Electronic Supplementary Material (ESI) for Digital Discovery.


# Electronic Supplementary Information

# Artificial Neural Networks and Data Fusion Enable Concentration Predictions for Inline Process Analytics

Peter Sagmeister,[a,b] Robin Hierzegger,[a,b] Jason D. Williams,[a,b] C. Oliver Kappe*[a,b] and Stefan Kowarik*[b]

[a]Center for Continuous Flow Synthesis and Processing (CCFLOW), Research Center Pharmaceutical Engineering GmbH (RCPE), Inffeldgasse 13, Graz, Austria

[b]Institute of Chemistry, University of Graz, Heinrichstrasse 28, 8010 Graz, Austria.

Email: oliver.kappe@uni-graz.at; stefan.kowarik@uni-graz.at

**Table of Contents**

# 1 Experimental Details

## 1.1 General Experimental Details

Solvents and chemicals were obtained from commercial suppliers and were used without any further purification unless otherwise noted. In the flow setup, standard PFA tubing (0.8 mm or 1.6 mm i.d.), fittings, T-pieces manufactured from PTFE or PEEK were used as connectors. Solvents and chemicals were obtained from commercial suppliers and were used without any further purification unless otherwise noted. Reference materials for 2ClBA, 5N-2ClBA, 3-NSA, 5-NSA were obtained from TCI in >98% purity and 3N-2ClBA was obtained from Apollo Scientific (99% purity).

## 1.2 Reactor Platform and Connectivity for Process Data

The reactor platform, connectivity of the instruments, UHPLC method, and the process runs are described in reference S1. The concentration for the intermediates and impurities after the NMR and UV/vis measurements were recalculated from the final measurements of the UHPLC placed at the reactor outlet. The UHPLC concentrations for **2ClBA**, **3N-2ClBA**, **5N-2ClBA**, **3-NSA**, **5-NSA** for the stability run and the run with dynamic changes can be found in Table S1 - Table S2 and Table S3 - Table S4, respectively.

**Table S1.** Online UHPLC results for the stability run.

| Injection | Timestamp (h) | [2ClBA] (mol/L) | [3N-2ClBA] (mol/L) | [5N-2ClBA] (mol/L) | [3-NSA] (mol/L) | [5-NSA] (mol/L) |
|---|---|---|---|---|---|---|
| 1 | 0.015 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 2 | 0.144 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 3 | 0.273 | 0.000 | 0.000 | 0.001 | 0.001 | 0.019 |
| 4 | 0.402 | 0.000 | 0.001 | 0.015 | 0.006 | 0.105 |
| 5 | 0.531 | 0.000 | 0.001 | 0.019 | 0.010 | 0.147 |
| 6 | 0.659 | 0.000 | 0.001 | 0.019 | 0.011 | 0.153 |
| 7 | 0.788 | 0.006 | 0.001 | 0.019 | 0.009 | 0.157 |
| 8 | 0.916 | 0.007 | 0.001 | 0.019 | 0.011 | 0.159 |
| 9 | 1.046 | 0.007 | 0.001 | 0.019 | 0.011 | 0.160 |
| 10 | 1.175 | 0.007 | 0.001 | 0.018 | 0.009 | 0.163 |
| 11 | 1.304 | 0.007 | 0.001 | 0.018 | 0.009 | 0.163 |
| 12 | 1.433 | 0.007 | 0.001 | 0.018 | 0.009 | 0.163 |
| 13 | 1.561 | 0.007 | 0.002 | 0.018 | 0.012 | 0.163 |
| 14 | 1.691 | 0.007 | 0.001 | 0.018 | 0.009 | 0.163 |
| 15 | 1.819 | 0.007 | 0.001 | 0.017 | 0.011 | 0.160 |
| 16 | 1.948 | 0.007 | 0.001 | 0.018 | 0.012 | 0.165 |
| 17 | 2.077 | 0.007 | 0.001 | 0.019 | 0.009 | 0.166 |
| 18 | 2.206 | 0.007 | 0.001 | 0.020 | 0.009 | 0.166 |
| 19 | 2.334 | 0.008 | 0.001 | 0.018 | 0.009 | 0.166 |
| 20 | 2.463 | 0.008 | 0.002 | 0.018 | 0.009 | 0.166 |
| 21 | 2.592 | 0.008 | 0.002 | 0.018 | 0.009 | 0.162 |
| 22 | 2.721 | 0.008 | 0.002 | 0.017 | 0.011 | 0.162 |
| 23 | 2.850 | 0.008 | 0.002 | 0.018 | 0.011 | 0.159 |
| 24 | 2.978 | 0.008 | 0.002 | 0.015 | 0.011 | 0.157 |
| 25 | 3.107 | 0.008 | 0.002 | 0.014 | 0.009 | 0.163 |
| 26 | 3.236 | 0.008 | 0.002 | 0.014 | 0.011 | 0.159 |
| 27 | 3.365 | 0.008 | 0.002 | 0.014 | 0.009 | 0.163 |
| 28 | 3.494 | 0.008 | 0.002 | 0.014 | 0.009 | 0.165 |
| 29 | 3.623 | 0.008 | 0.002 | 0.016 | 0.010 | 0.155 |

**Table S2.** Online UHPLC results for the stability run.

| Injection | Timestamp (h) | [2ClBA] (mol/L) | [3N-2ClBA] (mol/L) | [5N-2ClBA] (mol/L) | [3-NSA] (mol/L) | [5-NSA] (mol/L) |
|---|---|---|---|---|---|---|
| 30 | 3.751 | 0.008 | 0.002 | 0.015 | 0.011 | 0.159 |
| 31 | 3.880 | 0.008 | 0.002 | 0.015 | 0.011 | 0.160 |
| 32 | 4.009 | 0.008 | 0.002 | 0.014 | 0.009 | 0.163 |
| 33 | 4.138 | 0.008 | 0.002 | 0.014 | 0.009 | 0.161 |
| 34 | 4.267 | 0.009 | 0.002 | 0.015 | 0.009 | 0.162 |
| 35 | 4.395 | 0.009 | 0.002 | 0.012 | 0.008 | 0.139 |
| 36 | 4.524 | 0.012 | 0.000 | 0.004 | 0.002 | 0.044 |
| 37 | 4.653 | 0.010 | 0.000 | 0.001 | 0.001 | 0.014 |
| 38 | 4.783 | 0.009 | 0.000 | 0.000 | 0.000 | 0.008 |
| 39 | 4.912 | 0.007 | 0.000 | 0.000 | 0.000 | 0.005 |
| 40 | 5.041 | 0.007 | 0.000 | 0.000 | 0.000 | 0.003 |
| 41 | 5.171 | 0.007 | 0.000 | 0.000 | 0.000 | 0.003 |
| 42 | 5.302 | 0.007 | 0.000 | 0.000 | 0.000 | 0.003 |
| 43 | 5.431 | 0.006 | 0.000 | 0.000 | 0.000 | 0.002 |
| 44 | 5.560 | 0.000 | 0.000 | 0.000 | 0.000 | 0.002 |

**Table S3.** Online UHPLC results for the run with dynamic changes.

| Injection | Timestamp (h) | [2ClBA] (mol/L) | [3N-2ClBA] (mol/L) | [5N-2ClBA] (mol/L) | [3-NSA] (mol/L) | [5-NSA] (mol/L) |
|---|---|---|---|---|---|---|
| 1 | 0.007 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 2 | 0.136 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 3 | 0.264 | 0.000 | 0.000 | 0.000 | 0.002 | 0.000 |
| 4 | 0.393 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 5 | 0.521 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 6 | 0.650 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 7 | 0.778 | 0.000 | 0.001 | 0.006 | 0.001 | 0.009 |
| 8 | 0.906 | 0.000 | 0.005 | 0.087 | 0.005 | 0.060 |
| 9 | 1.035 | 0.000 | 0.009 | 0.106 | 0.006 | 0.073 |
| 10 | 1.164 | 0.000 | 0.010 | 0.117 | 0.006 | 0.076 |
| 11 | 1.293 | 0.000 | 0.010 | 0.124 | 0.007 | 0.080 |
| 12 | 1.421 | 0.000 | 0.010 | 0.107 | 0.008 | 0.094 |
| 13 | 1.549 | 0.000 | 0.006 | 0.072 | 0.012 | 0.129 |
| 14 | 1.678 | 0.000 | 0.003 | 0.042 | 0.016 | 0.155 |
| 15 | 1.806 | 0.000 | 0.001 | 0.028 | 0.018 | 0.166 |
| 16 | 1.935 | 0.000 | 0.001 | 0.024 | 0.015 | 0.170 |
| 17 | 2.064 | 0.000 | 0.000 | 0.023 | 0.013 | 0.171 |
| 18 | 2.192 | 0.000 | 0.000 | 0.018 | 0.010 | 0.148 |
| 19 | 2.321 | 0.000 | 0.000 | 0.017 | 0.009 | 0.141 |
| 20 | 2.449 | 0.006 | 0.000 | 0.016 | 0.009 | 0.138 |
| 21 | 2.578 | 0.006 | 0.000 | 0.018 | 0.009 | 0.145 |
| 22 | 2.707 | 0.006 | 0.000 | 0.019 | 0.011 | 0.164 |
| 23 | 2.836 | 0.007 | 0.000 | 0.019 | 0.011 | 0.170 |
| 24 | 2.964 | 0.007 | 0.000 | 0.018 | 0.010 | 0.168 |
| 25 | 3.093 | 0.008 | 0.000 | 0.017 | 0.010 | 0.170 |
| 26 | 3.221 | 0.011 | 0.000 | 0.017 | 0.009 | 0.167 |
| 27 | 3.351 | 0.014 | 0.000 | 0.016 | 0.008 | 0.162 |
| 28 | 3.479 | 0.015 | 0.000 | 0.016 | 0.008 | 0.162 |
| 29 | 3.608 | 0.016 | 0.000 | 0.016 | 0.009 | 0.166 |
| 30 | 3.737 | 0.016 | 0.000 | 0.016 | 0.008 | 0.164 |
| 31 | 3.866 | 0.017 | 0.000 | 0.016 | 0.008 | 0.166 |
| 32 | 3.994 | 0.017 | 0.000 | 0.015 | 0.008 | 0.159 |
| 33 | 4.123 | 0.017 | 0.000 | 0.016 | 0.008 | 0.166 |
| 34 | 4.252 | 0.017 | 0.000 | 0.015 | 0.008 | 0.165 |
| 35 | 4.384 | 0.017 | 0.000 | 0.014 | 0.008 | 0.163 |
| 36 | 4.513 | 0.017 | 0.000 | 0.013 | 0.008 | 0.163 |

**Table S4.** Online UHPLC results for the run with dynamic changes.

| Injection | Timestamp (h) | [2ClBA] (mol/L) | [3N-2ClBA] (mol/L) | [5N-2ClBA] (mol/L) | [3-NSA] (mol/L) | [5-NSA] (mol/L) |
|---|---|---|---|---|---|---|
| 37 | 4.641 | 0.018 | 0.000 | 0.013 | 0.008 | 0.164 |
| 38 | 4.770 | 0.017 | 0.000 | 0.015 | 0.008 | 0.165 |
| 39 | 4.899 | 0.017 | 0.000 | 0.016 | 0.008 | 0.166 |
| 40 | 5.027 | 0.017 | 0.002 | 0.016 | 0.009 | 0.169 |
| 41 | 5.158 | 0.017 | 0.000 | 0.016 | 0.010 | 0.167 |
| 42 | 5.286 | 0.017 | 0.004 | 0.025 | 0.009 | 0.156 |
| 43 | 5.415 | 0.011 | 0.012 | 0.050 | 0.007 | 0.109 |
| 44 | 5.544 | 0.008 | 0.016 | 0.060 | 0.005 | 0.085 |
| 45 | 5.672 | 0.007 | 0.017 | 0.058 | 0.005 | 0.082 |
| 46 | 5.802 | 0.007 | 0.017 | 0.058 | 0.006 | 0.081 |

## 1.3 PAT Instrument Details

### 1.3.1 General Information for NMR

NMR spectra were acquired on a low field benchtop 43.795 MHz NMR spectrometer (Magritek, Spinsolve Ultra 43 MHz). A glass flow-through cell (internal volume = 800 µL, length = 550 mm) was inserted through the benchtop NMR to enable inline measurements for reaction monitoring. The spectra were recorded with a pulse angle of 90 °, acquisition time of 6.4 s, repetition time of 10.0 s and a single scan. The recorded spectra were preprocessed with PEAXACT 5.4 software (S-PACT) by phasing (Auto, Negative Peak Penalization) and baseline correction (Straight Line Subtraction). The highest peak (water) in the spectrum was aligned with the reference value of 5.00 ppm and the global range of the spectrum was reduced to 7.00 and 9.00 ppm.

The description of the indirect hard model (IHM) and validation errors can be obtained from reference S1.

The partial least squares (PLS) model was developed in PEAXACT by using the experimentally measured concentration values as training data. The spectra were classified into groups based on their concentration level for cross validation (leave-one-group out approach) during the training. The ranks of each individual species were adjusted accordingly, to obtain the lowest possible rank by low root-mean-square error of calibration ($RMSE_{PLS\_C}$) and low root-mean-square error of cross validation ($RMSE_{PLS\_CV}$). The ranks, $RMSE_{PLS\_C}$, and $RMSE_{PLS\_CV}$ for 2ClBA, 3N-2ClBA, and 5N-2ClBA are listed in Table S5.

**Table S5.** PLS model, overview of the ranks, $RMSE_{PLS\_C}$ and $RMSE_{PLS\_CV}$ for **2ClBA**, **3N-2ClBA,** and **5N-2ClBA**.

| | Ranks | $RMSE_{PLS\_C}$ (mM) | $RMSE_{PLS\_CV}$ (mM) |
|---|---|---|---|
| **2ClBA** | 7 | 1.5 | 7.9 |
| **3N-2ClBA** | 7 | 4.1 | 41.6 |
| **5N-2ClBA** | 7 | 6.1 | 52.0 |

### 1.3.2 Individual Component NMR Spectra

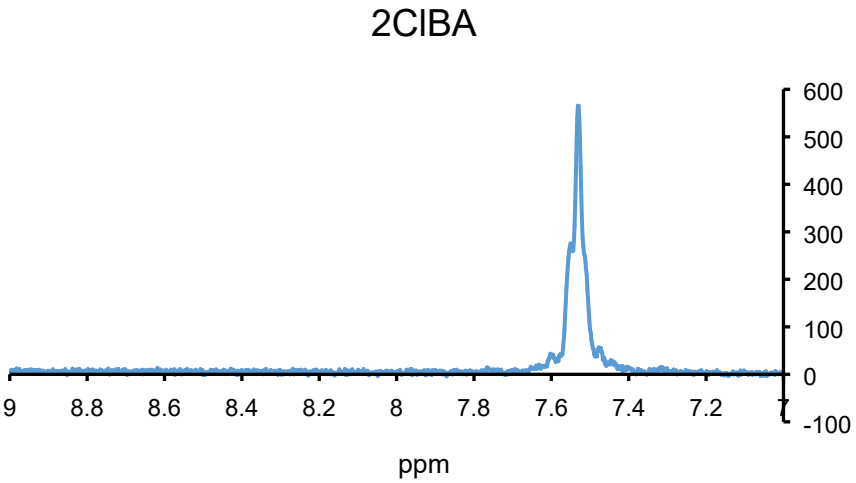



**Figure S1.** NMR spectrum of 2ClBA as a single component (0.15 M concentration).

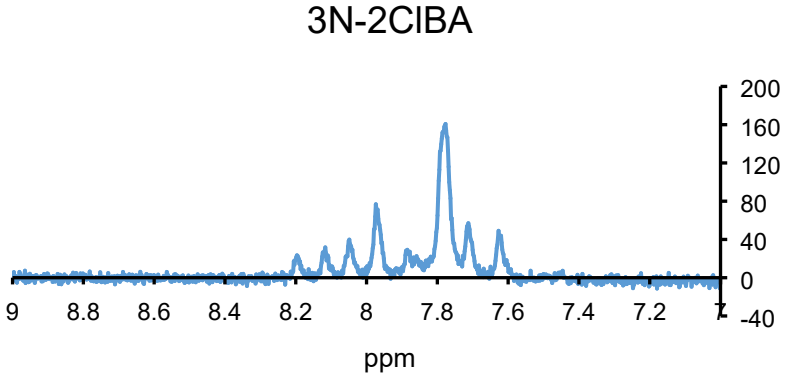



**Figure S2.** NMR spectrum of 3N-2ClBA as a single component (0.15 M concentration).

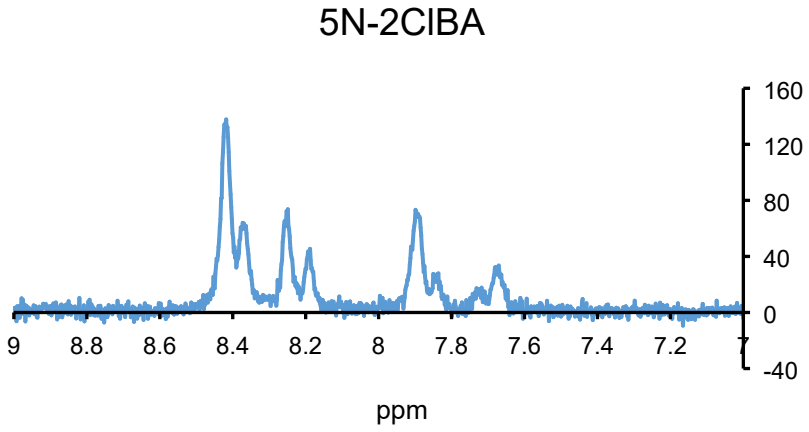



**Figure S3**. NMR spectrum of 5N-2ClBA as a single component (0.15 M concentration).

### 1.3.3 General Information for UV/vis Spectroscopy

The UV/vis spectra were acquired using a fiber-coupled Avantes Starline AvaSpec-ULS2048 spectrometer, with an Avantes AvaLight-DHc lamp as light source (deuterium and halogen lamp). The spectrometer was controlled using Avasoft 8.7 software and spectra were exported as csv file after each experiment. The spectra were recorded between a range from λ 199 – 769 nm, with an integration time of 20 ms and an average of 100 measurements per data point. Inline measurements were enabled by using a flow-through cell made out of a four-way connector (PEEK) and a bifurcated fiber-based reflection probe (FCR-7UVIR200-2-1.5x100). A detailed description of the flow cell can be found in reference S1.

### 1.3.4 Individual Component UV/vis Spectra

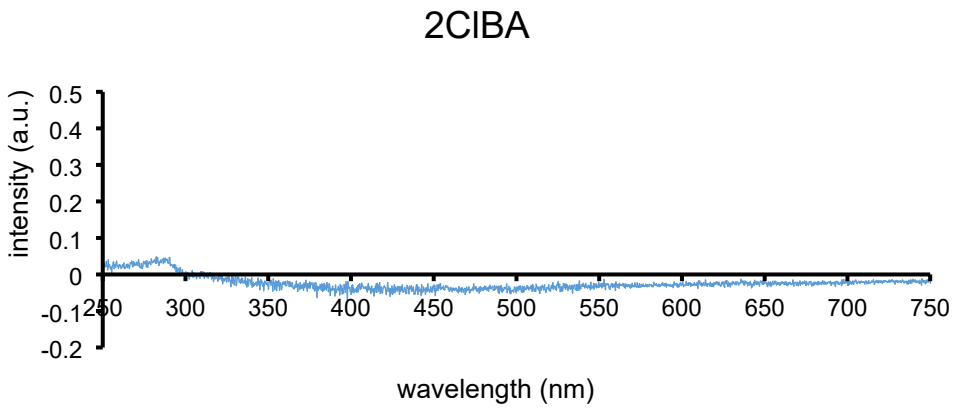



**Figure S4.** UV/vis spectrum of 2ClBA as a single component.

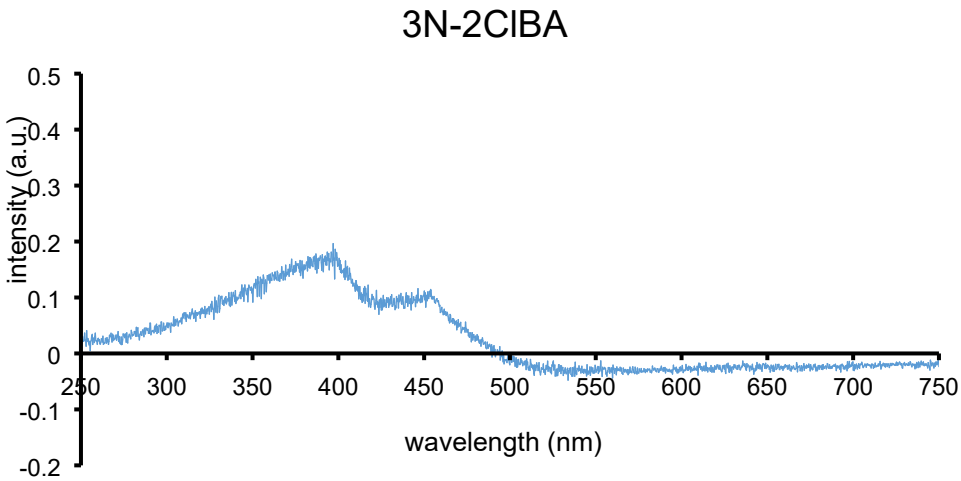



**Figure S5.** UV/vis spectrum of 3N-2ClBA as a single component.

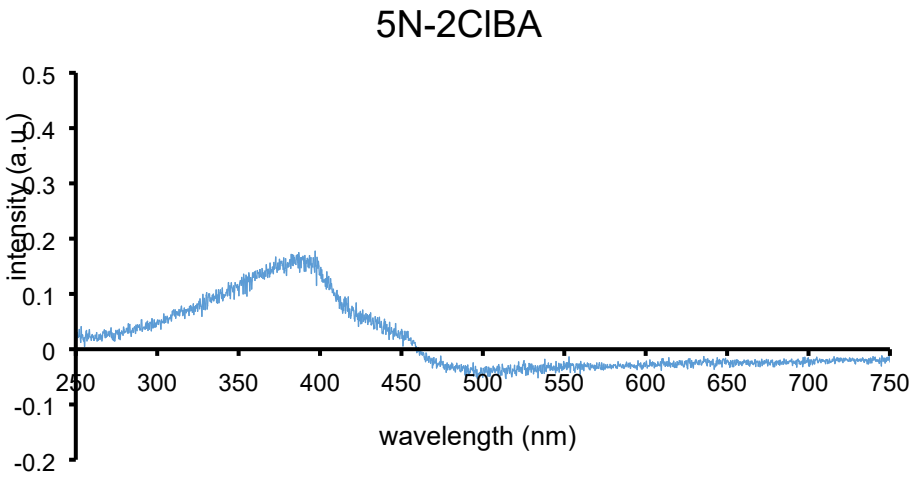



**Figure S6.** UV/vis spectrum of 5N-2ClBA as a single component.

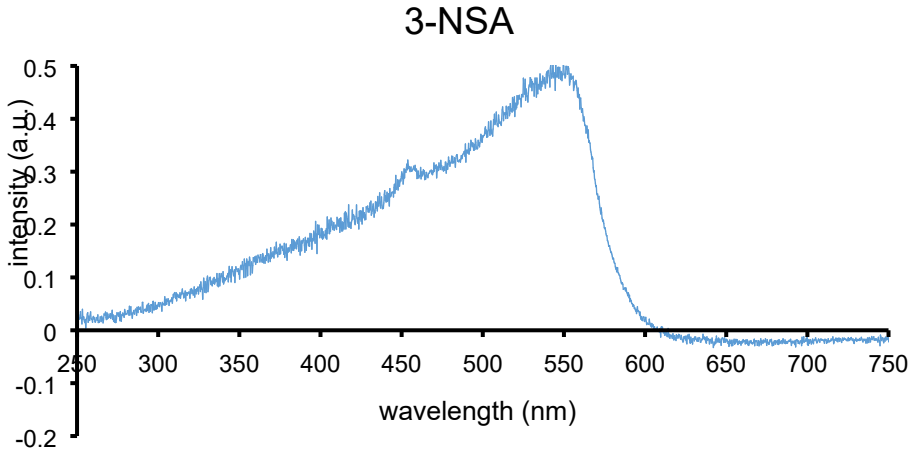



**Figure S7.** UV/vis spectrum of 3-NSA as a single component.

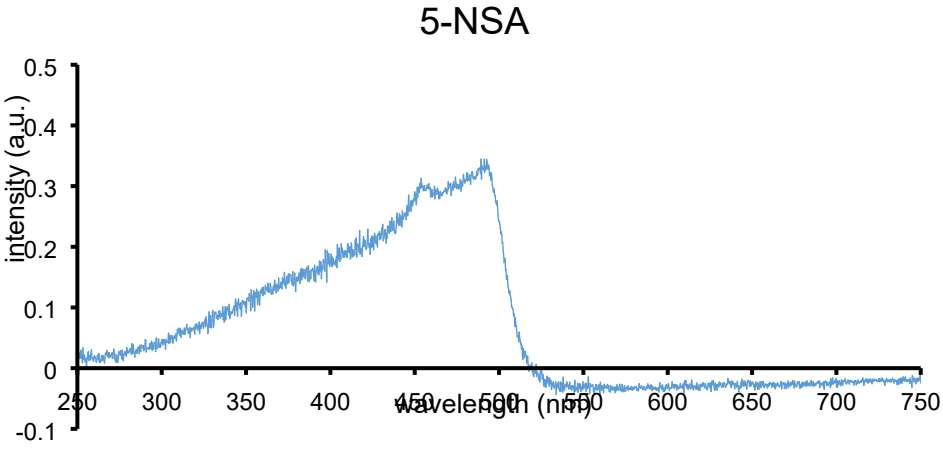



**Figure S8.** UV/vis spectrum of 5-NSA as a single component.

# 2 Development of Neural Network for NMR

## 2.1 Training Data

### 2.1.1 Experimentally Measured Concentration Levels

The concentration level solutions were prepared by weighing the correct amount of 2ClBA, 3N-2ClBA, and 5N-2ClBA into a 10 mL volumetric flask (Table S 6). The flask was filled up with 0.5 M NaOH to the 10 mL mark. The solutions were recirculated with a peristaltic pump (Ismatec, ISM834C) through the benchtop NMR. The pump speed was 10 rpm, which approximately corresponds to a flow rate of 1 mL/min.

**Table S 6.** Overview of the prepared solutions used as pure component spectra and as training data for the ANN, PLS and IHM models.

| | 2ClBA (mM) | 3N-2ClBA (mM) | 5N-2ClBA (mM) |
|---|---|---|---|
| Pure_2ClBA | 251.7 | 0 | 0 |
| Pure_3N-2ClBA | 0 | 105.1 | 0 |
| Pure_5N-2ClBA | 0 | 0 | 249.5 |
| Level_1 | 152.1 | 15.1 | 121.4 |
| Level_2 | 33.9 | 54.7 | 217.8 |
| Level_3 | 222.1 | 24.5 | 61.9 |
| Level_4 | 8.8 | 37.7 | 255.5 |
| Level_5 | 18.8 | 35.6 | 194.0 |
| Level_6 | 6.5 | 16.1 | 227.6 |
| Level_7 | 0 | 0 | 0 |

### 2.1.2 Simulation of NMR Spectra

The pure spectra and code for simulating NMR spectra is available from the authors or on GitHub.[S2]

### 2.1.3 Continuous Validation Data Set

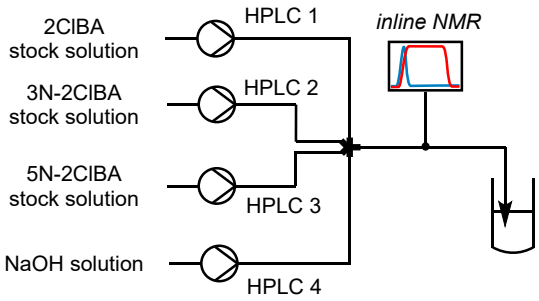



**Figure S9.** Process scheme for the measurement of the continuous validation set.

The 0.800 mol/L stock solution for **2ClBA** was prepared by weighing 12.5377 g of **2ClBA** into a 100 mL volumetric flask. The 0.250 mol/L stock solution for **3N-2ClBA** was prepared by weighing 2.5244 g of **3N-2ClBA** into a 50 mL volumetric flask. The 0.597 mol/L stock solution for **5N-2ClBA** was prepared by weighing 12.032 g into a 100 mL volumetric flask. The 0.1 mol/L NaOH solution was prepared by weighing approximately 4 g into a 1000 mL volumetric flask.

The four solutions were pumped with Knauer AZURA P 4.1S HPLC pumps (10 mL/min pump head, made of Hastelloy C or stainless steel and an integrated pressure sensor) and mixed in a 6-way mixer with one blocked port. The total flow rate of all four pumps stayed constant at 1.0 mL/min. The mixed solution was passed through the benchtop NMR flow through cell and the outlet was collected.

The continuous validation set was dynamically recorded by varying the flow rates for each individual HPLC pump. The flow rates of the HPLC pumps and the calculated concentrations of **2ClBA**, **3N-2ClBA,** and **5N-2ClBA** are depicted in Figure S10.

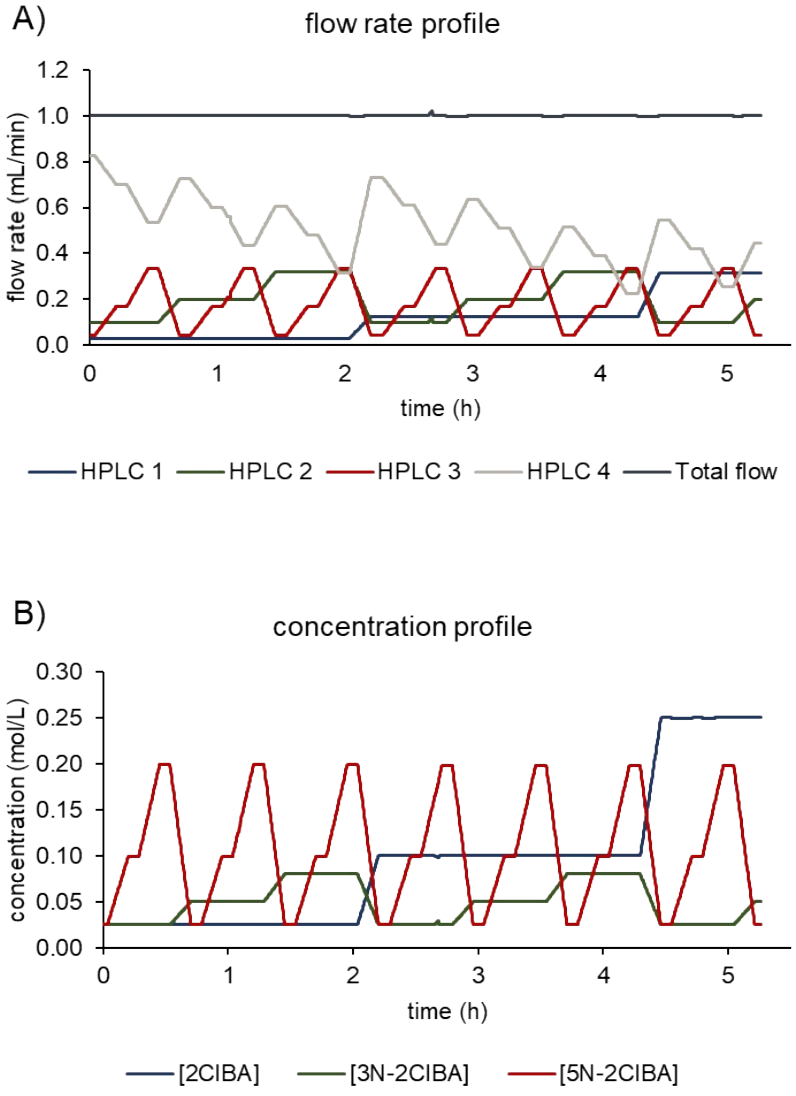



**Figure S10.** Flow rate **(A)** and concentration profile **(B)** for experiment to obtain the dynamic validation set.

### 2.1.4 Comparison of Simulated to Experimentally Recorded NMR Spectra

The comparison between the simulated and experimentally measured spectra are depicted in Figure S11. The spectra have been slightly shifted for better comparison.

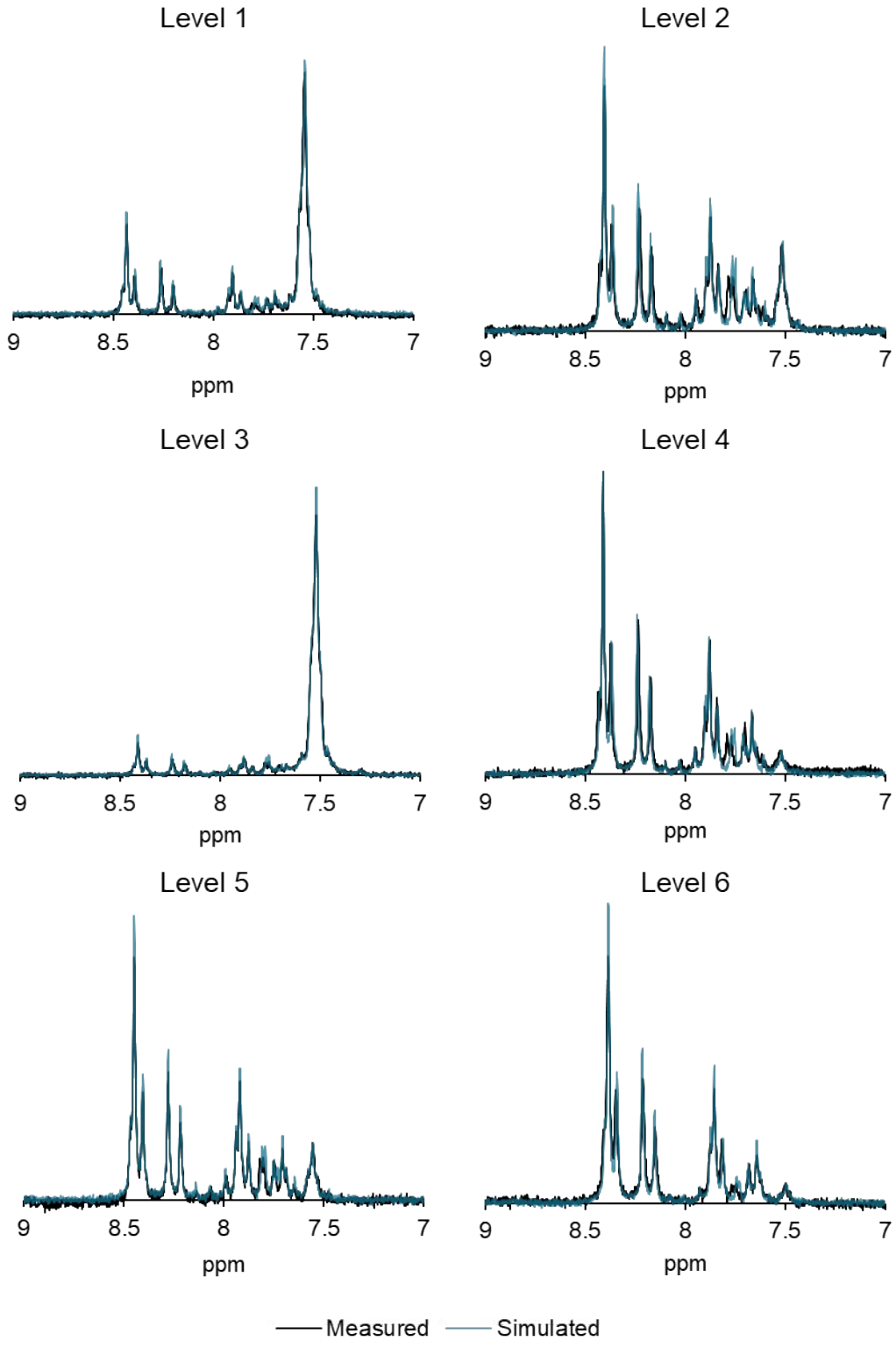



**Figure S11.** Comparison of the simulated spectra to the experimentally recorded spectra**.**

## 2.2 Neural Network Architecture

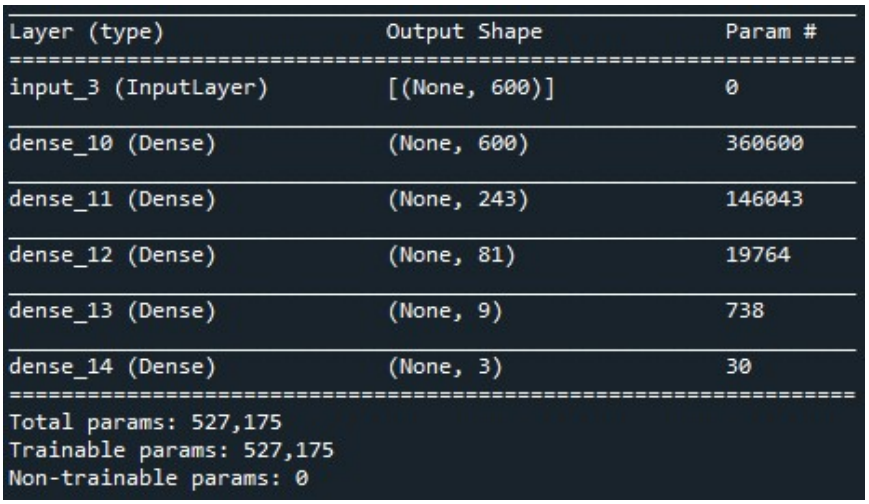
The models and the python code for training can be obtained from GitHub.[S2]

### 2.2.1 Final Model for the Fully Dense Neural Network

A total of 527175 parameters could be adjusted during training. The “adam” optimizer was used and monitored the validation loss. A detailed description of the model can be found in Figure S12 and Figure S13.

```
model_path_name = 'NMR_fully_final.hdf5'
# Architecture of the NMR model
visible = Input(shape=(600,)) # input layer (spectrum size)
hidden11 = Dense(600,activation='relu')(visible) #dense layer 1 connected to the input layer (tune neurons and activation function)
hidden12 = Dense(243, activation='relu')(hidden11) #dense layer 2 connected to the densee layer 1 (tune neurons and activation function)
hidden13 = Dense(81, activation='relu')(hidden12)  #dense layer 3 connected to the densee layer 2 (tune neurons and activation function)
hidden14 = Dense(9, activation='relu')(hidden13) #dense layer 4 connected to the densee layer 3 (tune neurons and activation function)
output = Dense(3, activation='relu')(hidden14) #model output dense layer 5 connected to the dense layer 4 (3 neurons represent the 3 intermediates measured at this point, tune the activation function)
model = Model(inputs=visible, outputs=output) #assign inputs and outputs of the model
# summarize model
print(model.summary())
# compile the fully dense ANN model
model.compile(loss='mse', optimizer='adam', metrics=['accuracy'])
# define learning and checkpointer
learning = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=20, min_lr=0.001)
checkpointer = ModelCheckpoint(filepath = path2 + model_path_name, monitor='val_loss', verbose=1, save_best_only=True)
#%% training of the model with validation data
Epochs = 2000
hist = model.fit(training_X, training_Y, epochs=Epochs, batch_size=500, verbose=1, validation_data=(validation_X, validation_Y) , callbacks=[checkpointer, learning])
```

**Figure S12.** Python code for the final fully dense neural network.



| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_3 (InputLayer) | [(None, 600)] | 0 |
| dense_10 (Dense) | (None, 600) | 360600 |
| dense_11 (Dense) | (None, 243) | 146043 |
| dense_12 (Dense) | (None, 81) | 19764 |
| dense_13 (Dense) | (None, 9) | 738 |
| dense_14 (Dense) | (None, 3) | 30 |

Total params: 527,175
Trainable params: 527,175
Non-trainable params: 0

**Figure S13.** Model summary model showing the trainable parameters for each layer.

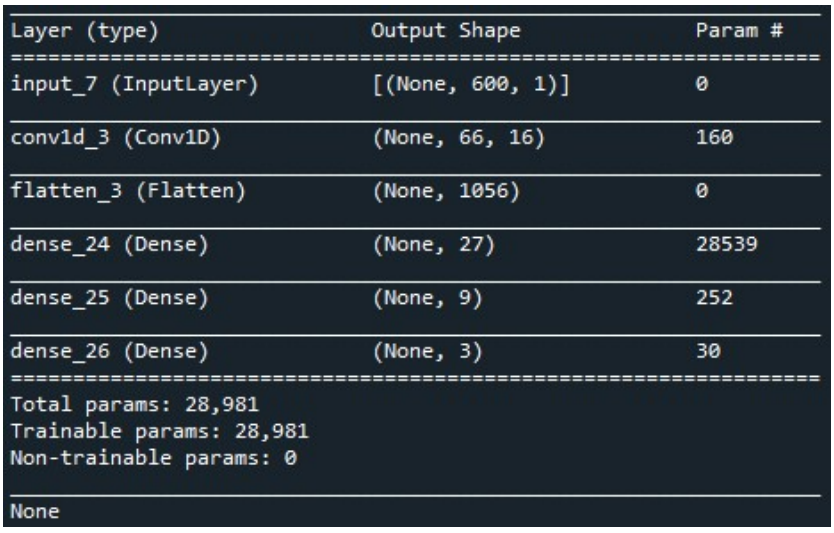
### 2.2.2 Final Model for the Convolutional Neural Network (Conv1D)

A total of 28981 parameters could be adjusted during training. The "adam" optimizer was used and monitored the validation loss. A detailed description of the model can be found in Figure S14 and Figure S15.

```
model_path_name = 'NMR_conv1D_final.hdf5'
#Model for NMR
visible = Input(shape=(600,1)) #input layer (spectrum size)
conv1 = Conv1D(filters=16, kernel_size=9, strides=9, activation='relu')(visible)
#convolutional layer (tune filters, kernel size, strides and the activation function)
flat = Flatten()(conv1) #flatten of the convolutional layer
hidden11 = Dense(27, activation='relu')(flat) #dense layer 1 connected to the flatten
layer (tune neurons and activation function)
hidden12 = Dense(9, activation='relu')(hidden11) #dense layer 2 connected to the
densee layer 1 (tune neurons and activation function)
output = Dense(3, activation='relu')(hidden12) #model output dense layer 3 connected
to the dense layer 2 (3 neurons represent the 3 intermediates measured at this point,
tune the activation function)
model = Model(inputs=visible, outputs=output) #assign inputs and outputs of the model
# summarize model
print(model.summary())
# compile the conv1D ANN model
learning_rate = 0.001
optimizer = Adam(lr=learning_rate, beta_1=0.9, beta_2=0.999, epsilon=10e-8, decay=0,
amsgrad=False)
model.compile(loss='mse', optimizer=optimizer, metrics=['accuracy'])
# define learning and checkpointer
learning = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=20,
min_lr=0.001)
```

```
checkpointer = ModelCheckpoint(filepath = path2 + model_path_name,
monitor='val_loss', verbose=1, save_best_only=True)
#%% training of the model with validation data
Epochs = 2000
hist = model.fit(training_X, training_Y, epochs=Epochs, batch_size=500, verbose=1,
validation_data=(validation_X, validation_Y) , callbacks=[checkpointer, learning])
```

**Figure S14.** Python code for the final conv1D neural network.



| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_7 (InputLayer) | [(None, 600, 1)] | 0 |
| conv1d_3 (Conv1D) | (None, 66, 16) | 160 |
| flatten_3 (Flatten) | (None, 1056) | 0 |
| dense_24 (Dense) | (None, 27) | 28539 |
| dense_25 (Dense) | (None, 9) | 252 |
| dense_26 (Dense) | (None, 3) | 30 |

Total params: 28,981
Trainable params: 28,981
Non-trainable params: 0

None

**Figure S15.** Model summary model showing the trainable parameters for each layer.

### 2.2.3 Final Model for the Convolutional Neural Network (Locally Connected 1D)

The best 1D locally connected ANN model was comprised of 1D locally connected convolutional layer with 16 filters, kernel size of 9, stride size of 9. The output was flattened and followed by 3 fully dense layers of 27, 9 and 3 neurons, respectively (Figure S16). The activation function for the convolutional layer and the two following dense layers were an exponential linear unit (elu) function. The output layer was connected with a rectified linear unit (relu) activation function to avoid negative predictions. A total number of 39381 parameters could be adjusted during the training. The root mean square error on the 7 experimental measured concentration levels ($RMSE_{Vexp}$) was calculated to be 3.1 mM for **2ClBA**, 1.7 mM for **3N-2ClBA** and 4.6 mM **5N-2ClBA** for the locally connected 1D network.

```
model_path_name = 'NMR_locally_conncected1D_final.hdf5'
#Model for NMR
visible = Input(shape=(600,1)) #input layer (spectrum size)
conv1 = LocallyConnected1D(filters=16, kernel_size=9, strides=9, activation='elu')
(visible) #convolutional layer (tune filters, kernel size, strides and the activation
function)
flat = Flatten()(conv1) #flatten of the convolutional layer
hidden11 = Dense(27, activation='elu')(flat) #dense layer 1 connected to the flatten
layer (tune neurons and activation function)
```

```
hidden12 = Dense(9, activation='elu')(hidden11) #dense layer 2 connected to the densee layer 1 (tune neurons and activation function)
output = Dense(3, activation='relu')(hidden12) #model output dense layer 3 connected to the dense layer 2 (3 neurons represent the 3 intermediates measured at this point, tune the activation function)
model = Model(inputs=visible, outputs=output) #assign inputs and outputs of the model
# summarize model
print(model.summary())
# compile the locally connected 1D ANN model
learning_rate = 0.0001
optimizer = Adam(lr=learning_rate, beta_1=0.9, beta_2=0.999, epsilon=10e-8, decay=0, amsgrad=False)
model.compile(loss='mse', optimizer=optimizer, metrics=['accuracy'])
# define learning and checkpointer
learning = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=20, min_lr=0.001)
checkpointer = ModelCheckpoint(filepath = path2 + model_path_name, monitor='val_loss', verbose=1, save_best_only=True)
#%% training of the model with validation data
Epochs = 2000
hist = model.fit(training_X, training_Y, epochs=Epochs, batch_size=500, verbose=1, validation_data=(validation_X, validation_Y) , callbacks=[checkpointer, learning])
```

**Figure S16.** Python code for the final locally connected 1D neural network.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 600, 1)]          0
_________________________________________________________________
locally_connected1d (Locally (None, 66, 16)            10560
_________________________________________________________________
flatten (Flatten)            (None, 1056)              0
_________________________________________________________________
dense (Dense)                (None, 27)                28539
_________________________________________________________________
dense_1 (Dense)              (None, 9)                 252
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 30
=================================================================
Total params: 39,381
Trainable params: 39,381
Non-trainable params: 0
_________________________________________________________________
None
```

**Figure S17.** Model summary model showing the trainable parameters for each layer.

## 2.3 Prediction of Process Data

### 2.3.1 Stability Run

The prediction of **2ClBA**, **3N-2ClBA**, **5N-2ClBA** during the stability run using a conv1D layer is depicted in Figure S18. The predictions using a locally connected 1D layer are visualized in Figure S20. Parity plots between IHM and ANN are displayed in Figure S19 and Figure S21

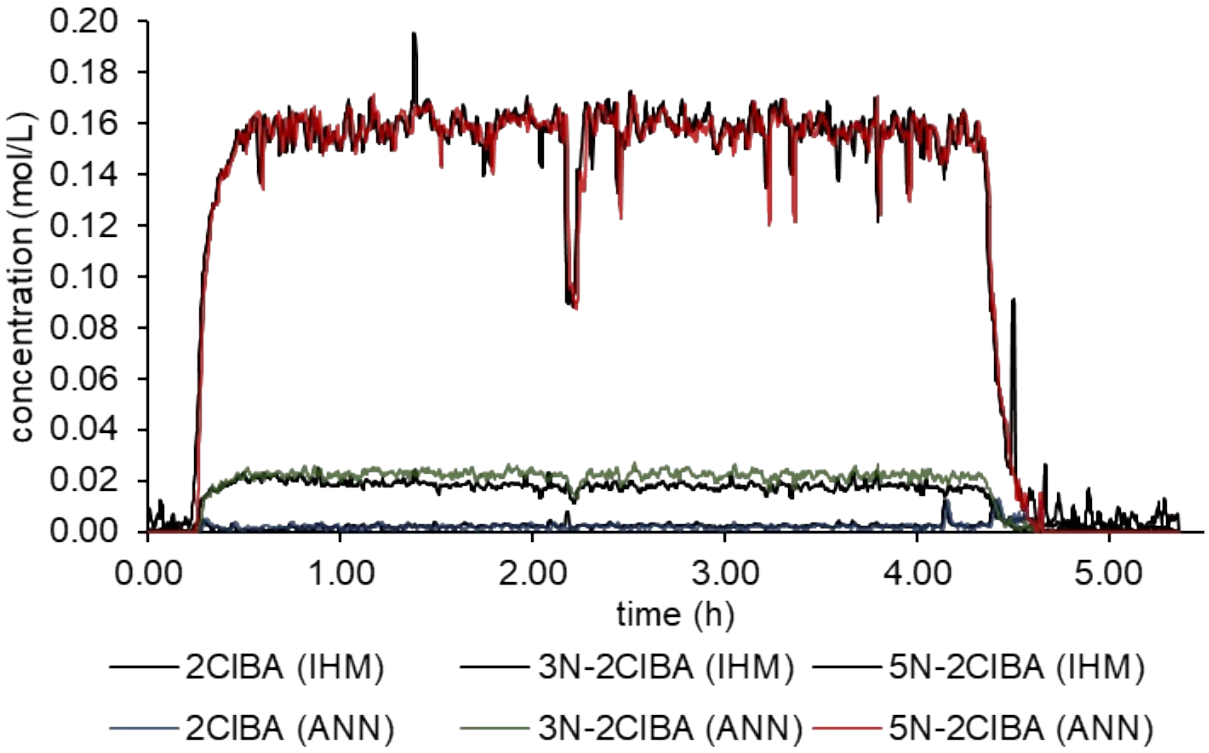



**Figure S18.** Prediction of the stability run using IHM and the conv1D ANN for NMR.

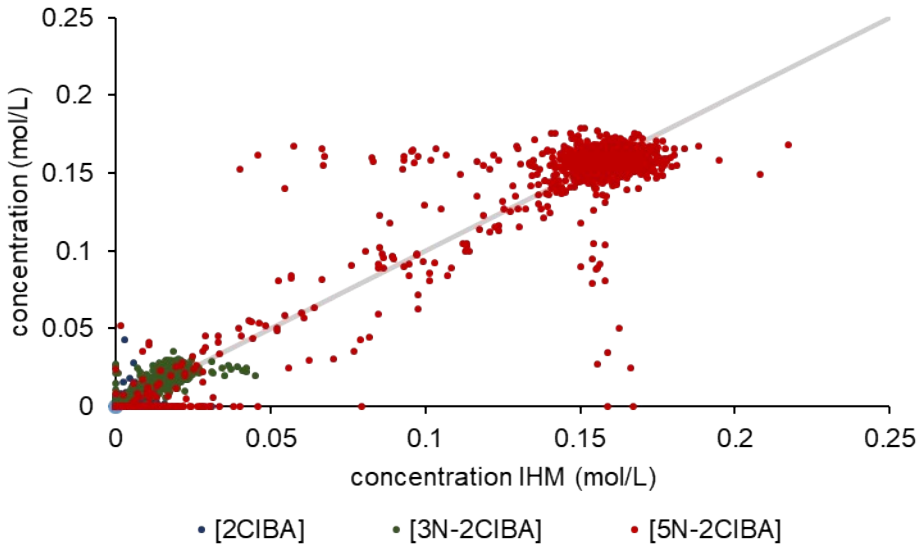



**Figure S19.** Parity plot of the obtained concentration predictions using IHM versus the conv1D ANN for the stability run. The coefficient of determination ($R^2$) is 0.125, 0.807 and 0.920 for 2ClBA, 3N-2ClBA, and 5N-2ClBA, respectively.

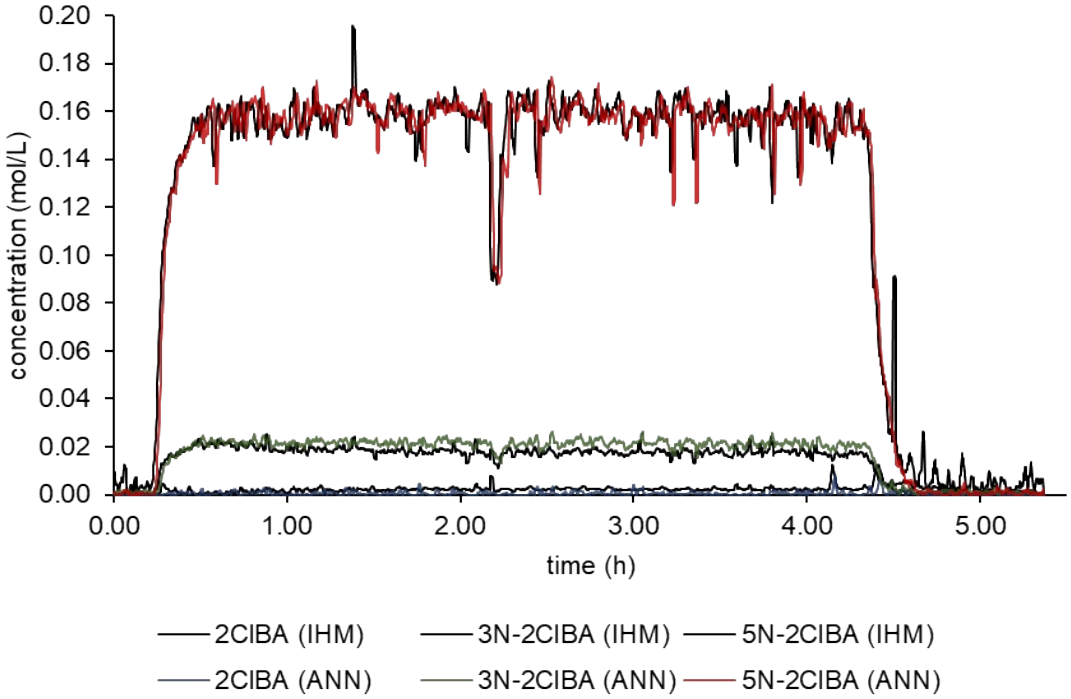



**Figure S20.** Prediction of the stability run using IHM and the locally connected 1D ANN for NMR.

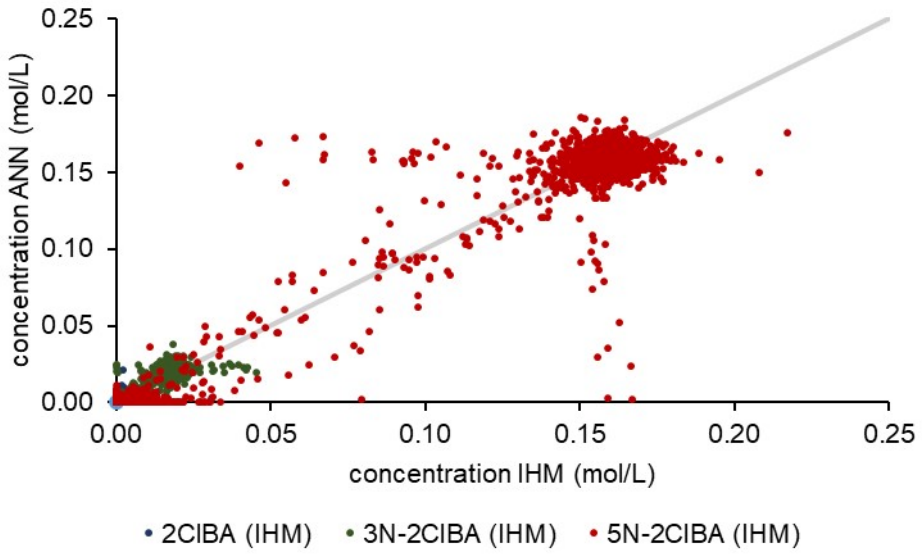



**Figure S21.** Parity plot of the obtained concentration predictions using IHM versus the locally connected 1D ANN for the stability run. The coefficient of determination ($R^2$) is 0.025, 0.802 and 0.917 for 2ClBA, 3N-2ClBA, and 5N-2ClBA, respectively.

### 2.3.2 Run with Dynamic Changes

The prediction of **2ClBA**, **3N-2ClBA**, **5N-2ClBA** during the run with dynamic changes using a locally connected 1D layer is depicted in Figure S23 (similar predictions from the conv1D ANN are found in manuscript Fig. 4C). Parity plots between IHM and ANN are displayed in Figure S22 and Figure S24.

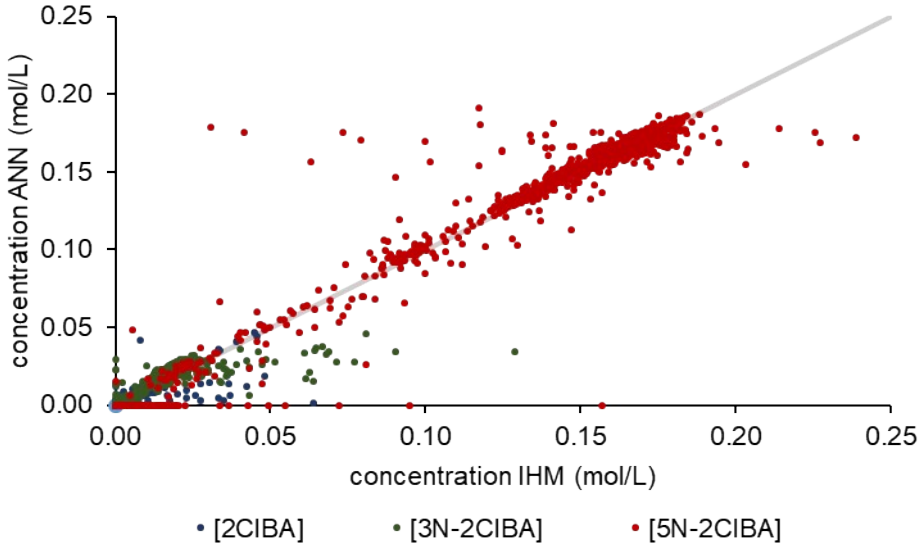



**Figure S22.** Parity plot of the obtained concentration predictions using IHM versus the conv1D ANN for the run with dynamic changes The coefficient of determination ($R^2$) is 0.764, 0.667 and 0.0466 for 2ClBA, 3N-2ClBA, and 5N-2ClBA, respectively.

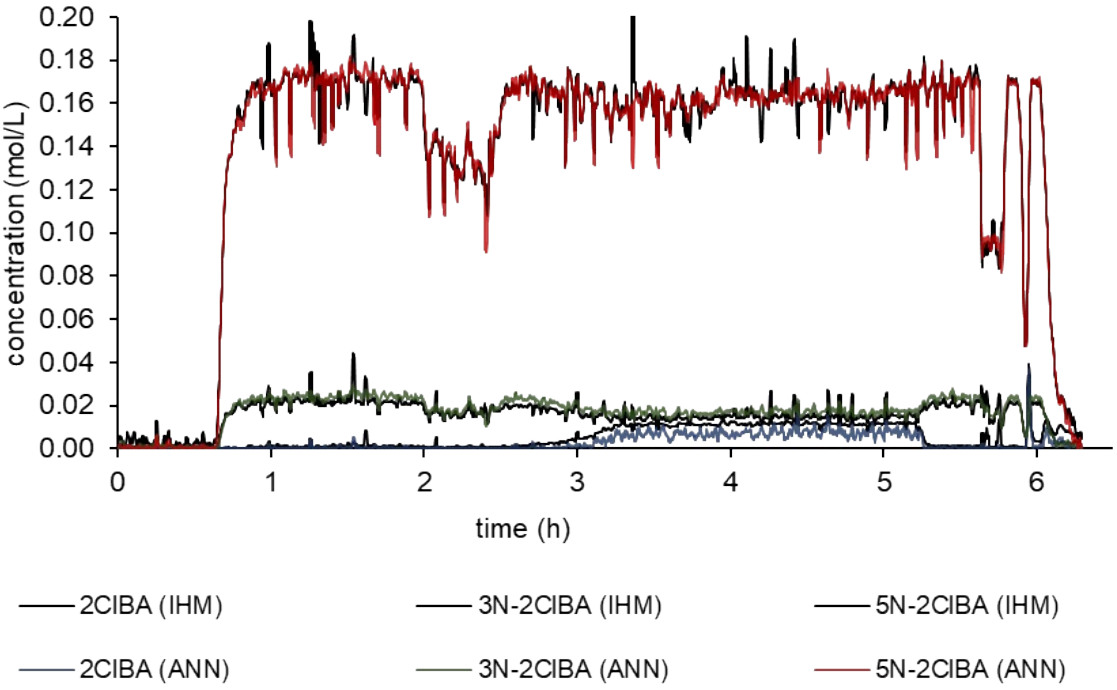



**Figure S23.** Prediction of the run with dynamic changes using IHM and the locally connected 1D ANN for NMR.

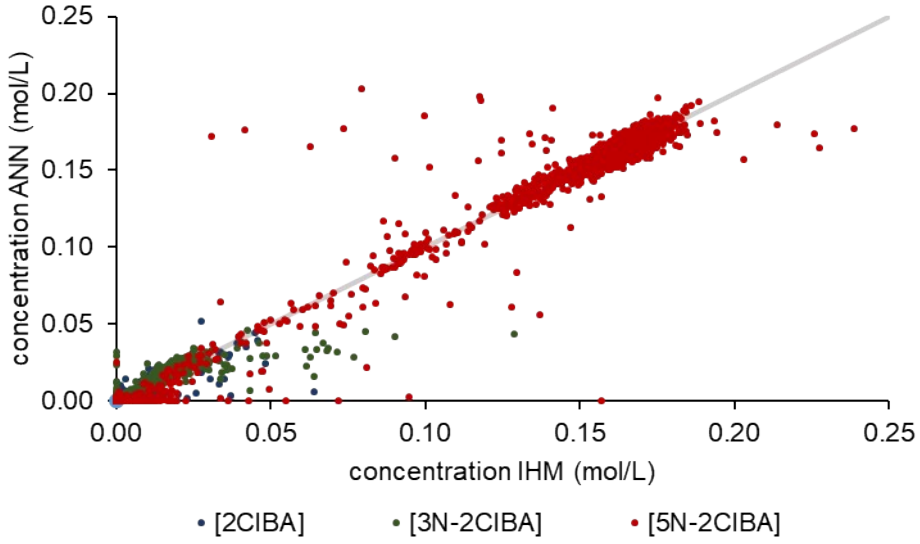
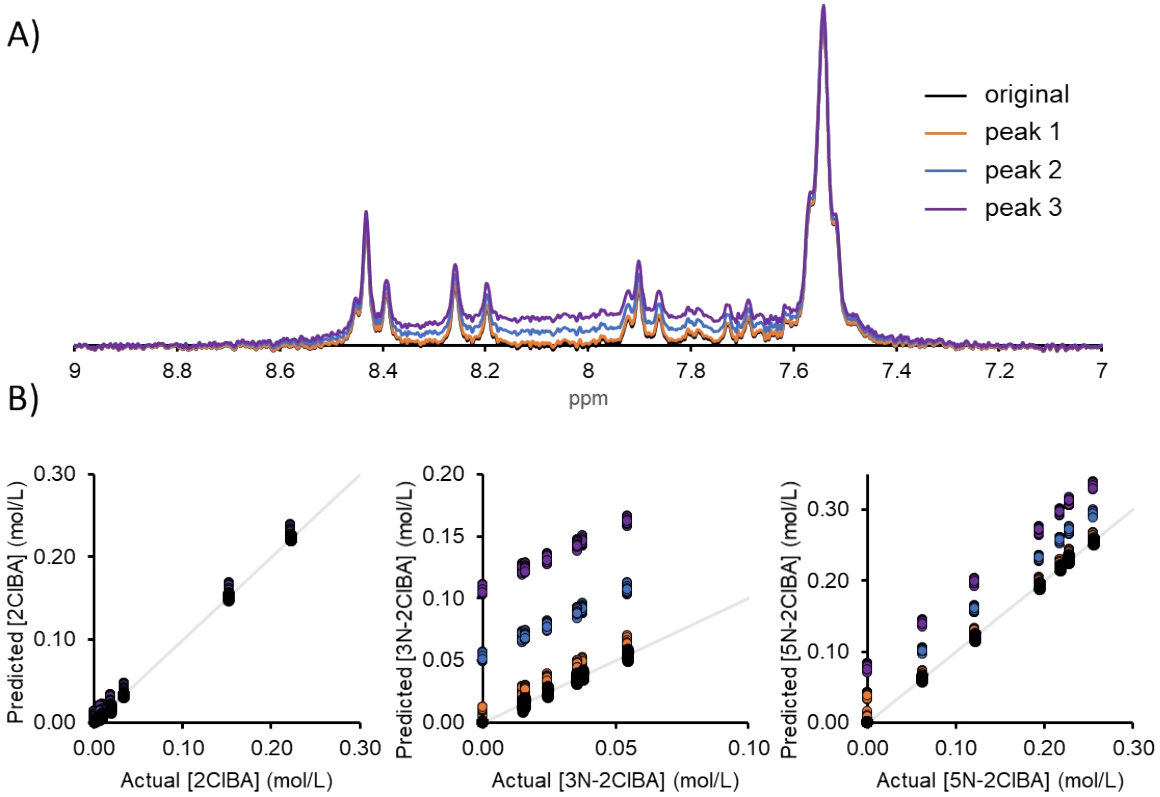



**Figure S24.** Parity plot of the obtained concentration predictions using IHM versus the locally connected 1D ANN for the run with dynamic changes. The coefficient of determination ($R^2$) is 0.623, 0.712 and 0.501 for 2ClBA, 3N-2ClBA, and 5N-2ClBA, respectively.

## 2.4 Evaluation of the ANN Robustness by adding Peaks to the Spectrum

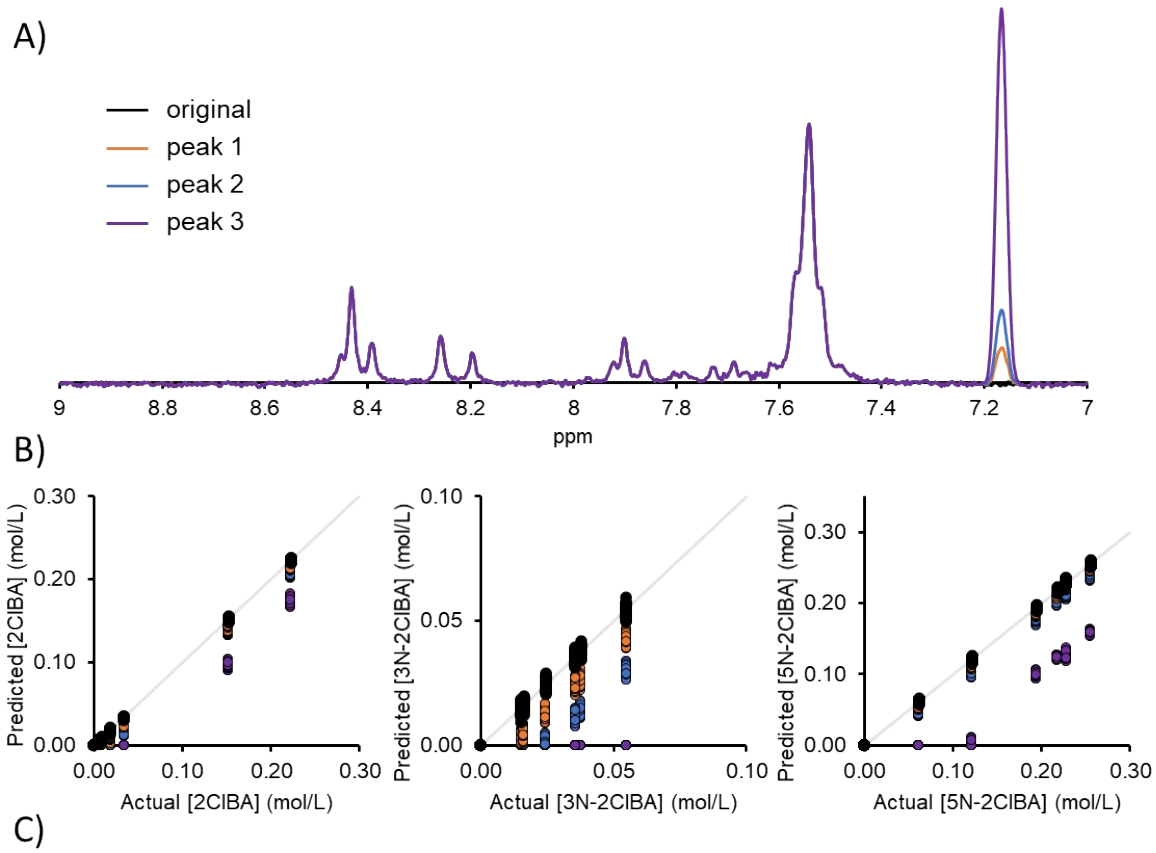
In python, an additional Gaussian peak was added to the experimentally measured concentration level spectra, to evaluate the robustness of the final Conv1D model. The simulated peak had a broad gaussian shape with different intensities (Figure S25A). Additionally, sharper peaks with different intensities and position shift were evaluated (Figure S26A, Figure S27A, and Figure S28A). The parity plots are provided (Figure S25B to Figure S28B) and the calculated root mean square errors for each component was calculated (Figure S25C to Figure S28C).




C)

| | original RMSE (mM) | peak 1 RMSE (mM) | peak 2 RMSE (mM) | peak 3 RMSE (mM) |
|---|---|---|---|---|
| 2ClBA | 1.2 | 1.7 | 6.4 | 12.4 |
| 3N-2ClBA | 1.5 | 10.1 | 53.6 | 107.8 |
| 5N-2ClBA | 2.0 | 8.0 | 39.7 | 79.1 |

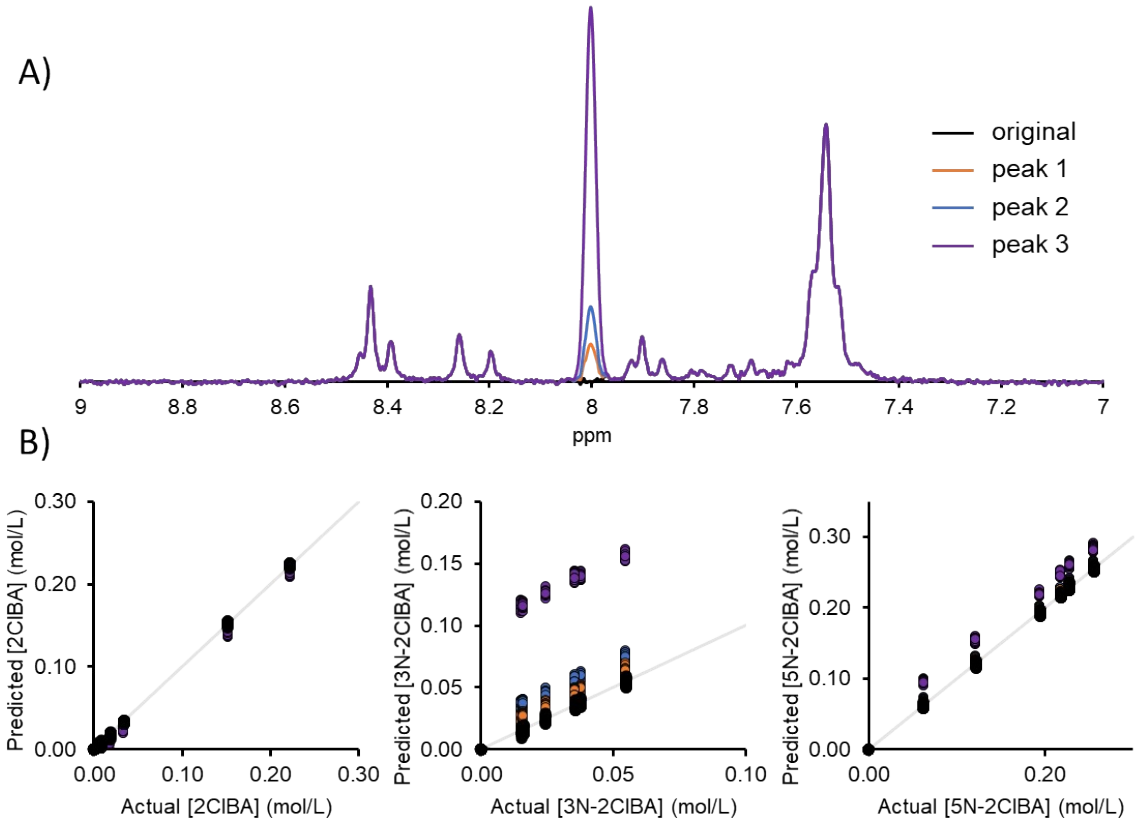
**Figure S25. A) Example spectrum with different simulated peaks. B) Parity plots for 2ClBA, 3N-2ClBA, and 5N-2ClBA. C) Table of the root mean square error for 2ClBA, 3N-2ClBA, and 5N-2ClBA.**




C)

| | original RMSE (mM) | peak 1 RMSE (mM) | peak 2 RMSE (mM) | peak 3 RMSE (mM) |
|---|---|---|---|---|
| 2ClBA | 1.2 | 8.2 | 14.7 | 32.5 |
| 3N-2ClBA | 1.5 | 11.0 | 20.0 | 31.0 |
| 5N-2ClBA | 2.0 | 8.0 | 17.0 | 88.6 |

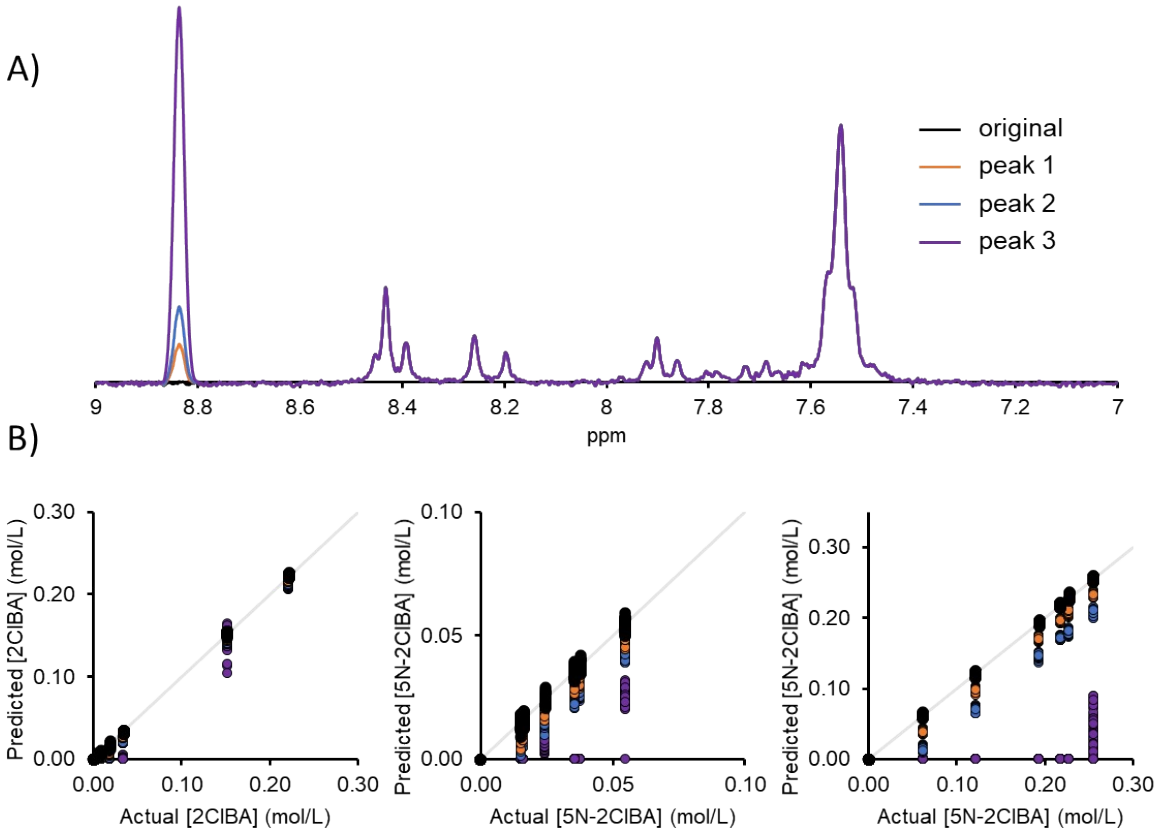
**Figure S26.** A) Example spectrum with different simulated peaks. B) Parity plots for 2ClBA, 3N-2ClBA, and 5N-2ClBA. C) Table of the root mean square error for 2ClBA, 3N-2ClBA, and 5N-2ClBA.




C)

| | original RMSE (mM) | peak 1 RMSE (mM) | peak 2 RMSE (mM) | peak 3 RMSE (mM) |
|---|---|---|---|---|
| 2ClBA | 1.2 | 1.4 | 1.7 | 8.4 |
| 3N-2ClBA | 1.5 | 9.4 | 19.1 | 93.9 |
| 5N-2ClBA | 2.0 | 3.5 | 6.0 | 28.3 |

**Figure S27.** A) Example spectrum with different simulated peaks. B) Parity plots for 2ClBA, 3N-2ClBA, and 5N-2ClBA. C) Table of the root mean square error for 2ClBA, 3N-2ClBA, and 5N-2ClBA.




C)

| | original | peak 1 | peak 2 | peak 3 |
|---|---|---|---|---|
| 2ClBA | 1.2 | 5.5 | 9.7 | 16.3 |
| 3N-2ClBA | 1.5 | 5.4 | 9.5 | 25.1 |
| 5N-2ClBA | 2.0 | 21.0 | 44.4 | 172.2 |

**Figure S28.** A) Example spectrum with different simulated peaks. B) Parity plots for 2ClBA, 3N-2ClBA, and 5N-2ClBA. C) Table of the root mean square error for 2ClBA, 3N-2ClBA, and 5N-2ClBA.

# 3 Data Fusion of NMR and UV/vis

## 3.1 Training Data

The UV/vis data for the data fusion ANN can be found in reference S1. The corresponding NMR spectra were simulated based on the approach in section 2.1.2.

### 3.1.1 Multi-Dimensional Dynamic Experiment

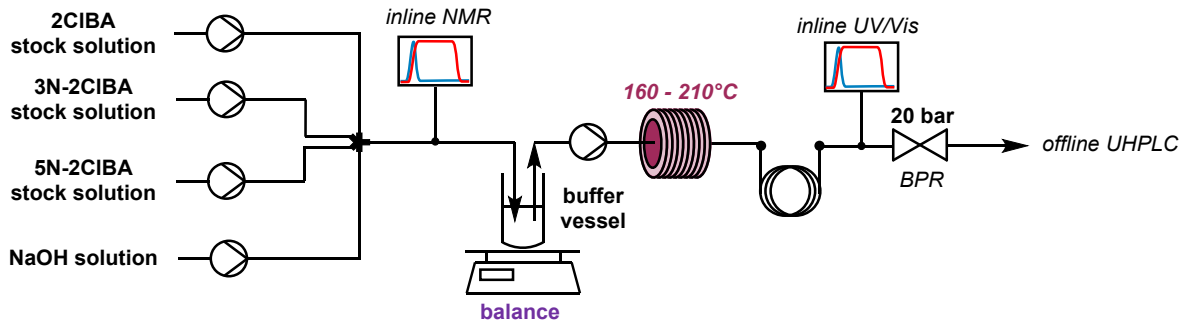



**Figure S29.** Process scheme for the measurement of the training data for the data fusion ANN.

The 0.800 mol/L stock solution for **2ClBA** was prepared by weighing 12.5377 g of **2ClBA** into a 100 mL volumetric flask. The 0.250 mol/L stock solution for **3N-2ClBA** was prepared by weighing 2.5244 g of **3N-2ClBA** into a 50 mL volumetric flask. The 0.597 mol/L stock solution for **5N-2ClBA** was prepared by weighing 12.032 g into a 100 mL volumetric flask. The 0.1 mol/L NaOH solution was prepared by weighing approximately 4 g into a 1000 mL volumetric flask.

The four solutions were pumped with Knauer AZURA P 4.1S HPLC pumps (10 mL/min pump head, made of Hastelloy C or stainless steel and an integrated pressure sensor) and mixed in a 6-way mixer with one blocked port. The total flow rate of all four pumps stayed constant at 1.0 mL/min, as individual flow rates were varied throughout the experiment (Figure S30, A). The mixed solution passed through the benchtop NMR flow through cell and the outlet was collected in a buffer vessel. The solution in the buffer vessel was pumped with Knauer Auzura HPLC pump through a 20 mL stainless steel coil placed on a coil heater (Unigsis, HotCoil, UQ1025-1). The temperature of the coil heater was ramped up and down during the experiment (Figure S30B). The reaction mixture was cooled to room temperature after the coil heater and passed through an inline UV-vis flow cell. . The UV-vis spectra were recorded using a fiber-coupled Avantes Starline AvaSpec-ULS2048 spectrometer. The outlet of the flow cell was connected to a Swagelok KCB series backpressure regulator (KCB1H0A2B5P60000), which was set to 20 bar. The process outlet was sampled every three minutes for offline UHPLC measurements (Table S7).

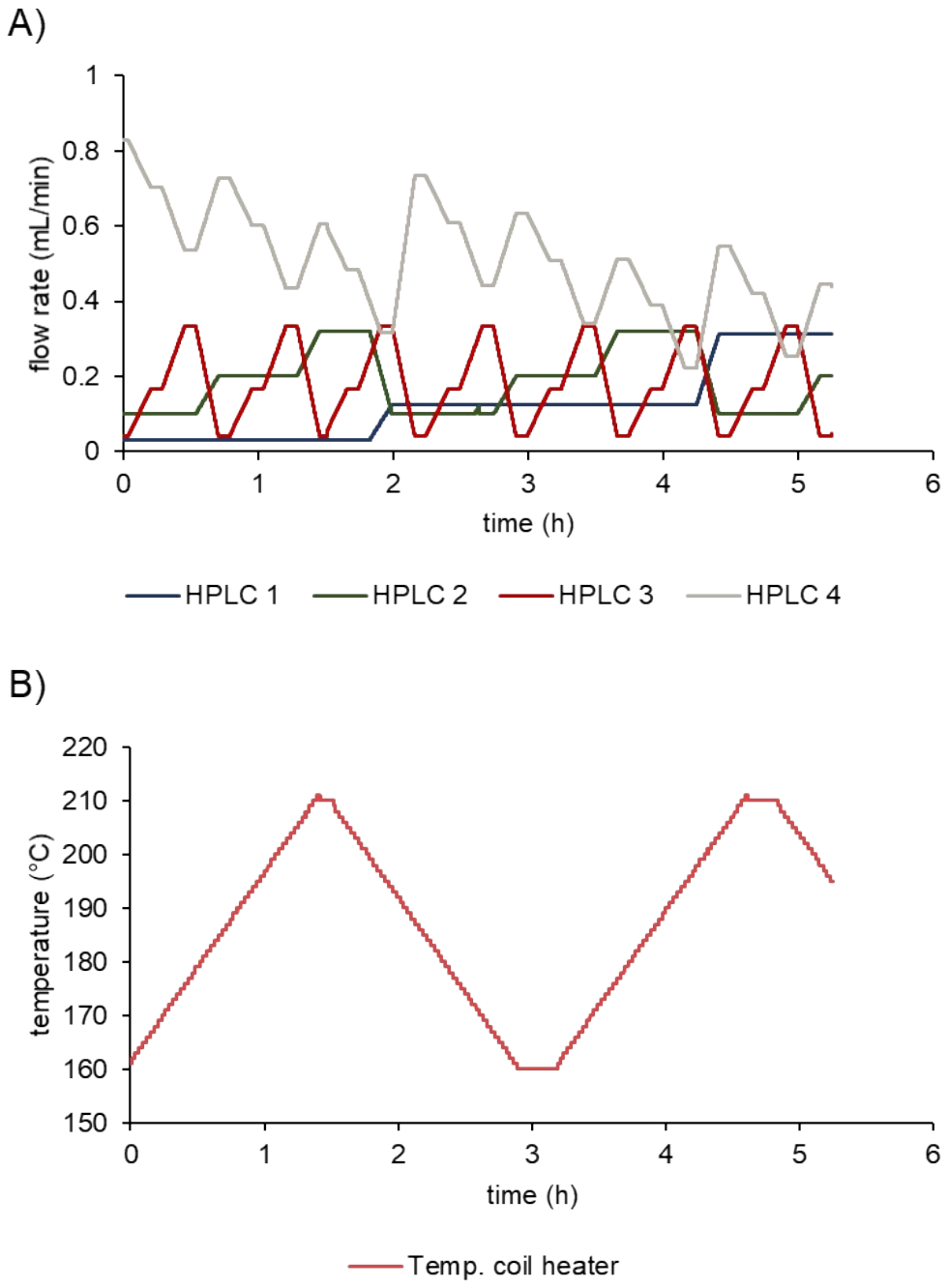



**Figure S30.** Flow rate **(A)** and temperature profile **(B)** for the multidimensional dynamic experiment.

**Table S7.** Offline UHPLC results for the multidimensional dynamic experiment.

| Fraction | Timestamp (h) | [2ClBA] (mol/L) | [3N-2ClBA] (mol/L) | [5N-2ClBA] (mol/L) | [3-NSA] (mol/L) | [5-NSA] (mol/L) |
|---|---|---|---|---|---|---|
| 1 | 0.01 | 0.039 | 0.008 | 0.006 | 0.022 | 0.035 |
| 2 | 0.06 | 0.037 | 0.008 | 0.005 | 0.022 | 0.045 |
| 3 | 0.11 | 0.037 | 0.008 | 0.005 | 0.022 | 0.062 |
| 4 | 0.16 | 0.034 | 0.007 | 0.005 | 0.022 | 0.078 |
| 5 | 0.21 | 0.031 | 0.007 | 0.004 | 0.023 | 0.092 |
| 6 | 0.26 | 0.029 | 0.006 | 0.004 | 0.023 | 0.112 |
| 7 | 0.31 | 0.030 | 0.006 | 0.004 | 0.023 | 0.097 |
| 8 | 0.36 | 0.029 | 0.008 | 0.005 | 0.022 | 0.136 |
| 9 | 0.41 | 0.028 | 0.009 | 0.007 | 0.021 | 0.158 |
| 10 | 0.46 | 0.027 | 0.010 | 0.007 | 0.020 | 0.177 |
| 11 | 0.51 | 0.026 | 0.009 | 0.006 | 0.021 | 0.180 |
| 12 | 0.56 | 0.026 | 0.008 | 0.004 | 0.025 | 0.165 |
| 13 | 0.61 | 0.026 | 0.005 | 0.002 | 0.033 | 0.130 |
| 14 | 0.66 | 0.025 | 0.002 | 0.002 | 0.042 | 0.084 |
| 15 | 0.71 | 0.024 | 0.001 | 0.002 | 0.047 | 0.045 |
| 16 | 0.76 | 0.024 | 0.001 | 0.002 | 0.049 | 0.036 |
| 17 | 0.81 | 0.025 | 0.002 | 0.002 | 0.050 | 0.046 |
| 18 | 0.86 | 0.025 | 0.001 | 0.003 | 0.051 | 0.065 |
| 19 | 0.91 | 0.025 | 0.002 | 0.003 | 0.051 | 0.081 |
| 20 | 0.96 | 0.025 | 0.001 | 0.002 | 0.051 | 0.093 |
| 21 | 1.01 | 0.025 | 0.002 | 0.003 | 0.051 | 0.102 |
| 22 | 1.06 | 0.025 | 0.002 | 0.003 | 0.050 | 0.117 |
| 23 | 1.11 | 0.026 | 0.004 | 0.004 | 0.049 | 0.141 |
| 24 | 1.16 | 0.026 | 0.007 | 0.005 | 0.045 | 0.165 |
| 25 | 1.21 | 0.025 | 0.010 | 0.006 | 0.041 | 0.180 |
| 26 | 1.26 | 0.026 | 0.010 | 0.006 | 0.043 | 0.184 |
| 27 | 1.31 | 0.026 | 0.007 | 0.005 | 0.049 | 0.161 |
| 28 | 1.36 | 0.025 | 0.004 | 0.004 | 0.059 | 0.129 |
| 29 | 1.41 | 0.025 | 0.002 | 0.004 | 0.069 | 0.085 |
| 30 | 1.46 | 0.024 | 0.002 | 0.004 | 0.076 | 0.047 |
| 31 | 1.51 | 0.024 | 0.002 | 0.004 | 0.079 | 0.038 |
| 32 | 1.52 | 0.029 | 0.002 | 0.004 | 0.070 | 0.047 |
| 33 | 1.57 | 0.027 | 0.003 | 0.004 | 0.074 | 0.062 |
| 34 | 1.62 | 0.028 | 0.006 | 0.004 | 0.072 | 0.077 |
| 35 | 1.67 | 0.027 | 0.008 | 0.004 | 0.070 | 0.090 |
| 36 | 1.72 | 0.026 | 0.011 | 0.005 | 0.071 | 0.097 |
| 37 | 1.77 | 0.026 | 0.014 | 0.005 | 0.068 | 0.102 |
| 38 | 1.82 | 0.026 | 0.019 | 0.008 | 0.061 | 0.115 |
| 39 | 1.87 | 0.032 | 0.024 | 0.014 | 0.049 | 0.126 |
| 40 | 1.92 | 0.048 | 0.027 | 0.023 | 0.039 | 0.146 |
| 41 | 1.97 | 0.071 | 0.025 | 0.034 | 0.028 | 0.161 |
| 42 | 2.02 | 0.099 | 0.018 | 0.033 | 0.020 | 0.164 |
| 43 | 2.07 | 0.107 | 0.012 | 0.013 | 0.021 | 0.133 |
| 44 | 2.12 | 0.104 | 0.006 | 0.005 | 0.022 | 0.077 |
| 45 | 2.17 | 0.103 | 0.005 | 0.004 | 0.023 | 0.042 |
| 46 | 2.22 | 0.101 | 0.006 | 0.004 | 0.022 | 0.037 |
| 47 | 2.27 | 0.101 | 0.007 | 0.005 | 0.021 | 0.050 |
| 48 | 2.32 | 0.101 | 0.009 | 0.007 | 0.019 | 0.063 |
| 49 | 2.37 | 0.100 | 0.011 | 0.012 | 0.017 | 0.074 |
| 50 | 2.42 | 0.101 | 0.013 | 0.015 | 0.016 | 0.081 |
| 51 | 2.47 | 0.102 | 0.014 | 0.019 | 0.015 | 0.083 |
| 52 | 2.52 | 0.100 | 0.014 | 0.025 | 0.013 | 0.086 |
| 53 | 2.57 | 0.099 | 0.015 | 0.036 | 0.011 | 0.096 |
| 54 | 2.62 | 0.099 | 0.018 | 0.056 | 0.010 | 0.103 |
| 55 | 2.67 | 0.099 | 0.019 | 0.078 | 0.009 | 0.108 |
| 56 | 2.72 | 0.101 | 0.020 | 0.088 | 0.008 | 0.108 |

**Table S8.** Offline UHPLC results for the multi-dimensional dynamic experiment.

| Fraction | Timestamp (h) | [2ClBA] (mol/L) | [3N-2ClBA] (mol/L) | [5N-2ClBA] (mol/L) | [3-NSA] (mol/L) | [5-NSA] (mol/L) |
|---|---|---|---|---|---|---|
| 57 | 2.77 | 0.099 | 0.021 | 0.074 | 0.009 | 0.096 |
| 58 | 2.82 | 0.103 | 0.024 | 0.049 | 0.013 | 0.077 |
| 59 | 2.87 | 0.100 | 0.025 | 0.026 | 0.018 | 0.051 |
| 60 | 2.92 | 0.102 | 0.026 | 0.015 | 0.023 | 0.032 |
| 61 | 2.97 | 0.103 | 0.026 | 0.012 | 0.026 | 0.029 |
| 62 | 3.02 | 0.102 | 0.026 | 0.014 | 0.027 | 0.037 |
| 63 | 3.07 | 0.103 | 0.026 | 0.017 | 0.027 | 0.053 |
| 64 | 3.12 | 0.101 | 0.026 | 0.020 | 0.026 | 0.066 |
| 65 | 3.17 | 0.102 | 0.026 | 0.022 | 0.026 | 0.078 |
| 66 | 3.22 | 0.103 | 0.025 | 0.021 | 0.028 | 0.085 |
| 67 | 3.27 | 0.103 | 0.025 | 0.024 | 0.027 | 0.097 |
| 68 | 3.32 | 0.099 | 0.026 | 0.032 | 0.025 | 0.109 |
| 69 | 3.37 | 0.100 | 0.029 | 0.043 | 0.023 | 0.122 |
| 70 | 3.42 | 0.103 | 0.030 | 0.035 | 0.030 | 0.135 |
| 71 | 3.47 | 0.102 | 0.031 | 0.054 | 0.022 | 0.138 |
| 72 | 3.52 | 0.102 | 0.031 | 0.050 | 0.024 | 0.143 |
| 73 | 3.57 | 0.103 | 0.025 | 0.016 | 0.043 | 0.109 |
| 74 | 3.62 | 0.104 | 0.017 | 0.008 | 0.059 | 0.071 |
| 75 | 3.67 | 0.102 | 0.012 | 0.006 | 0.070 | 0.040 |
| 76 | 3.72 | 0.100 | 0.009 | 0.005 | 0.072 | 0.034 |
| 77 | 3.77 | 0.102 | 0.010 | 0.006 | 0.073 | 0.047 |
| 78 | 3.82 | 0.103 | 0.013 | 0.006 | 0.071 | 0.067 |
| 79 | 3.87 | 0.103 | 0.016 | 0.007 | 0.067 | 0.085 |
| 80 | 3.92 | 0.104 | 0.018 | 0.008 | 0.066 | 0.097 |
| 81 | 3.97 | 0.104 | 0.018 | 0.008 | 0.065 | 0.101 |
| 82 | 4.02 | 0.103 | 0.024 | 0.013 | 0.058 | 0.116 |
| 83 | 4.07 | 0.102 | 0.032 | 0.024 | 0.048 | 0.128 |
| 84 | 4.12 | 0.103 | 0.041 | 0.043 | 0.040 | 0.137 |
| 85 | 4.17 | 0.101 | 0.045 | 0.055 | 0.034 | 0.137 |
| 86 | 4.22 | 0.107 | 0.042 | 0.050 | 0.035 | 0.141 |
| 87 | 4.27 | 0.137 | 0.025 | 0.023 | 0.043 | 0.137 |
| 88 | 4.32 | 0.178 | 0.009 | 0.010 | 0.047 | 0.109 |
| 89 | 4.37 | 0.209 | 0.005 | 0.010 | 0.034 | 0.064 |
| 90 | 4.42 | 0.244 | 0.003 | 0.007 | 0.030 | 0.041 |
| 91 | 4.47 | 0.266 | 0.002 | 0.002 | 0.028 | 0.036 |
| 92 | 4.52 | 0.259 | 0.003 | 0.004 | 0.026 | 0.045 |
| 93 | 4.57 | 0.259 | 0.003 | 0.004 | 0.026 | 0.045 |
| 94 | 4.62 | 0.263 | 0.002 | 0.002 | 0.025 | 0.084 |
| 95 | 4.67 | 0.267 | 0.002 | 0.002 | 0.025 | 0.097 |
| 96 | 4.72 | 0.265 | 0.002 | 0.001 | 0.024 | 0.102 |
| 97 | 4.77 | 0.267 | 0.004 | 0.002 | 0.023 | 0.121 |
| 98 | 4.82 | 0.258 | 0.007 | 0.011 | 0.020 | 0.141 |
| 99 | 4.87 | 0.245 | 0.011 | 0.027 | 0.015 | 0.146 |
| 100 | 4.92 | 0.256 | 0.015 | 0.044 | 0.013 | 0.154 |
| 101 | 4.97 | 0.258 | 0.016 | 0.046 | 0.013 | 0.151 |
| 102 | 5.02 | 0.258 | 0.014 | 0.027 | 0.018 | 0.139 |
| 103 | 5.07 | 0.252 | 0.011 | 0.013 | 0.029 | 0.106 |
| 104 | 5.12 | 0.260 | 0.008 | 0.007 | 0.039 | 0.065 |
| 105 | 5.17 | 0.266 | 0.007 | 0.004 | 0.045 | 0.040 |
| 106 | 5.22 | 0.261 | 0.006 | 0.003 | 0.043 | 0.032 |
| 107 | 5.27 | 0.262 | 0.010 | 0.004 | 0.040 | 0.043 |

## 3.2 Neural Network Architecture

### 3.2.1 Final Model for Data Fusion Using Locally Connected 1D for NMR and UV/vis

Python code for the final model using a locally connected 1D layers for NMR data and UV/vis data can be found below and on GitHub. The number of trainable parameters for every layer can be found in Figure S31.

```
model_path_name = 'datafusion_locally_NMR_locally_UV_final.hdf5'
# import NMR
visible_NMR = Input(shape=(600,1))
# import UV/vis
visible_UV = Input(shape=(187,1)
#architecture for the ANN for NMR
conv1_NMR = LocallyConnected1D(filters=16, kernel_size=9, strides=9, activation
='elu')(visible_NMR) #convolutional layer for NMR (tune filters, kernel size, strides
and the activation function)
flat_NMR = Flatten()(conv1_NMR) #flatten of the NMR convolutional layer
hidden11_NMR = Dense(27, activation='elu')(flat_NMR) #dense layer 1 NMR connected to
the flatten layer of NMR (tune neurons and activation function)
hidden12_NMR = Dense(9, activation='elu')(hidden11_NMR) #dense layer 2 NMR connceted
to the dense layer 1 NMR (tune neurons and activation function)
output_NMR = Dense(3, activation='relu')(hidden12_NMR) #output NMR layer dense layer
connected to dense layer 2 NMR (3 neurons represent the 3 intermediates measured at
this point, tune the activation function)

#architecture for the ANN for UV/vis
conv1_UV = LocallyConnected1D(filters=10, kernel_size=5, strides=2, activation
='elu')(visible_UV) #convolutional layer for UV (tune filters, kernel size, strides
and the activation function)
flat_UV = Flatten()(conv1_UV) #flatten of the UV convolutional layer
hidden11_UV = Dense(64, activation='relu')(flat_UV) #dense layer 1 UV connected to
the flatten layer of UV (tune neurons and activation function)
output_UV = Dense(32, activation='relu')(hidden11_UV) #dense layer 2 UV (output of
UV) connected to dense layer 1 UV (tune neurons and activation function)

#architecture for the ANN to merge ANN for NMR and ANN for UV/vis
merge1 = concatenate([output_NMR,output_UV]) #combining the output of the UV and NMR
networks to a new input for a NN
hidden11 = Dense(99, activation='relu')(merge1) #dense layer 1 comb connected to the
input layer (tune neurons and activation function)
hidden12 = Dense(64, activation='relu')(hidden11) #dense layer 2 comb connected to
dense layer 1 comb (tune neurons and activation function)
hidden13 = Dense(16, activation='relu')(hidden12) #dense layer 3 comb connected to
dense layer 2 comb (tune neurons and activation function)
```

```
output = Dense(4, activation='relu')(hidden13) #output layer as dense layer connected the dense layer 3 comb (4 neurons represent the 4 intermediates measured at this point, tune the activation function)
model = Model(inputs=[visible_NMR , visible_UV], outputs=[output_NMR, output]) #assign inputs and outputs of the model

# summarize layers
print(model.summary())
# compile the ANN model
learning_rate = 0.001
optimizer = Adam(lr=learning_rate, beta_1=0.9, beta_2=0.999, epsilon=10e-8, decay=0, amsgrad=False)
model.compile(loss='mse', optimizer=optimizer, metrics=['accuracy'])
# define learning and checkpointer
learning = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=20, min_lr=0.001)
checkpointer = ModelCheckpoint(filepath = path4 + model_path_name, monitor='val_loss', verbose=1, save_best_only=True)

#%% training of the model with validation data
Epochs = 1000
hist = model.fit([training_X_NMR_norm, training_X_UV_norm],[training_Y_NMR_norm, training_Y_UV_norm], epochs=Epochs, batch_size=1000, verbose=1, validation_data = ([val_X_NMR_norm, val_X_UV_norm],[val_Y_NMR_norm, val_Y_UV_norm]), callbacks = [checkpointer]) #, learning])
```

| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| input_4 (InputLayer) | [(None, 600, 1)] | 0 | |
| locally_connected1d_2 (LocallyC | (None, 66, 16) | 10560 | input_4[0][0] |
| input_5 (InputLayer) | [(None, 187, 1)] | 0 | |
| flatten_3 (Flatten) | (None, 1056) | 0 | locally_connected1d_2[0][0] |
| locally_connected1d_3 (LocallyC | (None, 92, 10) | 5520 | input_5[0][0] |
| dense_12 (Dense) | (None, 27) | 28539 | flatten_3[0][0] |
| flatten_4 (Flatten) | (None, 920) | 0 | locally_connected1d_3[0][0] |
| dense_13 (Dense) | (None, 9) | 252 | dense_12[0][0] |
| dense_15 (Dense) | (None, 64) | 58944 | flatten_4[0][0] |
| dense_14 (Dense) | (None, 3) | 30 | dense_13[0][0] |
| dense_16 (Dense) | (None, 32) | 2080 | dense_15[0][0] |
| concatenate_1 (Concatenate) | (None, 35) | 0 | dense_14[0][0]<br>dense_16[0][0] |
| dense_17 (Dense) | (None, 99) | 3564 | concatenate_1[0][0] |
| dense_19 (Dense) | (None, 16) | 1600 | dense_17[0][0] |
| dense_20 (Dense) | (None, 4) | 68 | dense_19[0][0] |

Total params: 111,157
Trainable params: 111,157
Non-trainable params: 0

None

**Figure S31.** Model summary showing the trainable parameters for each layer.

### 3.2.2 Final Model for Data Fusion Using Conv1D Layer for NMR and Locally Connected 1D Layer for UV/vis

The python code for the final model using a conv1D layer for NMR data and a locally connected 1D layers UV/vis data can be found below and on GitHub. The amount of trainable parameters for every layer can be found in Figure S32.

```
model_path_name = 'datafusion_conv1D_NMR_locally_UV_final.hdf5'
# import NMR
visible_NMR = Input(shape=(600,1))
# import UV/vis
visible_UV = Input(shape=(187,1))

#architecture for the ANN for NMR
conv1_NMR = Conv1D(filters=16, kernel_size=9, strides=9, activation
='relu')(visible_NMR) #convolutional layer for NMR (tune filters, kernel size, strides
and the activation function)
```

```
flat_NMR = Flatten()(conv1_NMR) #flatten of the NMR convolutional layer
hidden11_NMR = Dense(27, activation='relu')(flat_NMR) #dense layer 1 NMR connected to the flatten layer of NMR (tune neurons and activation function)
hidden12_NMR = Dense(9, activation='relu')(hidden11_NMR) #dense layer 2 NMR connceted to the dense layer 1 NMR (tune neurons and activation function)
output_NMR = Dense(3, activation='relu')(hidden12_NMR) #output NMR layer dense layer connected to dense layer 2 NMR (3 neurons represent the 3 intermediates measured at this point, tune the activation function)

#architecture for the ANN for UV/vis
conv1_UV = LocallyConnected1D(filters=10, kernel_size=5, strides=2, activation ='elu')(visible_UV) #convolutional layer for UV (tune filters, kernel size, strides and the activation function)
flat_UV = Flatten()(conv1_UV) #flatten of the UV convolutional layer
hidden11_UV = Dense(64, activation='relu')(flat_UV) #dense layer 1 UV connected to the flatten layer of UV (tune neurons and activation function)
output_UV = Dense(32, activation='relu')(hidden11_UV) #dense layer 2 UV (output of UV) connected to dense layer 1 UV (tune neurons and activation function)

#architecture for the ANN to merge ANN for NMR and ANN for UV/vis
merge1 = concatenate([output_NMR,output_UV]) #combining the output of the UV and NMR networks to a new input for a NN
hidden11 = Dense(99, activation='relu')(merge1) #dense layer 1 comb connected to the input layer (tune neurons and activation function)
hidden12 = Dense(64, activation='relu')(hidden11) #dense layer 2 comb connected to dense layer 1 comb (tune neurons and activation function)
hidden13 = Dense(16, activation='relu')(hidden12) #dense layer 3 comb connected to dense layer 2 comb (tune neurons and activation function)
output = Dense(4, activation='relu')(hidden13) #output layer as dense layer connected the dense layer 3 comb (4 neurons represent the 4 intermediates measured at this point, tune the activation function)
model = Model(inputs=[visible_NMR , visible_UV], outputs=[output_NMR, output]) #assign inputs and outputs of the model

# summarize layers
print(model.summary())
# compile the ANN model
learning_rate = 0.001
optimizer = Adam(lr=learning_rate, beta_1=0.9, beta_2=0.999, epsilon=10e-8, decay=0, amsgrad=False)
model.compile(loss='mse', optimizer=optimizer, metrics=['accuracy'])
# define learning and checkpointer
learning = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=20, min_lr=0.001)
checkpointer = ModelCheckpoint(filepath = path4 + model_path_name, monitor='val_loss', verbose=1, save_best_only=True)
#%% training of the model with validation data
```

```
Epochs = 1000
hist = model.fit([training_X_NMR_norm, training_X_UV_norm],[training_Y_NMR_norm, training_Y_UV_norm], epochs=Epochs, batch_size=1000, verbose=1, validation_data = ([val_X_NMR_norm, val_X_UV_norm],[val_Y_NMR_norm, val_Y_UV_norm]), callbacks = [checkpointer, learning])
```

```
Layer (type)                 Output Shape              Param #     Connected to
==================================================================================================
input_2 (InputLayer)         [(None, 600, 1)]          0
__________________________________________________________________________________________________
conv1d (Conv1D)              (None, 66, 16)            160         input_2[0][0]
__________________________________________________________________________________________________
input_3 (InputLayer)         [(None, 187, 1)]          0
__________________________________________________________________________________________________
flatten_1 (Flatten)          (None, 1056)              0           conv1d[0][0]
__________________________________________________________________________________________________
locally_connected1d_1 (LocallyC (None, 92, 10)         5520        input_3[0][0]
__________________________________________________________________________________________________
dense_3 (Dense)              (None, 27)                28539       flatten_1[0][0]
__________________________________________________________________________________________________
flatten_2 (Flatten)          (None, 920)               0           locally_connected1d_1[0][0]
__________________________________________________________________________________________________
dense_4 (Dense)              (None, 9)                 252         dense_3[0][0]
__________________________________________________________________________________________________
dense_6 (Dense)              (None, 64)                58944       flatten_2[0][0]
__________________________________________________________________________________________________
dense_5 (Dense)              (None, 3)                 30          dense_4[0][0]
__________________________________________________________________________________________________
dense_7 (Dense)              (None, 32)                2080        dense_6[0][0]
__________________________________________________________________________________________________
concatenate (Concatenate)    (None, 35)                0           dense_5[0][0]
                                                                   dense_7[0][0]
__________________________________________________________________________________________________
dense_8 (Dense)              (None, 99)                3564        concatenate[0][0]
__________________________________________________________________________________________________
dense_10 (Dense)             (None, 16)                1600        dense_8[0][0]
__________________________________________________________________________________________________
dense_11 (Dense)             (None, 4)                 68          dense_10[0][0]
==================================================================================================
Total params: 100,757
Trainable params: 100,757
Non-trainable params: 0
__________________________________________________________________________________________________
None
```

**Figure S32.** Model summary showing the trainable parameters for each layer.

## 3.3 Prediction of Process Data

### 3.3.1 Stability Run (ANN Using Locally Connected 1D Layers for NMR and UV/vis Data)

The prediction of **2ClBA**, **3N-2ClBA**, **5N-2ClBA** after the nitration step during the stability run using a locally connected 1D layer for the ANN1 (NMR) is depicted in Figure S33.

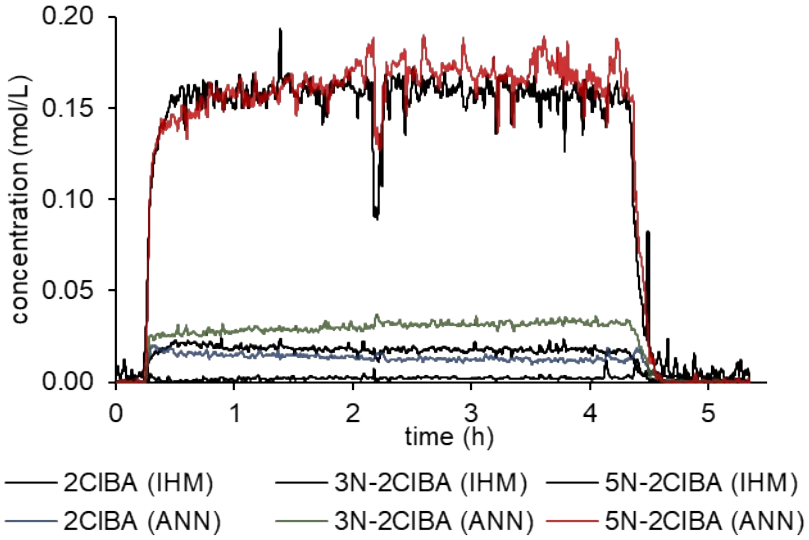



**Figure S33.** Predictions for the NMR data of the stability run using a locally connected 1D layer for the NMR data.

### 3.3.2 Stability Run (ANN Using Conv1D Layer for NMR Data and Locally Connected 1D Layer for UV/vis data)

The prediction for the process mixture (**2ClBA**, **3N-2ClBA**, **5N-2ClBA, 3-NSA, and 5-NSA**) after the hydrolysis step for the stability run is depicted in Figure S34. The prediction of **2ClBA**, **3N-2ClBA**, **5N-2ClBA** after the nitration step during the stability run using a locally conv1D layer for the ANN1 (NMR) is depicted in Figure S35.

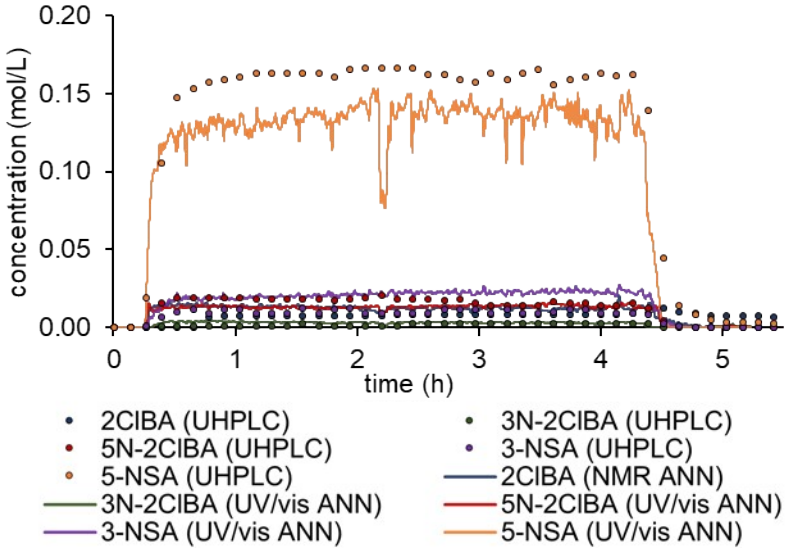



**Figure S34.** Predictions of the stability run using the ANN with a conv1D layer for the NMR data and a locally connected 1D layer for the UV/vis data.

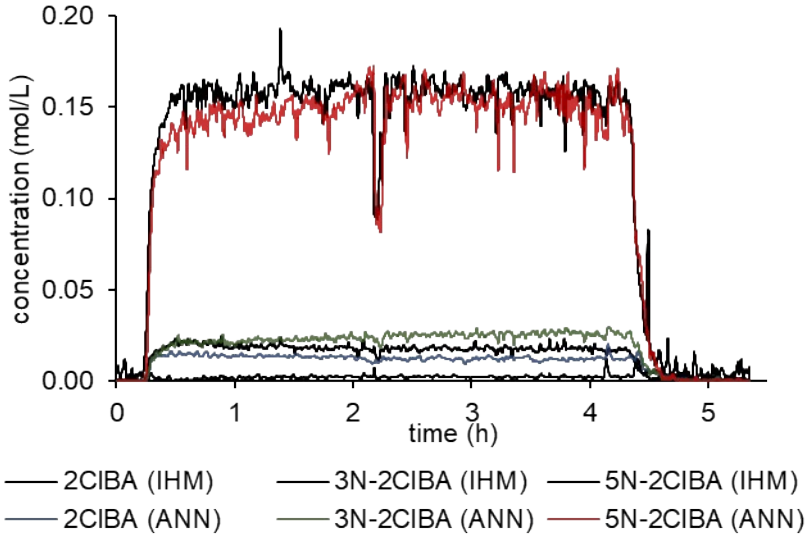



**Figure S35.** Predictions for the NMR data of the stability run using a conv1D layer for the NMR data.

### 3.3.3 Run with Dynamic Changes (ANN Using Locally Connected 1D Layers for NMR and UV/vis Data)

The prediction of **2ClBA**, **3N-2ClBA**, **5N-2ClBA** after the nitration step during the run with dynamic changes using a locally connected 1D layer for the ANN1 (NMR) is depicted in Figure S36.

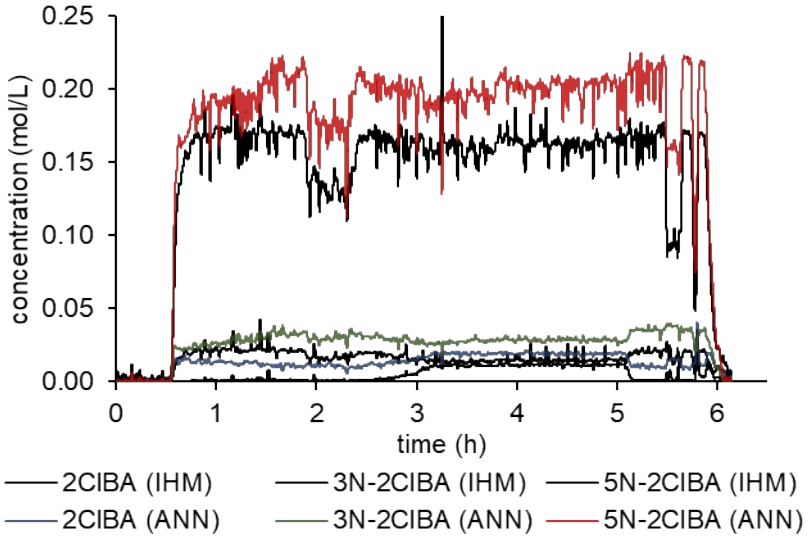



**Figure S36.** Predictions for the NMR data of the run with dynamic changes using a locally connected 1D layer for the NMR data.

### 3.3.4 Run with Dynamic Changes (ANN Using Conv1D Layer for NMR Data and Locally Connected 1D Layer for UV/vis Data)

The predictions for the process mixture (**2ClBA**, **3N-2ClBA**, **5N-2ClBA, 3-NSA, and 5-NSA**) after the hydrolysis step for the run with dynamic changes is depicted in Figure S37. The prediction of **2ClBA**, **3N-2ClBA**, **5N-2ClBA** after the nitration step during the run with dynamic changes using a locally conv1D layer for the ANN1 (NMR) is depicted in Figure S38.

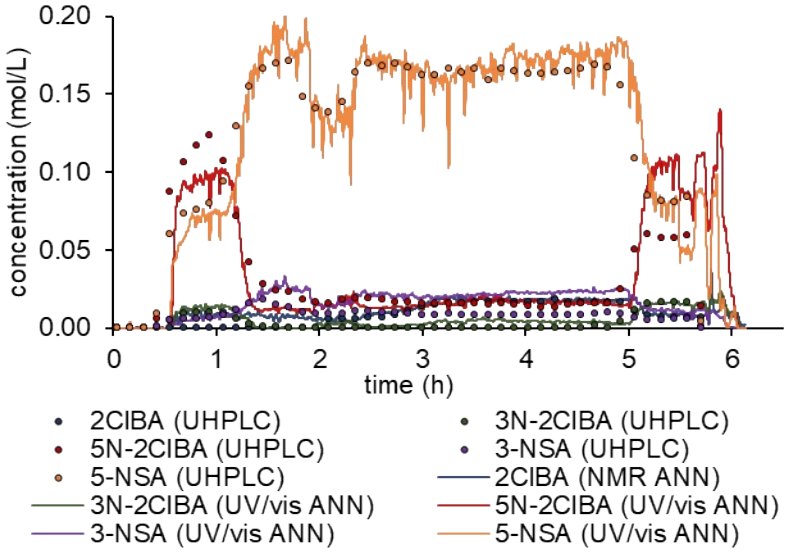



**Figure S37.** Predictions of the run with dynamic changes using the ANN with a conv1D layer for the NMR data and a locally connected 1D layer for the UV/vis data.

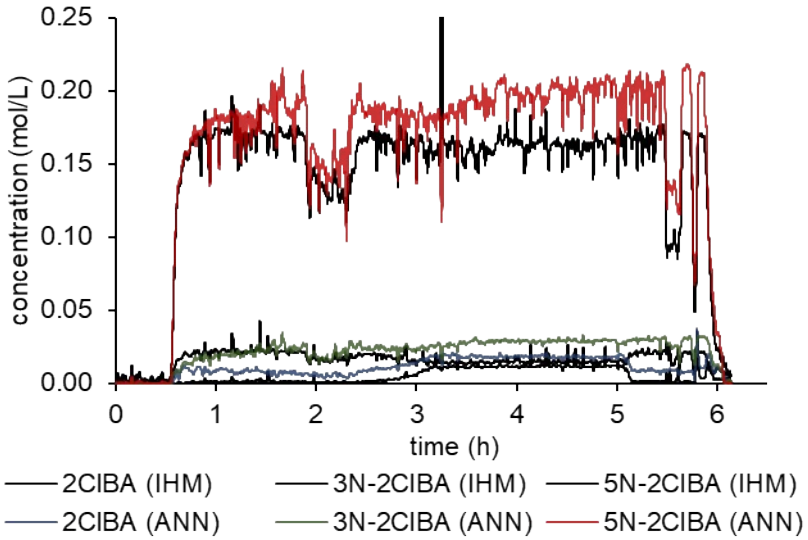



**Figure S38.** Predictions for the NMR data of the run with dynamic changes using a conv1D layer for the NMR data.

### 3.3.5 Multidimensional Dynamic Experiment (ANN Using Locally Connected 1D Layers for NMR and UV/vis Data)

The prediction of **2ClBA**, **3N-2ClBA**, **5N-2ClBA** after the nitration step during the multidimensional dynamic experiment using a locally connected 1D layer for the ANN1 (NMR) is depicted in Figure S39.

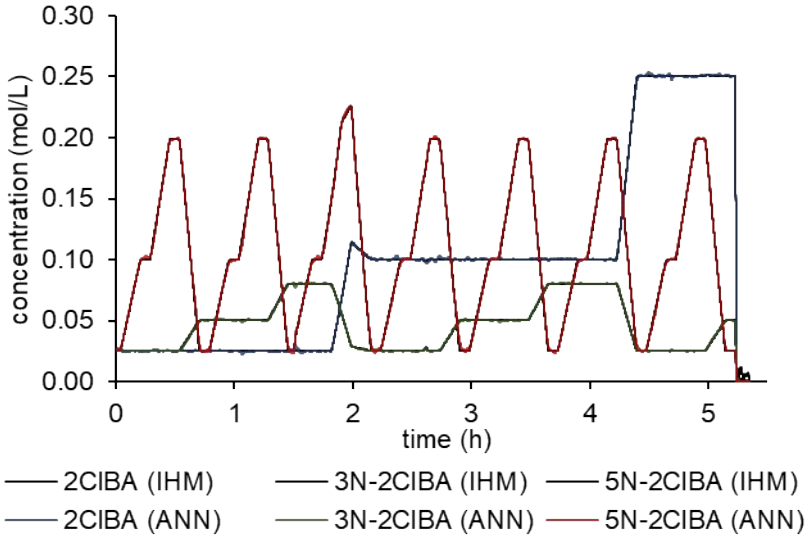



**Figure S39.** Predictions for the NMR data of multidimensional experiment using a locally connected 1D layer for the NMR data.

### 3.3.6 Multidimensional Dynamic Experiment (ANN Using Conv1D Layer for NMR Data and Locally Connected 1D Layer for UV/vis Data)

The predictions for the process mixture (**2ClBA**, **3N-2ClBA**, **5N-2ClBA, 3-NSA, and 5-NSA**) after the hydrolysis step multi-dimensional dynamic experiment is depicted in Figure S40. The prediction of **2ClBA**, **3N-2ClBA**, **5N-2ClBA** after the nitration step during the run with dynamic changes using a locally conv1D layer for the ANN1 (NMR) is depicted in Figure S41.

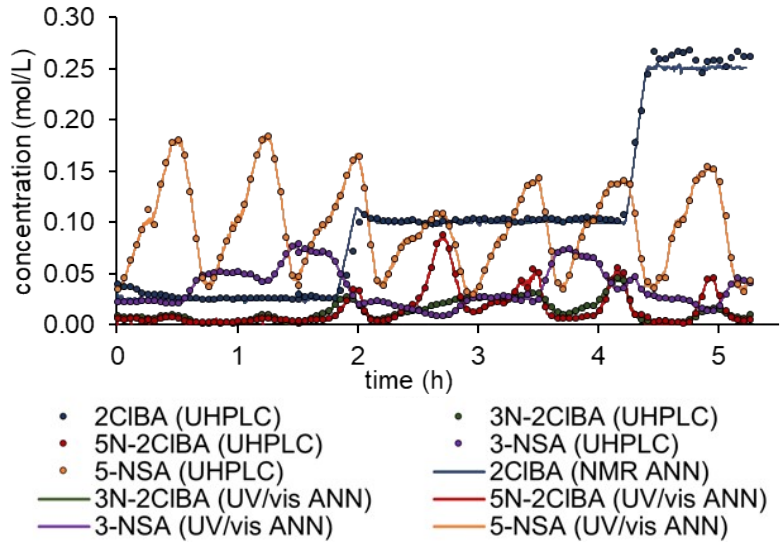



**Figure S40.** Predictions of the multi-dimensional dynamic experiment using the ANN with a conv1D layer for the NMR data and a locally connected 1D layer for the UV/vis data.

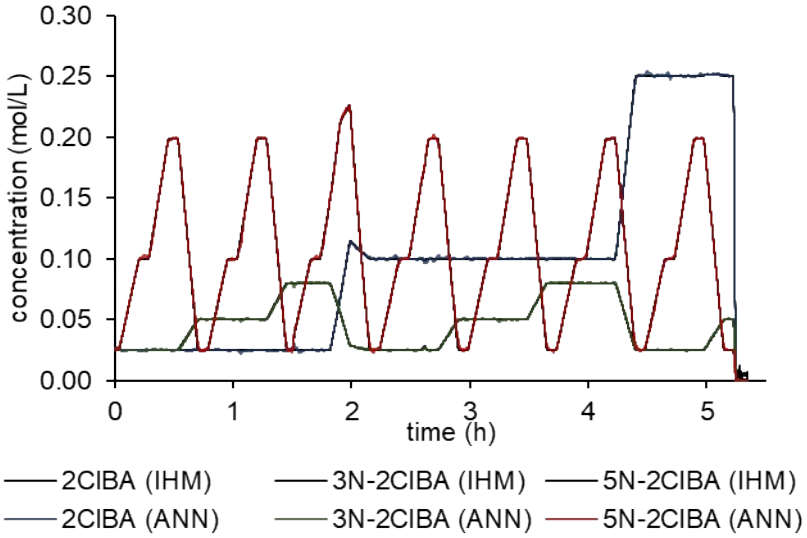



**Figure S41.** Predictions for the NMR data of multidimensional experiment using a conv1D layer for the NMR data.

## 4 References

S1 P. Sagmeister, R. Lebl, I. Castillo, J. Rehrl, J. Kruisz, M. Sipek, M. Horn, S. Sacher, D. Cantillo, J. D. Williams and C. O. Kappe, *Angew. Chemie Int. Ed.*, 2021, **60**, 8139–8148.

S2 https://github.com/SagmeisterPeter/ANN_Data_Fusion