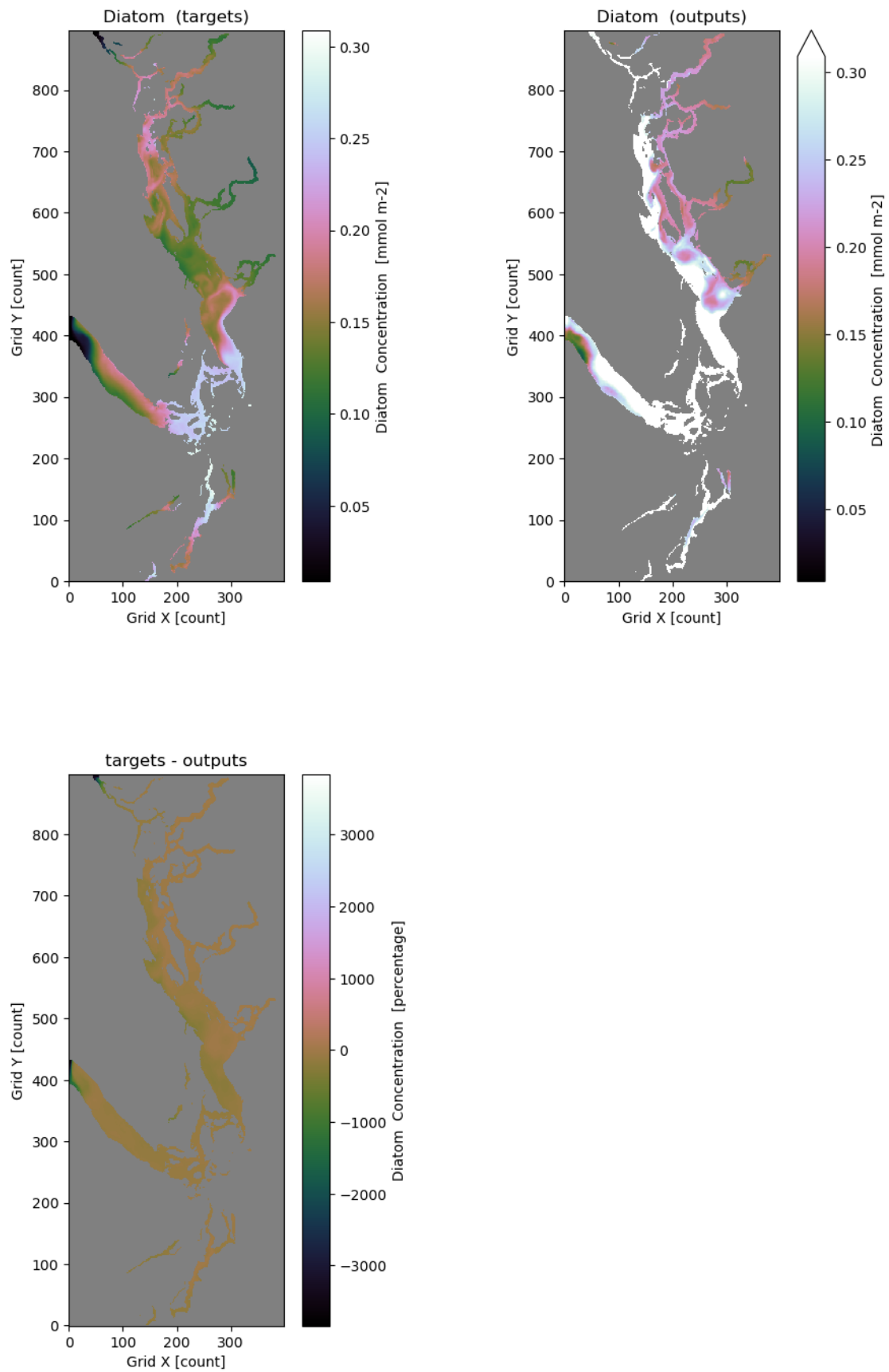


The amount of data points is 46479
The slope of the best fitting line is 1.802
The correlation coefficient is: 0.671
The mean square error is: 0.04004

Diatom 2007-02-22

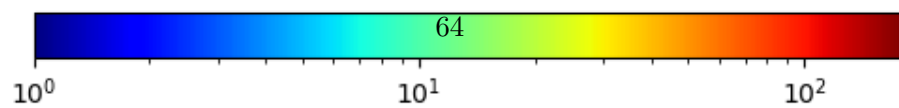
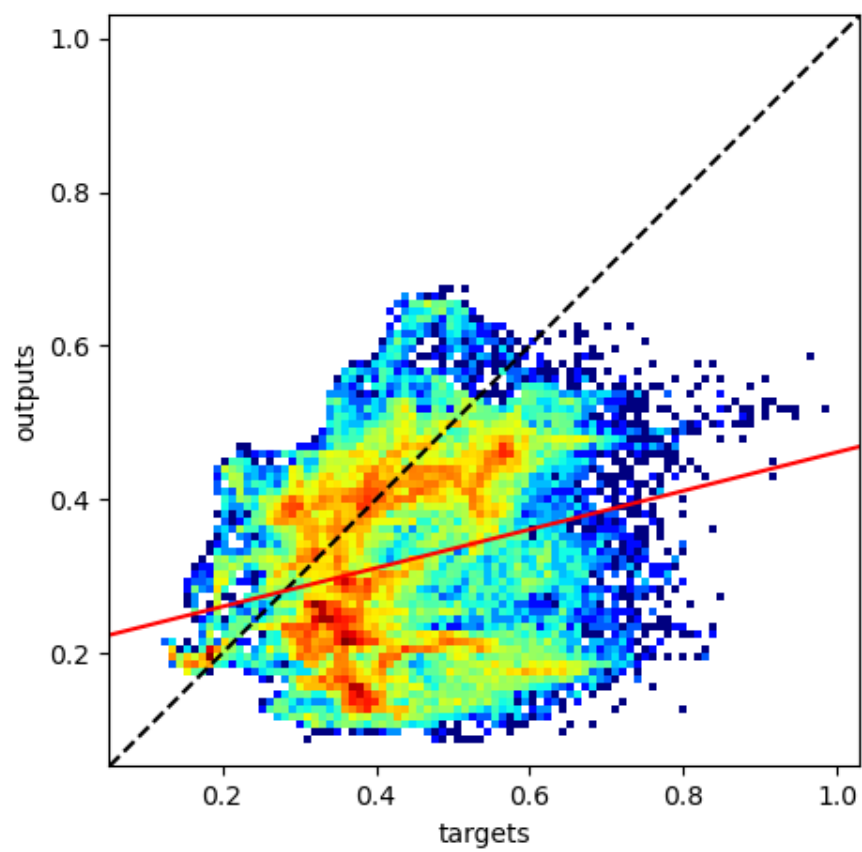
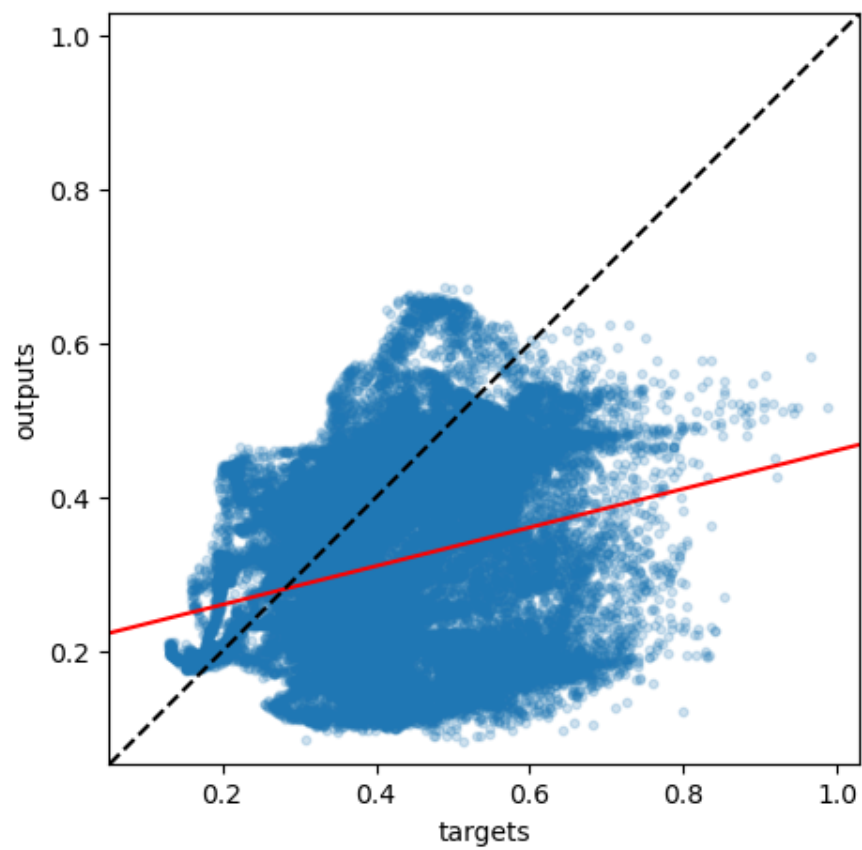


2007-02-22

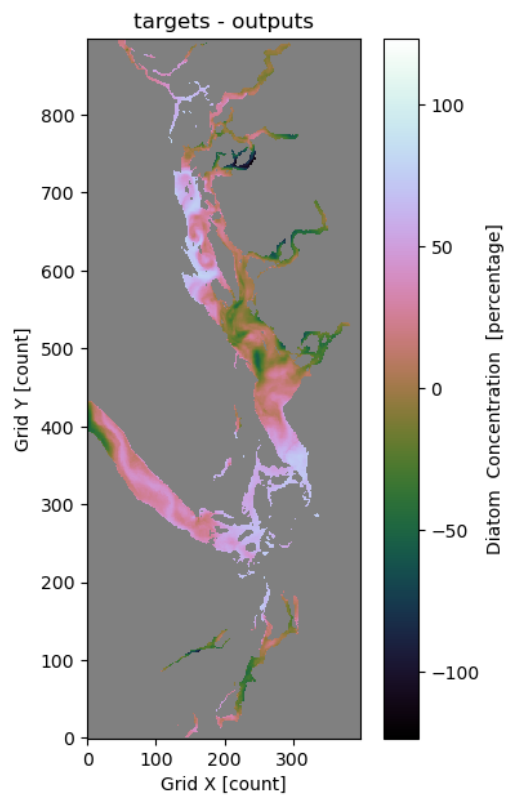
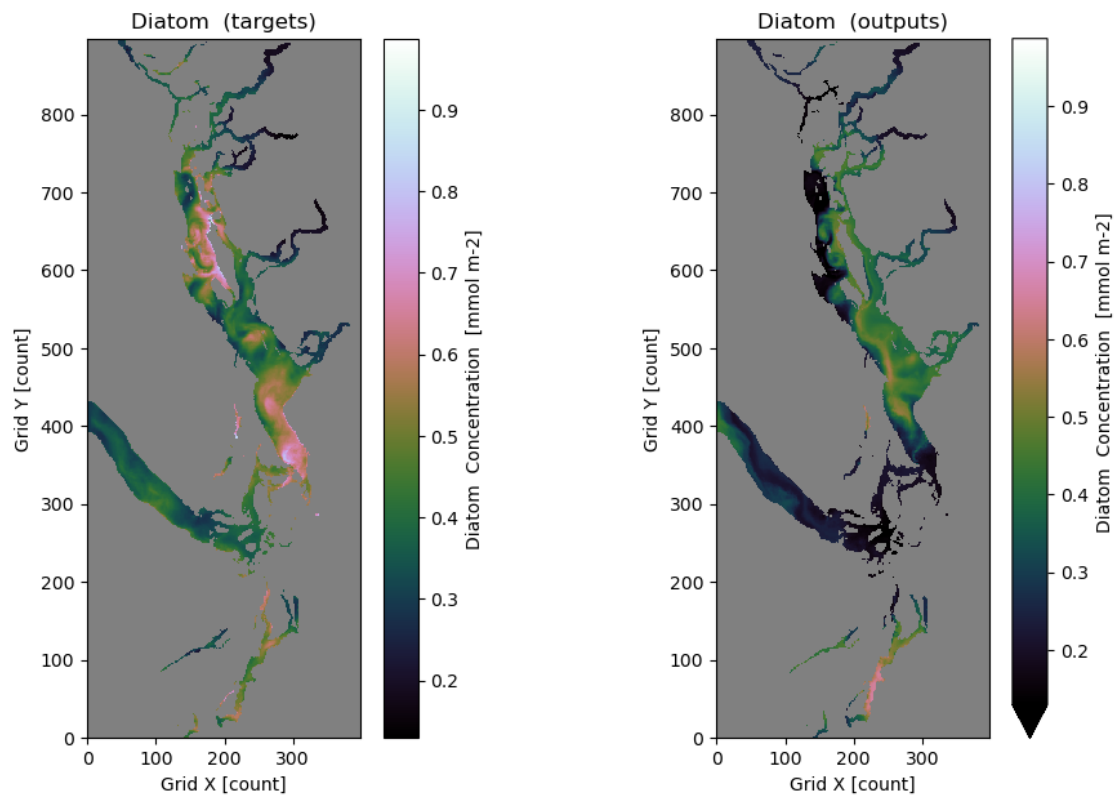


The amount of data points is 46479
The slope of the best fitting line is 0.251
The correlation coefficient is: 0.249
The mean square error is: 0.0325

Diatom 2007-04-23

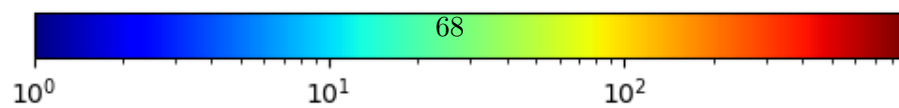
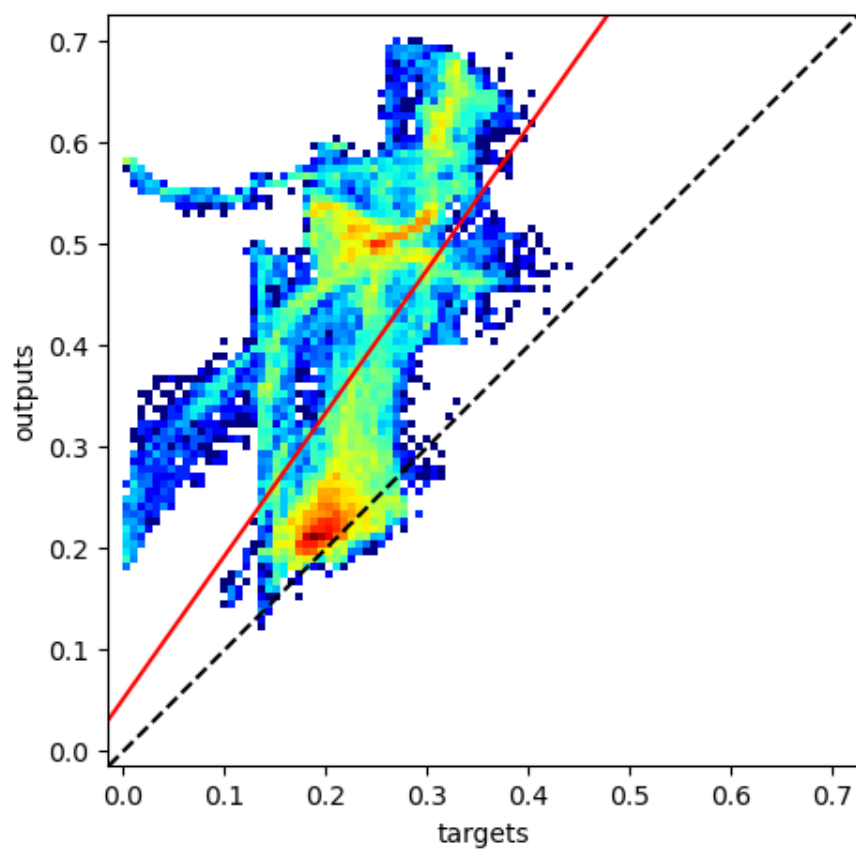
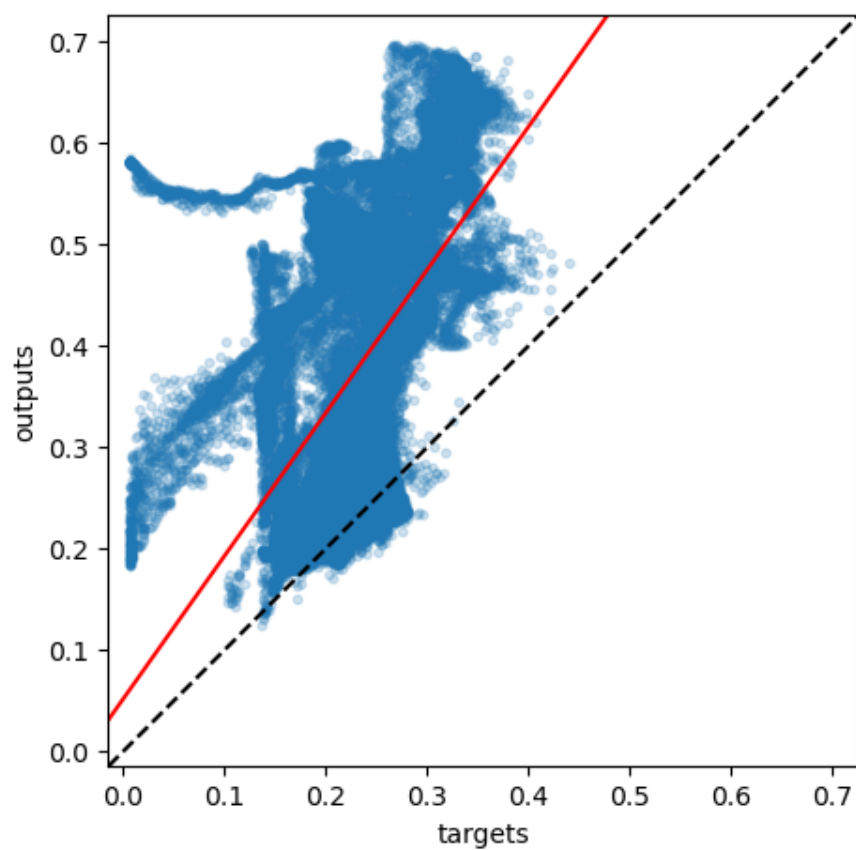


2007-04-23

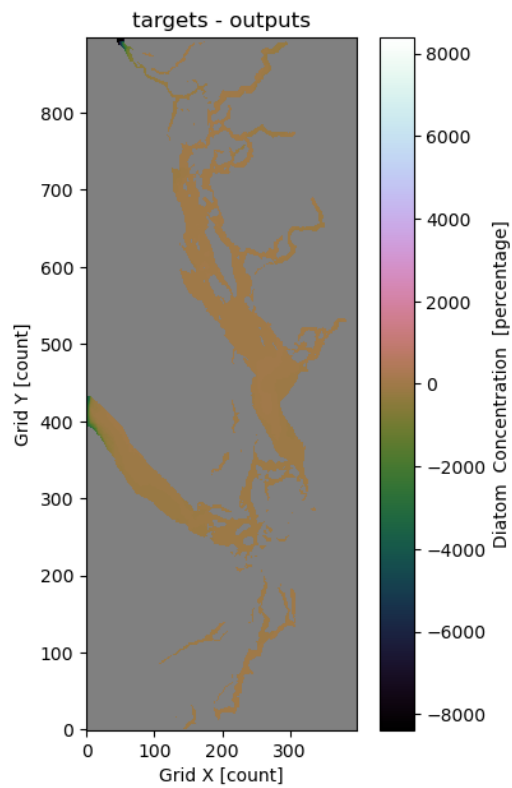
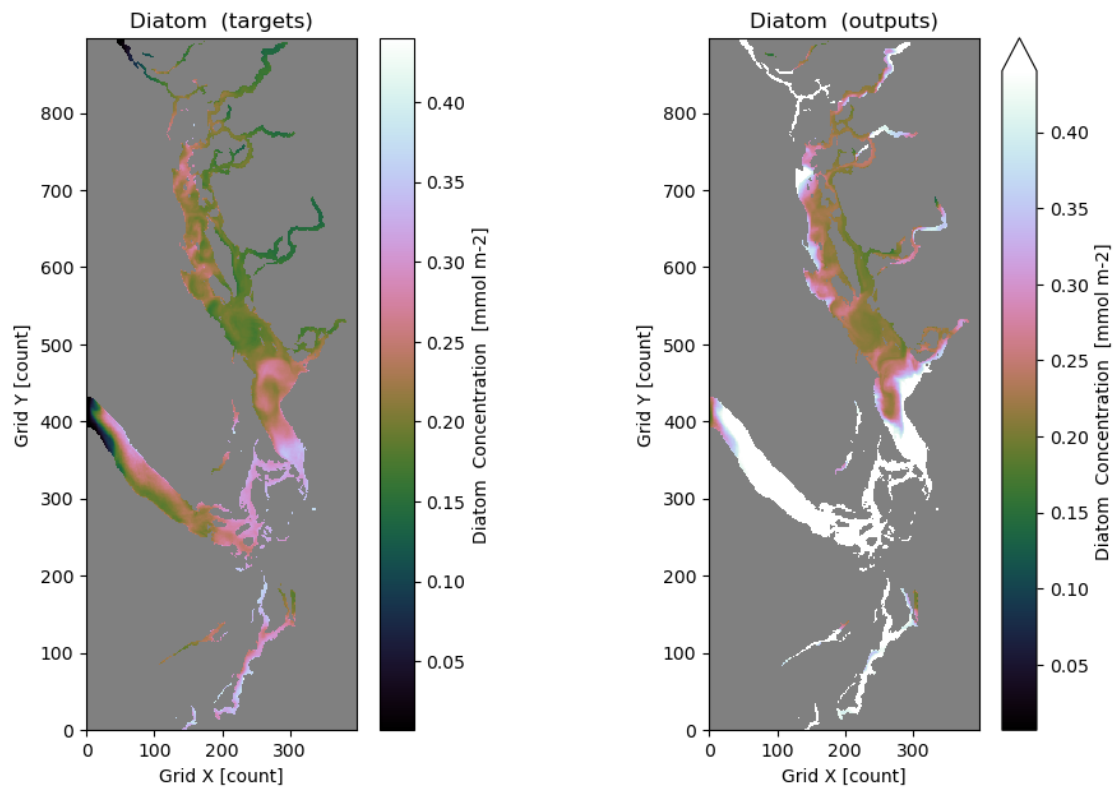


The amount of data points is 46479
The slope of the best fitting line is 1.411
The correlation coefficient is: 0.562
The mean square error is: 0.03584

Diatom 2008-02-20

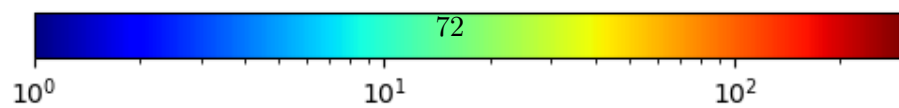
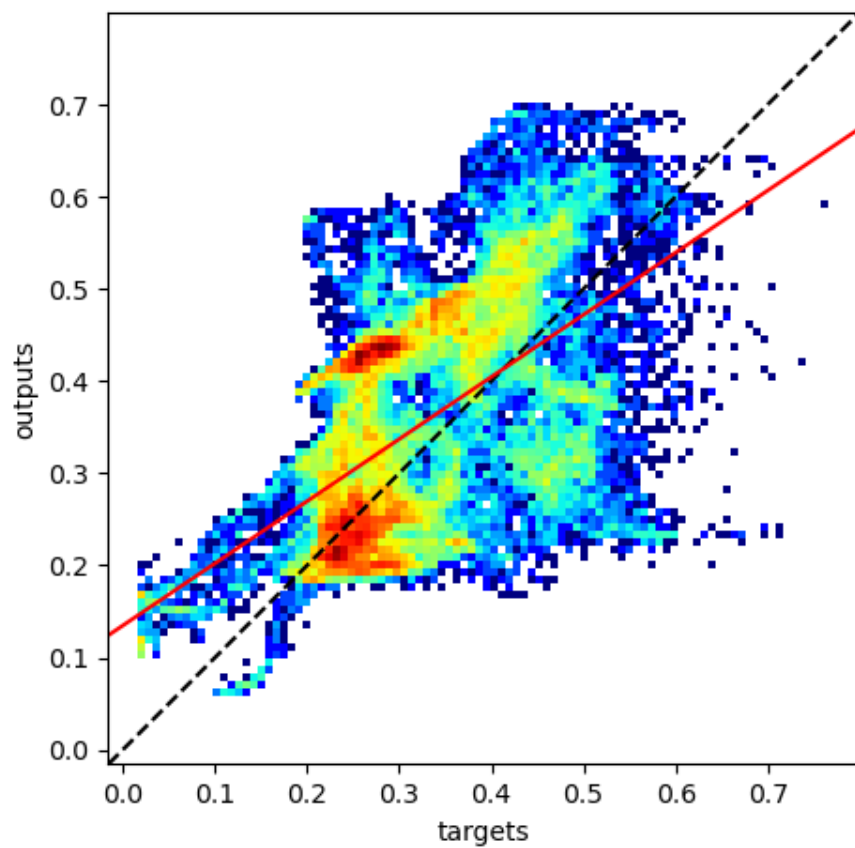
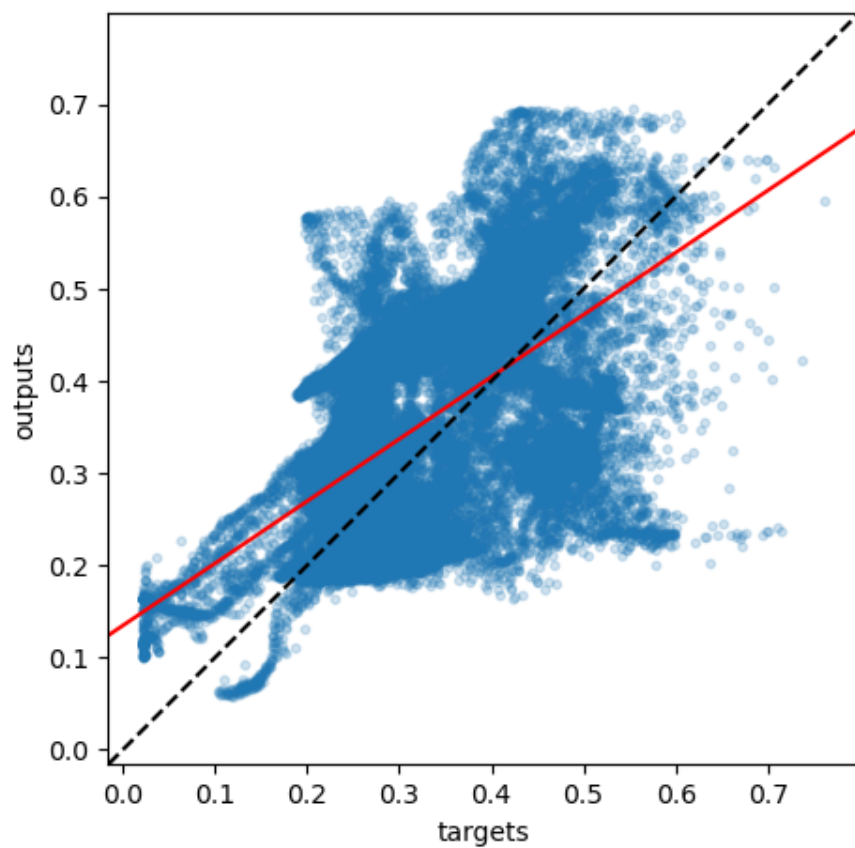


2008-02-20

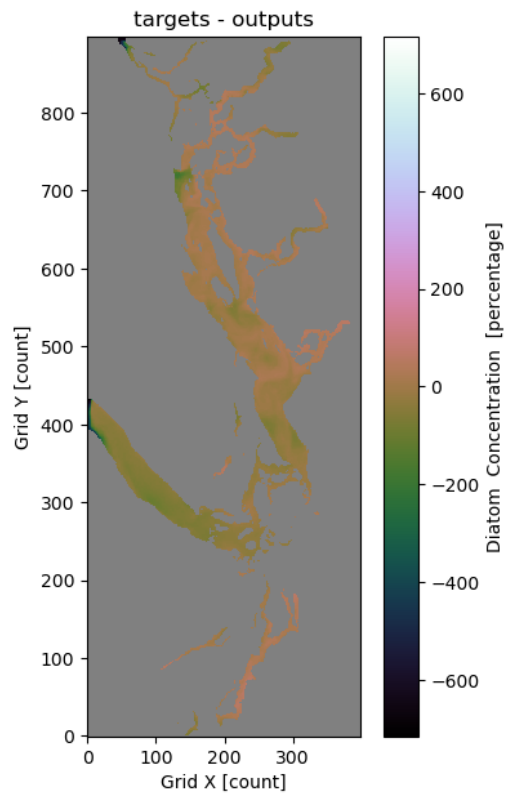
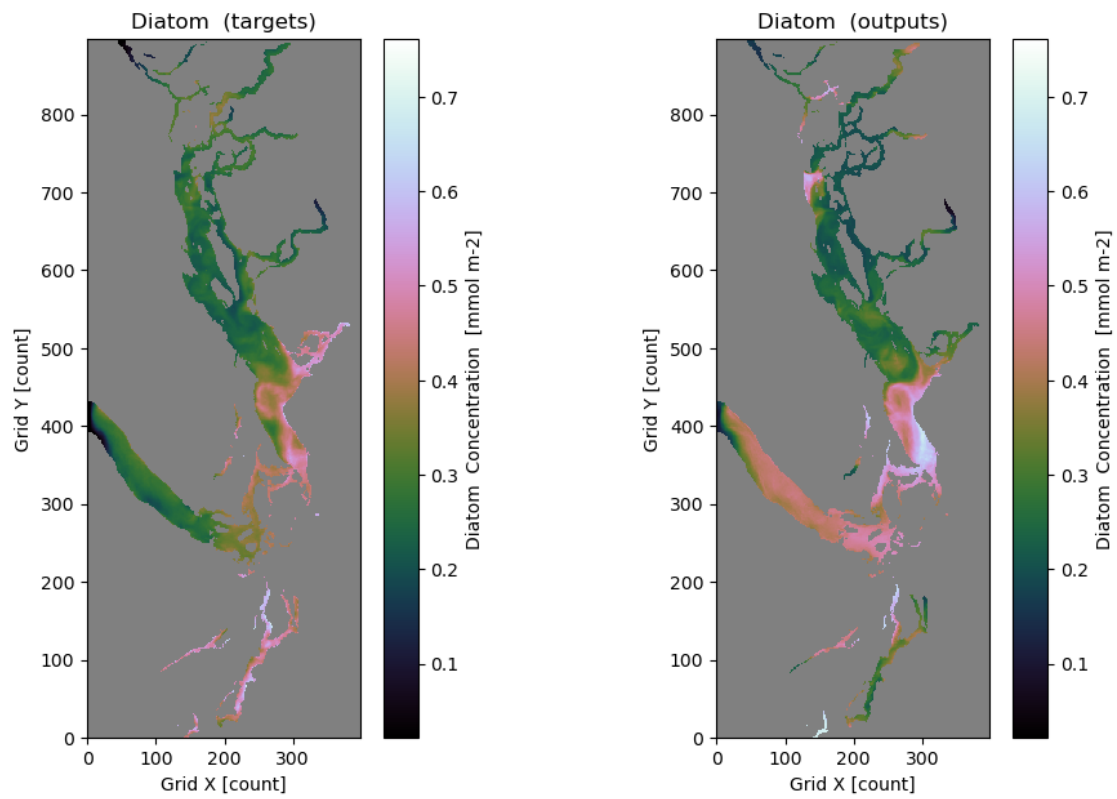


The amount of data points is 46479
The slope of the best fitting line is 0.674
The correlation coefficient is: 0.552
The mean square error is: 0.01206

Diatom 2008-03-09

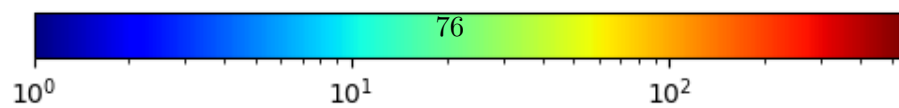
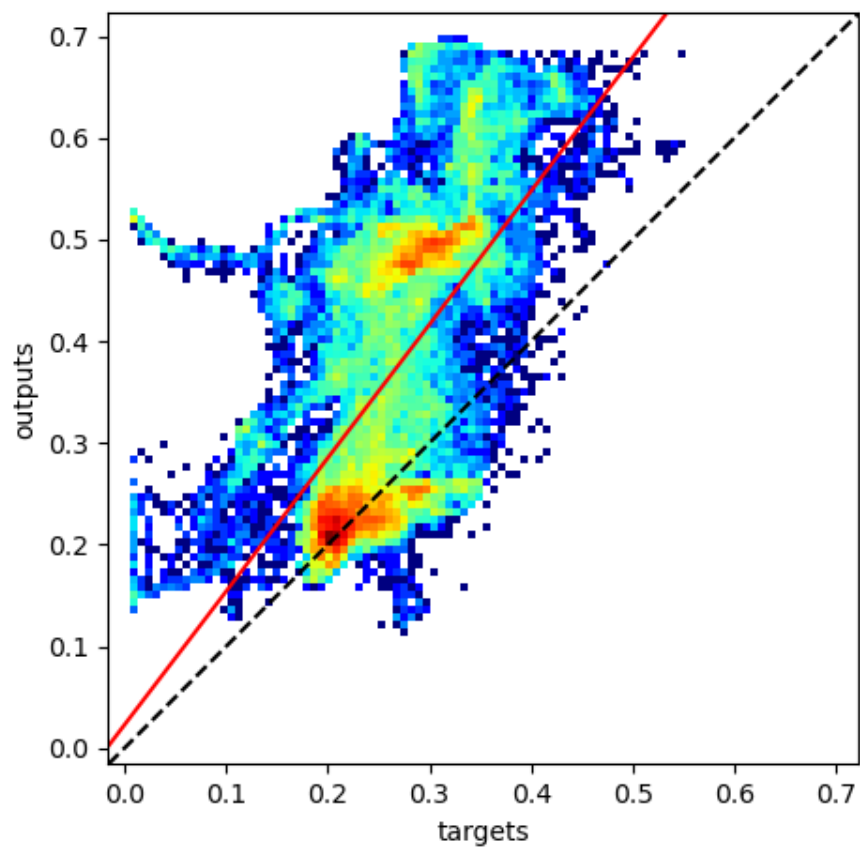
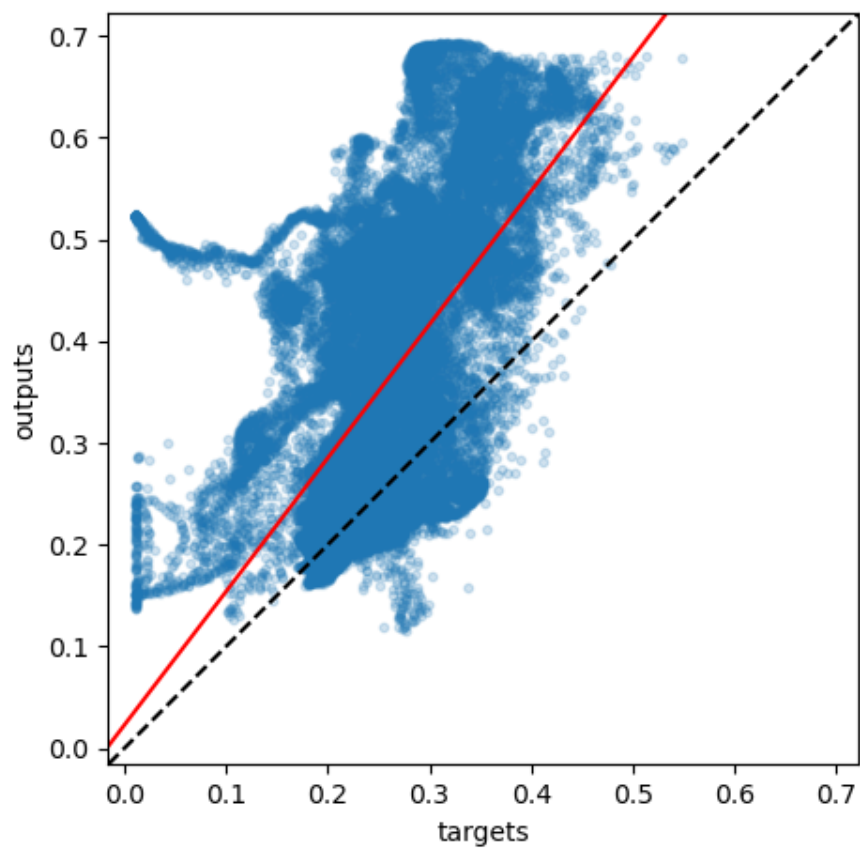


2008-03-09

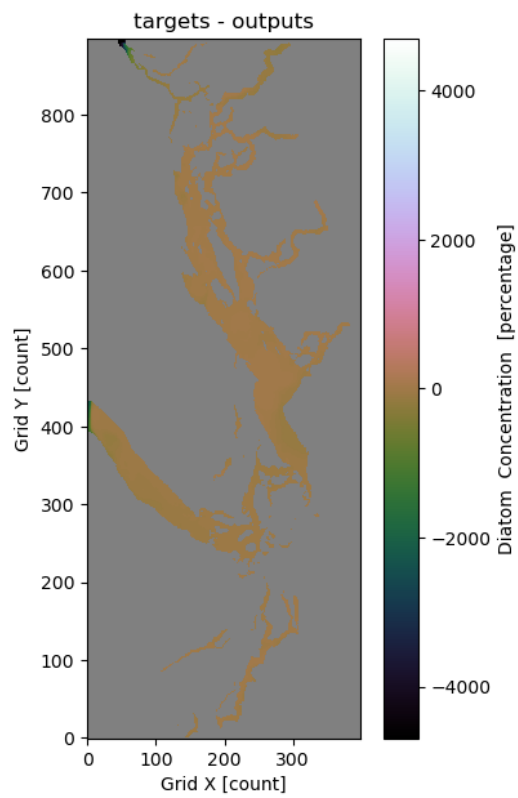
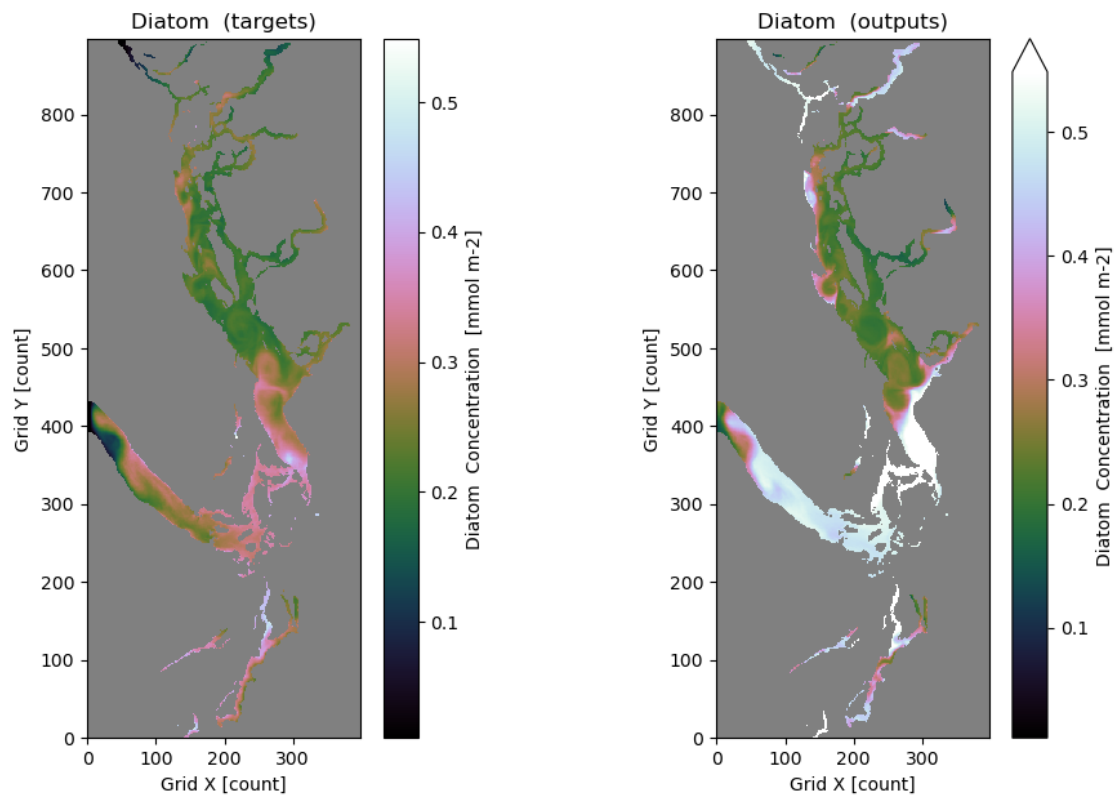


The amount of data points is 46479
The slope of the best fitting line is 1.312
The correlation coefficient is: 0.607
The mean square error is: 0.02382

Diatom 2008-02-25

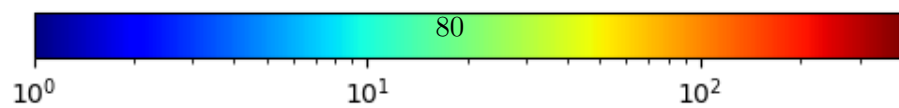
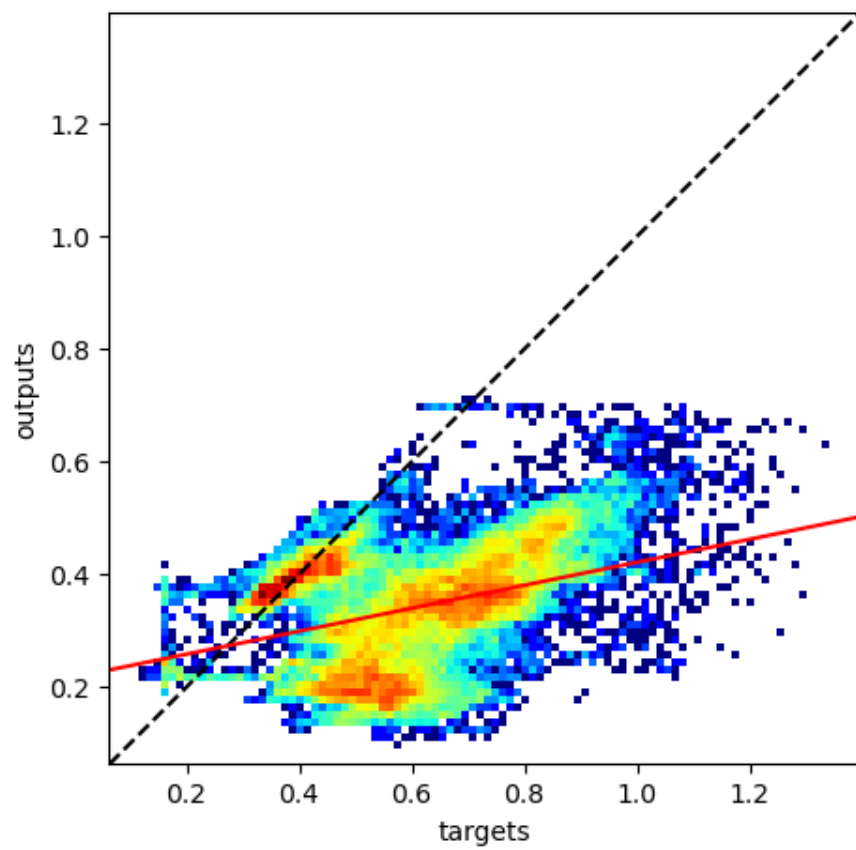
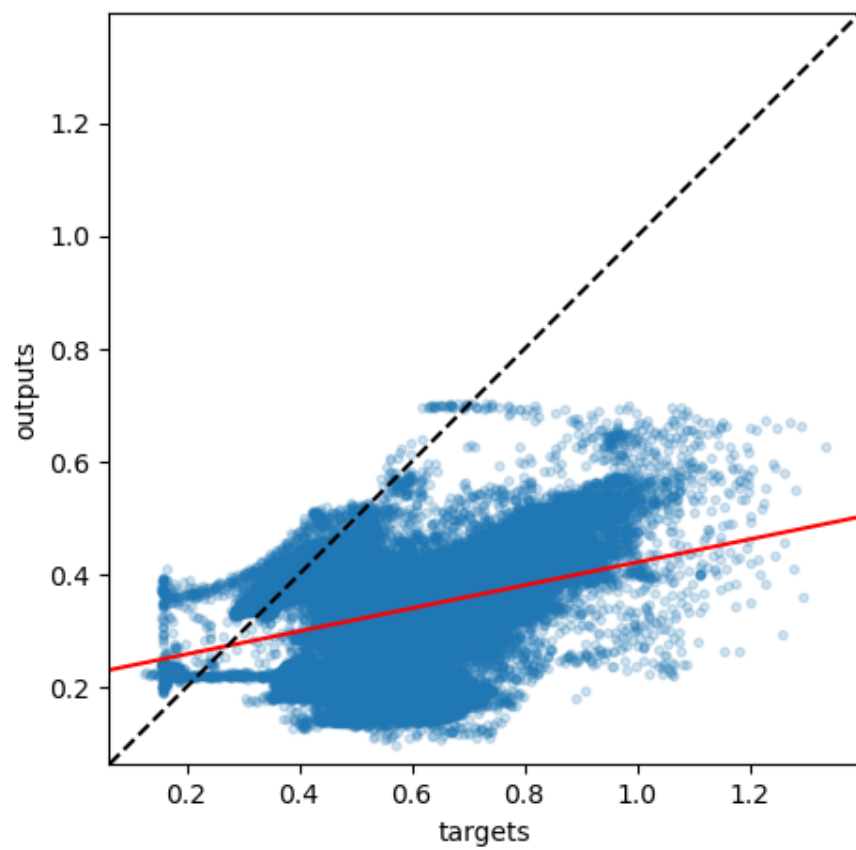


2008-02-25

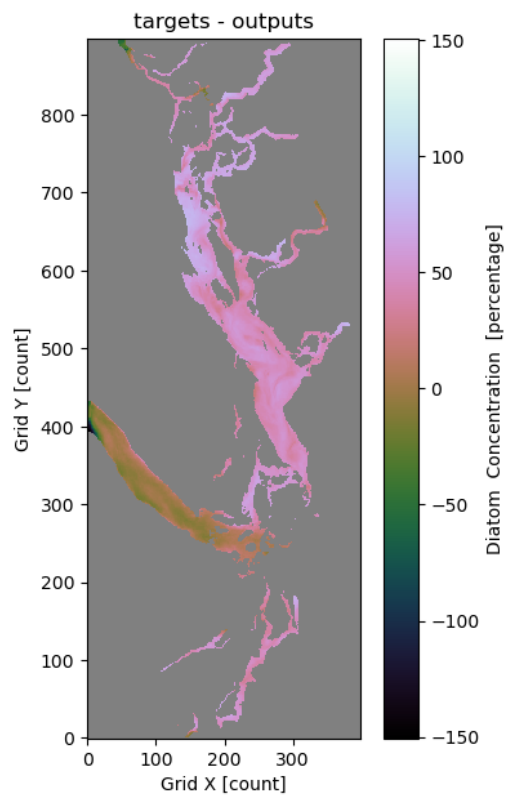
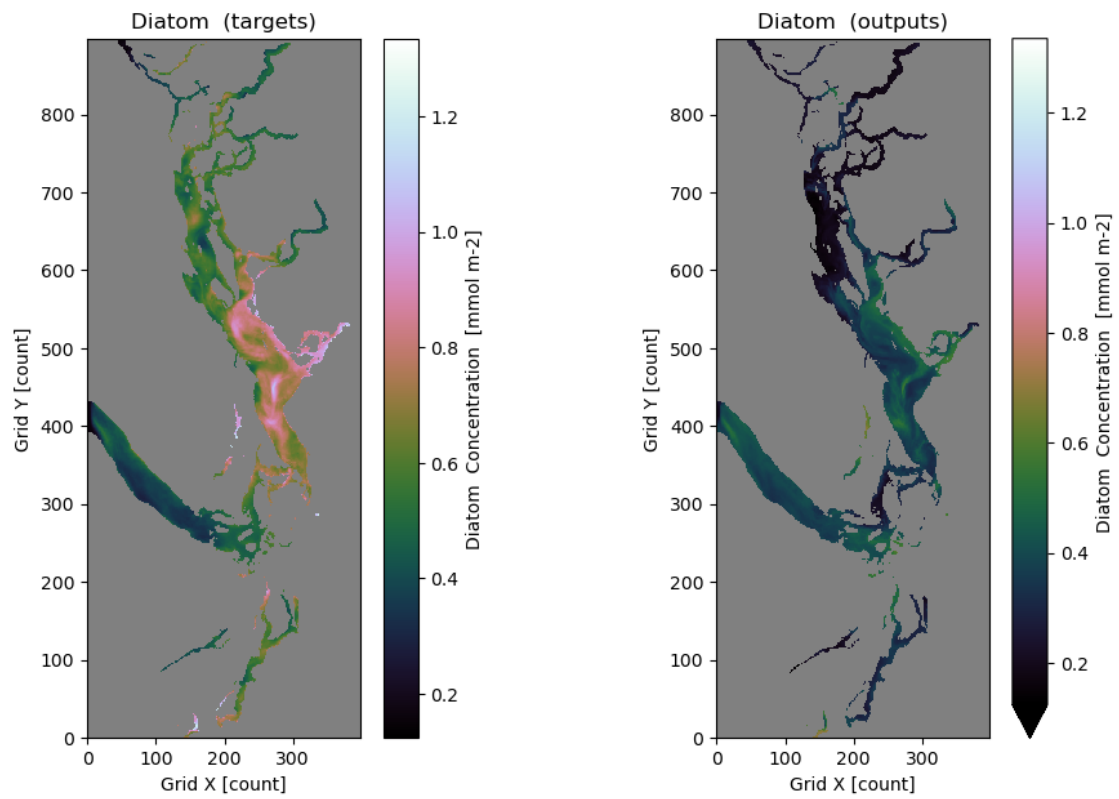


The amount of data points is 46479
The slope of the best fitting line is 0.204
The correlation coefficient is: 0.352
The mean square error is: 0.09093

Diatom 2007-04-04

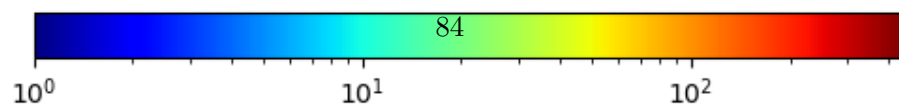
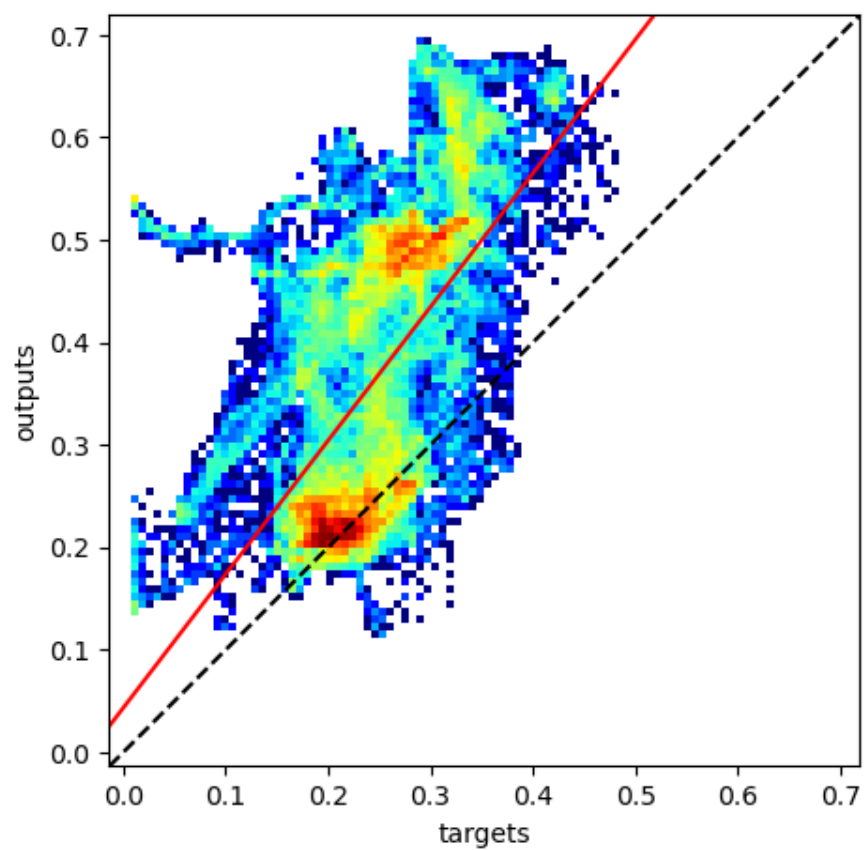
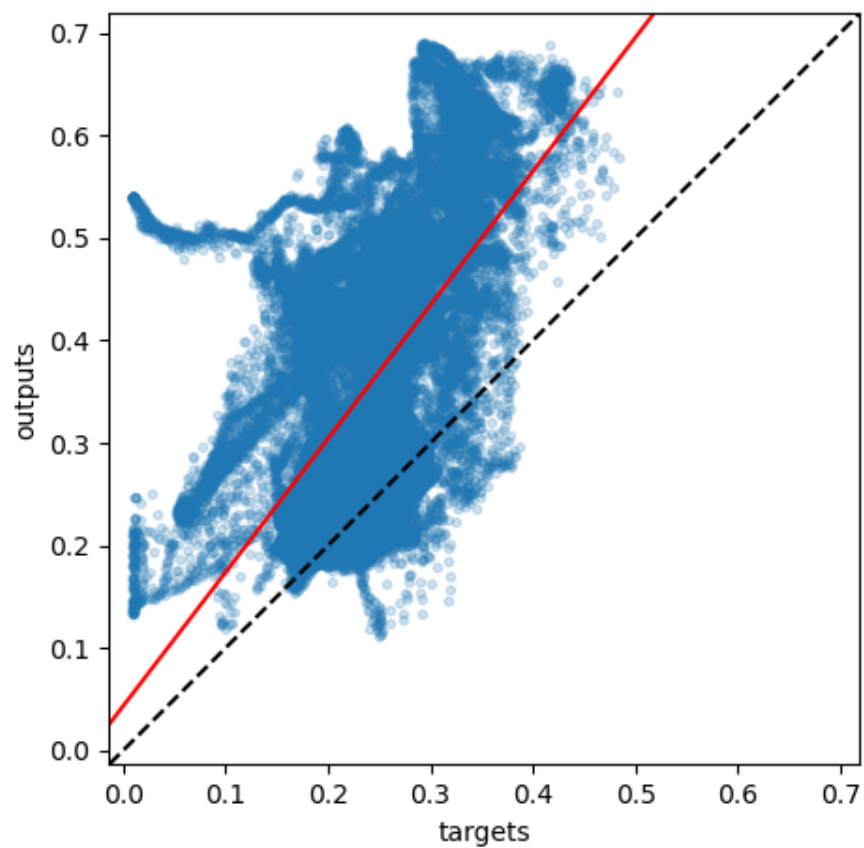


2007-04-04

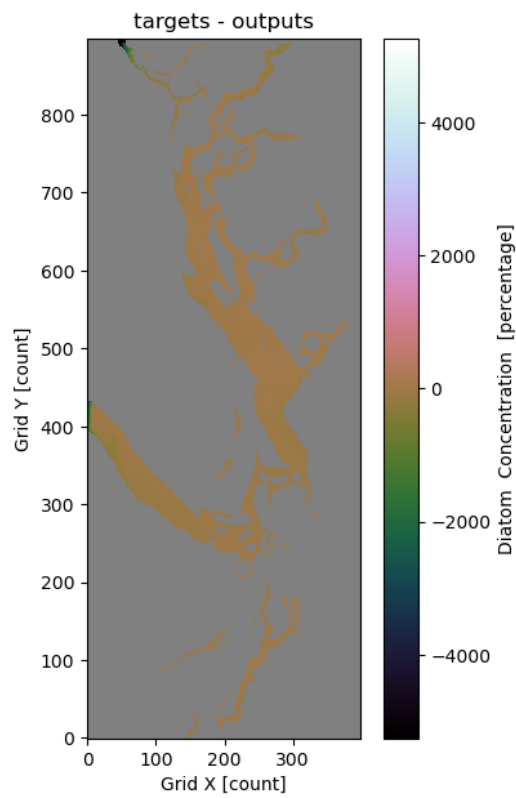
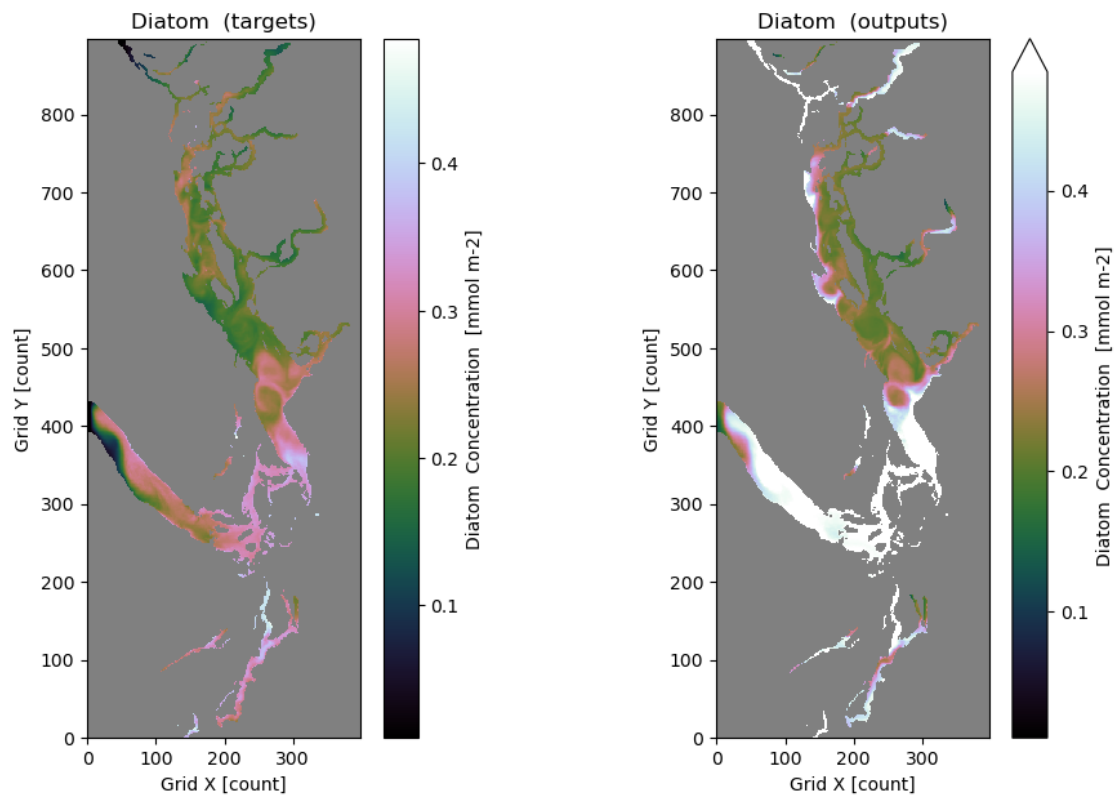


The amount of data points is 46479
The slope of the best fitting line is 1.304
The correlation coefficient is: 0.599
The mean square error is: 0.02682

Diatom 2008-02-24



2008-02-24



[]: