Electronic Supplementary Material (ESI) for Reaction Chemistry & Engineering.

Unlocking the potential of late-stage functionalisation: An accurate and fully automated method for the rapid characterisation of multiple regioisomeric products.

Jeffery Richardson*[a], Gary Sharman*[a], Francisco Martínez-Olid[b], Santiago Cañellas[c] and Jose Enrique Gomez[c].

[a] Discovery Research and Technologies, Eli Lilly and Company, Erl Wood Manor, Sunninghill Road, Windlesham, Surrey, GU20 6PH, United Kingdom.

[b] Discovery Research and Technologies, Eli Lilly and Company, Centro de Investigación Lilly, Avenida de la Industria 30, 28108 Alcobendas-Madrid, Spain.

[c] Institute of Chemical Research of Catalonia (ICIQ), The Barcelona Institute of Science and Technology, Avda. Països Catalans 16, E-43007 Tarragona, Spain.

richardson_jeffery@lilly.com, sharman_gary_gs@lilly.com

Supporting information

General methods

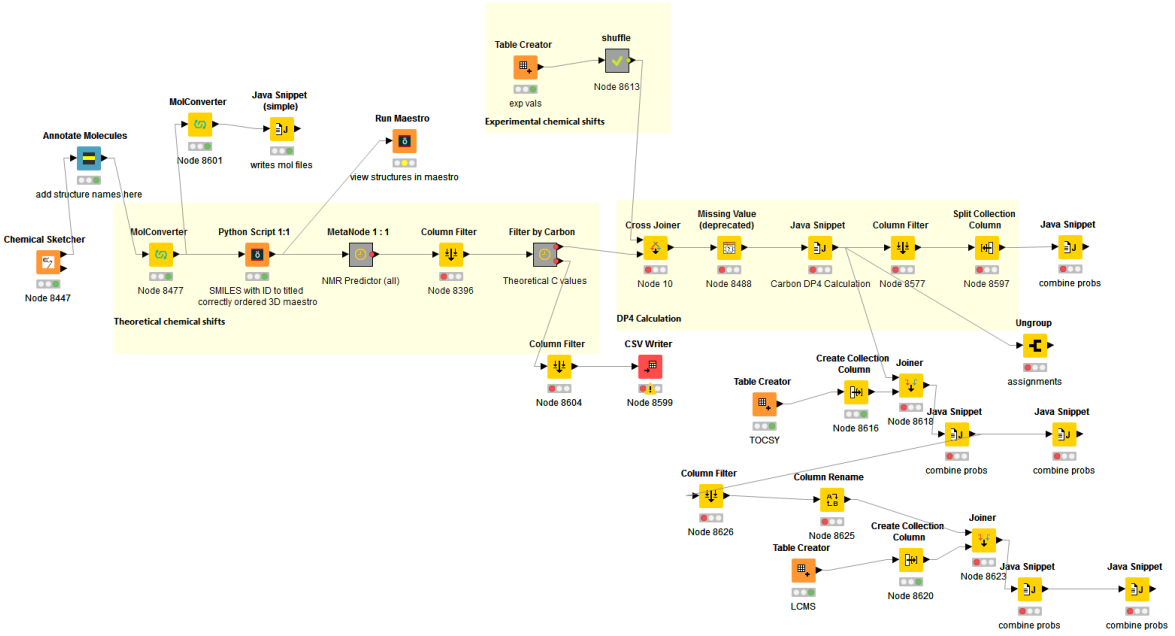
All reagents were purchased from commercial suppliers and used as received. Solvents were purchased from Aldrich, anhydrous, SureSeal quality, and used with no further purification.

Flash column chromatography was carried out using silica gel columns with a Teledyne ISCO CombiFlash Companion system. ^{1}H and ^{13}C NMR spectra were recorded on a Bruker AV-HD 400 or Bruker Avance III 500 spectrometer. Signal positions were recorded in δ ppm with the abbreviations s, d, t, q, dd, dt and m denoting singlet, doublet, triplet, quartet, doublet of doublets, doublet of triplets and multiplet respectively. All ^{1}H NMR chemical shifts were referenced to $SiMe_4$ as an internal standard (0.00 ppm). All ^{13}C NMR chemical shifts in $CDCl_3$ were referenced to the residual solvent peak at 77.00 ppm. All coupling constants, J, are quoted in Hz. HSQC spectra were acquired using a standard Bruker pulse sequence incorporating multiplicity editing and echo-antiecho coherence selection. Typically 2048 f2 data points were acquired in 128 f1 increments, employing 50% NUS sampling. TOCSY spectra were acquired using a standard Bruker MLEV sequence. 2k f2 points and 128 f2 increments were typically acquired, again with 50% NUS sampling. Note that the 2D NMR parameters are essentially standard and non-critical to this approach, such that any typical open access HSQC and TOCSY would suffice. Infra-red spectra were recorded on a Nexus FT-IR spectrometer with Nicolet OMNI sampler using a neat sample. All LCMS analyses were performed using an Agilent 1200 Infinity Series Liquid Chromatography (LC) system, consisting of a 1260 HiP degasser (G4225A), 1260 Binary Pump (G1312B), 1290 auto-sampler (G4226A), 1290 thermo-stated column compartment (G1316C)

and a 1260 Diode Array Detector (G4212B) coupled to an Agilent 6150 single quadrupole mass spectrometry (MS) detector. The injection volume was set to 1 µL. The UV (DAD) acquisition was performed at 40 Hz, with a scan range of 190-400 nm (in 5 nm steps). A 1:1 flow split was used before the MS detector. The MS was operated with an electro-spray ionization source (ESI) in both positive & negative ion mode. The nebulizer pressure was set to 50 psi, the drying gas temperature and flow to 350 °C and 12.0 L/min respectively. The capillary voltages used were 4000 V in positive mode and 3500 V in negative mode. The MS acquisition range was set to 100-800 m/z with a step size of 0.2 m/z in both polarity modes. Fragmentor voltage was set to 70 (ESI+) or 120 (ESI-), Gain to 0.40 (ESI+) or 1.00 (ESI-) and the ion count threshold to 4000 (ESI+) or 1000 (ESI-). The overall MS scan cycle time was 0.15 s/cycle. Data acquisition was performed with Agilent Chemstation software. Analyses were carried out on a Waters XBridge C18 column of 50 mm length, 2.1 mm internal diameter and 3.5 µm particle size, eluting from 95:5 to 5:95 of pH 9 adjusted 10 mM NH_4HCO_3 (aq) in MeCN over 1.5 min and holding for 0.5 min.

KNIME workflows and DP4 calculations

All KNIME workflows were written and executed in version 3.3 of the software typically using a regular PC with modest processor (intel i3) and 4Gb or RAM. The Schrodinger nodes were part of this installation, these require a separate Schrodinger license and are not part of the KNIME open source package. It should be noted that the procedure does not rely on a particular software vendor or version; Other ab initio software packages could have been used, but the in-house availability and simplified KNIME interface for Schrodinger suited our automation strategy. Details of the calculations executed through these nodes are provided later in this document. Source code for DP4 was downloaded from the web site of Dr Jonathan Goodman's group. Additional classes required to implement the HSQC assignment and combination of the resulting probabilities were created and compiled using Netbeans version 8. The resulting compiled .jar file was then executed from a java snippet within the KNIME workflow.



Alternative computational structure elucidation approaches considered.

Automated structural elucidation (ASV) was considered as an alternative to the quantum mechanical shift prediction. However, this approach was quickly discarded as initial studies showed the score from ASV was essentially the same for many region-isomeric structures. This is perhaps not entirely surprising as ASV generally asks the question is the structure consistent with the experimental data. For many regioisomers, this is the case, and little differentiation is observed. Further the score from ASV is not directly interpretable as a probability, so further work would be needed to ask the question If an observed difference in score was significant or not.

We also considered using full assignments with DP4, but quickly rejected this approach as generating a full assignment almost inevitably presupposed the answer. That is, one would have needed to consider the structural possibilities and effectively decided which one was correct in order to be certain of the assignments.

Compounds analysed in Figures 5 and 13

Compounds analysed using this method in figures 5 and 13 were selected from the Lilly archive. They are all “drug-like molecules” where sufficient inventory was available for a sample to be taken without impacting the material retained in inventory. The molecules were selected to provide as much diversity as possible. Below is a summary of some key descriptors for this set.

MW: 273.286 to 552.649
Heavy atoms: 19-39
Number of rings: 1-6
Number of rotatable bonds 4-6
Aromatic rings 1-5

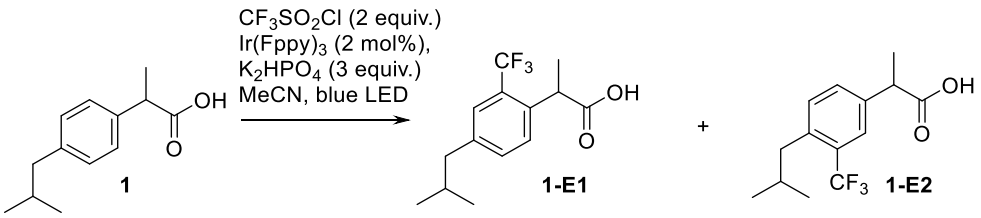
Additionally, this set of compounds contained the following easily identified functional groups:

tertiary amide
secondary amide
primary amide
basic amines
aryl chloride
aryl fluoride
aryl CF_3
aminopyrazole
sulfide
pyridine
sulfonamide
benzonitrile
tert aniline
benzothiophenes
phenol
thiazole
1,2,4-triazole
morpholine
urea
primary alcohol
secondary alcohol
tertiary alcohol
ether
pyrimidines
azetidines
thiophene
carboxylic acid
ketone
cyclopropane
sulfone

Trifluoromethylation of ibuprofen.

Prepared according to the procedure demonstrated by Macmillan and Nagib.(*1*)

2 isomers were obtained (**1-E1** and **1-E2**) experimentally, these had two possible structures (**1-S1** and **1-S2**)

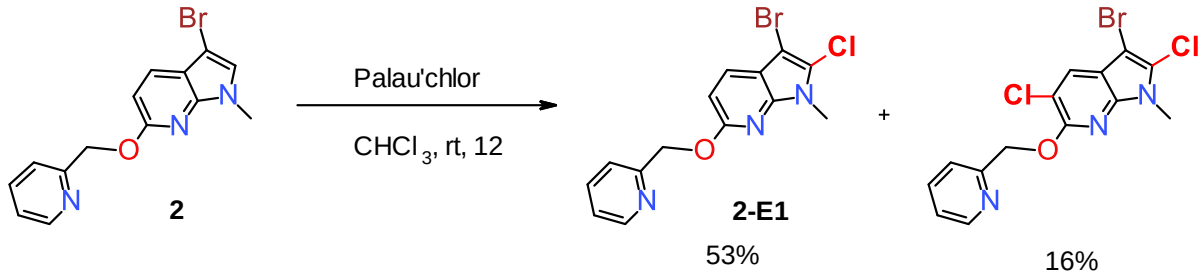


A 24 mL vial was equipped with a magnetic stir bar, $Ir(Fppy)_3$ (16 mg, 0.02 mmol, 2 mol%), and potassium hydrogen phosphate (485 mg, 2.78 mmol, 3 equiv.) in a glovebox under nitrogen. The vial was sealed before acetonitrile (8 mL) and the 2-(4-isobutylphenyl)propanoic acid (190 mg, 0.92 mmol) were added by syringe. The resulting solution was further degassed by nitrogen bubbling for 5 minutes and trifluoromethanesulfonyl chloride (312 mg, 1.85 mmol, 3 equiv.) was then added by syringe. The vial was sealed with parafilm and placed in the photoreactor (*2*) and stirred at 1000 rpm under irradiation with blue LED light for 24 h. LCMS indicated that desired products were formed and the reaction mixture was filtered, concentrated, dissolved in MeOH (to a total volume of 9.8 mL), filtered and purified by prep-HPLC (Phenomenex Gemini-NX 10 Micron 50*150mm C-18) (CH_3CN & Water both with 0.1 % formic acid, 15 % to 100% CH_3CN over 11 minutes at 120ml/min) (1 injection). (220 nm). This afforded a mixture of 2 regioisomeric products that were then separated by preparative SFC (method GS-NO2 column, 15-25 % MeOH (+40 mM NH_3)).

2-[4-isobutyl-3-(trifluoromethyl)phenyl]propanoic acid (1-E1). (6 mg, 2%) Colourless oil. **^{1}H NMR** (500 MHz, $CDCl_3$): δ = 7.41 (m, 2H), 7.3 (m, 1H), 4.16 (q, *J* = 7.1 Hz, 1H), 2.49 (d, *J* = 6.8 Hz, 2H), 1.87 (sept, *J* = 6.7 Hz, 1H), 1.51 (d, *J* = 7.1 Hz, 3H), 0.91 (d, *J* = 6.6 Hz, 6H). **^{13}C NMR** (125 MHz, $CDCl_3$): δ = 179.34, 141.22, 136.13, 133, 128.745, 128.095 (q, *J* = 29.1 Hz), 126.555 (q, *J* = 11.4 Hz), 124.53 (q, *J* = 273.0 Hz), 44.95, 40.5, 30.21, 22.43 (2C), 19.38. LRMS (ESI) m/z: $[M+NH_3]^+$292, $[M+Na]^+$297.

2-[4-isobutyl-3-(trifluoromethyl)phenyl]propanoic acid (1-E2). (31 mg, 12%) Colourless oil. **^{1}H NMR** (500 MHz, $CDCl_3$): δ = 11.09 (br s, 1H), 7.55 (d, *J* = 1.97 Hz, 1H), 7.41 (dd, *J* = 2.02, 7.97 Hz, 1H), 7.27 (d, *J* = 8.03 Hz, 1H), 3.76 (q, *J* = 7.20 Hz, 1H), 2.63 (dd, *J* = 1.44, 7.25 Hz, 2H), 1.94 (m, 1H), 1.53 (d, *J* = 7.19 Hz, 3H), 0.92 (d, *J* = 6.62 Hz, 6H). **^{13}C NMR** (125 MHz, $CDCl_3$): δ = 180.44, 139.805 (q, *J* = 1.4 Hz), 137.47, 132.12, 130.62, 129.24 (q, *J* = 29.5 Hz), 125.455, 124.61 (q, *J* = 274.5 Hz), 44.96, 41.2 (q, *J* = 1.4 Hz), 29.89, 22.59 (2C), 18.15. LRMS (ESI) m/z: $[M+H]^+$ 275.0, $[M+NH_3]^+$292, $[M+Na]^+$297.

3-bromo-2-chloro-1-methyl-6-(2-pyridylmethoxy)pyrrolo[2,3-b]pyridine (2-E1).



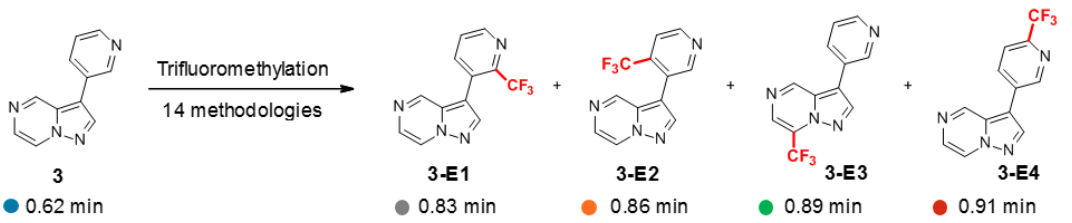
Prepared according to the procedure described by Baran et al. (3)

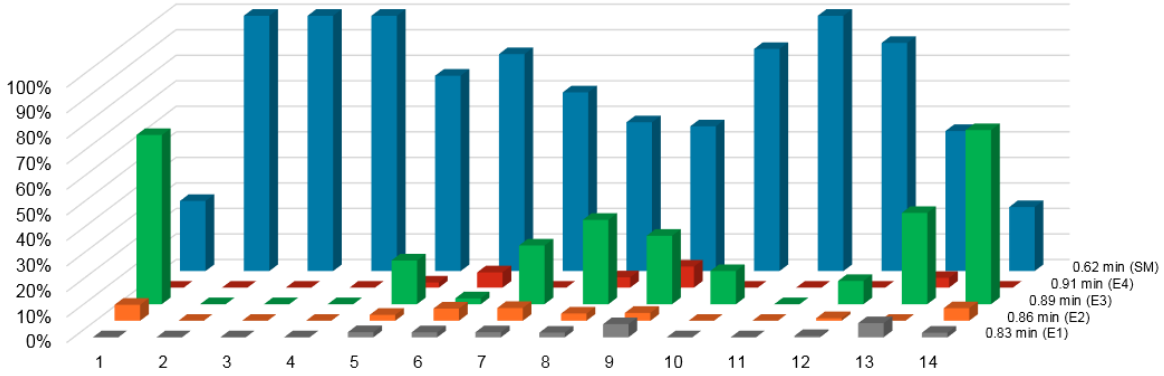
To a stirred solution of 3-bromo-1-methyl-6-(2-pyridylmethoxy)pyrrolo[2,3-b]pyridine (2) (102 mg, 0.32 mmol) in chloroform (2 mL) at r.t. was added Palau'chlor (85 mg, 0.38 mmol, 95% purity) and the mixture stirred at ambient temperature overnight. LCMS showed complete consumption of starting material and the reaction mixture was concentrated under reduced pressure. The residue was dissolved in MeOH (to a total volume of 9.8 mL), filtered and purified by prep-HPLC (Phenomenex Gemini-NX 10 Micron 50*150mm C-18) (acetonitrile & water adjusted to approx. pH 9 with conc. ammonium hydroxide solution [0.5 mL of conc. ammonium hydroxide per 2.5 L of water], 15 % to 100% acetonitrile over 11 minutes at 120 mL/min) (1 injection) to afford the title compound as a pink-white solid, (60 mg, 53% yield). **^{1}H NMR** (500 MHz, $CDCl_3$): δ = 8.6 (m, 1H), 7.69 (m, 1H), 7.65 (d, J = 8.41 Hz, 1H), 7.48 (d, J = 7.85 Hz, 1H), 7.2 (ddd, J = 7.56, 4.79, 1.14 Hz, 1H), 6.73 (dd, J = 8.45, 1.69 Hz, 1H), 5.57 (s, 2H), 3.7 (s, 3H). **^{13}C NMR** (125 MHz, $CDCl_3$): δ = 160.51, 157.83, 149.33, 143.32, 136.72, 129.94, 129.1, 122.49, 121.43, 113.87, 105.56, 86.99, 68.49, 29.18. LRMS (ESI) m/z: $[M+H]^+$ 352.0/353.0/354.0/355.0/356.0.

Trifluoromethylation regioselectivity study.

Compound **3** was subjected to a series of literature trifluoromethylation conditions to demonstrate the potential of this method for quickly mapping the reactivity profiles of various conditions in LSF applications. Compound **3** was chosen because of the high likelihood of obtaining multiple regioisomers. Table S1 summarises the conditions employed in this study.

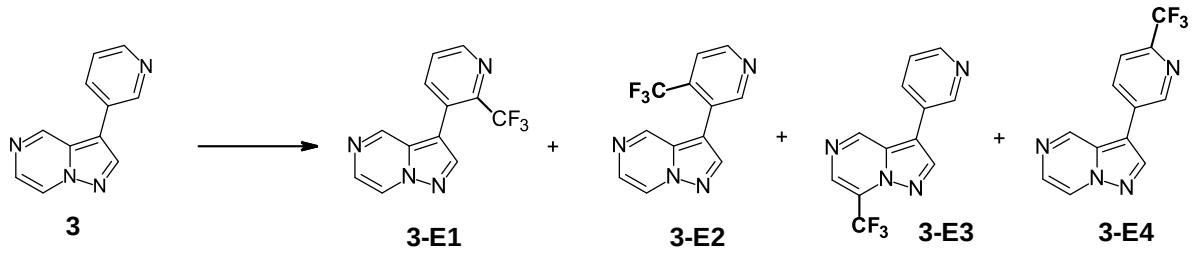
Table S1





Number	Conditions	Ref
1	CF_3SO_2Cl (8 equiv.), $Ir(Fppy)_3$ (5 mol%), K_2HPO_4 (3 equiv.), MeCN, blue LED	(*1*)
2	$TMSCF_3$ (5 equiv.), AgOTf (4 equiv.), KF (4 equiv.), DCE, 85 °C	(*4*)
3	$NaSO_2CF_3$ (4 equiv.), acetone, blue LED, rt	(*5*)
4	TFA (4 equiv), Ag_2CO_3 (0.4 equiv.), $K_2S_2O_8$ (2 equiv.), Na_2CO_3 (1.5 equiv.), H_2SO_4 (0.5 equiv.), MeCN	(*6*)
5	$TMSCF_3$ (2 equiv.), $PhI(OAc)_2$ (2 equiv.), AgF (0.25 equiv.), DMSO, rt	(*7*)
6	$TMSCF_3$ (2.4 equiv.), $PhI(OAc)_2$, (1.2 equiv.), 1,4-benzoquinone (0.2 equiv.), K_3PO_4 (2.4 equiv.), MeCN, 4Å MS, 70 °C	(*8*)
7	TFA (0.06 equiv.), CF_3SO_2Na (4 equiv.), Eosin Y (0.05 equiv.), visible light	(*9*)
8	PIFA (2 equiv.), CF_3SO_2Na (2 equiv.), MeCN, rt	(*10*)
9	Togni reagent (2 equiv.), $(TMS)_3SiCl$ (1 equiv.), MeCN, 70 °C	(*11*)
10	Togni reagent (2 equiv.), $MeReO_3$ (0.1 equiv.), $CHCl_3$, 70 °C	(*12*)
11	$PhCOCH(Me)SO_2CF_3$ (3 equiv.), MeCN, blue LED	(*13*)
12	CF_3SO_2Na (3 equiv.), $Na_2S_2O_8$ (1.5 equiv.), MeCN/H_2O (1:1), H_3PO_4, rt	(*14*)
13	Tf_2O (3 equiv.), $Ru(bpy)_3Cl_2$ (5 mol%), pyridine (3 equiv.), 1,2-DCE, blue LED, rt	(*15*)
14	CF_3SO_2Na (4 equiv.), tBuOOH (7 equiv.), $Cu(OTf)_2$ (0.1 equiv.), MeCN, rt	(*16*)

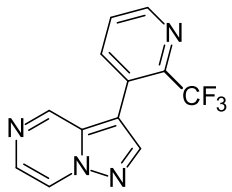
Synthesis and isolation of compounds 3-E1, 3-E2, 3-E3 and 3-E4.



Based on conditions described by Langlois *et al* (*16*). Compound **3** was obtained from the Lilly sample archive. Spectroscopic data can be found in the spectra and analytical data section.

A vial equipped with a magnetic stir bar was charged with $NaSO_2CF_3$ (133 mg, 0.818 mmol), 3-(3-pyridyl)pyrazolo[1,5-a]pyrazine (**3**) (40 mg, 0.204 mmol), $Cu(OTf)_2$ (7 mg, 0.019 mmol), CH_3CN (2 mL) and 7.22 M tBuOOH in water (180 mg, 1.4 mmol). The mixture was stirred at room temperature for 12 h. LCMS analysis of the reaction mixture showed a mixture of trifluoromethylated regioisomers. This was combined with crude reaction mixtures from experiments 1,5,7, 8,9, 12,13 & 14 (Table S1) and filtered through a silica plug, eluting with 10% MeOH/DCM). The eluted organics were combined and concentrated under reduced pressure. The residue obtained was dissolved MeOH (to a total volume of 9.8 mL), filtered and purified by prep-HPLC (Phenomenex Gemini-NX 10 Micron 50*150 mm C-18) (CH_3CN & Water adjusted to approx. pH 9 with conc. ammonium hydroxide solution [0.5 mL of conc. ammonium hydroxide per 2.5 L of water], 15 % to 100% CH_3CN over 11 minutes at 120 mL/min). Mixed fractions were then dissolved in MeOH (to a total volume of 9.8 mL), filtered and purified by prep-HPLC (Phenomenex Gemini-NX 10 Micron 50*150mm C-18) (CH_3CN & Water both with 0.1 % formic acid, 15 % to 100% CH_3CN over 11 minutes at 120 mL/min). 4 regioisomers were obtained and characterized as follows:

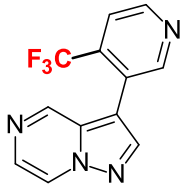
3-[2-(trifluoromethyl)-3-pyridyl]pyrazolo[1,5-a]pyrazine (**3-E1**, 2 mg)



3-E1

Obtained as a colourless oil. **^{1}H NMR** (500 MHz, $CDCl_3$) δ 8.92 (s, 1H), 8.80 (dd, J = 4.8, 1.6 Hz, 1H), 8.46 (dd, J = 4.7, 1.4 Hz, 1H), 8.12 (s, 1H), 7.98 (d, J = 4.8 Hz, 1H), 7.88 (ddd, J = 7.8, 1.6, 0.7 Hz, 1H), 7.63 (dd, J = 7.8, 4.7 Hz, 1H). **^{13}C NMR** (126 MHz, $CDCl_3$) δ 148.66, 143.69, 142.05, 141.23, 130.18, 126.37, 122.08, 109.33. [quaternary not observed]. **LRMS** (ESI) m/z: $[M+H]^+$ 265.2.

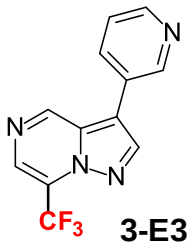
3-[4-(trifluoromethyl)-3-pyridyl]pyrazolo[1,5-a]pyrazine (**3-E2**, 10 mg)



3-E2

Obtained as a colourless solid. **^{1}H NMR** (500 MHz, $CDCl_3$) δ 8.96 (s, 1H), 8.87 (d, J = 5.1 Hz, 1H), 8.82 (s, 1H), 8.46 (dd, J = 4.7, 1.4 Hz, 1H), 8.13 (s, 1H), 7.99 (d, J = 4.9 Hz, 1H), 7.74 (d, J = 5.1 Hz, 1H). **^{13}C NMR** (126 MHz, $CDCl_3$) δ 153.29, 150.31, 143.77, 142.53, 130.26, 122.09, 120.28. [quaternary not observed] **LRMS** (ESI) m/z: $[M+H]^+$ 265.2.

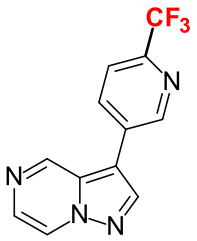
3-(3-pyridyl)-7-(trifluoromethyl)pyrazolo[1,5-a]pyrazine (**3-E3,** 5 mg)



3-E3

Obtained as a colourless oil. **^{1}H NMR** (500 MHz, $CDCl_3$) δ 9.41 (s, 1H), 8.93 (s, 1H), 8.68 (d, J = 5.3 Hz, 1H), 8.37 (s, 1H), 8.34 (s, 1H), 7.94 (ddd, J = 7.9, 2.3, 1.6 Hz, 1H), 7.51 – 7.44 (m, 1H). **^{13}C NMR** (126 MHz, $CDCl_3$) δ 149.47, 148.75, 147.62, 141.47, 135.14, 133.21, 129.41 (q, J = 5.3 Hz), 126.78, 124.31, 123.07 (q, J = 36.4 Hz), 120.39 (q, J = 273.2 Hz), 113.98. **LRMS** (ESI) m/z: $[M+H]^+$ 265.2.

3-[6-(trifluoromethyl)-3-pyridyl]pyrazolo[1,5-a]pyrazine (**3-E4**, 32 mg)

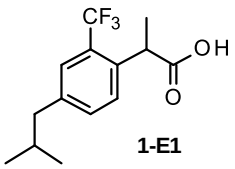


3-E4

Lot number: **60P-E18045-062-009-057**

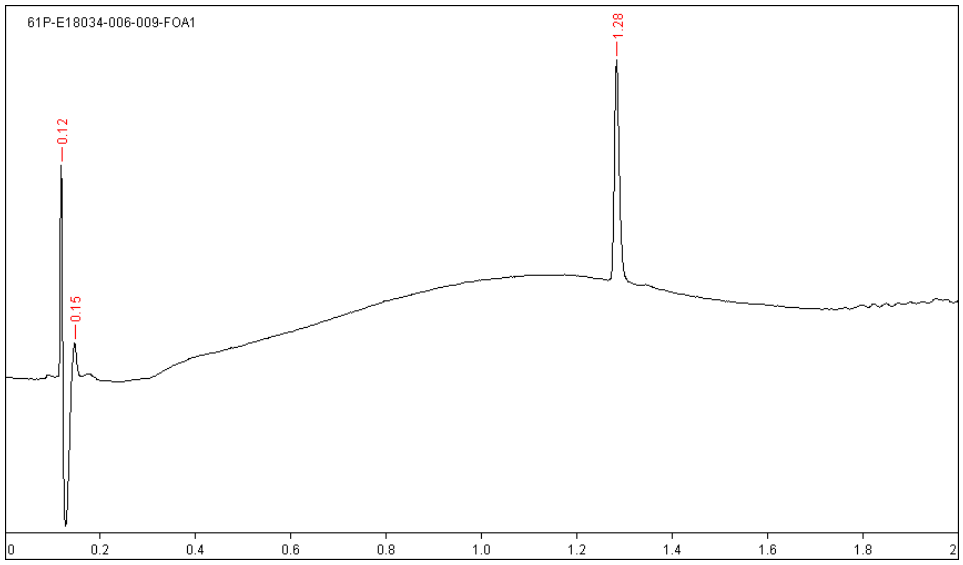
Obtained as a colourless solid. **^{1}H NMR** (500 MHz, $CDCl_3$) δ 9.33 (d, J = 1.4 Hz, 1H), 9.05 (d, J = 2.2 Hz, 1H), 8.48 (dd, J = 4.7, 1.5 Hz, 1H), 8.31 (s, 1H), 8.14 – 8.08 (m, 1H), 8.03 (d, J = 4.7 Hz, 1H), 7.83 (dd, J = 8.1, 0.9 Hz, 1H). **^{13}C NMR** (126 MHz, $CDCl_3$) δ 148.36, 146.84 (q, J = 35.3 Hz), 143.74, 140.88, 135.48, 132.87, 130.82, 130.65, 122.46, 121.68 (q), 121.11 (q, J = 2.7 Hz), 110.56. **LRMS** (ESI) m/z: $[M+H]^+$ 265.2.

Spectra and Analytical Data

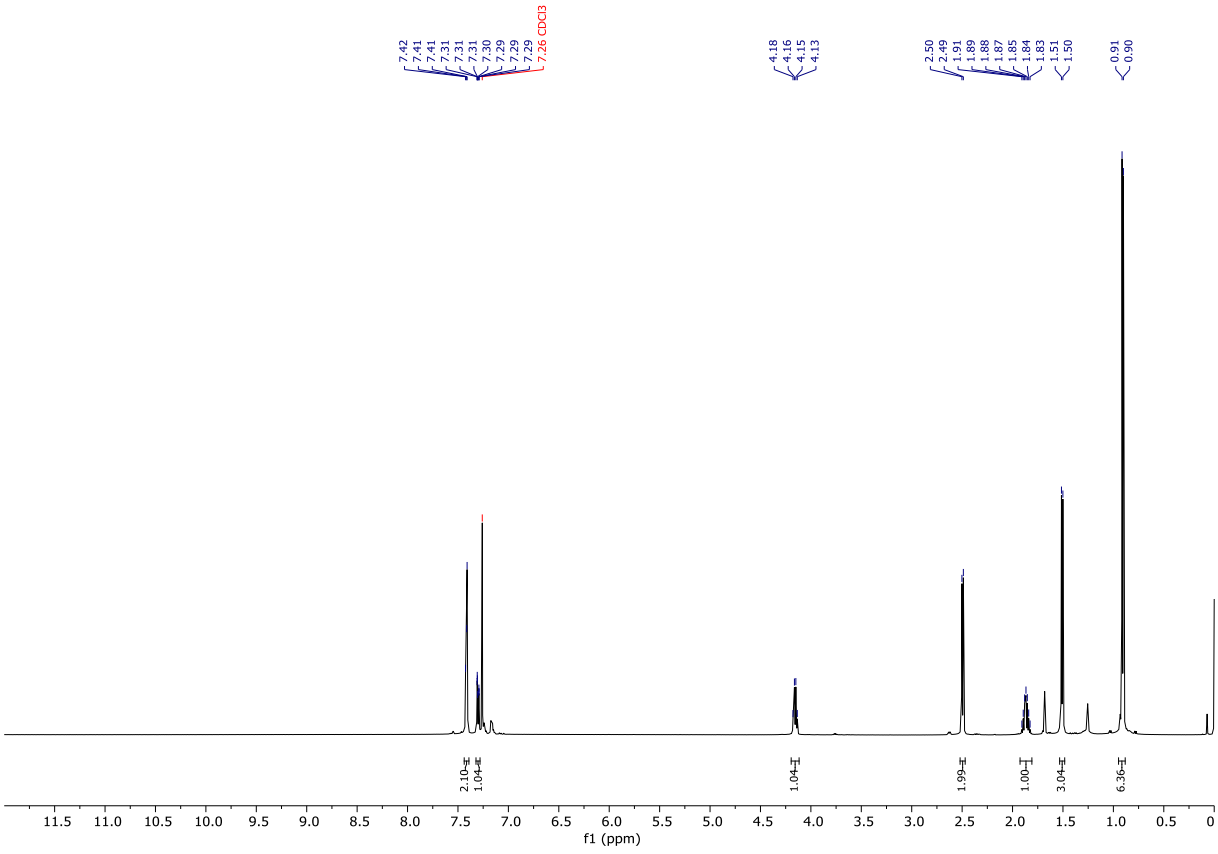


1-E1

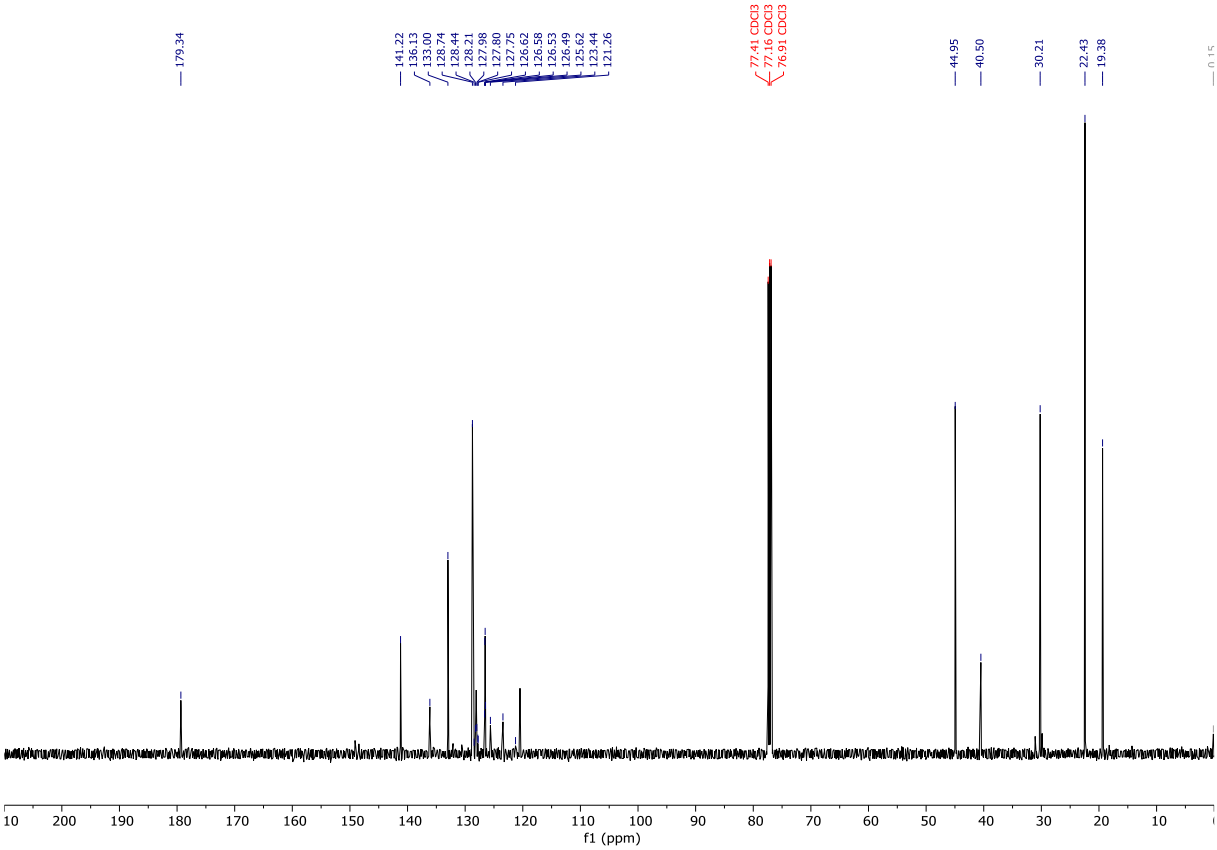
LCMS chromatogram



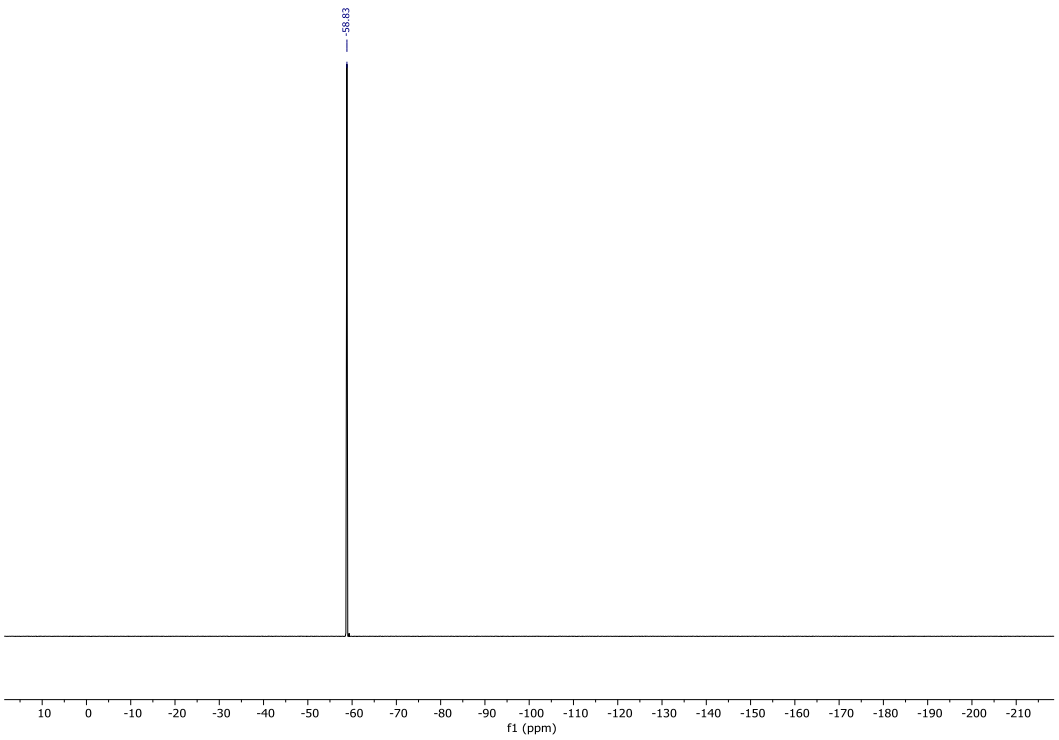
^{1}H-NMR (500MHz, $CDCl_3$):



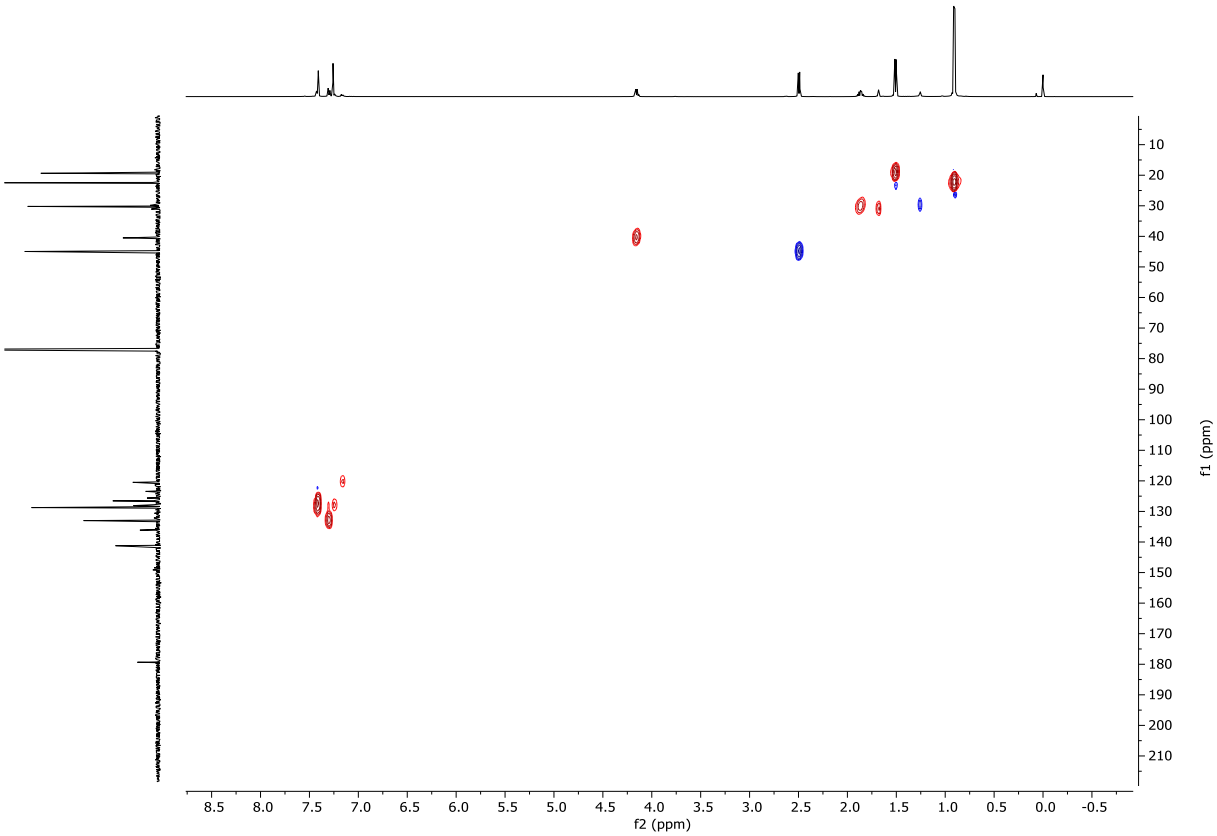
^{13}C-NMR (125MHz, $CDCl_3$):



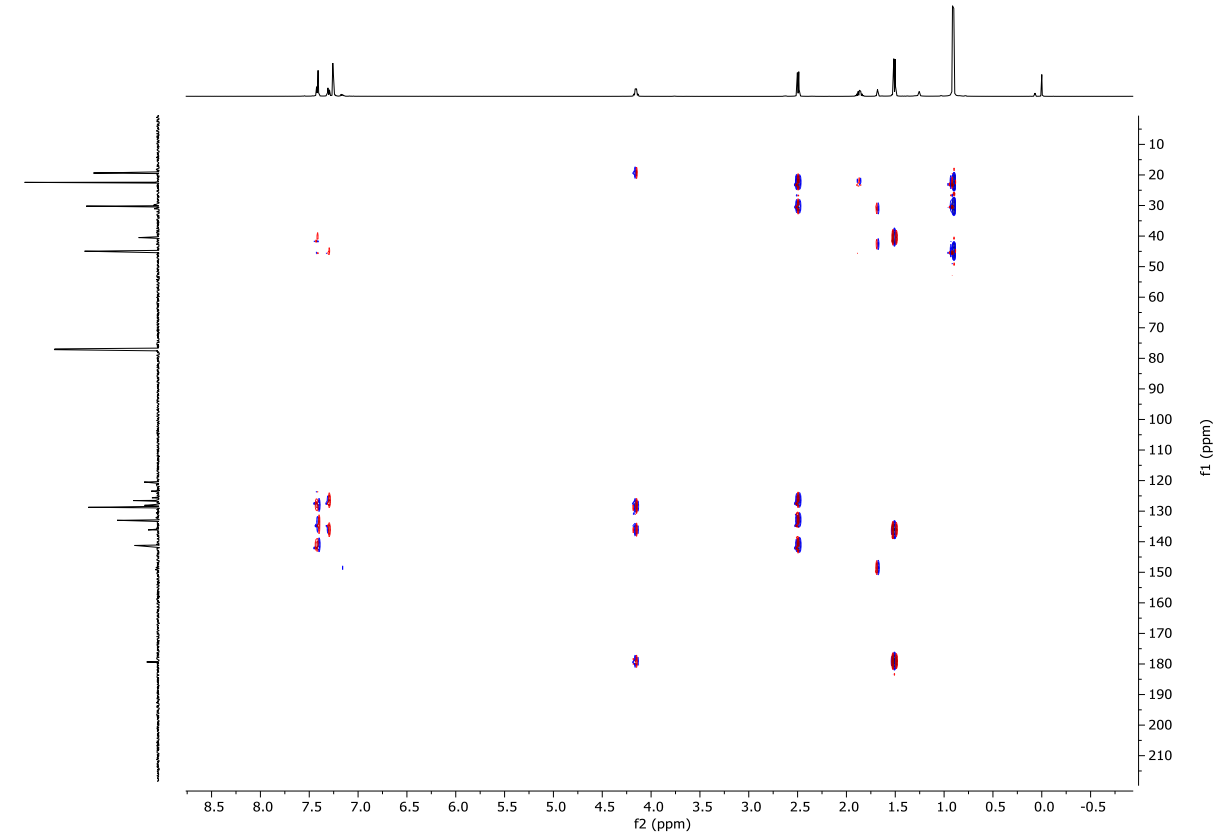
^{19}F-NMR (376MHz, $CDCl_3$):



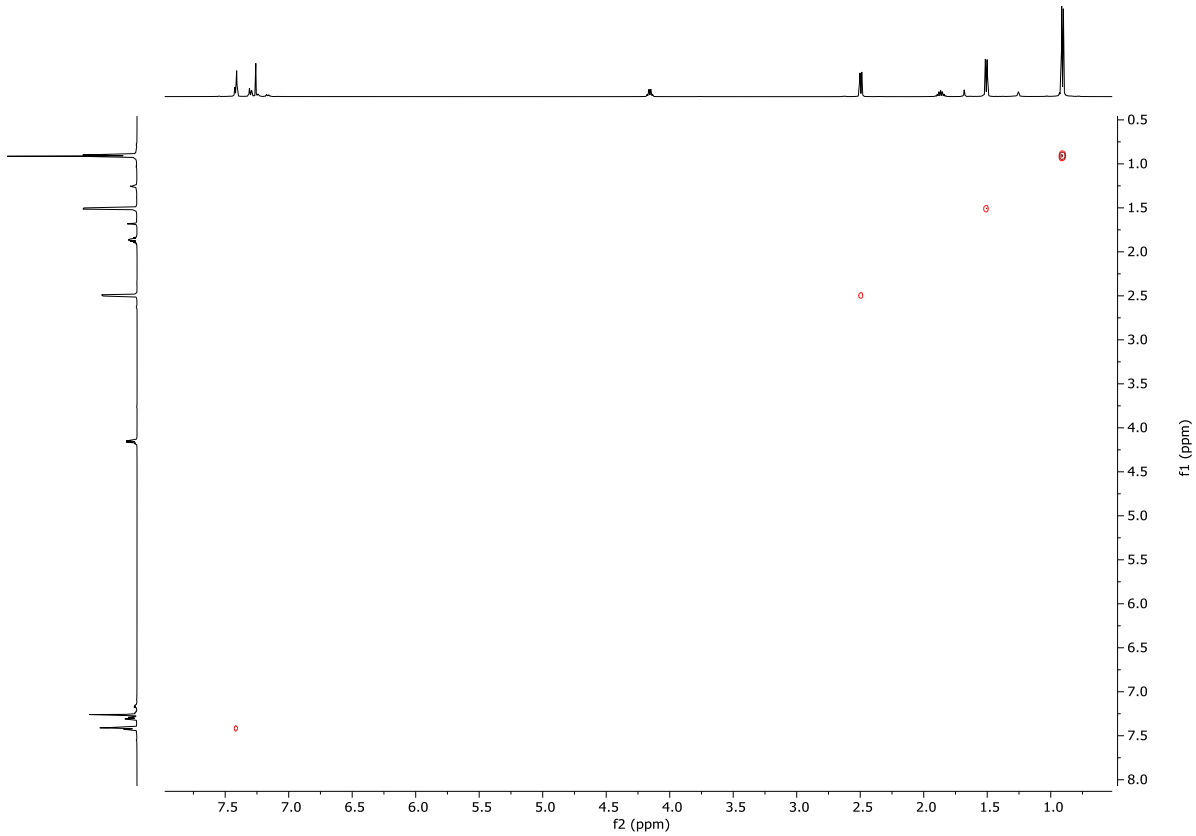
2D HSQC ($CDCl_3$):



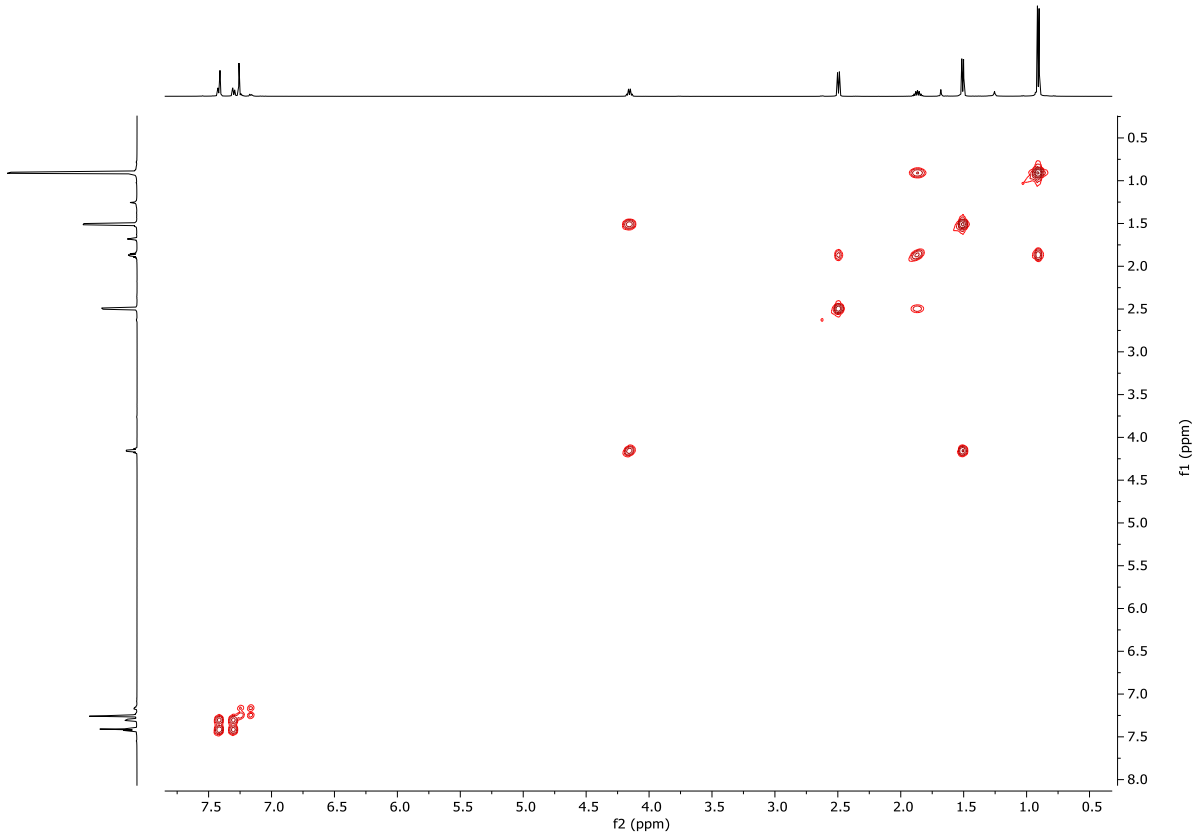
2D HMBC ($CDCl_3$):



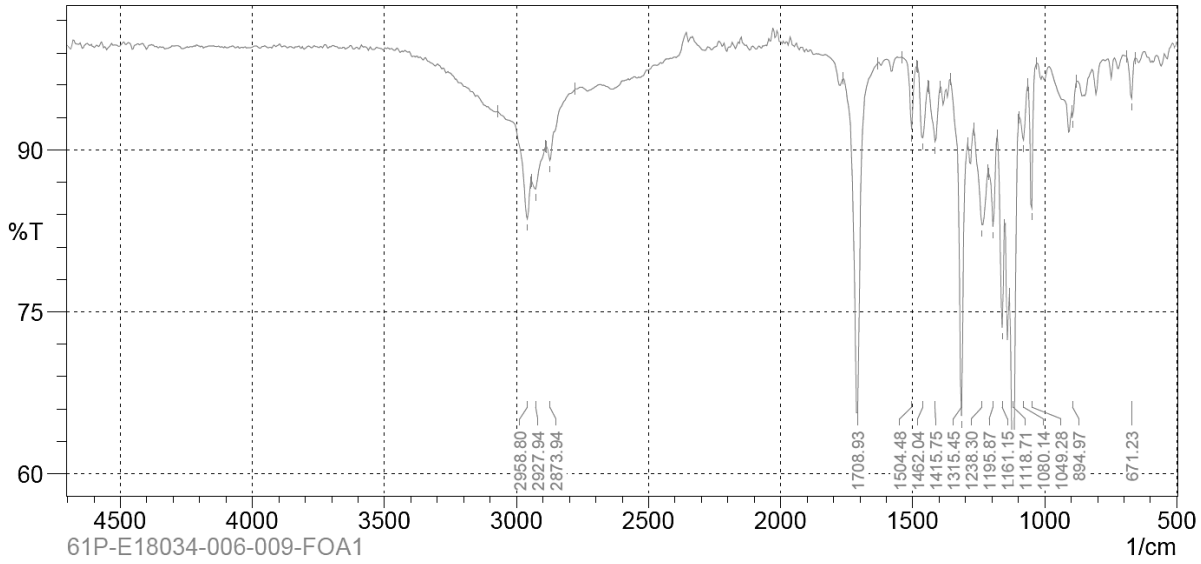
2D NOESY ($CDCl_3$):

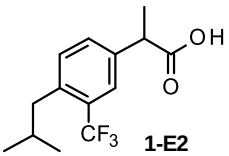


2D COSY ($CDCl_3$):



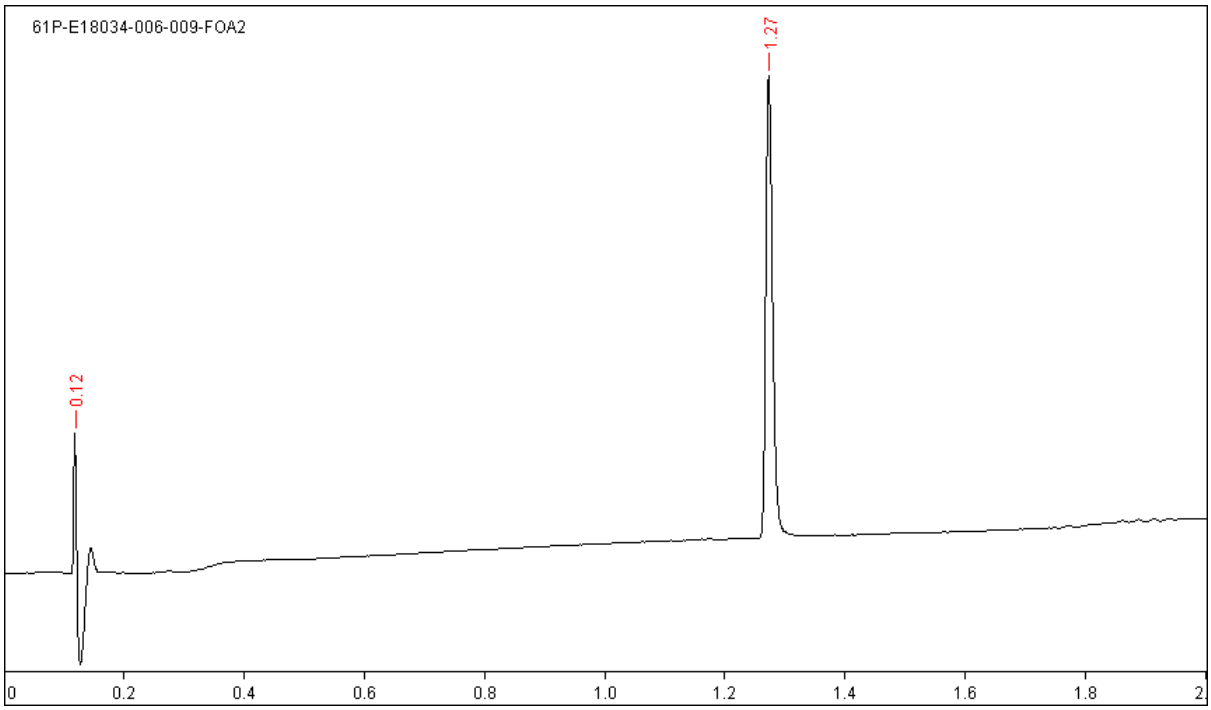
FT-IR:


90
%T
75
60
2958.80
2927.94
2873.94
1708.93
1504.48
1462.04
1415.75
1315.45
1238.30
1195.87
1161.15
1118.71
1080.14
1049.28
894.97
671.23
4500
4000
3500
3000
2500
2000
1500
1000
500
1/cm
61P-E18034-006-009-FOA1

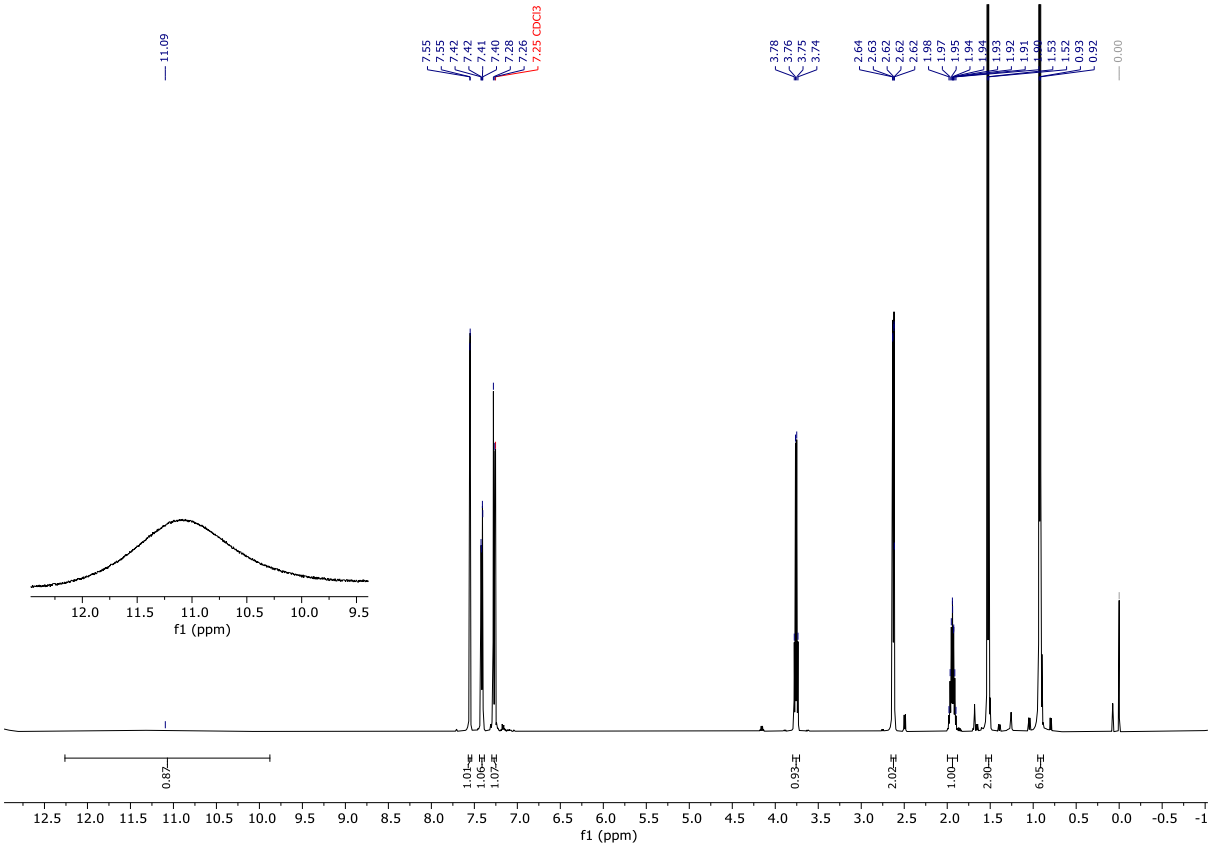


1-E2

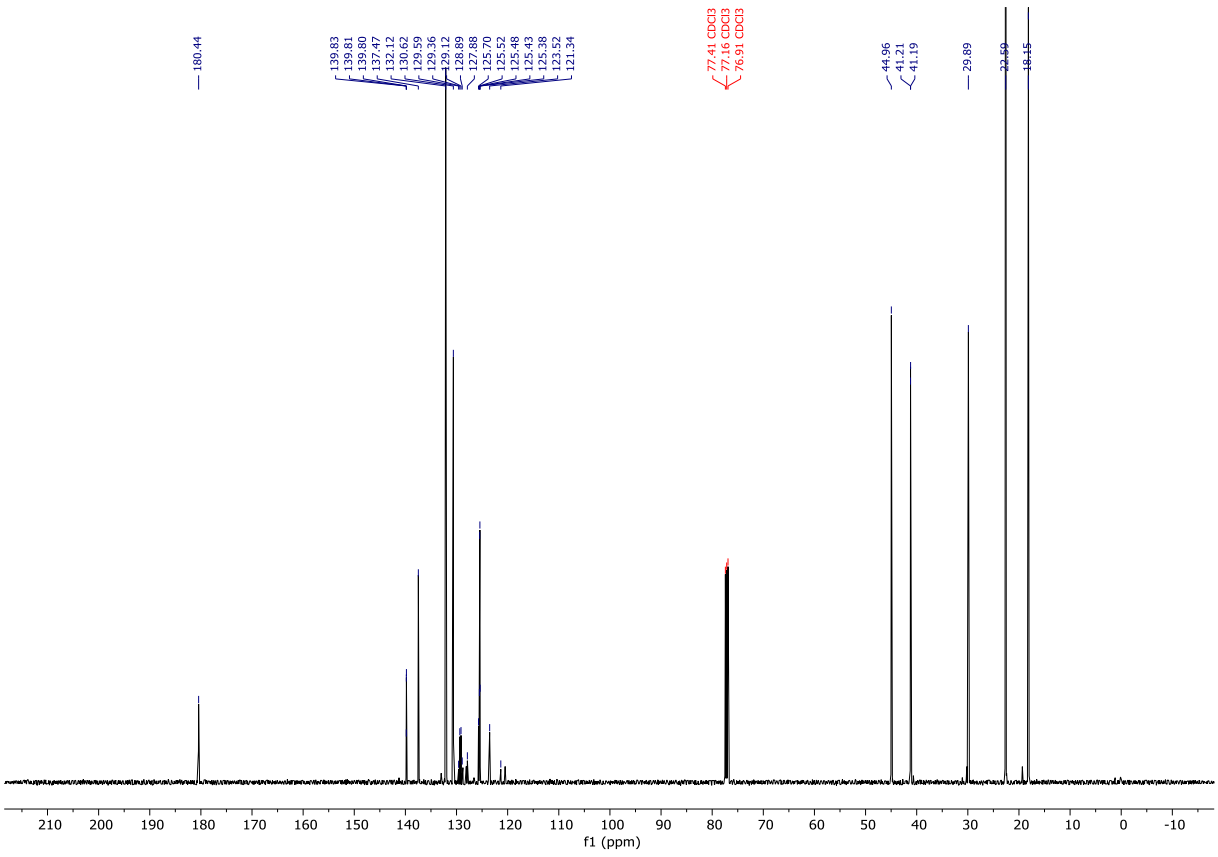
LCMS chromatogram



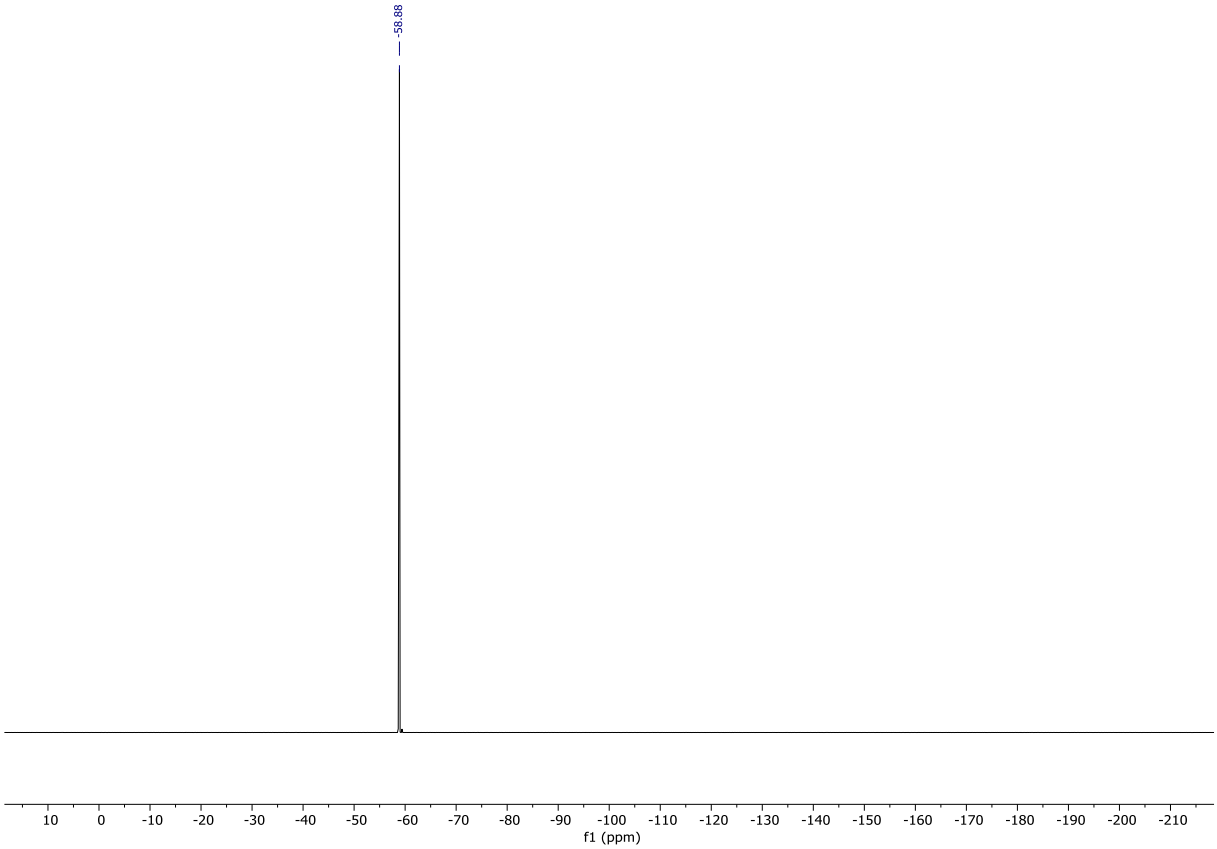
^{1}H-NMR (500MHz, $CDCl_3$):



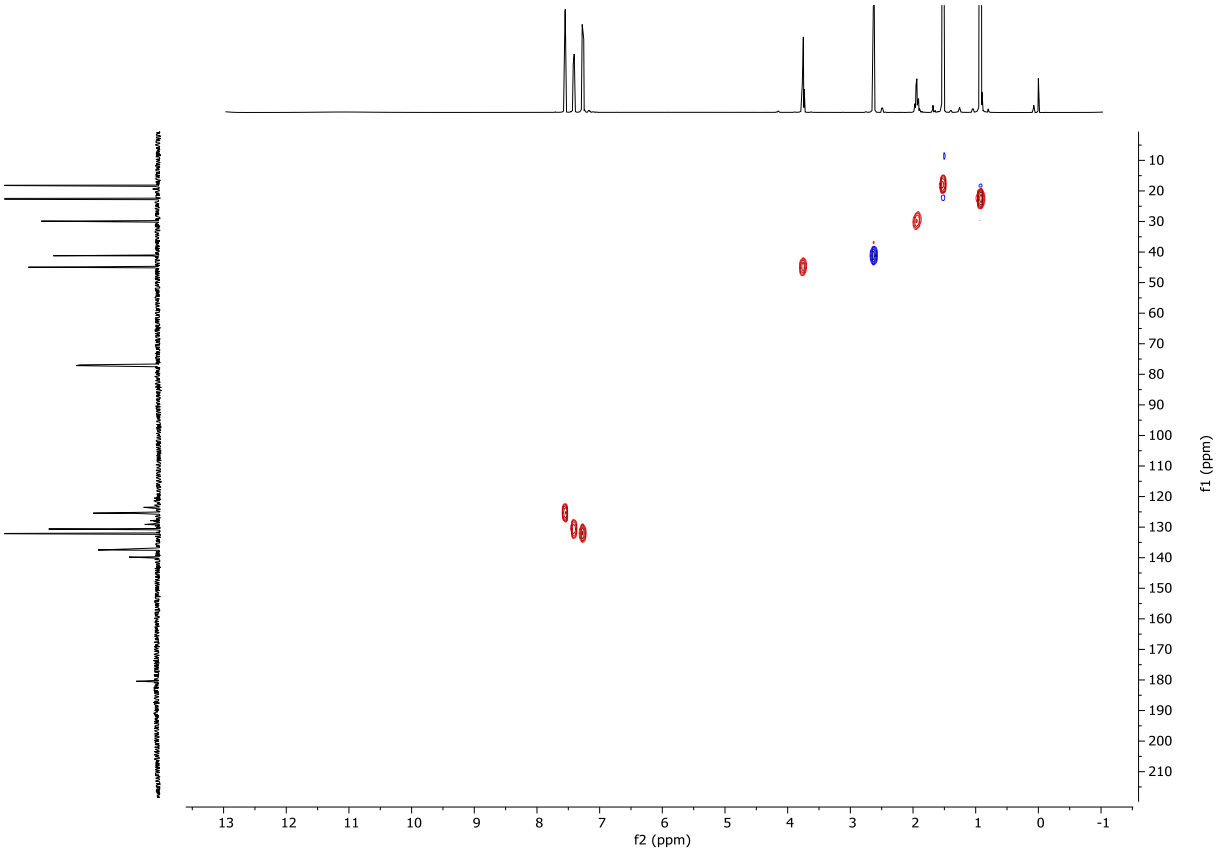
^{13}C-NMR (125MHz, $CDCl_3$):



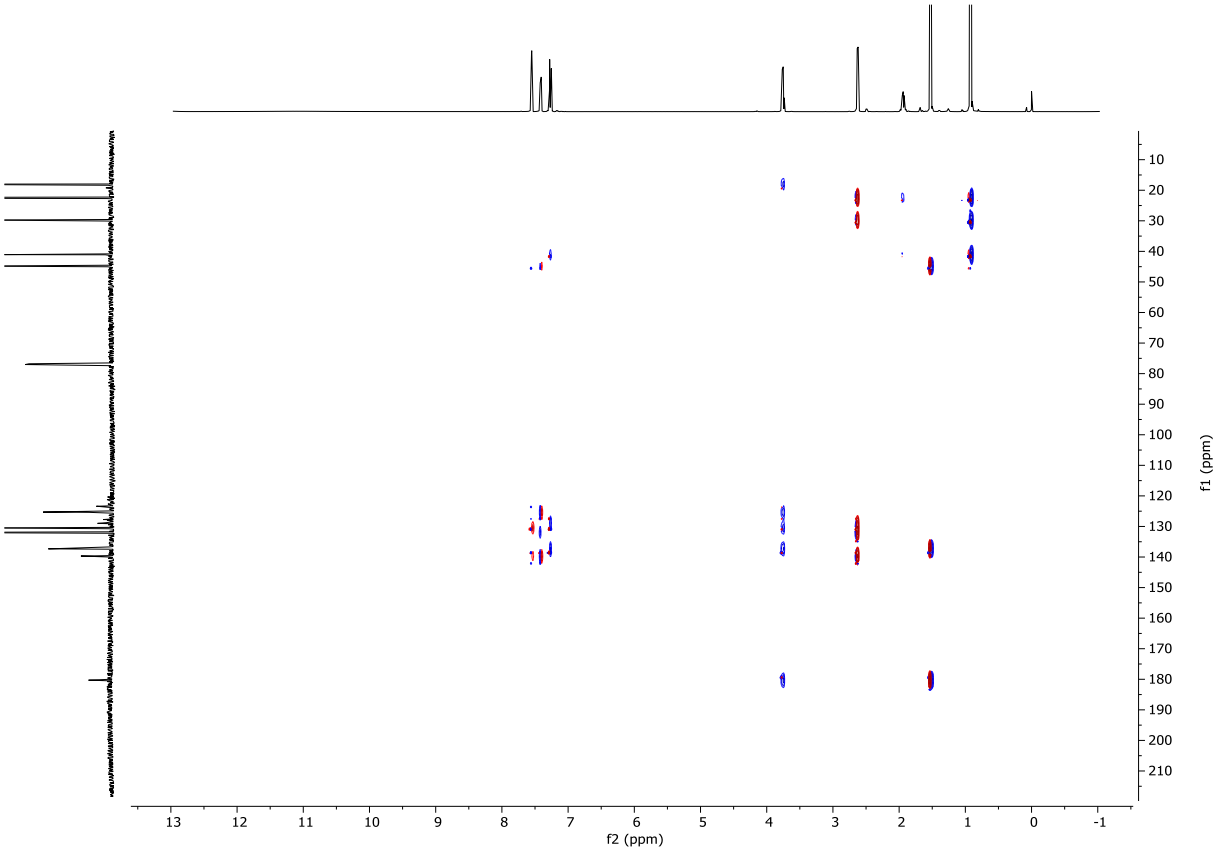
^{19}F-NMR (376MHz, $CDCl_3$):



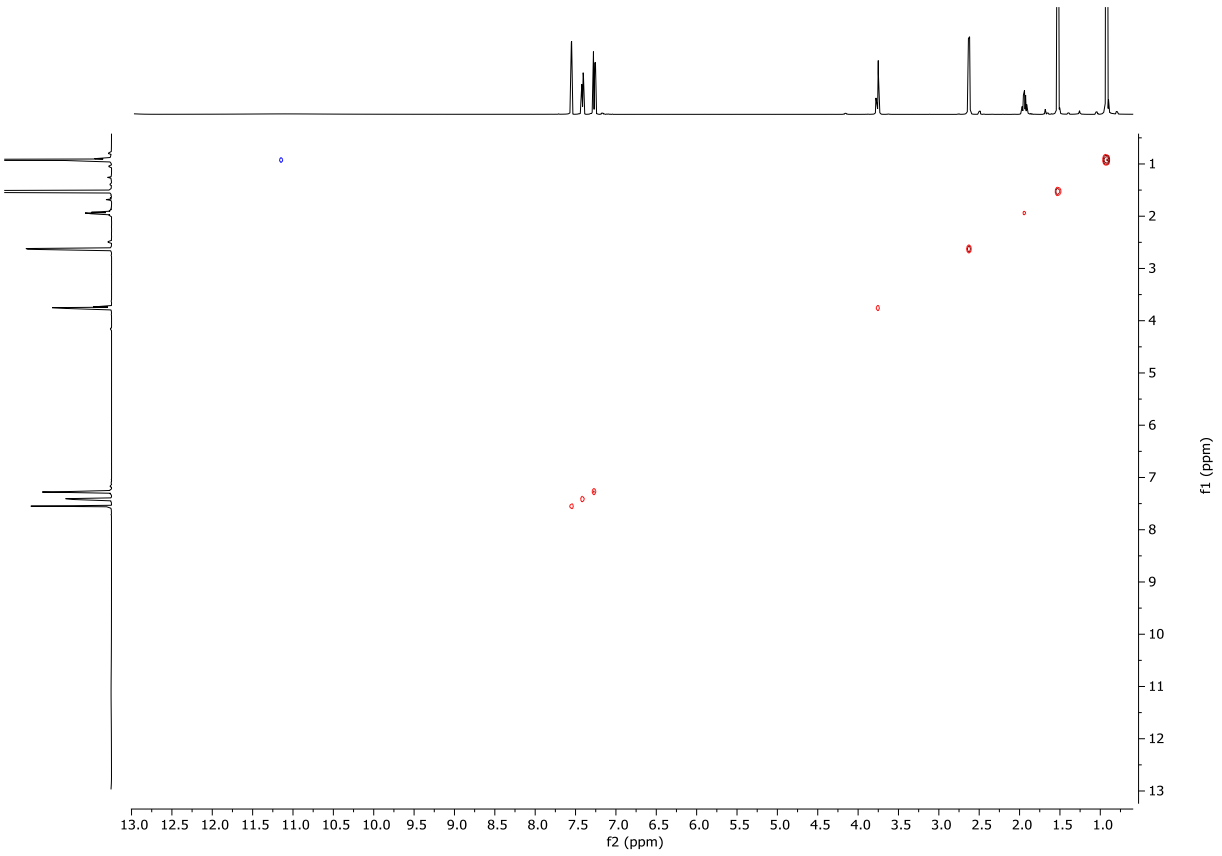
2D HSQC ($CDCl_3$):



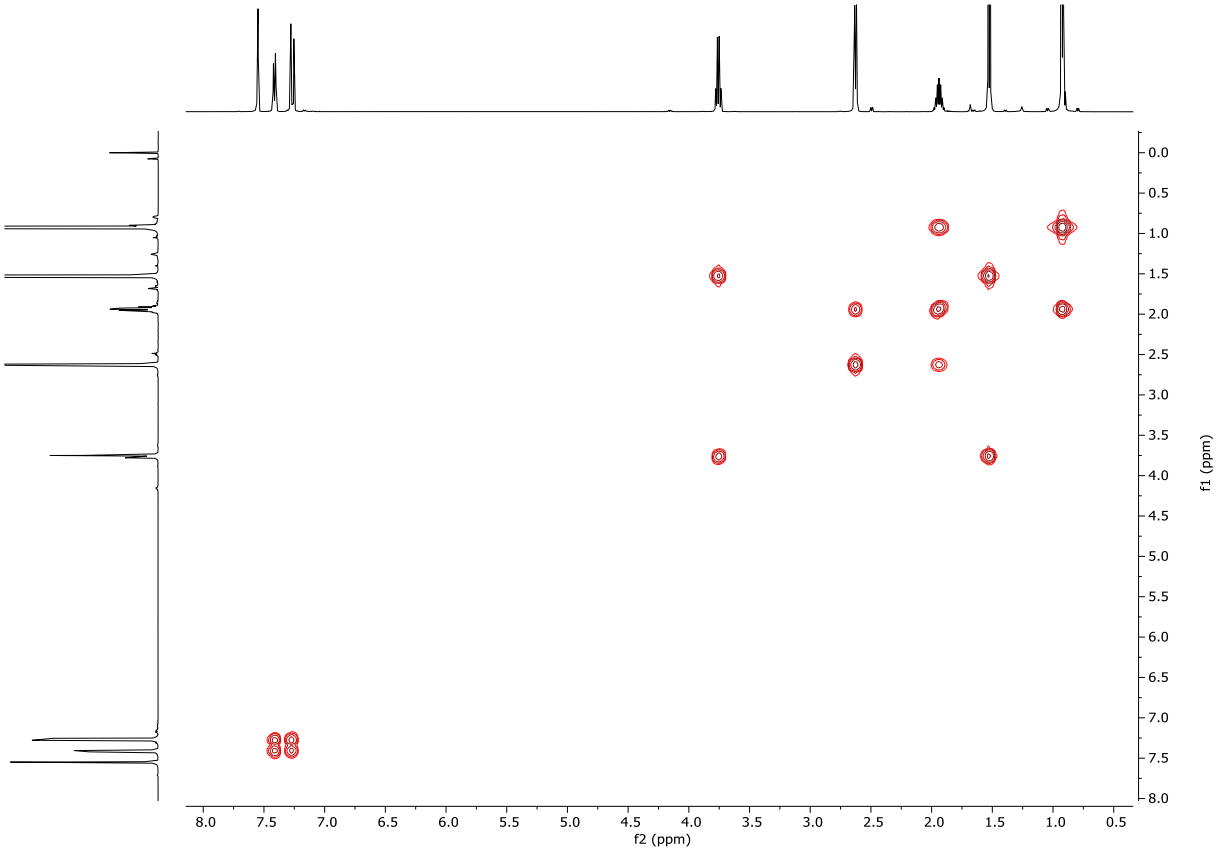
2D HMBC ($CDCl_3$):



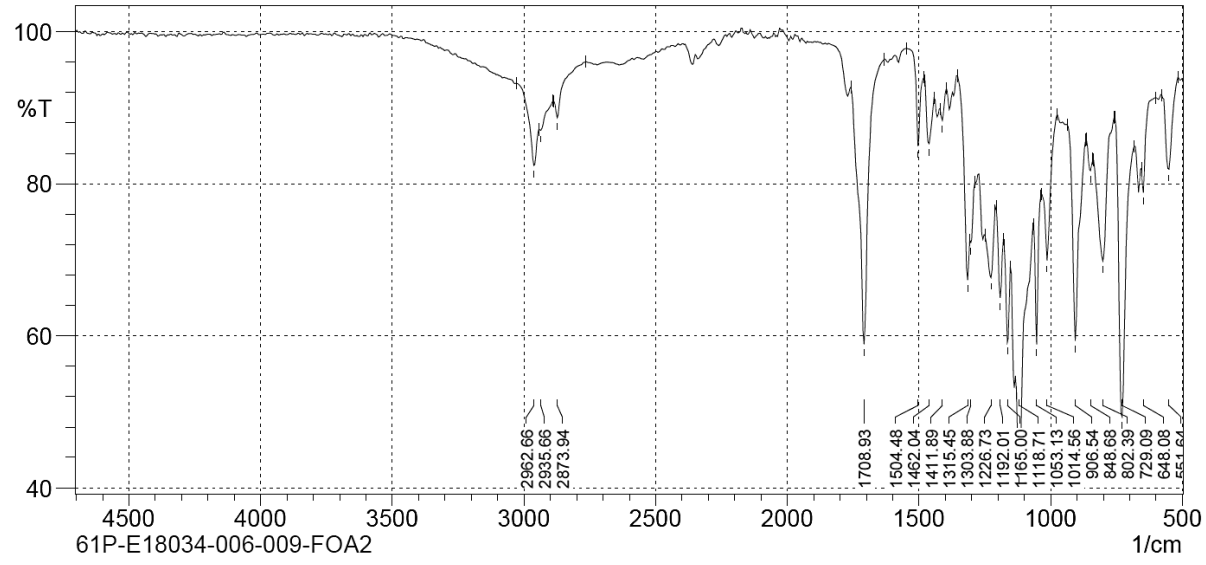
2D NOESY ($CDCl_3$):

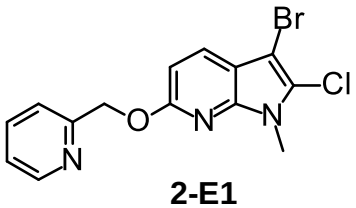


2D COSY ($CDCl_3$):



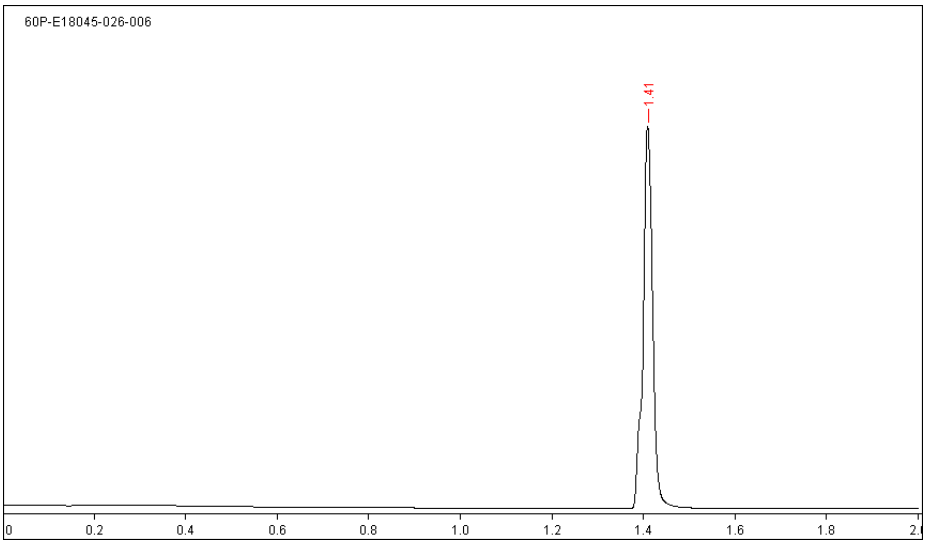
FT-IR



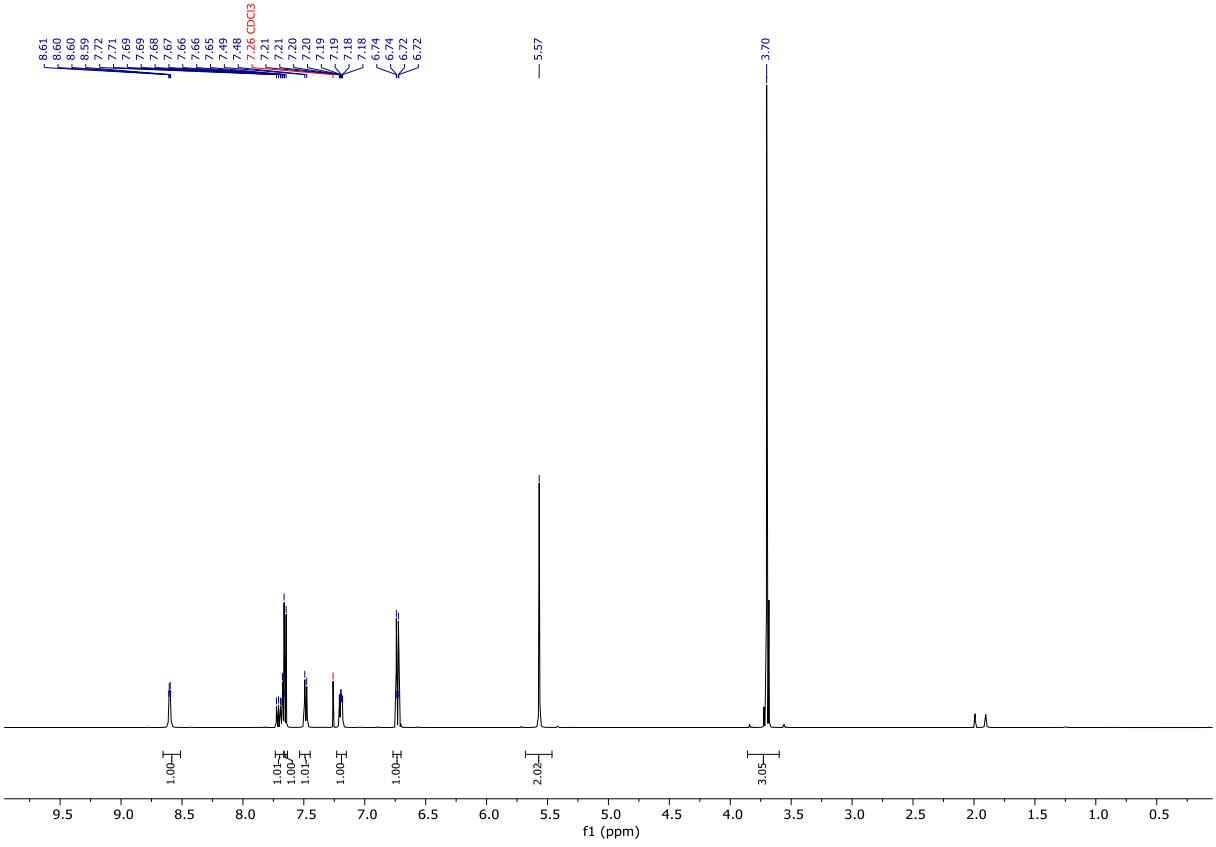


2-E1

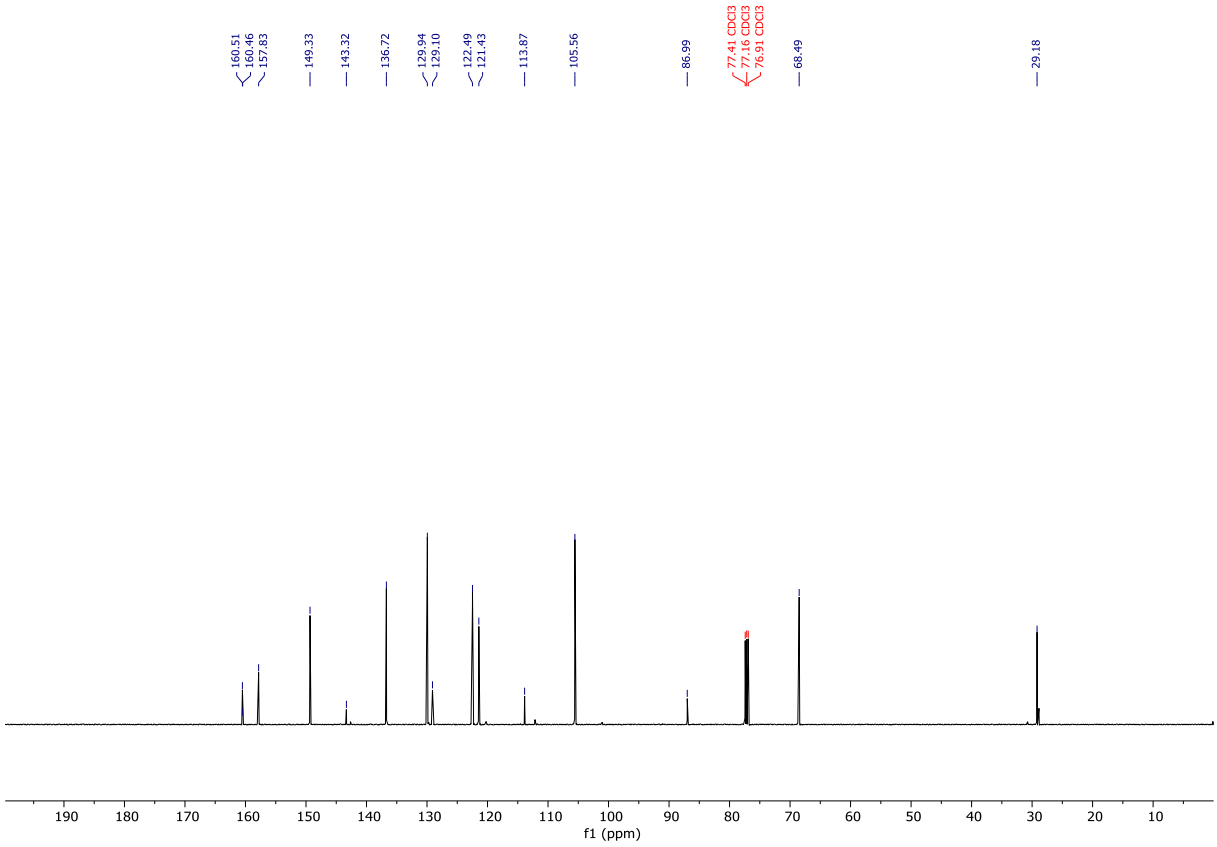
LC-MS:



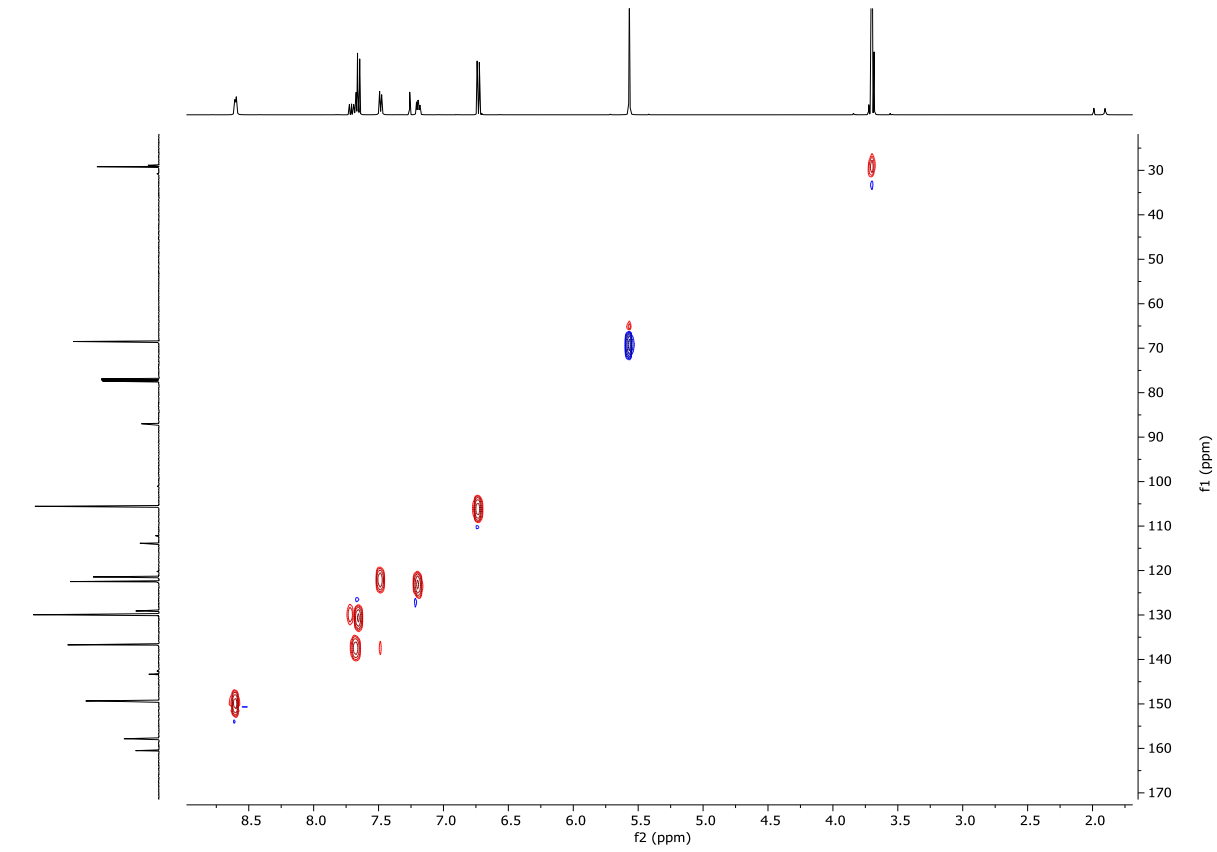
^{1}H-NMR (500MHz, $CDCl_3$):



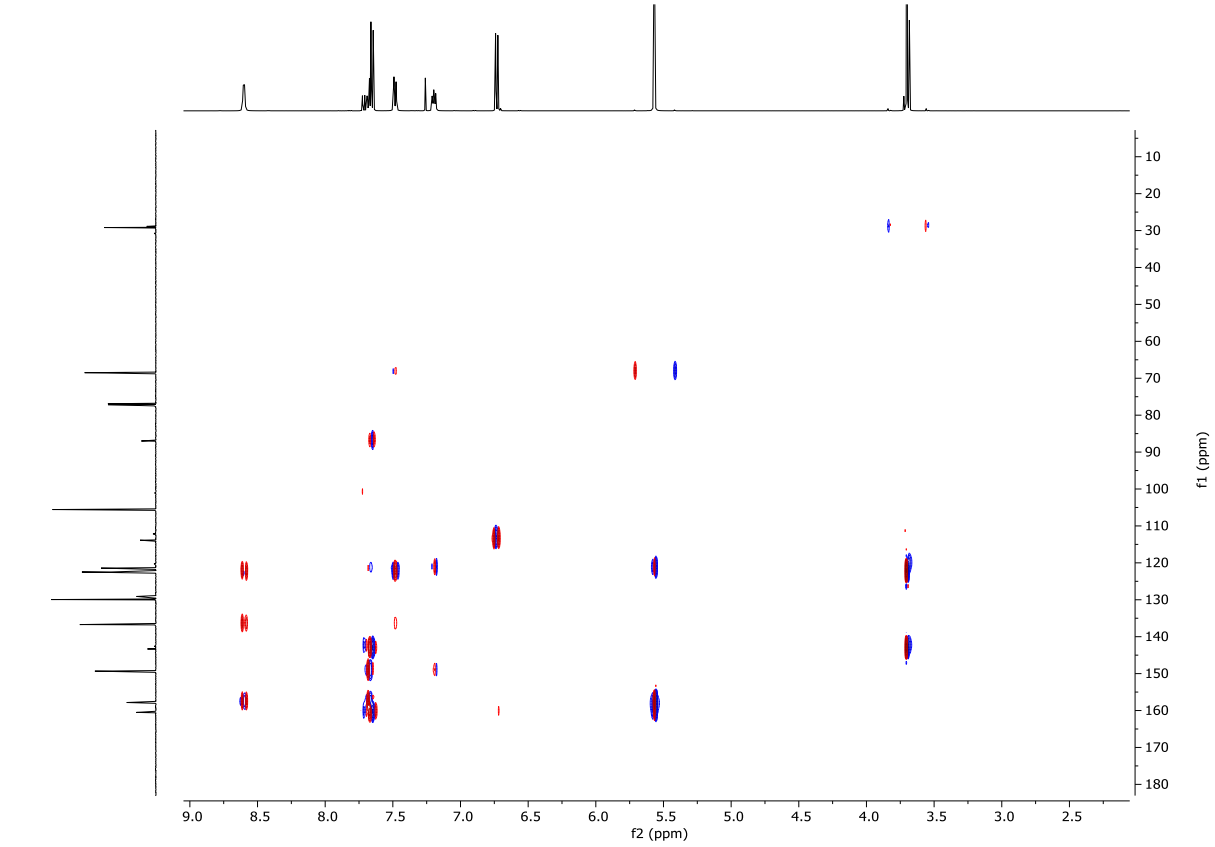
^{13}C-NMR (125MHz, $CDCl_3$):



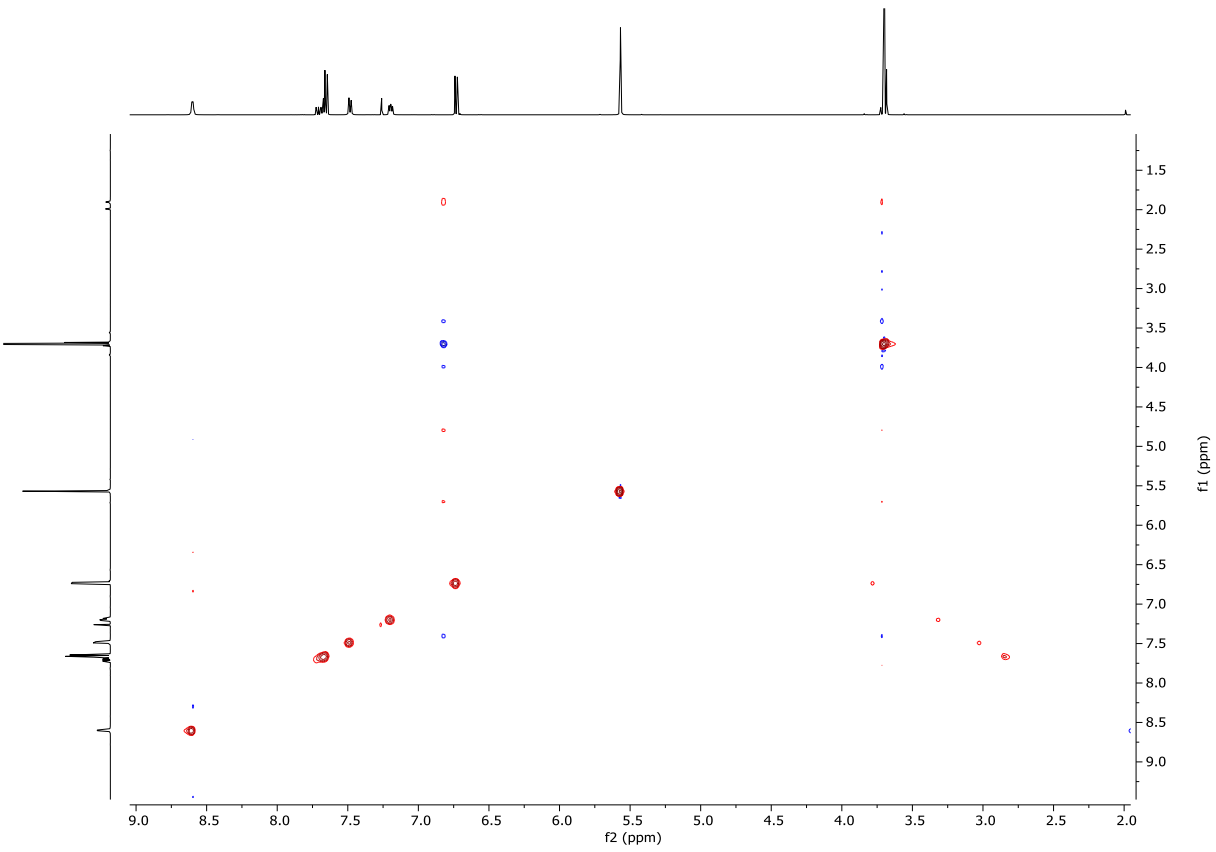
2D HSQC ($CDCl_3$):



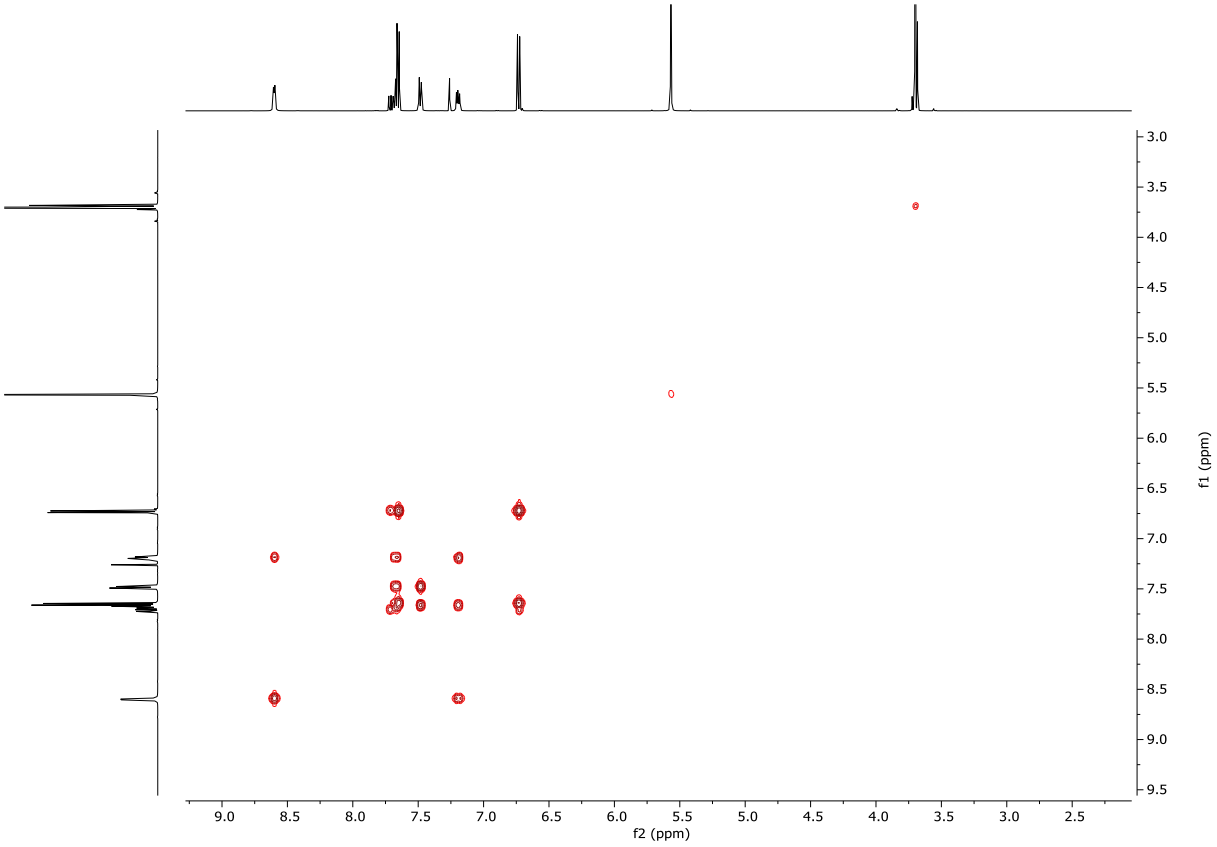
2D HMBC ($CDCl_3$):



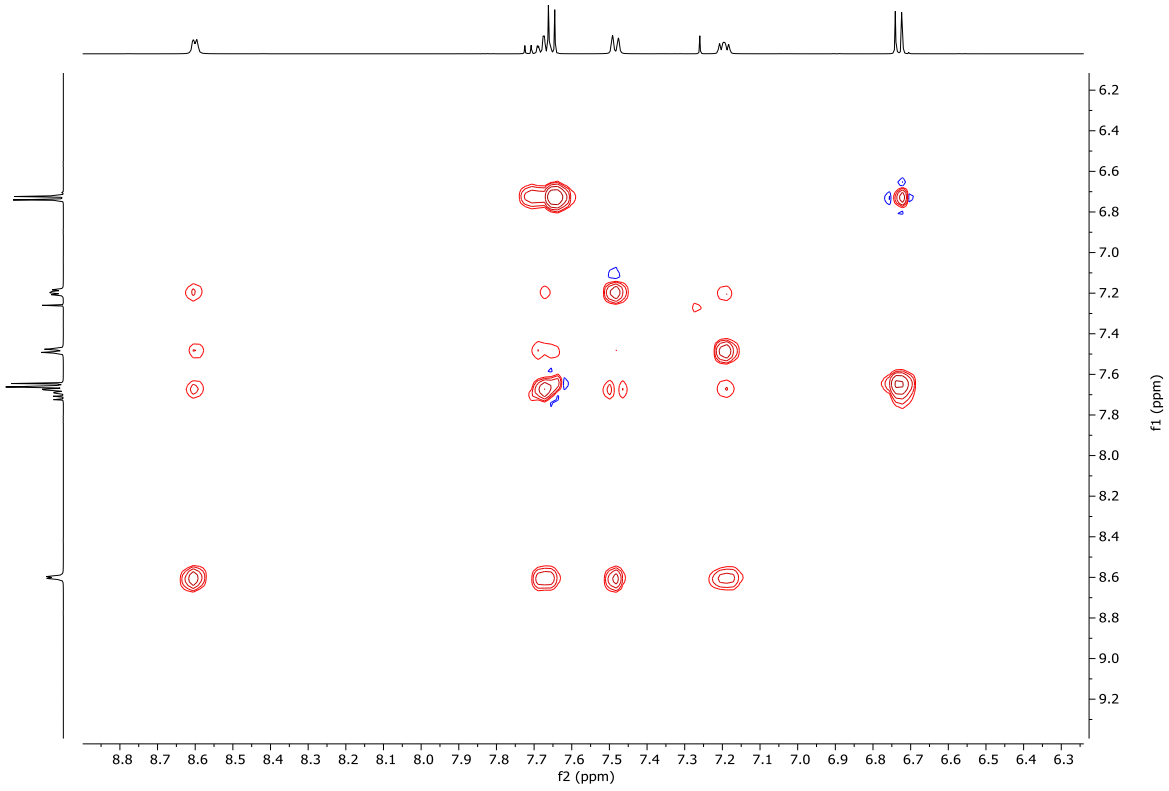
2D NOESY ($CDCl_3$):



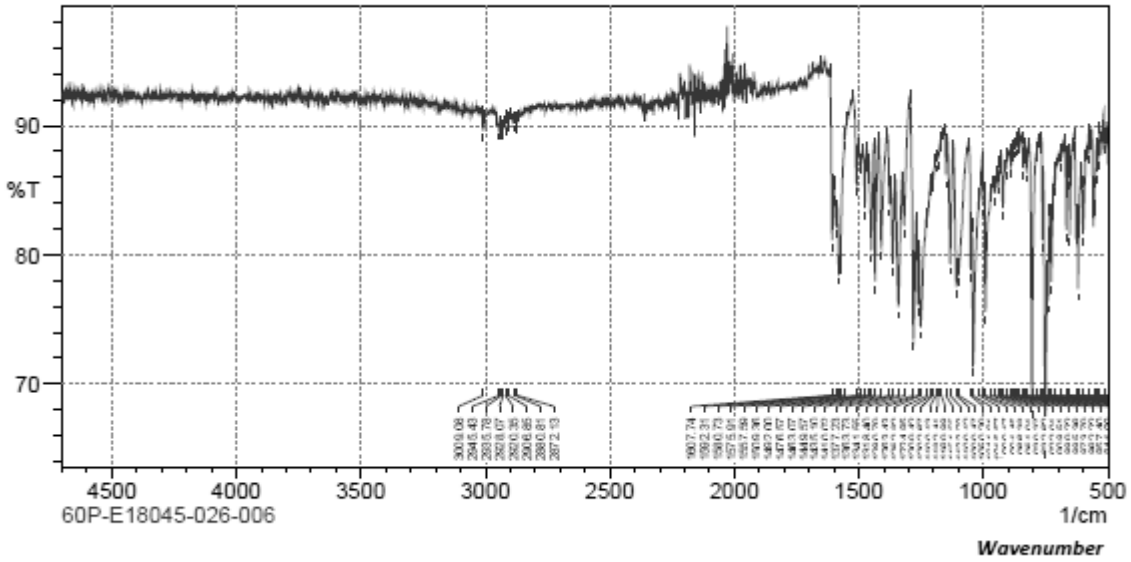
2D COSY ($CDCl_3$):

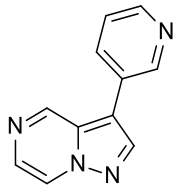


2D TOCSY ($CDCl_3$):



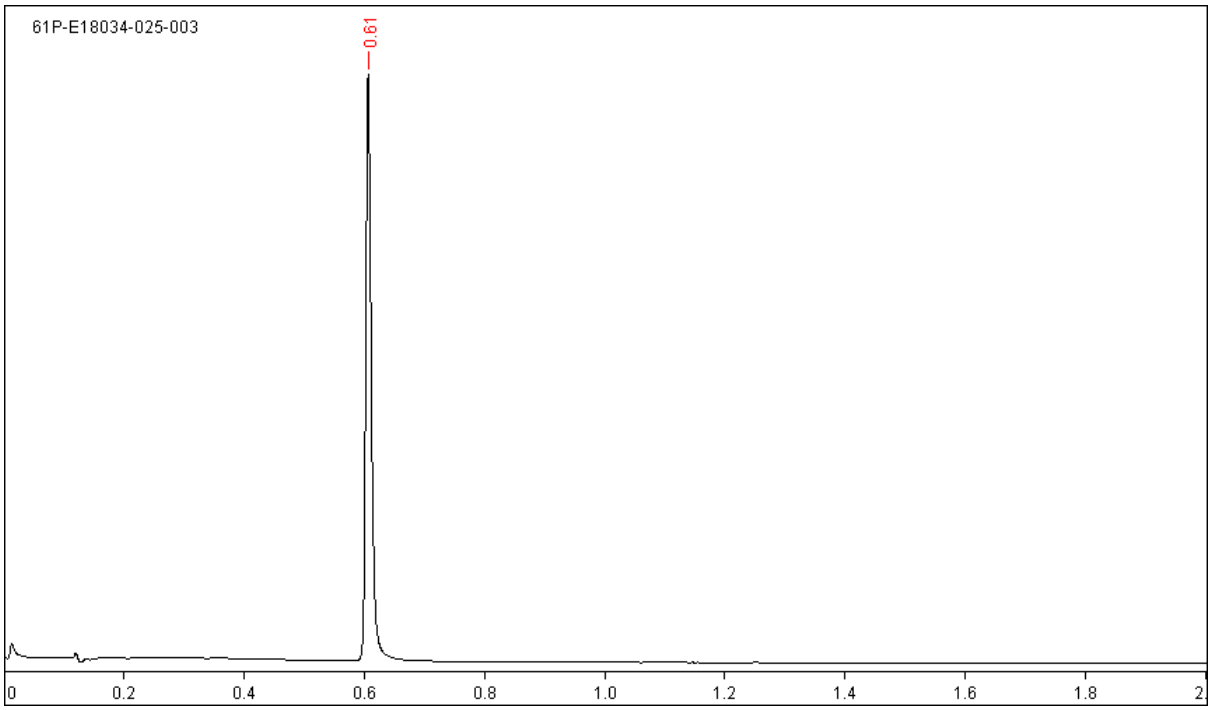
FT-IR


90
%T
80
70
4500
4000
3500
3000
2500
2000
1500
1000
500
1/cm
60P-E18045-026-006
Wavenumber

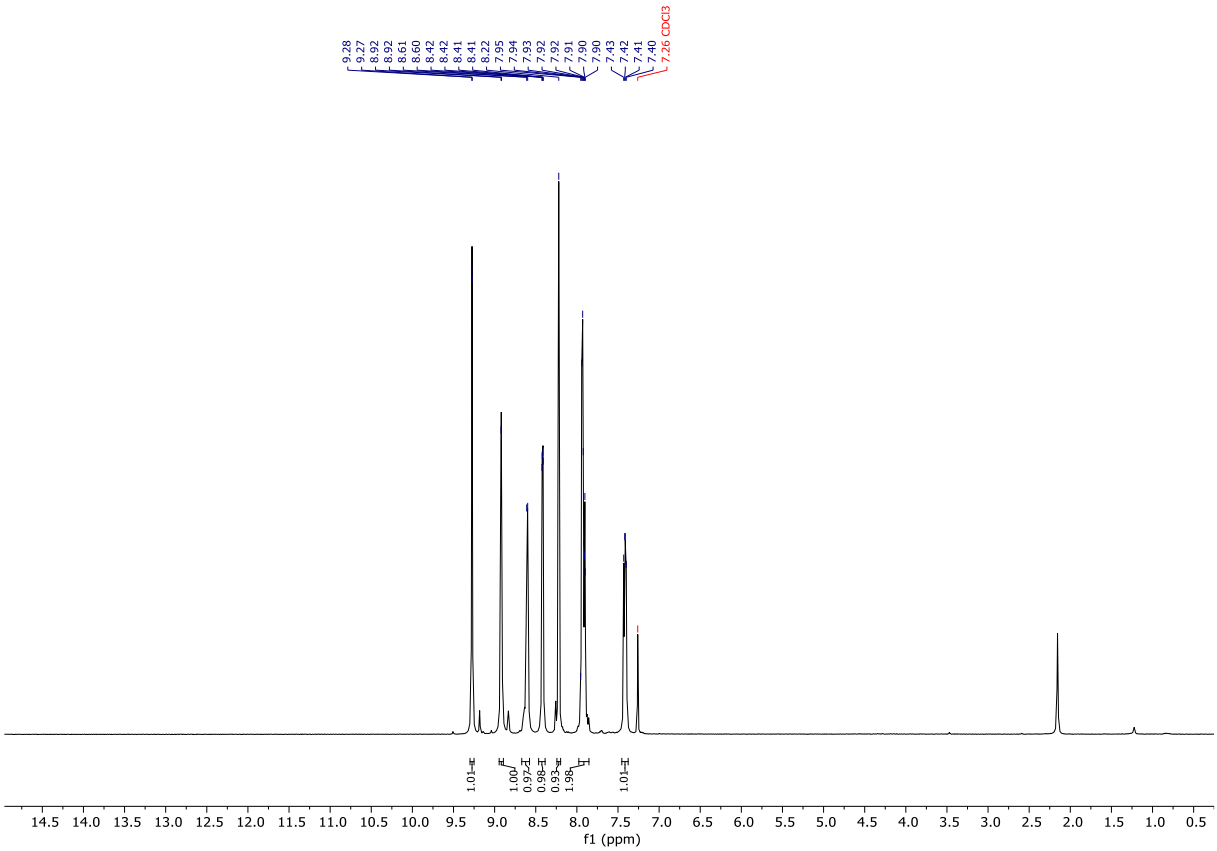


3

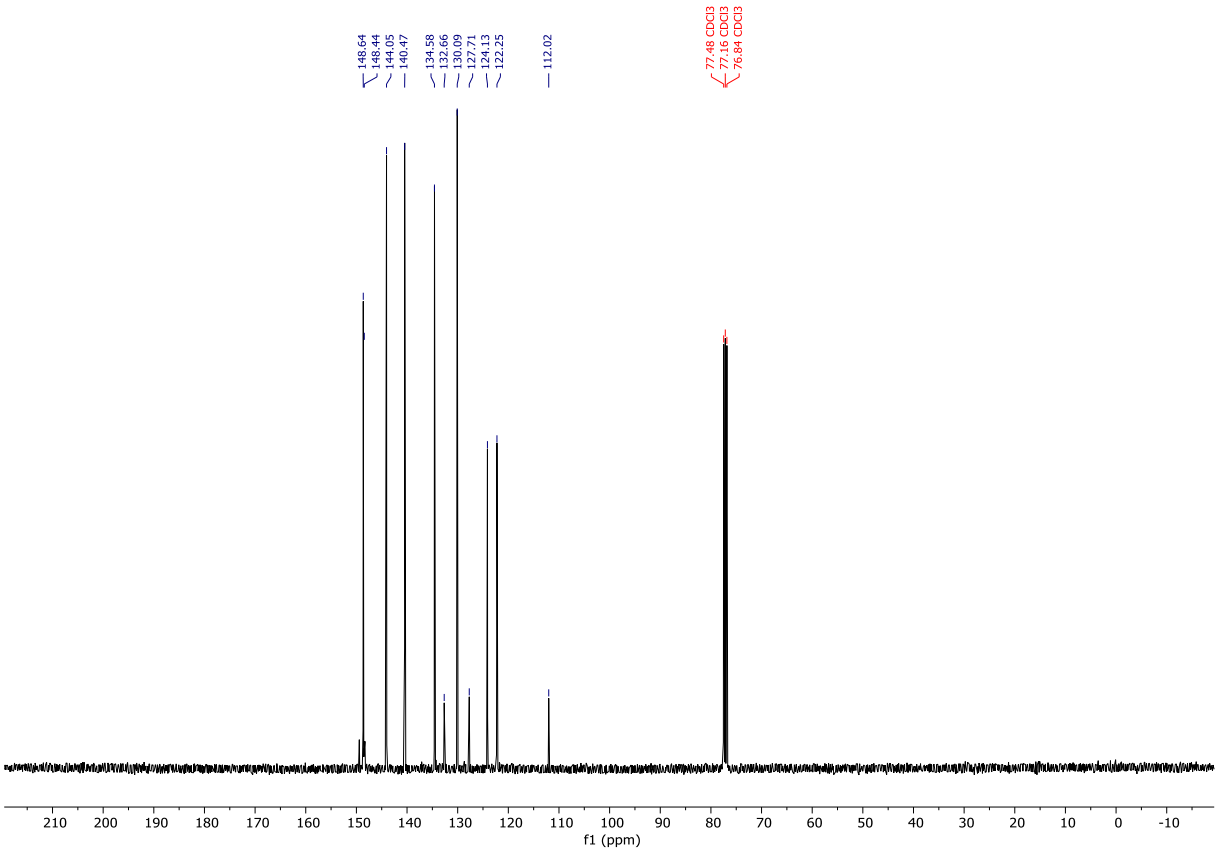
LC-MS (High pH method):



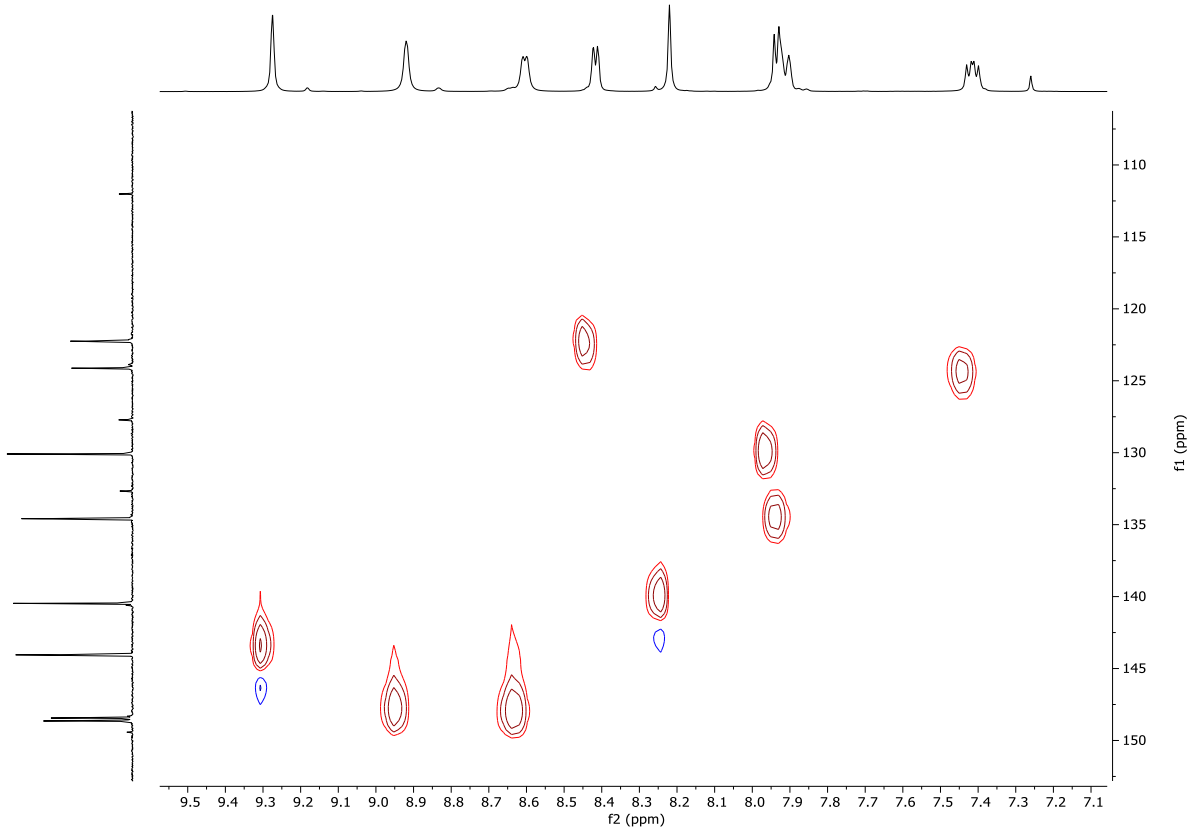
^{1}H-NMR (400MHz, $CDCl_3$):



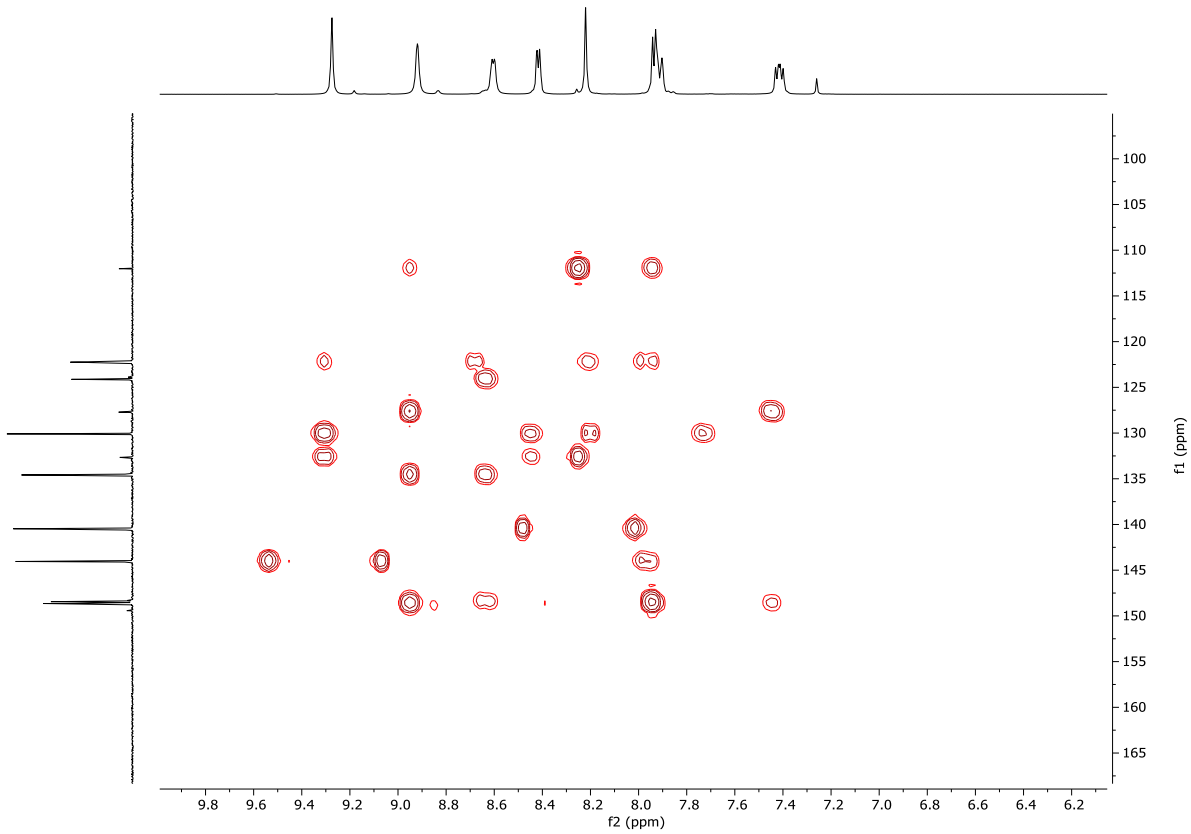
^{13}C-NMR (100MHz, $CDCl_3$):



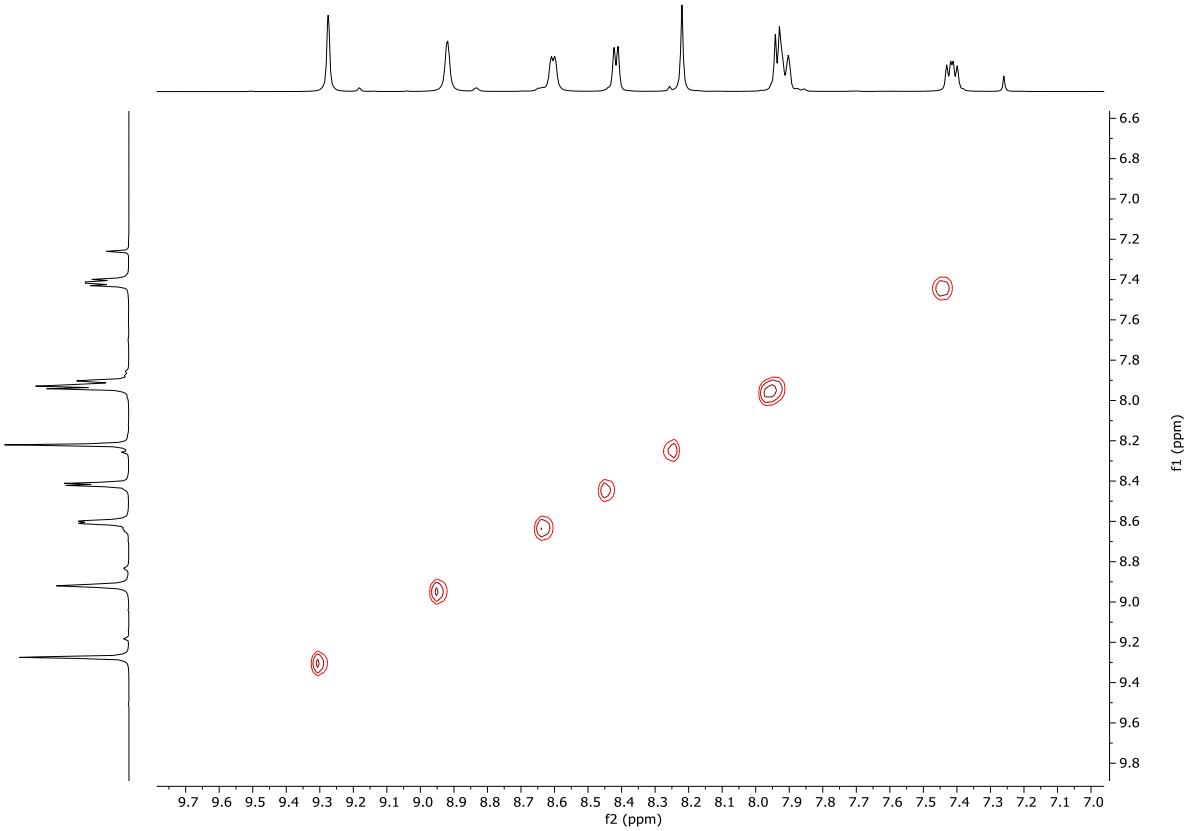
2D HSQC ($CDCl_3$):



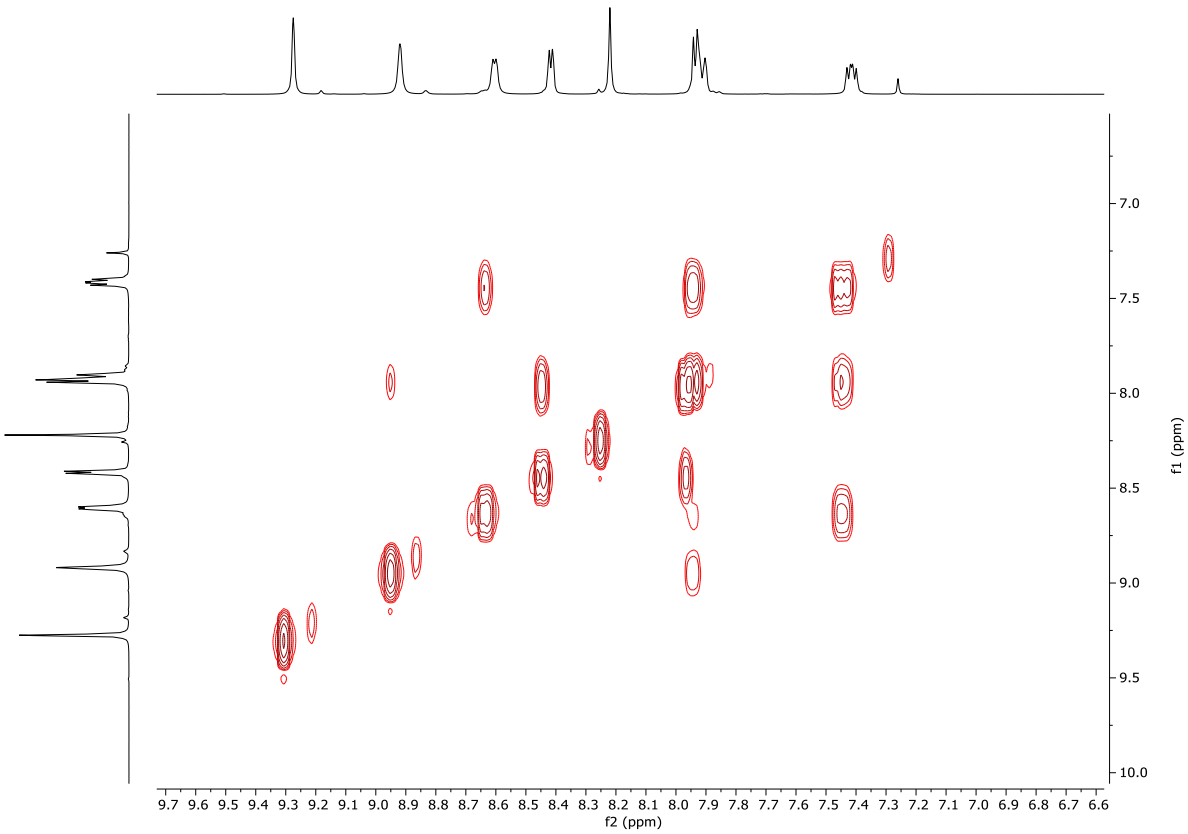
2D HMBC ($CDCl_3$):



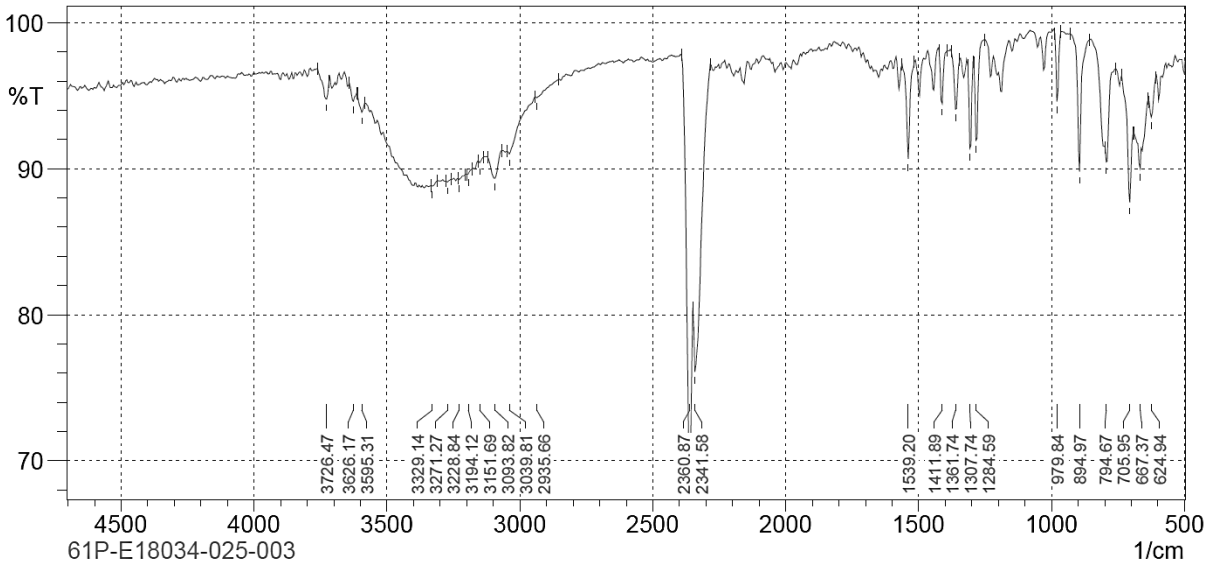
2D NOESY ($CDCl_3$):

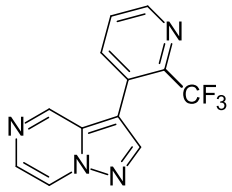


2D COSY ($CDCl_3$):



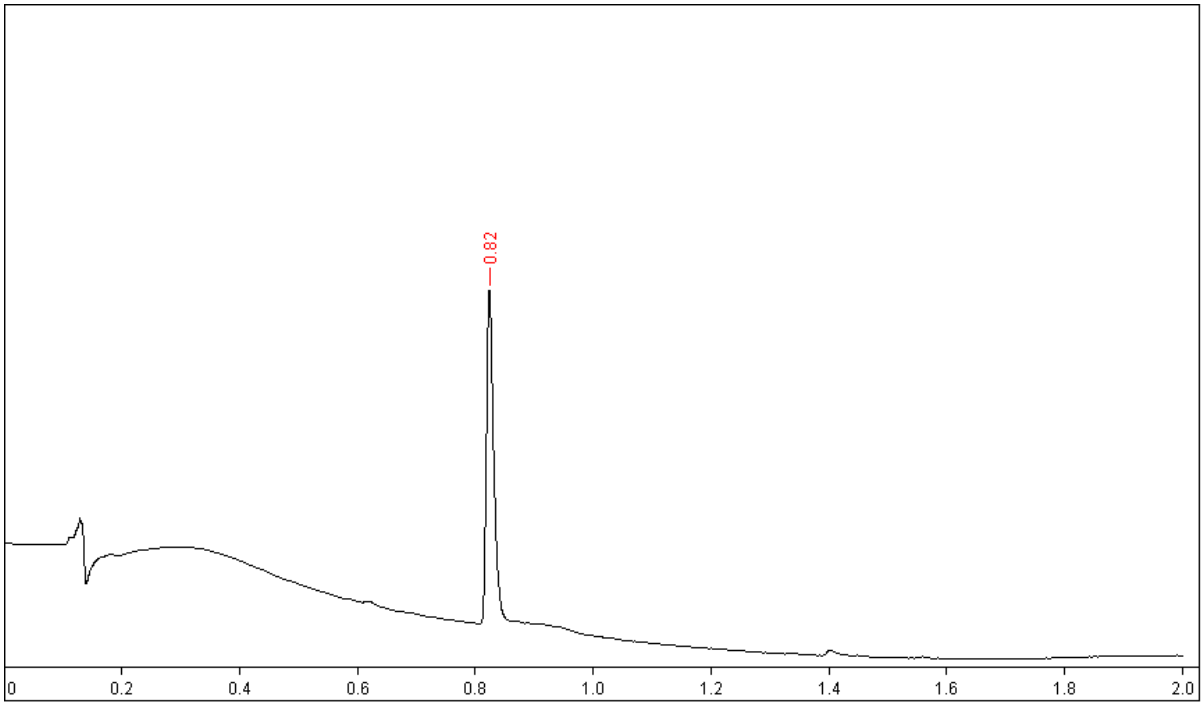
FT-IR


100
%T
90
80
70
3726.47
3626.17
3595.31
3329.14
3271.27
3228.84
3194.12
3151.69
3093.82
3039.81
2935.66
2360.87
2341.58
1539.20
1411.89
1361.74
1307.74
1284.59
979.84
894.97
794.67
705.95
667.37
624.94
4500
4000
3500
3000
2500
2000
1500
1000
500
1/cm
61P-E18034-025-003

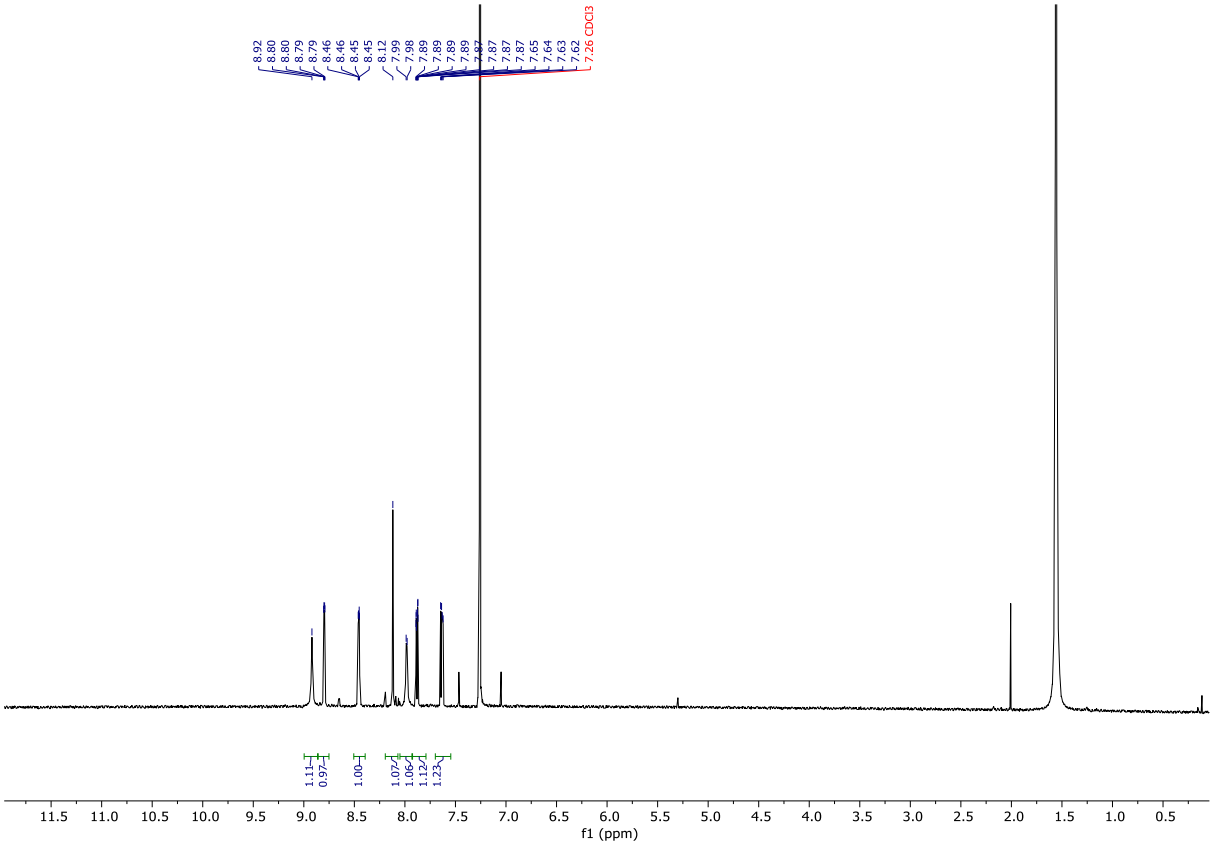


3-E1

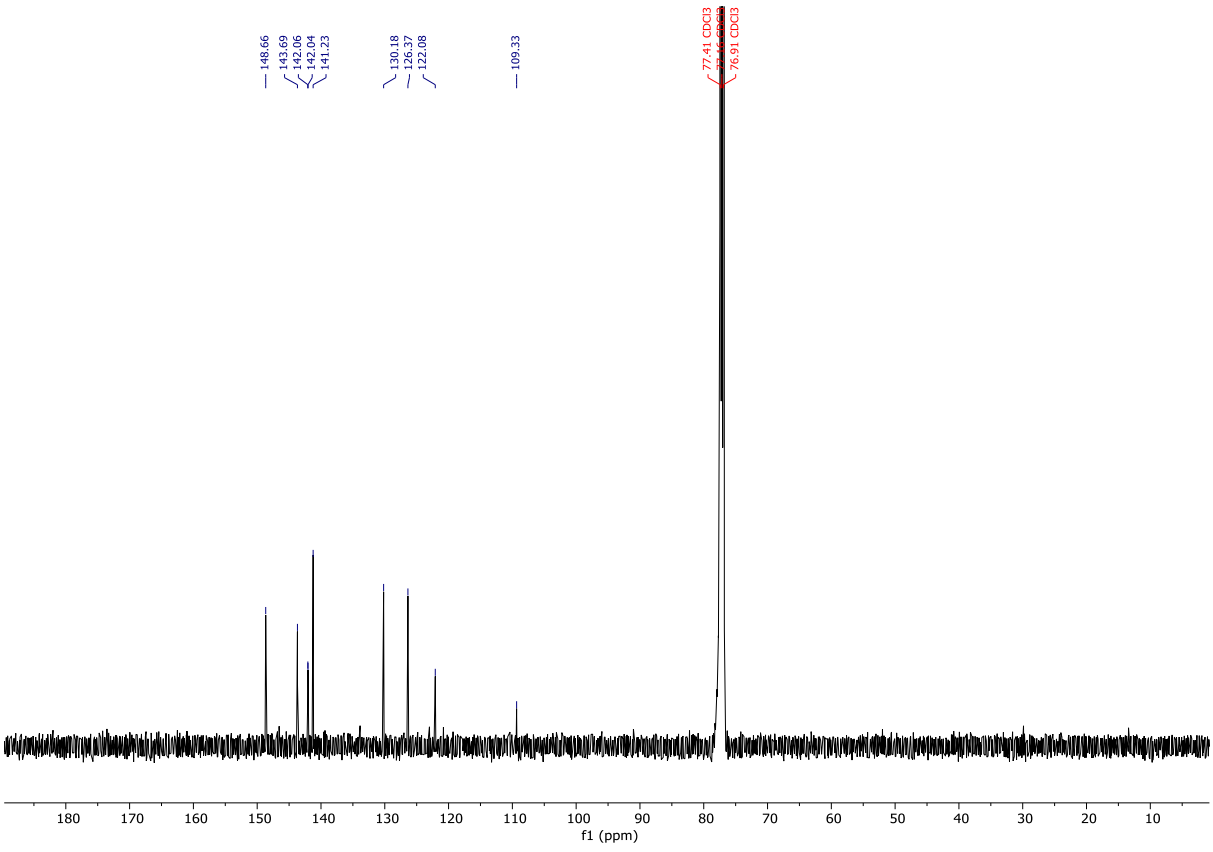
LC-MS (High pH method):



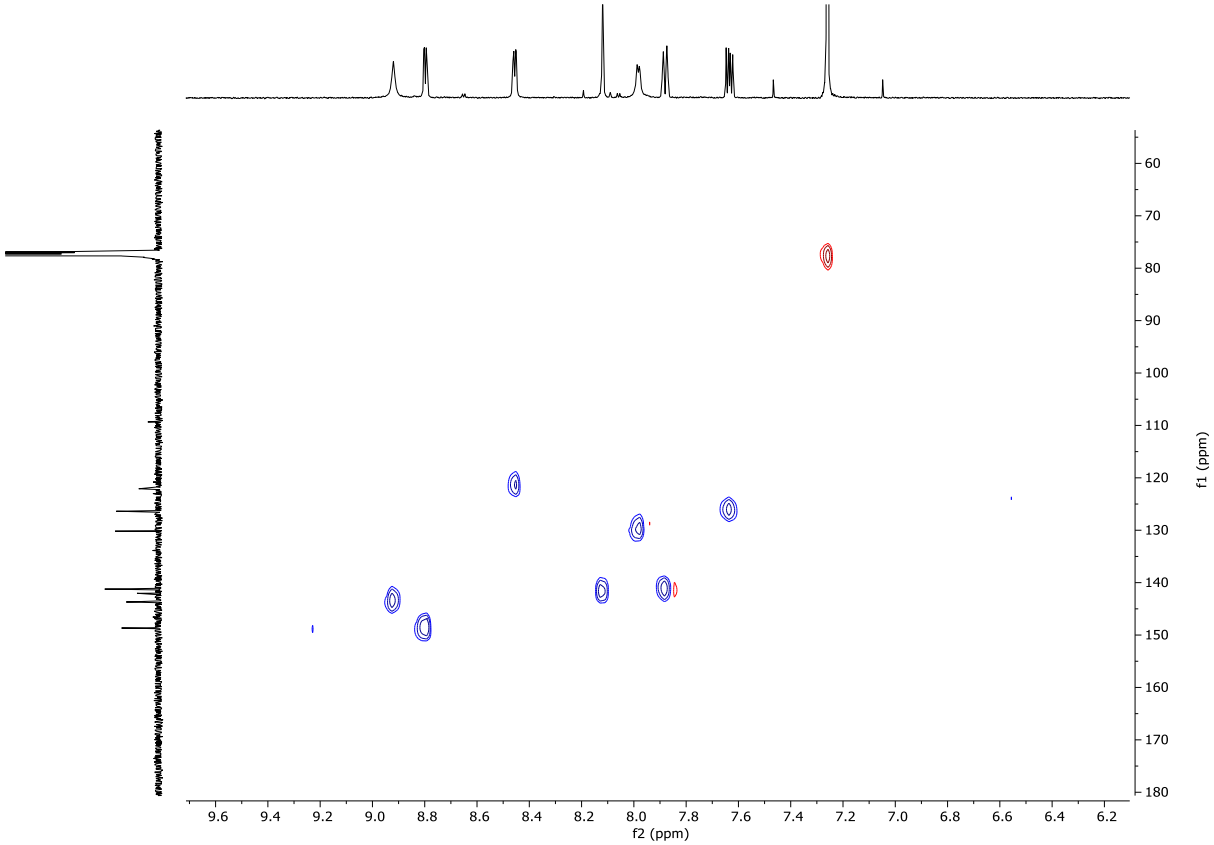
^{1}H-NMR (500MHz, $CDCl_3$):



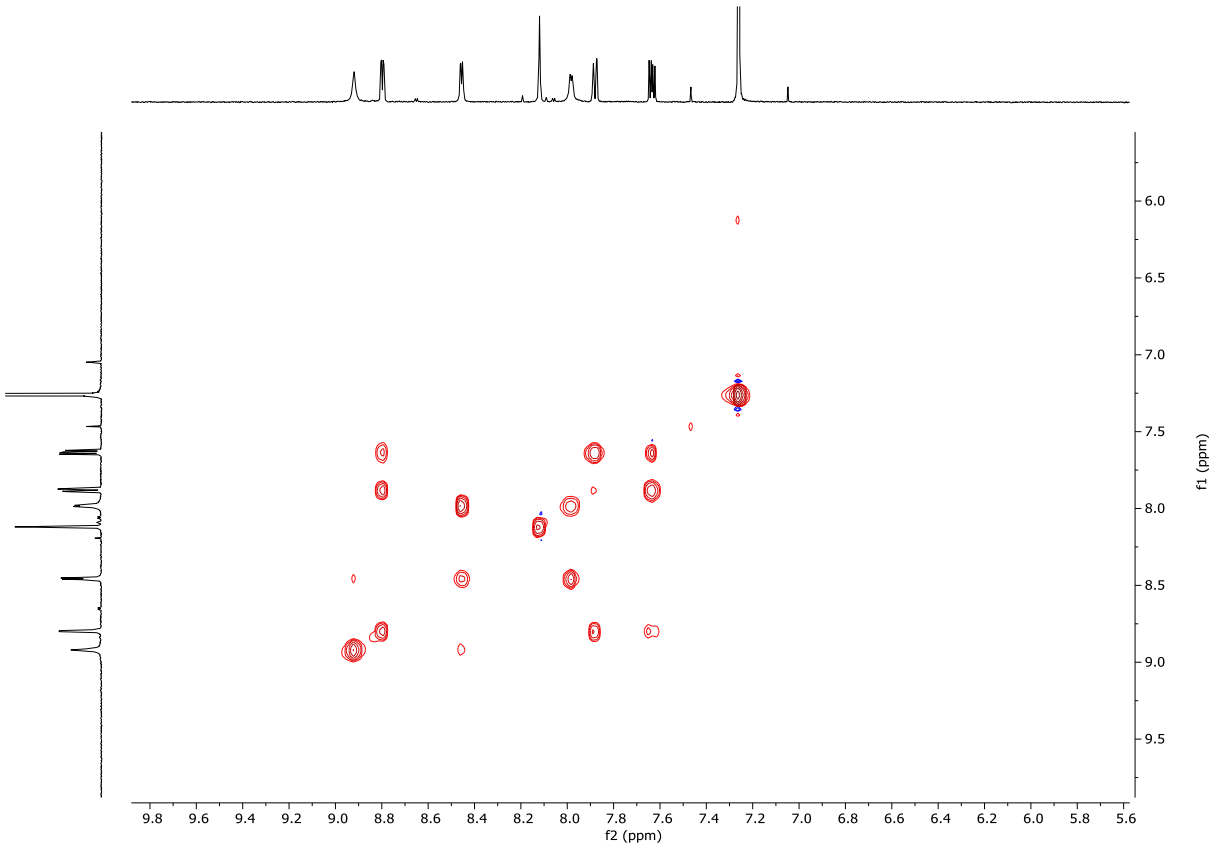
^{13}C-NMR (125MHz, $CDCl_3$):

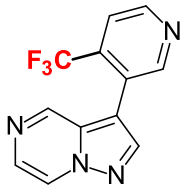


2D HSQC ($CDCl_3$):



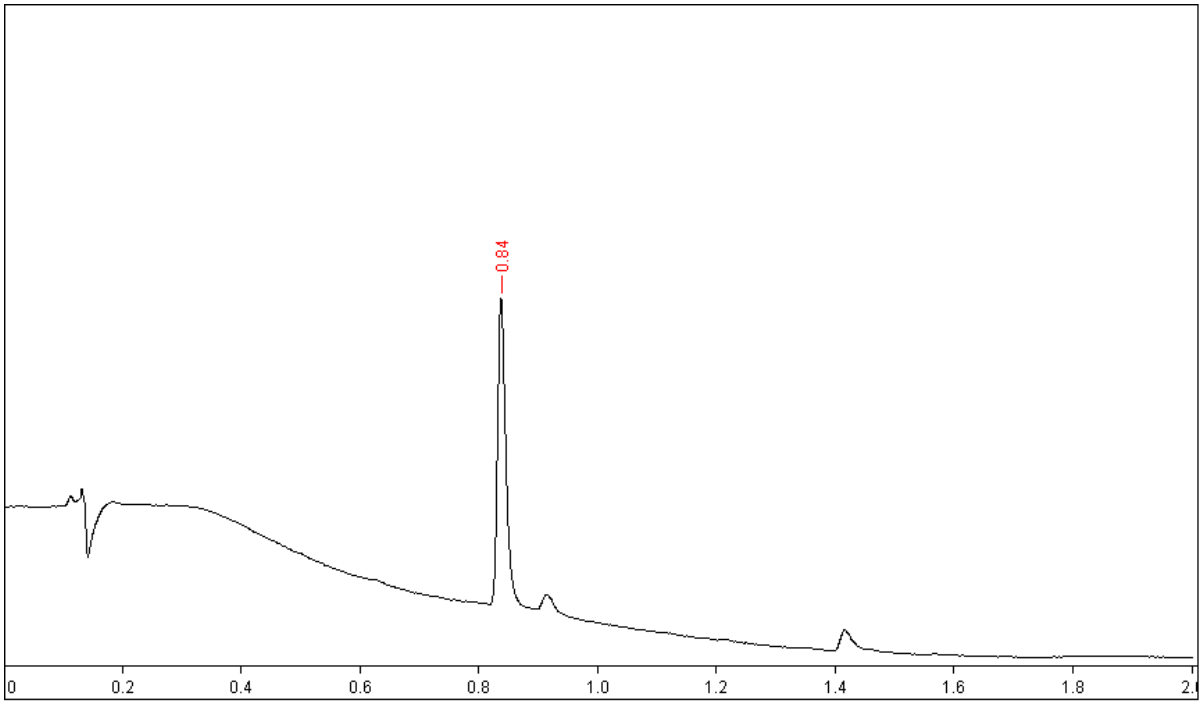
2D TOCSY ($CDCl_3$):



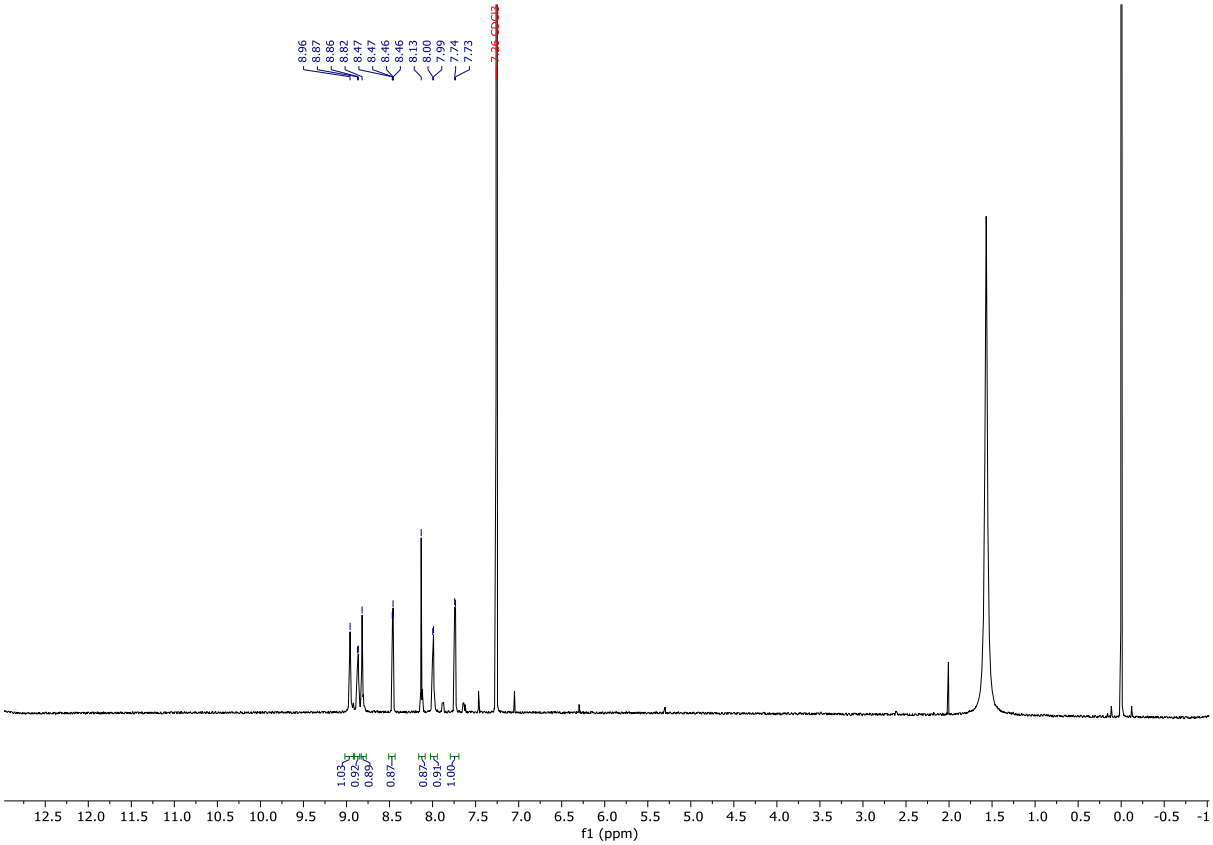


3-E2

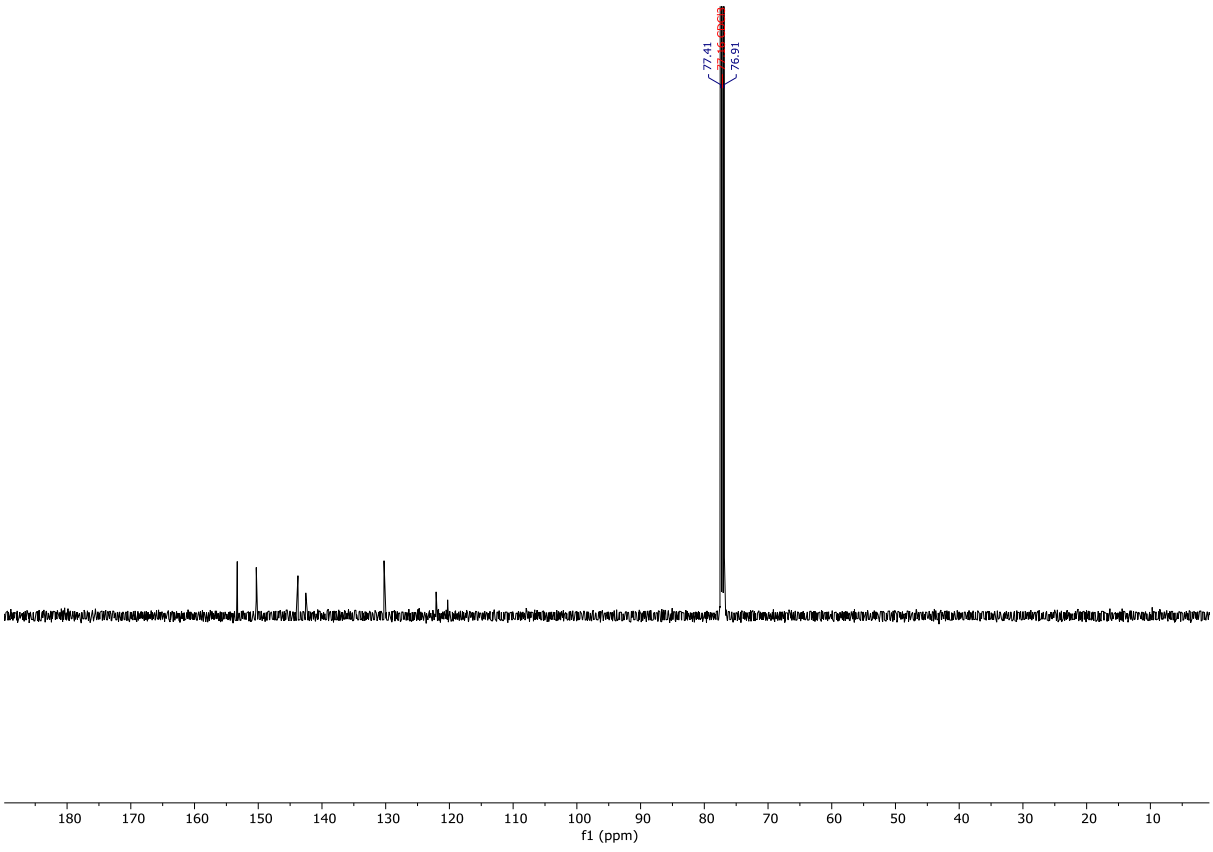
LC-MS (High pH method):



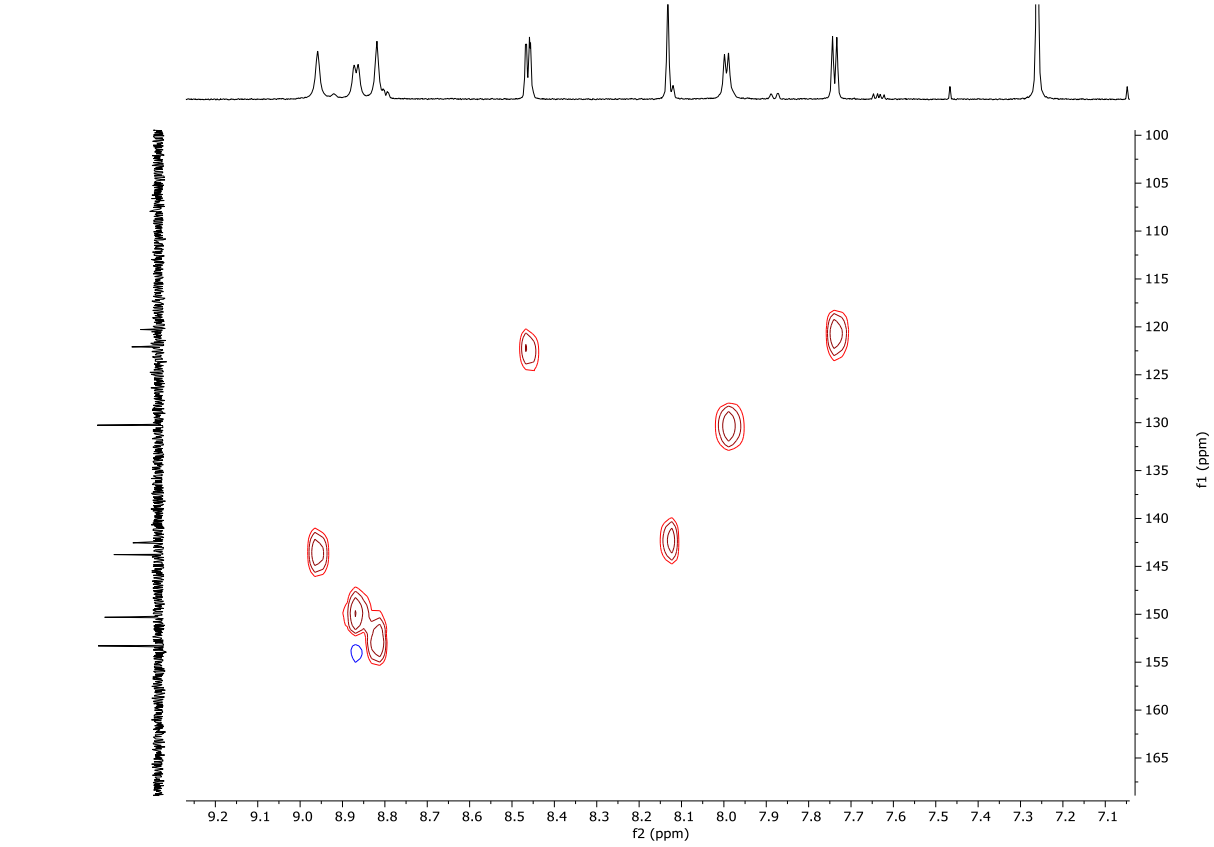
^{1}H-NMR (500MHz, $CDCl_3$):



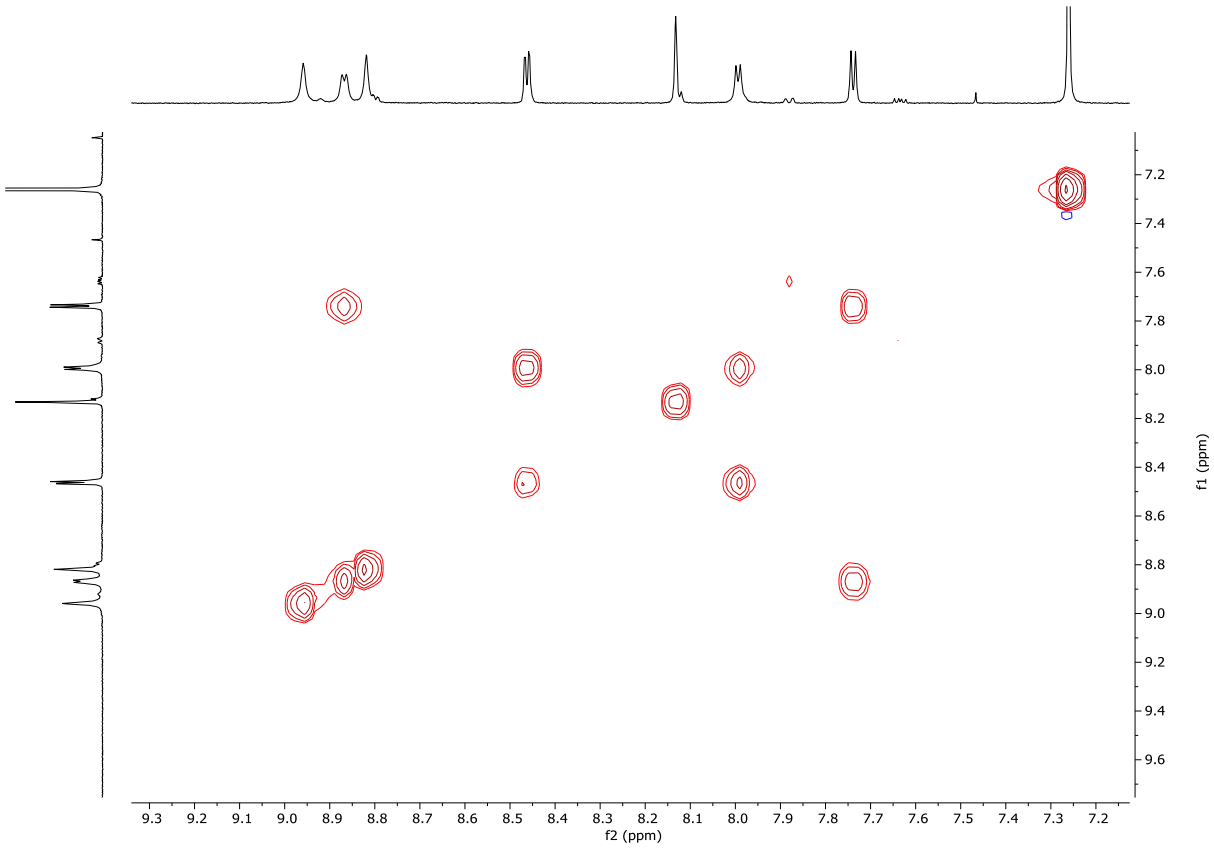
^{13}C-NMR (125MHz, $CDCl_3$):

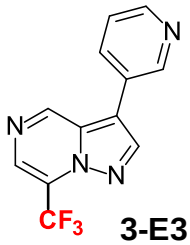


2D HSQC ($CDCl_3$):



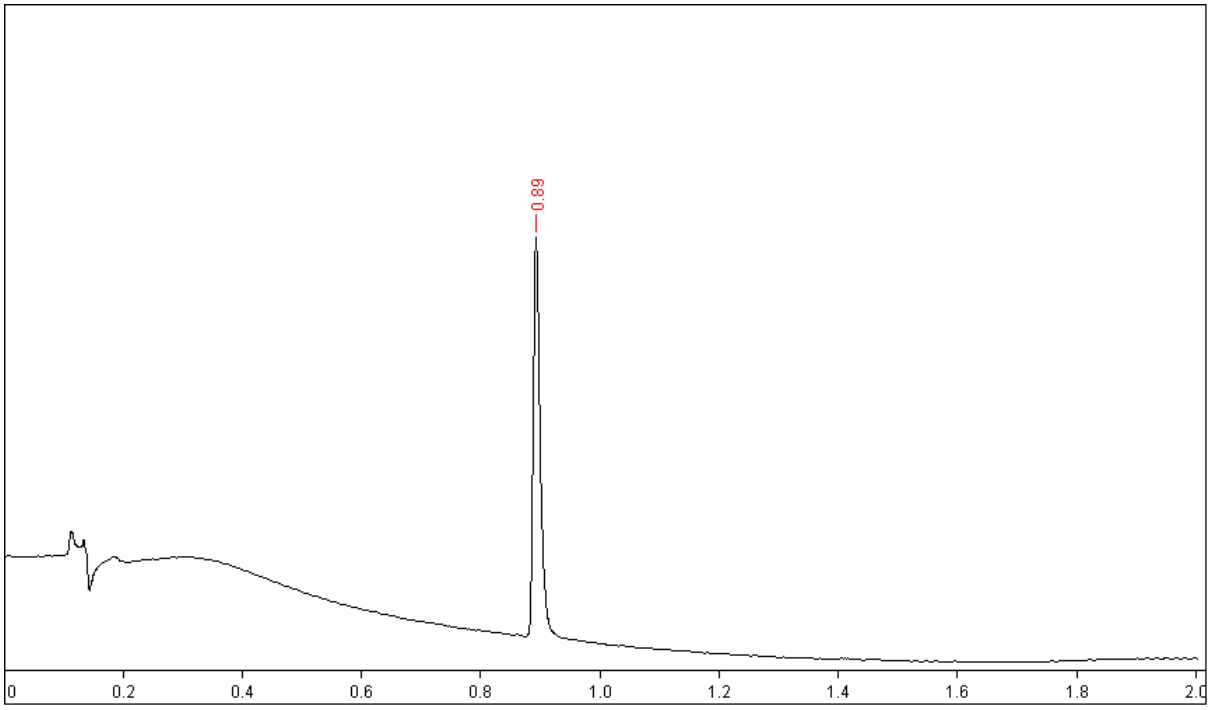
2D TOCSY ($CDCl_3$):



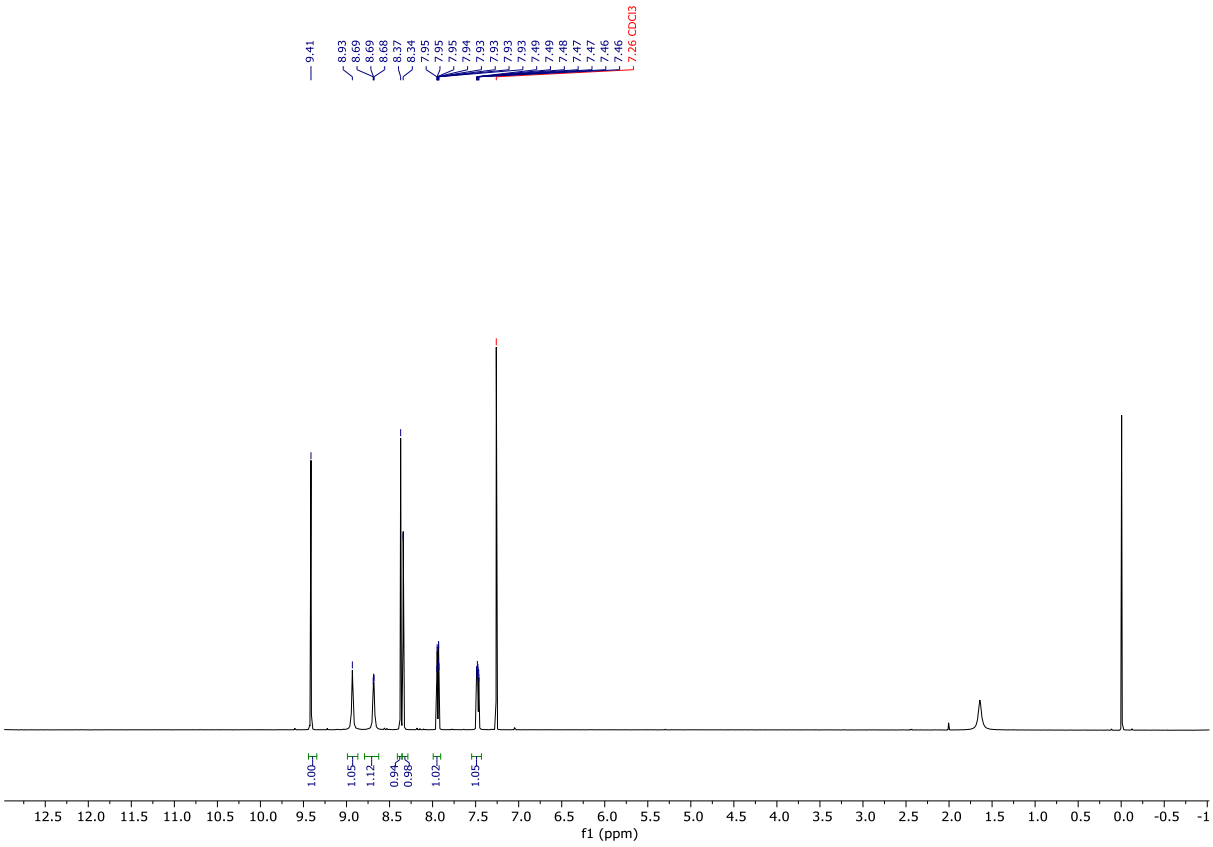


3-E3

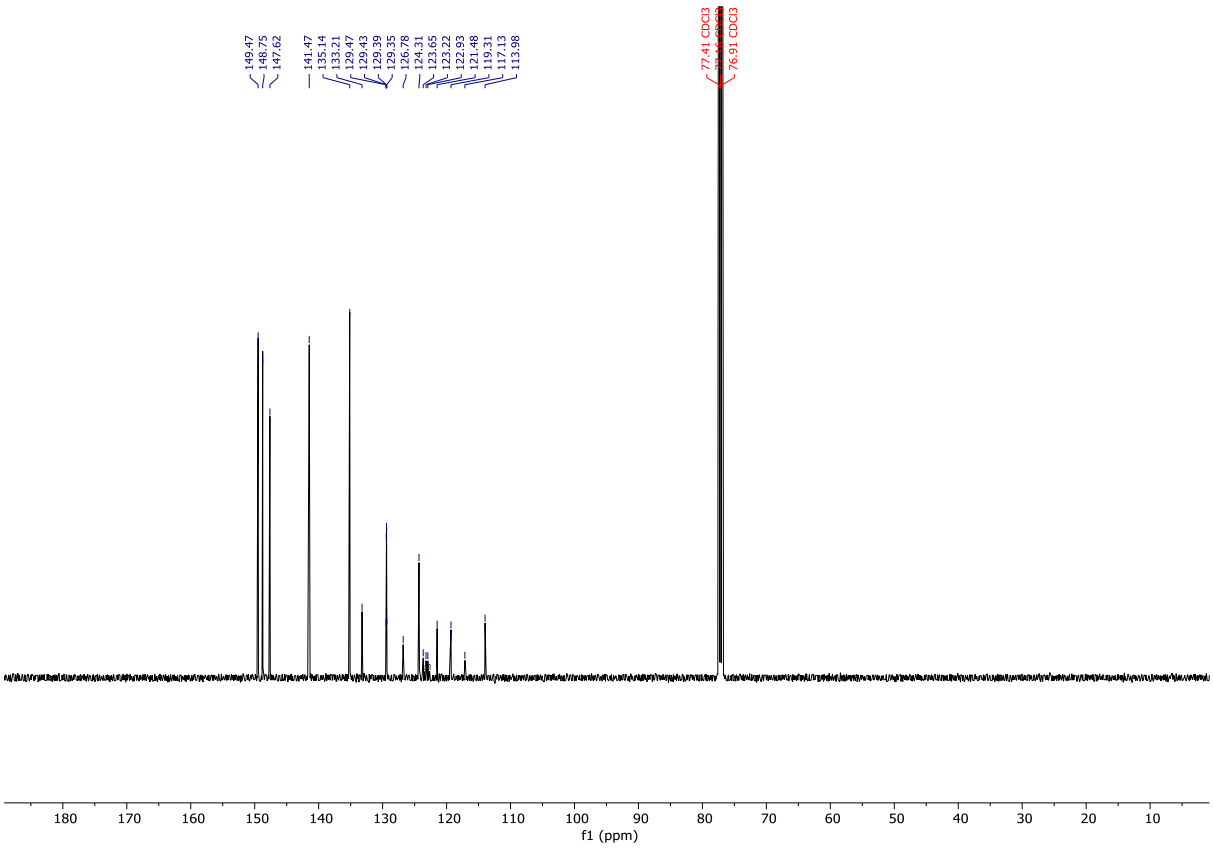
LC-MS (High pH method):



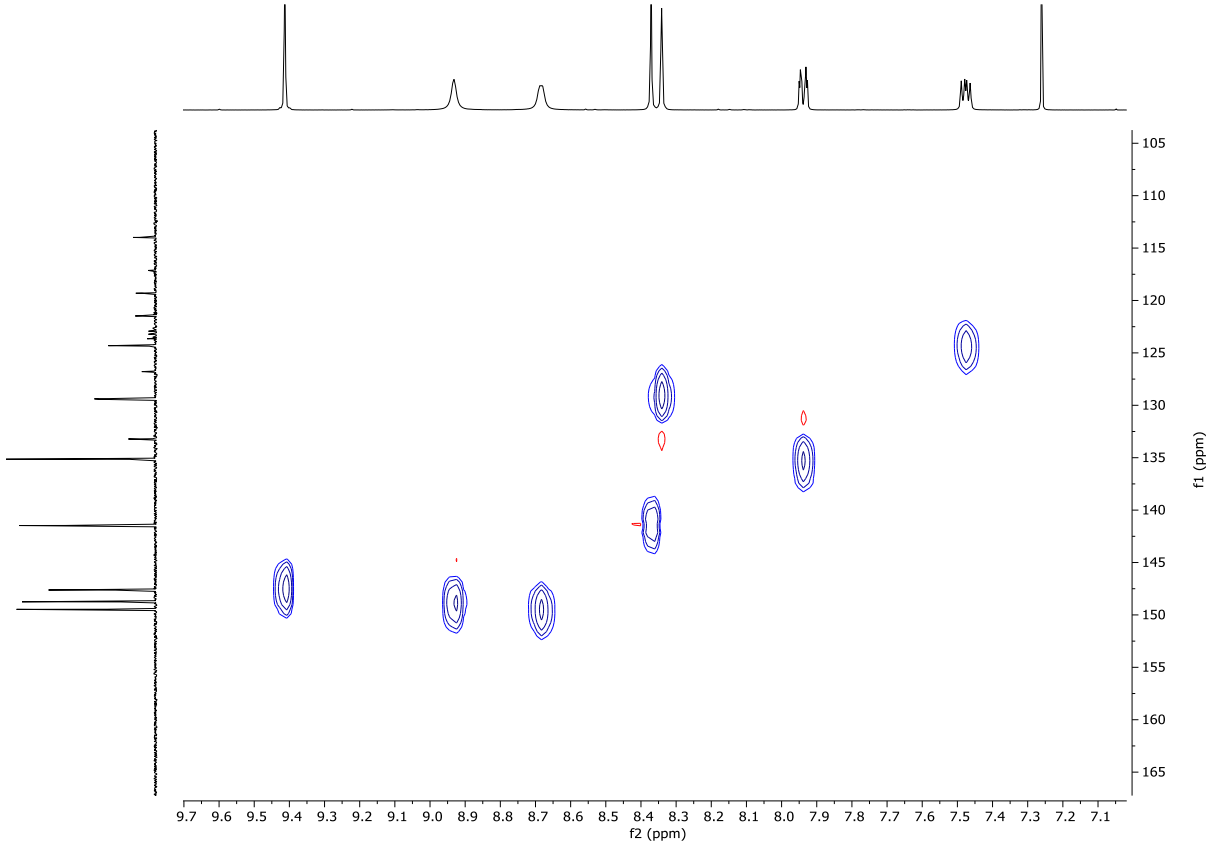
^{1}H-NMR (500MHz, $CDCl_3$):



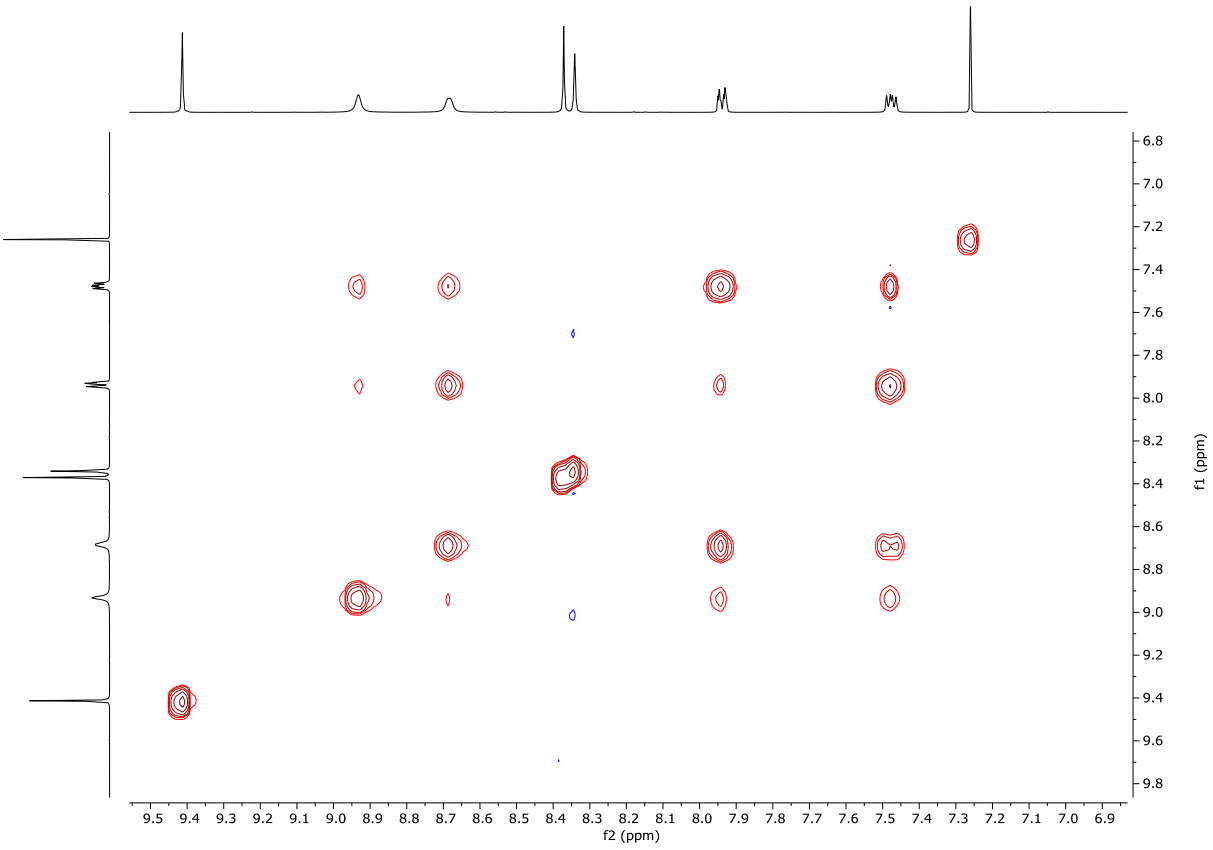
^{13}C-NMR (125MHz, $CDCl_3$):

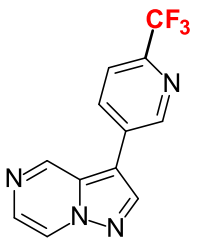


2D HSQC ($CDCl_3$):



2D TOCSY ($CDCl_3$):

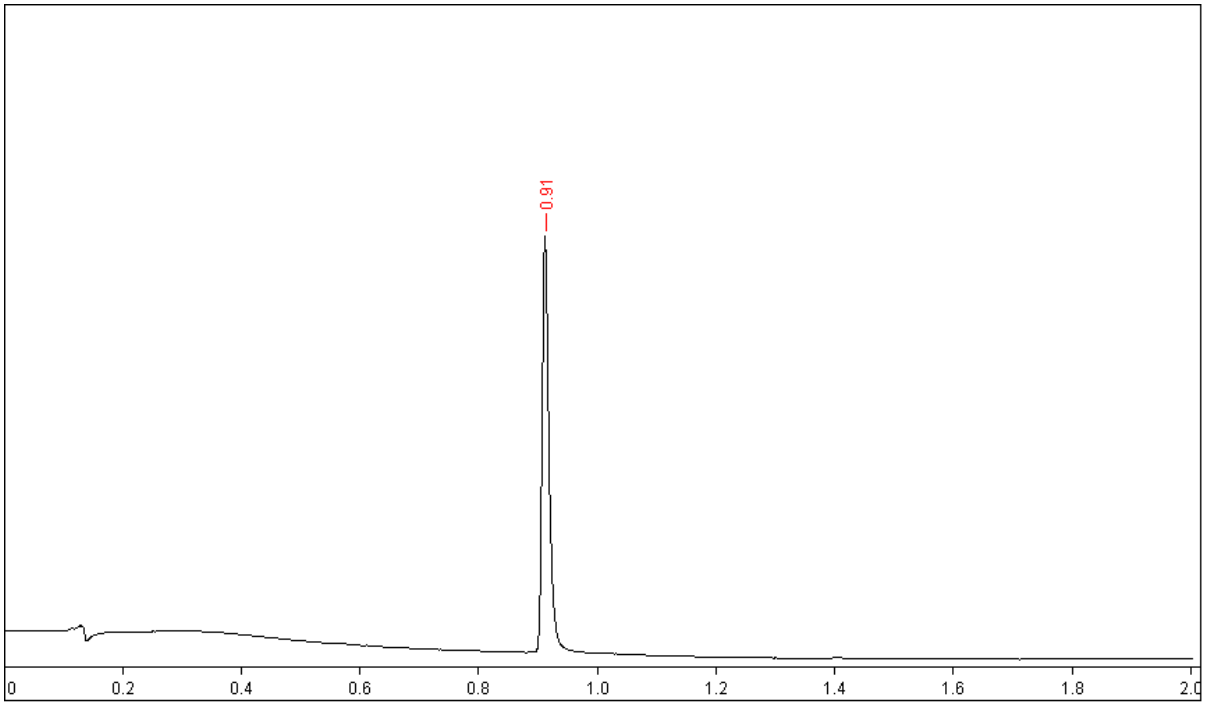




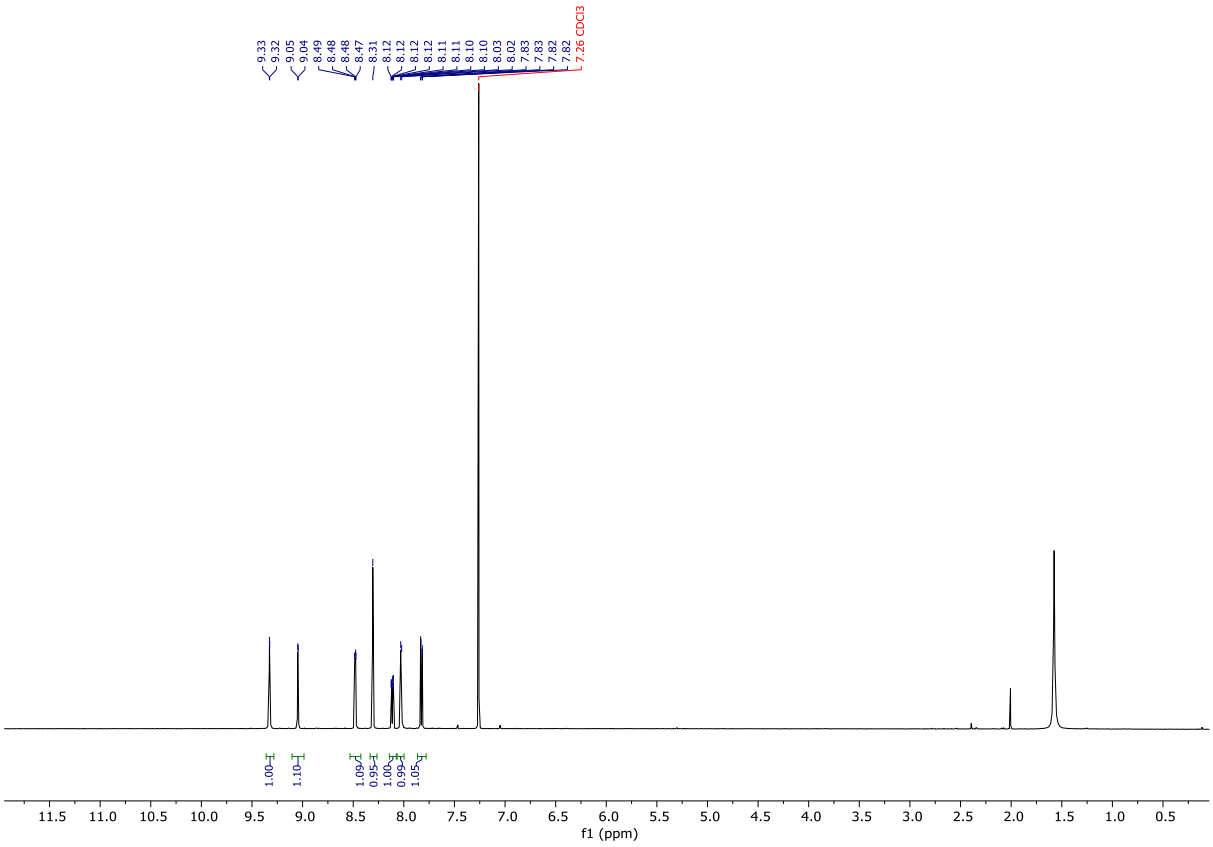
3-E4

Lot number: **60P-E18045-062-009-057**

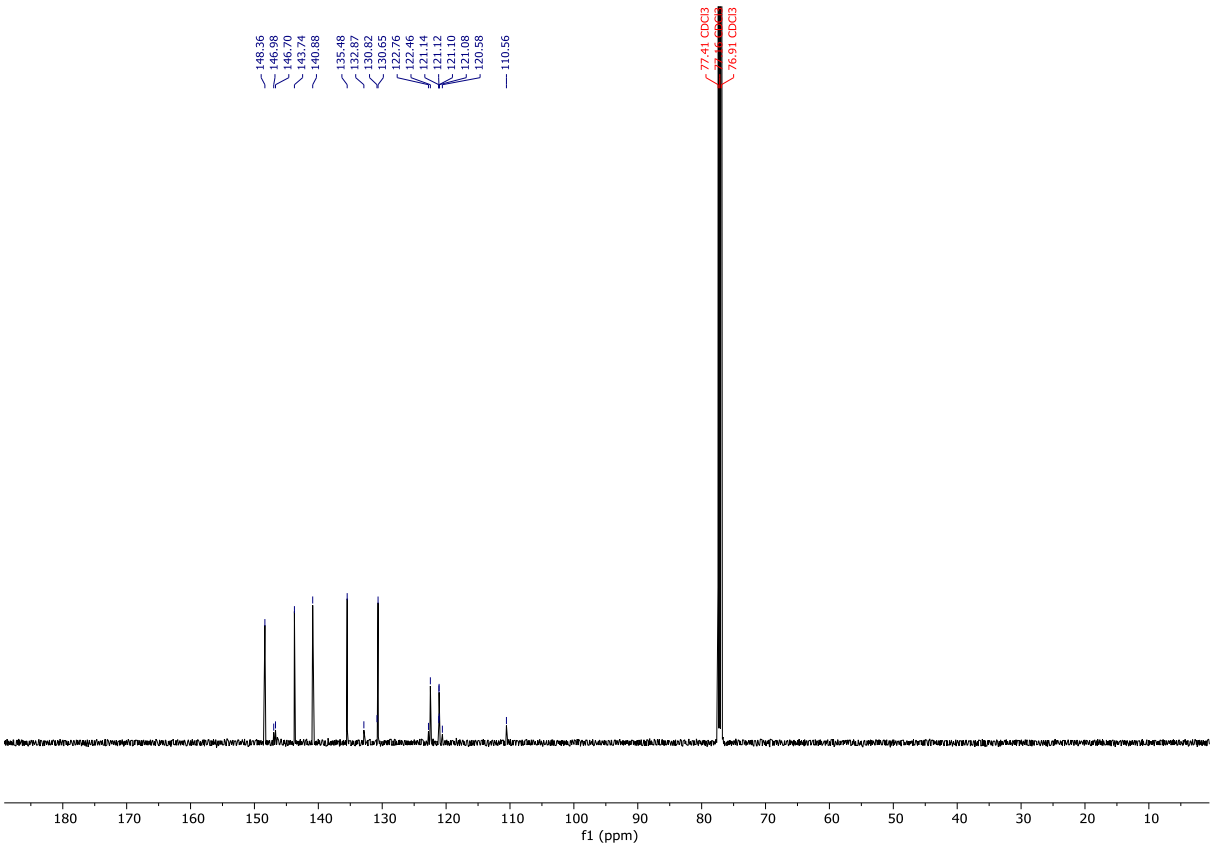
LC-MS (High pH method):



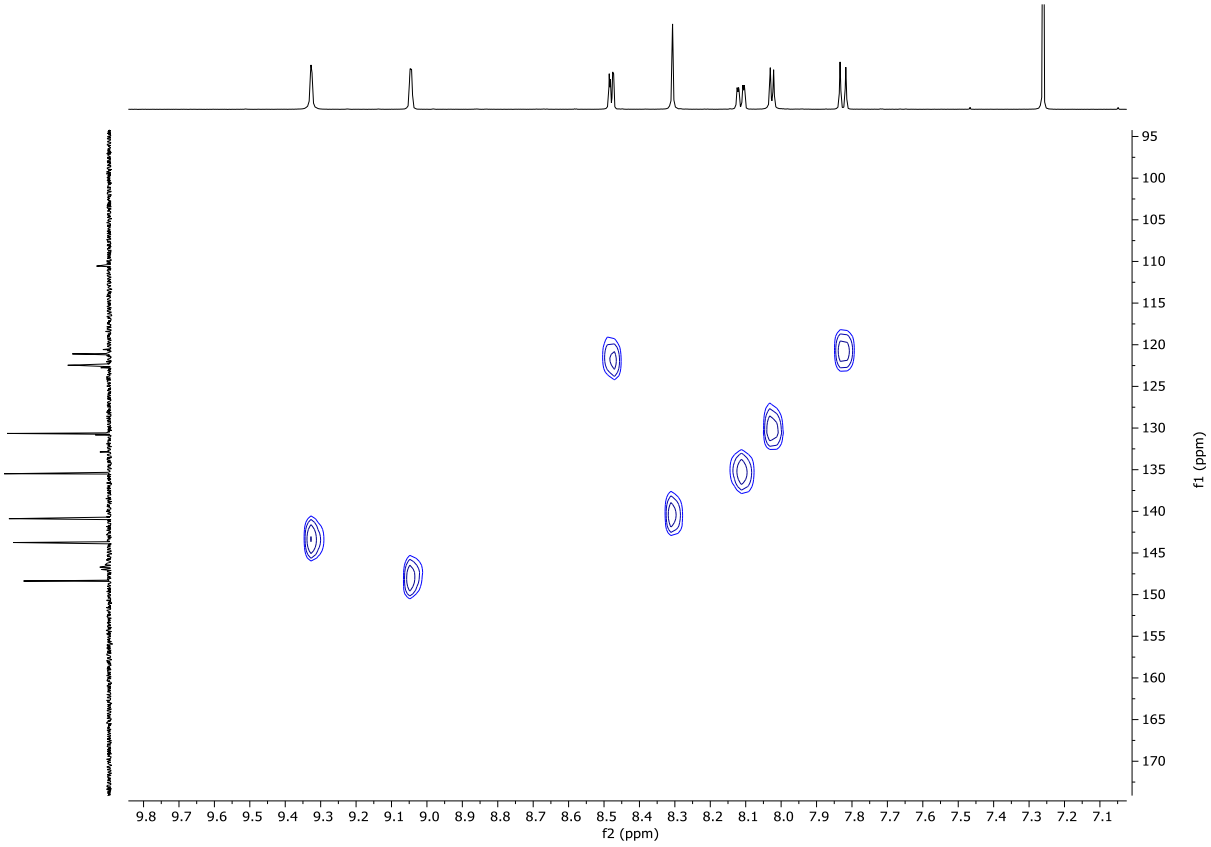
^{1}H-NMR (500MHz, $CDCl_3$):



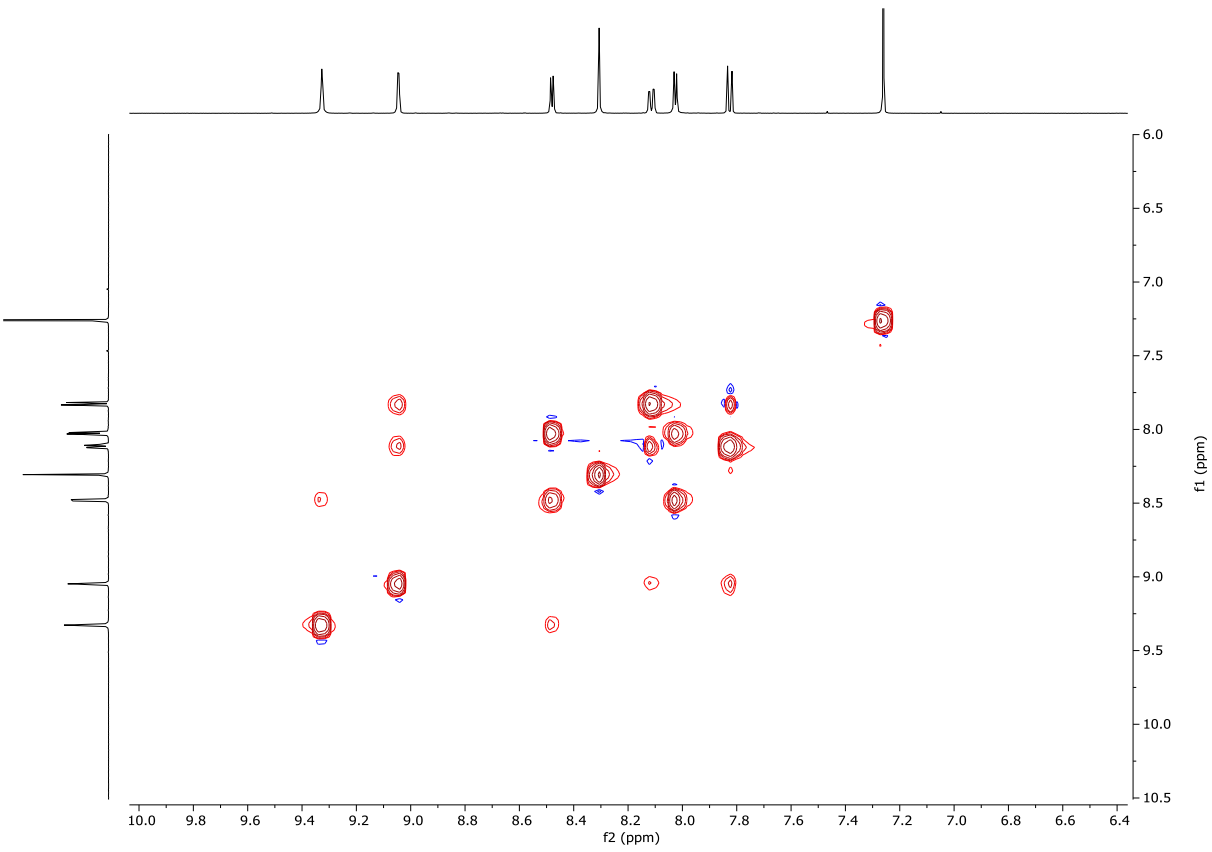
^{13}C-NMR (125MHz, $CDCl_3$):



2D HSQC ($CDCl_3$):



2D TOCSY ($CDCl_3$):



HSQC peak matching algorithm

The approach to auto assigning HSQC cross peaks is described in the main text. Details of the function used to carry this out are given below:

1. Assign the experimental peaks to calculated peaks in order, based on carbon shift.
2. If there are more calculated shifts than experimental, allow the additional calculated peaks to be removed so as to minimise the overall error:

$$\sum_{i=1}^{i=n}(calc(C)_i - experimental(C)_i)^2)$$

3. Allow the first experimental peak to change it assignment to all other calculated peaks. Calculate the error as:

$$(calc(C) - experimental(C))^2 + (calc(H) - experimental(H))^2 . {}^{\sigma_H^2}\!/_{\sigma_C^2}$$

   i.e., the Pythagorean distance between the experimental and calculated cross peak, scaled by the relative prediction errors for the two nuclei.

4. Calculate this same value for peaks 2-n by calling step 3 recursively and sum the total error at each point. If this error exceeds the best error so far, abandon this branch. That is if at the second peak we are already great than the best error summed error for all peaks, we can stop and so avoid a combinatorial explosion
5. Return the best assignment so arrived at.

The java code to implement this is reproduced below:

```
    public static NMRData convertExpHSQCToNMR(ExpData expData, NMRData calcData,
double factor) {
        int i = 0;
        int j = 0;
        int k = 0;
        int nexpShifts = expData.nShifts;
        int ncalcShifts = calcData.nShifts;
        int nLabels = expData.nLabels;

        expData.sort();
        calcData.sort();

        String[] calclabels = new String[ncalcShifts];
        String[] calcF2labels = new String[ncalcShifts];
        for (i = 0; i < ncalcShifts; i++) {
            calclabels[i] = calcData.items[i].label;
            calcF2labels[i] = calcData.items[i].labelF2;
        }

        int[] assignTemp = new int[nexpShifts];

        for (i = 0; i < nexpShifts; i++) {
            assignTemp[i] = expData.items[i].assign;
            if (assignTemp[i] == 0) {
                assignTemp[i] = 1;
            }
        }

        NMRDataItem[] items = new NMRDataItem[nexpShifts];
        //first assign nearest carbon
        System.out.println("-----------initial assignment in order-------");
        for (i = 0; i < nexpShifts; i++) //loop over the experimental shifts
        {
```

```
            items[i] = new NMRDataItem();
            items[i].shift = expData.items[i].shift;
            items[i].shiftF2 = expData.items[i].shiftF2;
            items[i].nH = expData.items[i].nH;
            items[i].label = calcData.items[i].label;
            items[i].labelF2 = calcData.items[i].labelF2;
            //System.out.println(expData.items[i].shift      +      "      "      +
calcData.items[i].shift + " " + (expData.items[i].shift - calcData.items[i].shift));
            double cdiff = expData.items[i].shift - calcData.items[i].shift;
            double hdiff = expData.items[i].shiftF2 - calcData.items[i].shiftF2;
            System.out.println(items[i].label + " "+ expData.items[i].shift + " " +
calcData.items[i].shift  +  "  "  +  cdiff  +  "  "  +  items[i].labelF2  +  "  "+
expData.items[i].shiftF2 + " " + calcData.items[i].shiftF2 + " " + hdiff);
        }


        double error = 0.0;
        for (i = 0; i < items.length; i++) //loop over the experimental shifts
        {
            error += getErrorVal(items[i], calcData.items[i], factor);
        }
        System.out.println("error: " + error);

        //any missing??
        int diff = ncalcShifts - nexpShifts;
        int[] missInd = new int[diff];
        for (int c2=0;c2<diff;c2++) {
            missInd[c2] = ncalcShifts-1-c2;
        }
        MyDouble bestError = new MyDouble(error);

        for (i = 0; i < diff; i++) {
            //System.out.println("missing shift iteration "+i);
            for (j = 0; j < nexpShifts; j++) {
                boolean used = false;
                for (int c2 : missInd) {
                    if (c2 == j) {
                        used = true;
                    }
                }
                if (used) {
                    continue;
                }
                error=0;
                for (int c = 0; c < items.length; c++) //loop over the experimental
shifts
                {
                    int jj = c;
                    for (int c2=0;c2<i;c2++) {
                        if (c >= missInd[c2]) {
                            jj++;
                        }
                    }
                    if(c>=j) {
                        jj++;
                    }

                    error += getErrorVal(items[c], calcData.items[jj], factor);
                }
                if (error < bestError.d) {
                    missInd[i] = j;
                    bestError.d = error;
                }
            }
        }
        System.out.println("--------- after missing taken into account ------------
---");
        for (i = 0; i < nexpShifts; i++) //loop over the experimental shifts
```

```
        {
            int ii = i;
            for (int c2 : missInd) {
                if (i>=c2) {
                    ii++;
                }
            }
            items[i].label = calcData.items[ii].label;
            items[i].labelF2 = calcData.items[ii].labelF2;
            double cdiff = expData.items[i].shift - calcData.items[ii].shift;
            double hdiff = expData.items[i].shiftF2 - calcData.items[ii].shiftF2;
            System.out.println(items[i].label + " "+ expData.items[i].shift + " " +
calcData.items[ii].shift  +  "  "  +  cdiff  +  "  "  +  items[i].labelF2  +  "  "+
expData.items[i].shiftF2 + " " + calcData.items[ii].shiftF2 + " " + hdiff);
        }
        System.out.println("error: " + bestError);
        NMRData myNMRData = new NMRData(items, expData.filename);
        //now try other combinations, stop if error worse than this
        NMRPossibility nmrp = new NMRPossibility();
        NMRPossibility bestnmrp = new NMRPossibility();
        Counter count = new Counter(0);
        doPerms(0,  0.0,  bestError,  nmrp,  bestnmrp,  myNMRData,  calcData,  factor,
count);
        System.out.println("-------do perms done: " + count.count + " iterations---
-----------");
        //System.out.println(bestnmrp.toString());
        for (i = 0; i < bestnmrp.pairs.size(); i++) {
            myNMRData.items[i] = bestnmrp.pairs.get(i).exp;
            myNMRData.items[i].label = bestnmrp.pairs.get(i).calc.label;
            myNMRData.items[i].labelF2 = bestnmrp.pairs.get(i).calc.labelF2;
            double       cdiff       =       bestnmrp.pairs.get(i).exp.shift       -
bestnmrp.pairs.get(i).calc.shift;
            double      hdiff      =      bestnmrp.pairs.get(i).exp.shiftF2      -
bestnmrp.pairs.get(i).calc.shiftF2;
            System.out.println(myNMRData.items[i].label        +        "        "+
bestnmrp.pairs.get(i).exp.shift  +  "  "  +  bestnmrp.pairs.get(i).calc.shift  +  "  "  +
cdiff + " " + myNMRData.items[i].labelF2 + " "+ bestnmrp.pairs.get(i).exp.shiftF2 +
" " + bestnmrp.pairs.get(i).calc.shiftF2 + " " + hdiff);

        }
        System.out.println("error: " + bestError);
        return myNMRData;
    }
```

TOCSY spin system counting algorithm

The algorithm we developed to count spin system size was implemented as a javascript which was executed from within Mestrelabs' MNova software. The implementation for the algorithm is as follows:

```
function tocsysss() {
      //select active spectrum
      var toc = nmr.activeSpectrum();
      var p = toc.peaks();
      print("peaks number: " + p.count)
      var spinSystems = new Array();
      //loop over peaks
      for(var i=0;i<p.count;i++) {
            //check if thi speak already in a spin system
            id = getGroupID(spinSystems,p.at(i))
            if(id==-1) {
                  //new spin system
                  var ss = new sSystem();
                  ss.shifts.push(p.at(i).delta(2));
                  print("added " + p.at(i).delta(2) + " to " + id)
```

```
                if(Math.abs(p.at(i).delta(1)-p.at(i).delta(2))>=tol) {
                      ss.shifts.push(p.at(i).delta(1));
                      print("added " + p.at(i).delta(1) + " to " + id)
                }
                spinSystems.push(ss);
          } else {
                //add to exisiting
                ss = spinSystems[id];
                addF2=true;
                addF1=true;
                for(j=0;j<ss.shifts.length;j++) {
                      if(Math.abs(ss.shifts[j]-p.at(i).delta(2))<tol) {
                            addF2=false;
                      }
                      if(Math.abs(ss.shifts[j]-p.at(i).delta(1))<tol) {
                            addF1=false;
                      }
                }
                if(addF2) {
                      ss.shifts.push(p.at(i).delta(2));
                }
                if(addF1) {
                      ss.shifts.push(p.at(i).delta(1));
                }
          }
    }
    //merge spin systems to make sure we don't have overlapping ones
    var fss = new Array();

    for (i=0;i<spinSystems.length;i++) {
          for(j=0;j<spinSystems[i].shifts.length;j++) {
                for (i2=i+1;i2<spinSystems.length;i2++) {
                      for(j2=0;j2<spinSystems[i2].shifts.length;j2++) {
                            if(Math.abs(spinSystems[i].shifts[j]-
                      spinSystems[i2].shifts[j2])<tol) {
                            print("merging!")
                            mergeSS(spinSystems[i],spinSystems[i2])
                            break;
                            }
                      }
                }
          }
    }
    //remove any empty spin systems
    for (i=0;i<spinSystems.length;i++) {
          if(spinSystems[i].shifts.length>0) {
                fss.push(spinSystems[i]);
          }
    }
    spinSystems=fss;
    print ("there are " + spinSystems.length + " spins systems");
    for(i=0;i<spinSystems.length;i++) {
          print("-------- " + i + " ------------")
          b = spinSystems[i];
          print("size = " + b.shifts.length)
          for(j=0;j<b.shifts.length;j++) {
                print("shift " + b.shifts[j]);
          }
    }
}
```

```
function mergeSS(ss1, ss2) {
	for(i=0;i<ss2.shifts.length;i++) {
		addme=true;
		for(j=0;j<ss1.shifts.length;j++) {
			if(Math.abs(ss2.shifts[i]-ss1.shifts[j])<tol) {
				addme=false;
			}
		}
		if(addme) {
			ss1.shifts.push(ss2.shifts[i]);
		}
	}
	ss2.shifts={}
}

function getGroupID(arr, peak) {
	//test if a new peak is already a member of a spin system. If so
return its ID. Return -1 if not found.
	tol = 0.01;
	for(i=0;i<arr.length;i++) {
		b = arr[i];
		for(j=0;j<b.shifts.length;j++) {
			if(Math.abs(peak.delta(1)-b.shifts[j])<tol ||
			Math.abs(peak.delta(2)-b.shifts[j])<tol) {
				return i;
			}
		}
	}
	return -1;
}

function sSystem() {
	//spin system object is simply an array of chemical shifts
	this.shifts= new Array();
}
```

Combination of probabilities

The contents of the knime java snippet used to combine the probabilities from DP4 output are listed below:

```
class DP4Combiner {

    ArrayList<Structure> structures;
    ArrayList<ExpResult> expResults;
    ArrayList<Possibility> possibilities;
    String[] strnames;
    String[] enames;
    int nStructures = 5;
    int nExps = 5;
    double rtProb=0.95;
    double[][] probs;

    /**
     * @param args the command line arguments
     */

     public DP4Combiner() {
       structures = new ArrayList<>();
       expResults= new ArrayList<>();
       possibilities = new ArrayList<>();
     }
```

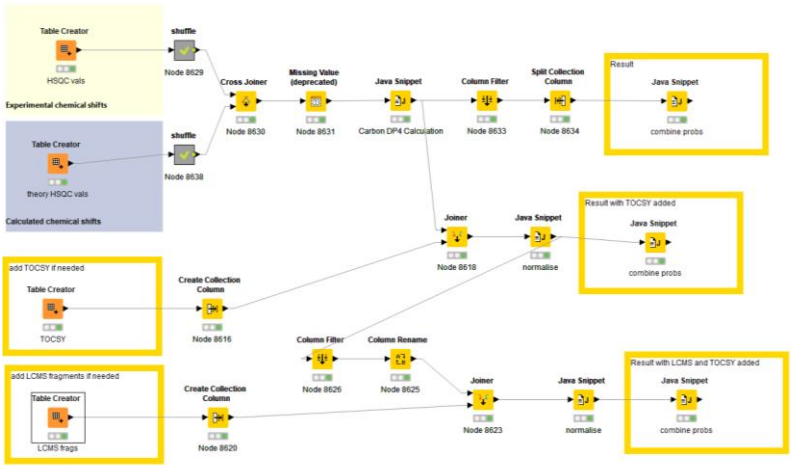
```
 public void setArray(int a, int b) {
   nExps=a;
   nStructures=b;
   probs = new double[nExps][nStructures];
 }


public String doit() {


    for (int i = 0; i < nStructures; i++) {
        Structure s1 = new Structure();
        s1.name = strnames[i];
        structures.add(s1);
    }


    for (int i = 0; i < nExps; i++) {
        ExpResult e1 = new ExpResult();
        e1.name = enames[i];
        expResults.add(e1);
        for(int j=0;j<nStructures;j++) {
            Match m = new Match();
            m.probability=probs[i][j];
            m.s = structures.get(j);
            e1.matches.add(m);
        }
    }

    doPerms(0, new Possibility());
    calcProbabilities();

    int count=0;
    Possibility best =null;
    String out="";

    for (Possibility p : possibilities) {
        String s = p.toString();

        System.out.println(s);
        count++;
        if(best == null || p.probability>best.probability) best=p;
    }
    System.out.println();
    System.out.println("there are " + count + " possibilities");
    out+="the best match is " + best.toString();
    return out;
}

public void calcProbabilities() {
    double tp = 0.0;

    for (Possibility p : possibilities) {
        double prob = 1;
        for (Pair pa : p.pairs) {

            prob = prob * pa.match.probability;
        }
        p.probability = prob;
        tp += prob;
    }

    for (Possibility p : possibilities) {
        p.probability = p.probability / tp;
    }
}
```

```
    public void doPerms(int level, Possibility p) {
        System.out.println("doPerms, level = " + level);
        ExpResult e = expResults.get(level);
        for (Match m : e.matches) {
            if (p.containsMatch(m)) {
                continue;
            }
            Possibility p2 = clonePossibility(p);
            Pair pair = new Pair();
            pair.expResult = e;
            pair.match = m;
            p2.pairs.add(pair);
            if (level < expResults.size() - 1) {
                doPerms(level + 1, p2);
            } else {
                possibilities.add(p2);
            }
        }

    }

    public Possibility clonePossibility(Possibility p) {
        Possibility p2 = new Possibility();
        for (Pair pa : p.pairs) {
            p2.pairs.add(pa);
        }
        return p2;
    }

}

class ExpResult {

    ArrayList<Match> matches;
    String name;

    ExpResult() {
        matches = new ArrayList<>();

    }
}

class Match {

    double probability;
    Structure s;
}

class Structure {

    String name;

}

class Possibility {

    ArrayList<Pair> pairs;
    double probability;

    Possibility() {
        pairs = new ArrayList<>();
    }

    public boolean containsMatch(Match m) {
        for (Pair p : pairs) {
            if (p.match.s.equals(m.s)) {
                return true;
            }
```

```
        }
        return false;
    }

    public String toString() {
        String s = "";
        NumberFormat nf = NumberFormat.getInstance();
        nf.setMinimumFractionDigits(4);
        nf.setMinimumFractionDigits(4);
            for (Pair pa : pairs) {
                s = s + pa.expResult.name + "(" + pa.match.s.name + ")-";
            }
            s = s + " " + nf.format(probability);
            return s;
    }
}

class Pair {

    Match match;
    ExpResult expResult;


}
DP4Combiner dp4c;

// expression start
    public void snippet() throws TypeException, ColumnException, Abort {
// Enter your code here:
    if(ROWINDEX==0) {
              dp4c = new DP4Combiner();
              dp4c.setArray(ROWCOUNT,c_probs.length);
              dp4c.strnames=c_strnames;
              dp4c.enames=new String[ROWCOUNT];
    }
    for(int i=0;i<c_probs.length;i++) {
              double d= Double.parseDouble(c_probs[i]);
              dp4c.probs[ROWINDEX][i]=d;

    }
    dp4c.enames[ROWINDEX]=c_ExpName;
    if(ROWINDEX==ROWCOUNT-1) {

       out_out = dp4c.doit();
    }




// expression end
```

Reusable version of the workflow

The workflow presented in the main part of the paper contains some proprietary nodes and some commercial ones requiring a license. In order to facilitate use of the workflow by others, we have produced a version which requires no extensions to KNIME and will run with the standard package which is available free from the knime website (https://www.knime.com/) under an open license.



A simplified workflow which allows execution of the workflow without proprietary or licensed nodes.

This workflow differs from the one we use in that the calculated chemical shifts and experimental ones need to be entered manually. Likewise, all possible structures need to be entered manually. These changes were made for this version of the workflow because the original version, outlined in the manuscript, relies on proprietary nodes and licences which would mean that this implementation would likely not work for most users. An interested user is of course free to replace the manually created table nodes with any script that produces the output in the same format from software they have available to them. The rest of the probability calculations, for pure NMR shift data or for additional data such as TOCSY or LCMS fragments as described in the main body of the paper, are identical to that we have used.

The workflow is provided as a KNIME archive file which can be imported into KNIME via 'file, import KNIME workflow' menu items. To use the workflow, you need only enter data for calculated and experimental NMR, and optionally TOCSY and LCMS fragments. Examples for some 'dummy' data are included in the workflow to demonstrate the format used. They are also reproduced here. General instructions on using KNIME can be found in the application documentation and numerous online help fora.

Experimental data

Experimental data for all isomers is entered in one table, with one line per HSQC cross peak. One only needs to enter the ^{13}C and ^{1}H shifts along with the number of protons this represents. This information is usually available from edited HSQC experiments and is only used to ensure CH's are not assigned to CH2's. If you do not have this information entering a 1 in all cases will work. The final column is a label which identifies the sample it pertains to, e.g.:

Dialog - 0:8628 - Table Creator (HSQC vals)

File

Table Creator Settings | Flow Variables | Job Manager Selection | Memory Policy

Input line:

	D C Shift	D H Shift	I nH	S ExpName
Row0	150	8.5	1	lot ref 1
Row1	38.89	4.36	3	lot ref 1
Row2	110	7.1	1	lot ref 1
Row3	140	8.0	1	lot ref 2
Row4	30	3.0	3	lot ref 2
Row5	120	7.5	1	lot ref 2
Row6				
Row7				

Calculated data

The calculated shifts are entered in essentially the same format. The name now refers to the structural possibility. In the final output it will be matched to the experimental sample name.

Dialog - 0:8637 - Table Creator (theory HSQC vals)

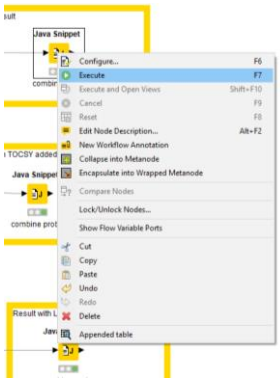
File

Table Creator Settings | Flow Variables | Job Manager Selection | Memory Policy

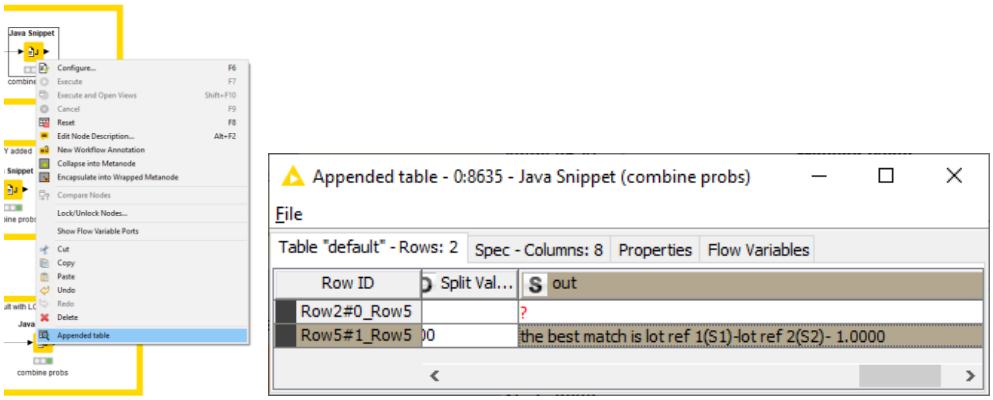
Input line:

	D CalcC	D calcH	I nH	S name
Row0	148	8.4	1	S1
Row1	40.0	4.4	3	S1
Row2	108	7.0	1	S1
Row3	138	7.9	1	S2
Row4	31	3.1	3	S2
Row5	119	7.55	1	S2
Row6				
Row7				

Once this data is entered, executing the workflow involves right clicking and choosing ‘execute’ on the ‘combine probs’ java snippet.



After this, choosing ‘appended table’ by right clicking the same node shows the results. The overall determination is shown in the final column

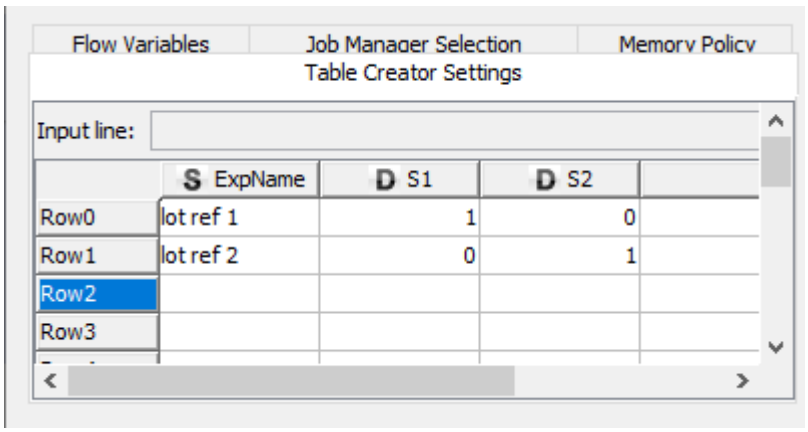


And is of the format:

the best match is lot ref 1(S1)-lot ref 2(S2)- 1.0000

i.e., the overall best match is shown, showing which supplied sample label goes which structure label along with the overall probability.

If the probabilities are not sufficiently high to make a clear determination based on HSQC data alone, one may enter TOCSY spin system size or LCMS fragmentation data as described in the main text. The format of the data entry in this workflow is shown below:



i.e, the name of each experimental sample, which much match that used in the HSQC experimental table, along with either a 1 or a 0 depending on whether the observed experimental data matches or does not match the proposed structure. The name of the proposed structure appears in the column header and this name must match that in the calculated NMR data table. For TOCSY data, match means the size of the spin systems based on the candidate structure matches that of the experimental data in question. For LCMS data, match means the fragmentation ions predicted for the candidate structure match that of the experimental data.

Molecular modelling and Ab initio calculation details

All molecular modelling was carried out using the Schrodinger suite of software, version 2017-4. Candidate structures were converted to 3D from a mol representation and then minimised, after which they were submitted to a mixed mode (Monte Carlo - low mode) conformational search, all using the OPLS-2005 forcefield with a constant dielectric and extended cut-off. The lowest energy structure resulting from this was geometry optimised using DFT theory with a 6-31G** (or LACVP if 3rd row element present) basis set. Chemical shifts were then calculated using the same basis sets; conversion of calculated shielding to chemical shifts was done within the software using the built-in regression function. It should be noted that these represent largely default settings within this software, and they were not highly optimised for this work. We considered it reassuring that even with these default settings, the approach we have described worked well, there is no doubt scope to improve the accuracy still further through their optimisation.

References

1. D. A. Nagib, D. W. C. Macmillan, Trifluoromethylation of arenes and heteroarenes by means of photoredox catalysis. *Nature*. **480**, 224–228 (2011).

2. C. C. Le *et al.*, A General Small-Scale Reactor to Enable Standardization and Acceleration of Photocatalytic Reactions. *ACS Cent. Sci.* **3**, 647–653 (2017).

3. R. A. Rodriguez *et al.*, Palauchlor: A practical and reactive chlorinating reagent. *J. Am. Chem. Soc.* **136**, 6908–6911 (2014).

4. Y. Ye, S. H. Lee, M. S. Sanford, Silver-mediated trifluoromethylation of arenes using TMSCF 3. *Org. Lett.* **13**, 5464–5467 (2011).

5. L. Li *et al.*, Simple and Clean Photoinduced Aromatic Trifluoromethylation Reaction. *J. Am. Chem. Soc.* **138**, 5809–5812 (2016).

6. G. Shi, C. Shao, S. Pan, J. Yu, Y. Zhang, Silver-Catalyzed C − H Tri fl uoromethylation of Arenes Using Tri fl uoroacetic Acid as the Tri fl uoromethylating Reagent. *Org. Lett.* **17**, 38–41 (2015).

7. S. Seo, J. B. Taylor, M. F. Greaney, Silver-catalysed trifluoromethylation of arenes at room temperature. *Chem. Commun.* **3**, 6385–6387 (2013).

8. X. Wu, L. Chu, F. Qing, PhI (OAc) 2 -mediated oxidative trifluoromethylation of arenes with CF 3 SiMe 3 under metal-free conditions. *Tetrahedron Lett.* **54**, 249–251 (2013).

9. L. Cui *et al.*, Metal-Free Direct C-H Perfluoroalkylation of Arenes and Heteroarenes Using a Photoredox Organocatalyst. *Adv. Synth. Catal.* **355**, 2203–2207 (2013).

10. Y.-D. Yang, K. Iwamoto, E. Tokunaga, N. Shibata, Transition-metal-free oxidative trifluoromethylation of unsymmetrical biaryls with trifluoromethanesulfinate. *Chem. Commun.* **49**, 5510–5512 (2013).

11. M. S. Wiehn, E. V Vinogradova, A. Togni, Electrophilic trifluoromethylation of arenes and N-heteroarenes using hypervalent iodine reagents. *J. Fluor. Chem.* **131**, 951–957 (2010).

12. E. Mejía, A. Togni, Rhenium-Catalyzed Trifluoromethylation of Arenes and Heteroarenes by Hypervalent Iodine Reagents. *ACS Catal.* **2**, 521–527 (2012).

13. P. Liu, W. Liu, C. J. Li, Catalyst-Free and Redox-Neutral Innate Trifluoromethylation and Alkylation of Aromatics Enabled by Light. *J. Am. Chem. Soc.* **139**, 14315–14321 (2017).

14. D. Wang, G. Deng, S. Chen, H. Gong, Catalyst-free direct C–H trifluoromethylation of arenes in water–acetonitrile. *Green Chem.* **18**, 5967–5970 (2016).

15. Y. Ouyang, X. Xu, F. Qing, Trifluoromethanesulfonic Anhydride as a Low-Cost and Versatile Trifluoromethylation Reagent. *Angew Chem Int Ed Engl*. **130**, 7042–7045 (2018).

16. B. R. Langlois, E. Laurent, N. Roidot, Trifluoromethylation of aromatic compounds with sodium trifluoromethanesulfinate under oxidative conditions. *Tetrahedron Lett.* **32**, 7525–7528 (1991).