Filtering

In [10]:

```
import pandas as pd
import numpy as np
import os
import gc
import matplotlib.pyplot as plt
import time

from functions.filtering import *
from functions.plotting import *
from functions.fitting import *

_FOLDER = "database/"
_FOLDER_2 = "figures/"
_FOLDER_3 = "results/"
SAVE_FIGURES = False

R2_limit = 0.99
```

Original data

In [11]:

```
drug_curves = pd.read_csv(_FOLDER+"normalised_dose_response_data.csv")

if "Unnamed: 0" in drug_curves:
    drug_curves.drop("Unnamed: 0", axis=1, inplace =True)

col_to_drop = ["per_slope_change_"+str(i) for i in range(8)]+₩
            ["slope_" + str(i) for i in range(9)]
drug_curves.drop(col_to_drop, axis=1, inplace=True)

conc_columns= ["fd_num_"+str(i) for i in range(10)]
response_norm = ['norm_cells_'+str(i) for i in range(10)]
CCL_names = dict(zip(drug_curves["COSMIC_ID"], drug_curves["CELL_LINE_NAME"]))
df= pd.read_csv(_FOLDER+'Drug_Features.csv')
drug_names = dict(zip(df["Drug ID"].values, df["Drug Name"].values))
del df
drug_curves["drug_name"] = drug_curves["DRUG_ID"].map(drug_names)
drug_curves["CCL_name"] = drug_curves["COSMIC_ID"].map(CCL_names)
drug_curves.shape
```

Out[11]:

```
(225384, 28)
```

Filtering 1: 4 stage filtering

```
# Description of filtering_sigmoid_curves:
"""
    filtering_scenario = [1,2,3,4]
    1. Ensure that all the response are less than 1

    2. Ensure that first and last points form plateus
    the minimal number of points are specified in the function arguments
    by default, two points for both lpateus are considered
    tolerance =0.05 values to ensure the points form a plateu
    first_columns_to_compare = [1, 2]  - first two columns for plateu
    last_columns_to_compare = [-1, -2] - last two columns for plateu

    3. Specify location of the plateus - first_points_lower_limit and last_points_upper_limit

    4. Cutting off ambiqueos data:
    Among all "middle" datapoints a subsequent point should not be higher than antecedent by
0.2
"""
```

In [12]:

```
%%time
# difference between middle points 0.2
df_filt_1234 = filtering_sigmoid_curves(drug_curves, filtering_scenario=[1,2,3,4], ₩
                        response_columns = response_norm, ₩
                        first_points_lower_limit = 0.8, last_points_upper_limit = 0.2,
                         middle_points_limit = -0.2)
df_filt_1234.to_csv(_FOLDER_3+"filt_1234_02.csv", index=False)
```

```
Original dataset: (225384, 28)
1st filtration (Ensure that all the response are less than 1): Filtered dataset:
(63325, 28)
2d filtration (Ensure that first and last points form plateus): Filtered dataset:
(6321, 30)
3d stage filtration (Specified location of the plateus): Filtered dataset: (2152,
30)
4th stage filtration (Cut off high ancedent points): Filtered dataset: (2108, 30)
CPU times: total: 656 ms
Wall time: 646 ms
```

In [13]:

```
%%time
df = df_filt_1234.copy()
fitting_function = "sigmoid_4_param"

r2, fit_param = fitting_column(df, df.index, x_columns=conc_columns, y_columns= response_norm,
                               fitting_function = fitting_function, default_param=True)
df[fitting_function+"_r2"] = r2
df[fitting_function] = fit_param
df = df[df[fitting_function+"_r2"]>R2_limit]
print(df_filt_1234.shape, df.shape)
df.to_csv(_FOLDER_3+"fit_1234_02.csv", index=False)
```

```
100%|██████████████████████████████████████████████████████████████
█| 2108/2108 [00:03<00:00, 633.77it/s]

<function sigmoid_4_param at 0x0000024E37FCC900>
(2108, 30) (1527, 32)
CPU times: total: 3.56 s
Wall time: 3.53 s
```

Filtering 2: auc>0.7 and spearman_r<0

```
# description of auc_filtration
"""
    1. Remove all the curves where the normalised response value is greater than one at zero
dosage.
    2. Leave only those curves with an Area Under the Curve (AUC) >0.7.
    3. Compute the Spearman correlation coefficient between the normalised response
    and the scaled dosage (so the x-axis and the y-axis).
    4. Further remove the curves for which the Spearman correlation coefficient is zero or
positive.
    5. Cut off samples with last response above 0.2
"""
```

In [14]:

```
%%time
df_filt_auc = pd.read_csv(_FOLDER_3+"filt_auc_02.csv")
df = df_filt_auc.copy()
fitting_function = "sigmoid_4_param"

r2, fit_param = fitting_column(df, df.index, x_columns=conc_columns, y_columns= response_norm,
                               fitting_function = fitting_function, default_param=True)
df[fitting_function+"_r2"] = r2
df[fitting_function] = fit_param
df = df[df[fitting_function+"_r2"]>R2_limit]
print(df_filt_auc.shape, df.shape)
df.to_csv(_FOLDER_3+"fit_auc_02.csv", index=False)
```

```
100%|████████████████████████████████████████████████████████████|
14084/14084 [01:52<00:00, 125.25it/s]

<function sigmoid_4_param at 0x0000024E37FCC900>
(14084, 31) (3062, 33)
CPU times: total: 1min 53s
Wall time: 1min 52s
```

Filtering 3: direct fitting

In [15]:

```
functions = [
             "fsigmoid",
             "sigmoid_2_param",
             "sigmoid_3_param",
             "sigmoid_4_param",
             "logistic_4_param",
            "ll4_4_param",
            "ll4R_4_param",
            "logLogist_3_param"]
```

In [16]:

```
%%time
df_no_filt = compare_fitting_functions(drug_curves, functions, conc_columns, response_norm,
                                        save_file_name = _FOLDER_3 +"fit_no_filt.csv")
```

```
(225384, 28)

 fsigmoid

100%|██████████████████████████████████████████████████████████████| 2
25384/225384 [04:58<00:00, 755.78it/s]

<function fsigmoid at 0x0000024E37FCC7C0>

 sigmoid_2_param

100%|██████████████████████████████████████████████████████████████| 2
25384/225384 [03:49<00:00, 982.67it/s]

<function sigmoid_2_param at 0x0000024E37FCC720>

 sigmoid_3_param

100%|██████████████████████████████████████████████████████████████| 2
25384/225384 [07:43<00:00, 486.34it/s]

<function sigmoid_3_param at 0x0000024E37FCC860>

 sigmoid_4_param

100%|██████████████████████████████████████████████████████████████| 2
25384/225384 [32:20<00:00, 116.14it/s]

<function sigmoid_4_param at 0x0000024E37FCC900>

 logistic_4_param

100%|██████████████████████████████████████████████████████████████| 2
25384/225384 [28:13<00:00, 133.11it/s]

<function logistic_4_param at 0x0000024E37FCCAE0>

 ll4_4_param

100%|██████████████████████████████████████████████████████████████| 2
25384/225384 [32:11<00:00, 116.69it/s]

<function ll4_4_param at 0x0000024E37FCC9A0>

 ll4R_4_param

 75%|███████████████████████████████████████████████               | 1
69960/225384 [27:08<05:44, 160.94it/s]
```

In [ ]:

```

```