

Fitting drug response curves with sigmoid function

In [4]:

```
import pandas as pd
import numpy as np
from tqdm import tqdm
import warnings
warnings.filterwarnings("ignore")
import os, sys
sys.path.insert(1, os.path.relpath("functions"))
from fitting import *
from plotting import *
R2_limit = 0.99

_FOLDER = "results/"
_FOLDER_2 = "database/"
```

Fitting data

In [5]:

```
df_auc = pd.read_csv(_FOLDER+"filt_auc_02.csv")
df_1234 = pd.read_csv(_FOLDER+"filt_1234_02.csv")
drug_curves = pd.read_csv(_FOLDER_2+"normalised_dose_response_data.csv")
conc_columns= ["fd_num_"+str(i) for i in range(10)]
response_norm = ['norm_cells_'+str(i) for i in range(10)]

df_auc.shape, df_1234.shape, drug_curves.shape
```

Out[5]:

```
((14084, 31), (2108, 30), (225384, 44))
```

In [6]:

```
functions = [
    "fsigmoid",
    "sigmoid_2_param",
    "sigmoid_3_param",
    "sigmoid_4_param",
    "logistic_4_param",
    "l14_4_param",
    "l14R_4_param",
    "logLogist_3_param"
]
```

In [7]:

```
def compare_fitting(df_raw, fitting_functions_list):
    df_results = pd.DataFrame(columns= ["R2>0.9", "R2>0.95", "R2>0.99"])
    for fitting_function in fitting_functions_list:
        df = df_raw.copy()
        print("Wn", fitting_function)
        r2, fit_param = fitting_column(df, df.index, x_columns=conc_columns, y_columns= response_
                                     fitting_function = fitting_function, default_param=True)
        df[fitting_function+"_r2"] = r2
        df[fitting_function] = fit_param
    #     df= df[df[fitting_function+"_r2"]>0]
    print("R2>0:", df.shape)
    print("R2>", R2_limit, df[df[fitting_function+"_r2"]>R2_limit].shape[0])
    df_results.loc[fitting_function, "R2>0.9"] = df[df[fitting_function+"_r2"]>0.9].shape[0]
    df_results.loc[fitting_function, "R2>0.95"] = df[df[fitting_function+"_r2"]>0.95].shape[0]
    df_results.loc[fitting_function, "R2>0.99"] = df[df[fitting_function+"_r2"]>0.99].shape[0]
    print("Number of samples with fitting <0.1:", df[df[fitting_function+"_r2"]<0.1].shape[0])
    print("")
    return df, df_results
```

```
%%time
df, df_results = compare_fitting(df_auc, functions)
df_results.to_csv(_FOLDER+"fit_auc_02_compare.csv", index=False)
df.to_csv(_FOLDER+"filt_auc_02_fit.csv", index=False)
```

```
100%|███████████████████████████████████████████████████████████  
14084/14084 [00:15<00:00, 907.74it/s]
```

```
100%|███████████████████████████████████████████████████████████████████  
14084/14084 [00:13<00:00, 1018.07it/s]
```

```
100%|███████████████████████████████████████████████████████████████|  
14084/14084 [00:23<00:00, 593.25 it/s]
```

```
100%|███████████████████████████████████████████|  
14084/14084 [02:08<00:00, 109.19it/s]
```

[illegible]


```
%%time
df, df_results_2 = compare_fitting(df_1234, functions)
df_results_2.to_csv(_FOLDER+"fit_1234_compare.csv", index=False)
df.to_csv(_FOLDER+"fitt_1234_fit.csv", index=False)
```

```
100%|███████████|  
| 2108/2108 [00:01<00:00, 1204.35it/s]
```

```
Number of samples with fitting <0.1: 0
```

```
100%|██████████████████████████████████████████████████████████████████████████|  
| 2108/2108 [00:01<00:00, 1279.44it/s]
```

```
Number of samples with fitting <0.1: 0
```

```
100%|███████████|  
█ | 2108/2108 [00:02<00:00, 756.22it/s]
```

Number of samples with fitting <0.1 : 0

```
100%|███████████████████████████████████████████████████████████████|  
█ | 2108/2108 [00:03<00:00, 666.34it/s]
```

Number of samples with fitting <0.1 : 63

```
100%|██████████| 2108/2108 [00:02<00:00, 704.55it/s]
```



```
%%time
df, df_results_3 = compare_fitting(drug_curves, functions)
df_results_3.to_csv(_FOLDER+"fit_no_fit_compare.csv", index=False)
df.to_csv(_FOLDER+"filt_fit.csv", index=False)
```

```
100%|██████████████████████████████████████████████████████████████████████████| 2  
25384/225384 [04:48<00:00, 781.67it/s]
```

```
100%|██████████████████████████████████████████████████████████████████████████| 2  
25384/225384 [03:58<00:00, 943.25it/s]
```

```
100%|██████████████████████████████████████████████████████████████████████████| 2  
25384/225384 [07:48<00:00, 481.28it/s]
```

```
100%|██████████████████████████████████████████████████████████████████████████| 2  
25384/225384 [28:14<00:00, 133.04it/s]
```

```

100%|███████████████████████████████████████████████████████████| 2
25384/225384 [ 19:55<00:00. 188.60it/s]

```


In [12]:

```
df_results_2
```

Out [12]:

	R2>0.9	R2>0.95	R2>0.99
fsigmoid	2058	1874	921
sigmoid_2_param	2058	1874	921
sigmoid_3_param	2098	2031	1220
sigmoid_4_param	2045	2017	1527
logistic_4_param	2103	2063	1473
ll4_4_param	2103	2063	1473
ll4R_4_param	2085	2056	1473
logLogist_3_param	2100	2051	1465

In [13]:

```
df_results_2.sort_values("R2>0.99")
```

Out [13]:

	R2>0.9	R2>0.95	R2>0.99
fsigmoid	2058	1874	921
sigmoid_2_param	2058	1874	921
sigmoid_3_param	2098	2031	1220
logLogist_3_param	2100	2051	1465
logistic_4_param	2103	2063	1473
ll4_4_param	2103	2063	1473
ll4R_4_param	2085	2056	1473
sigmoid_4_param	2045	2017	1527

In [14]:

```
df_results_3
```

Out [14]:

	R2>0.9	R2>0.95	R2>0.99
fsigmoid	53091	32241	6638
sigmoid_2_param	53063	32194	6613
sigmoid_3_param	70557	46861	11402
sigmoid_4_param	53846	42088	15169
logistic_4_param	73125	52855	16756
ll4_4_param	73134	53103	17072
ll4R_4_param	67848	49454	16614
logLogist_3_param	72883	53643	16218

In [15]:

```
df_results_3.sort_values("R2>0.99")
```

Out [15]:

	R2>0.9	R2>0.95	R2>0.99
sigmoid_2_param	53063	32194	6613
fsigmoid	53091	32241	6638
sigmoid_3_param	70557	46861	11402
sigmoid_4_param	53846	42088	15169
logLogist_3_param	72883	53643	16218
ll4R_4_param	67848	49454	16614
logistic_4_param	73125	52855	16756
ll4_4_param	73134	53103	17072

In []: