Solution-phase dimerization of an oblong shape-persistent macrocycle

Supporting Information

*Meng Chu, Ashley N. Scioneaux, and C. Scott Hartley**
Department of Chemistry & Biochemistry, Miami University, Oxford, OH 45056, USA

Table of Contents

Geometry Optimizations, Frontier Molecular Orbitals, and TD-DFT Calculations

Please note that Cartesian coordinates for all calculations (including dimers and trimers) are included in a separate plain text file.

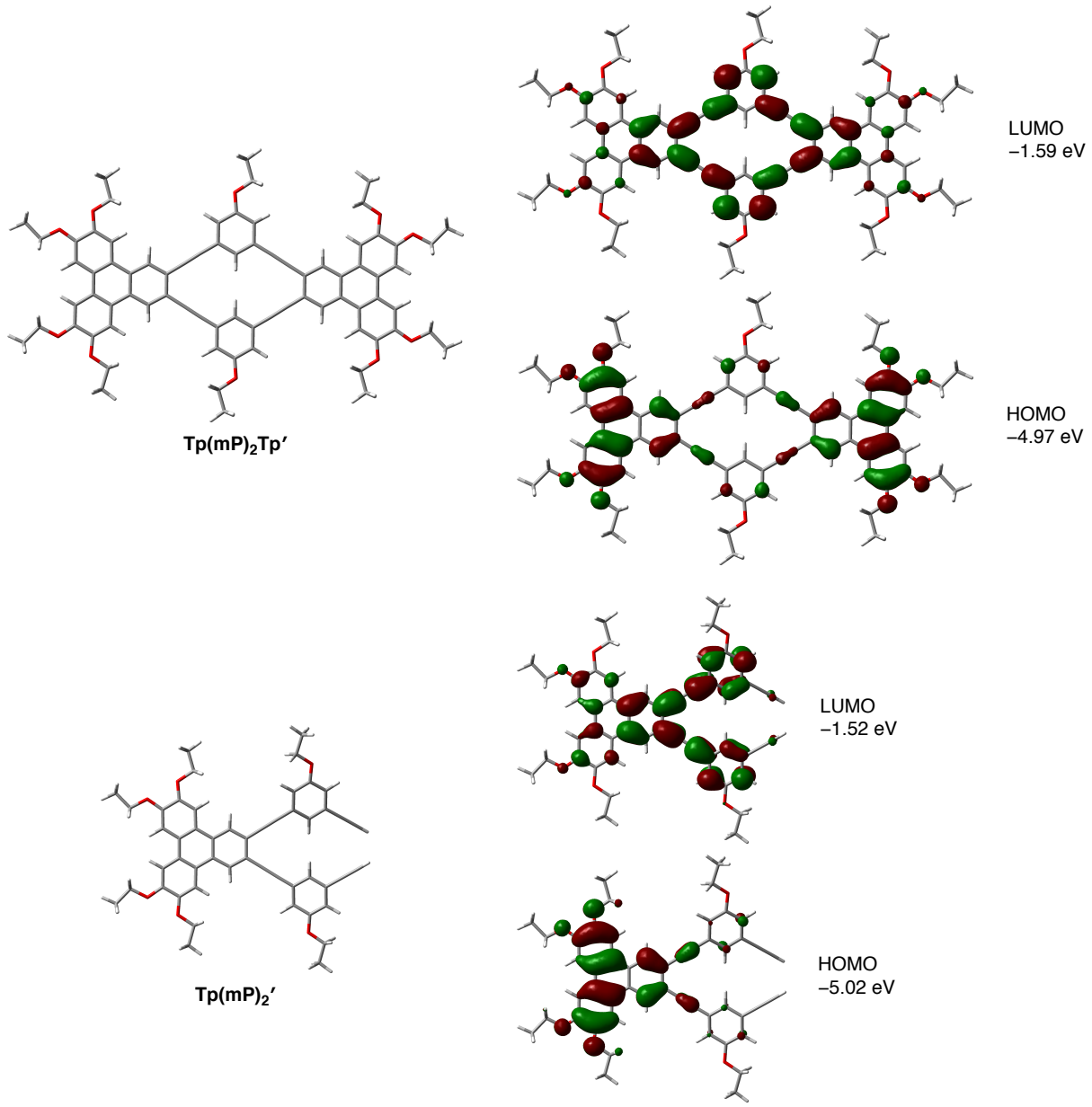

Figure S1. Frontier molecular orbitals of $\mathbf{Tp_2(mP)_2Tp'}$ and $\mathbf{Tp_2(mP)_2'}$.

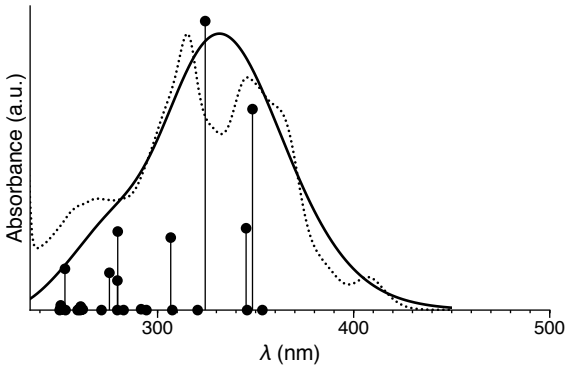

Figure S2. TD-DFT-predicted (PCM/TD/CAM-B3LYP/6-31+G(d)//B3LYP/6-31G(d)) UV–vis spectrum of **$Tp_2(mP)_2Tp'$**. The points represent the oscillator strengths of the 24 lowest-energy transitions. The predicted spectrum (solid line) was generated assuming a 0.333 eV half-width at half-height.

Structure	Imag. Freq.	Energy (E_h)	ZPC (E_h)	Total Energy (E_h)
$Tp(mP)_2Tp'$	0	−3689.116291	1.254094	−3687.862197
$Tp(mP)_2'$	0	−2382.937637	0.806538	−2382.131099
6	0	−1308.539497	0.485972	−1308.053525
7	0	−1460.828981	0.503078	−1460.325903
8	0	−1536.531917	0.484719	−1536.047198

Table S1. Energies of optimized geometries (B3LYP/6-31G(d)). Zero-point energies are unscaled.

NMR Concentration-Dependence Plots

Nonlinear Fitting to Eq 1

Python Script

The NMR chemical shifts were fit to eq 1 using a script written in Python 3 (v. 3.3.4) using the NumPy (v. 1.8.0) and SciPy (v. 0.13.3) modules.

```
""" A simple python script that uses the leastsq function from SciPy to carry out a nonlinear
regression. Outputs a bunch of different measures to be used in judging the fit. Requires an input
file with x and y columns, no headings. Pass filename as an argument.

"""
import numpy, sys
from scipy.optimize import leastsq

# Define the function here, using p as an array of parameters. Functions must be able to take arrays
# as arguments (use numpy version of exp, for example).
def func(p, x):
    return (p[0] - p[1])*(1+(1-numpy.sqrt(8*p[2]*x+1))/(4*p[2]*x)) + p[1]

# Initial guesses for the parameters.
p0 = [10, 5, 600] # Pd Pm Ke


datafile = open(sys.argv[1])
x, y = numpy.array([]), numpy.array([])
for line in datafile:
```

```
    curline = line.replace("\n", "").split() # Splits at any whitespace.
    x = numpy.append(x, float(curline[0]))
    y = numpy.append(y, float(curline[1]))

# Defines the error function for leastsq. In for regression, it is just the residuals.
def func_res(p, x, y):
    return y - func(p, x)

dof = len(x) - len(p0) # Degrees of freedom

fit_parameters, covariance_matrix, info, msg, success \
    = leastsq(func_res, p0, args=(x,y), full_output=True)

sum_squares_residuals = sum(info["fvec"]*info["fvec"])
sum_squares_mean_dev = sum((y - numpy.mean(y))**2)

# The errors for each parameter are obtained by multiplying the covariance matrix by the residual
# variance (= sum_squares_residuals / dof).
errors = []
for i in range(len(covariance_matrix)):
    errors.append(numpy.sqrt(covariance_matrix[i,i]*sum_squares_residuals/dof))

print("**Regression results for file \"{}\"**".format(sys.argv[1]))
print()

print("Data (x, y, yfit)")
print("=================")
for n in range(len(x)):
    print("{}, {}, {}".format(x[n], y[n], y[n] + info["fvec"][n]))
print()

print("Optimized parameters")
print("====================")
for n in range(len(fit_parameters)):
    print("{} +/- {}".format(fit_parameters[n], errors[n]))
print()

print("Regression data")
print("===============")

# See leastsq documentation for descriptions of flags.
print("Flag: {}".format(success))

print("Std Deviation of residuals: {}".format(numpy.sqrt(sum_squares_residuals/dof)))
print("chi2 (sum square residuals): {}".format(sum_squares_residuals))

# Ideally, (reduced chi2)/(std dev of measurement) = 1.
print("Reduced chi2 (chi2/dof): {}".format(sum_squares_residuals/dof))
print("R2 = {}".format(1 - sum_squares_residuals/sum_squares_mean_dev))
print("Adjusted R2 = {}".format(1 - (sum_squares_residuals/dof)/(sum_squares_mean_dev/(len(x)-1))))
```

```
print("Covariance matrix:")
print(covariance_matrix*sum_squares_residuals/dof)
print("Residuals:")
print(info["fvec"])
```

Sample Input File

The following is the input file for proton H_a in compound **Tp(mP)$_2$Tp**. The concentrations are in M and the chemical shifts in ppm.

```
0.0093984   8.3052
0.0046992   8.3872
0.0023496   8.4618
0.0011748   8.537
0.0005874   8.5974
0.0002937   8.6442
0.00014685  8.6771
7.3425e-5   8.6945
3.6713e-5   8.706
1.8356e-5   8.7129
```

Sample Output

The following output is obtained from the script when run on the input file above.

```
**Regression results for file "Peak1.txt"**

Data (x, y, yfit)
=================
0.0093984, 8.3052, 8.301016297739334
0.0046992, 8.3872, 8.392387766010664
0.0023496, 8.4618, 8.463269784016147
0.0011748, 8.537, 8.537880820218216
0.0005874, 8.5974, 8.59448351704852
0.0002937, 8.6442, 8.640989601932
0.00014685, 8.6771, 8.676504042039442
7.3425e-05, 8.6945, 8.693702638713068
3.6713e-05, 8.706, 8.707133752053007
1.8356e-05, 8.7129, 8.715931780186846

Optimized parameters
====================
8.051258848031216 +/- 0.01091157854594874
8.715026008346026 +/- 0.0020031447176610743
214.98860269726853 +/- 10.94585569834156

Regression data
===============
Flag: 1
Std Deviation of residuals: 0.00333022852907864
chi2 (sum square residuals): 7.763295439122498e-05
Reduced chi2 (chi2/dof): 1.1090422055889282e-05
R2 = 0.999587979883382
```

```
Adjusted R2 = 0.9994702598500625
Covariance matrix:
[[  1.19062546e-04   9.91418636e-06   1.10529116e-01]
 [  9.91418636e-06   4.01258876e-06   1.42906572e-02]
 [  1.10529116e-01   1.42906572e-02   1.19811757e+02]]
Residuals:
[-0.0041837   0.00518777  0.00146978  0.00088082 -0.00291648 -0.0032104
 -0.00059596 -0.00079736  0.00113375  0.00303178]
```

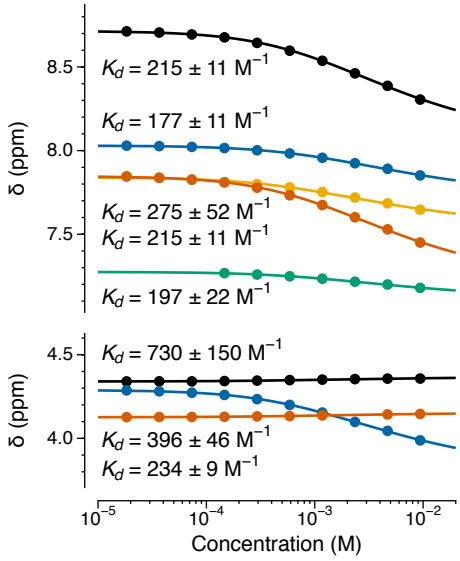

Figure S3. Concentration dependence of experimental NMR chemical shifts of **Tp(mP)$_2$Tp** in $CDCl_3$.

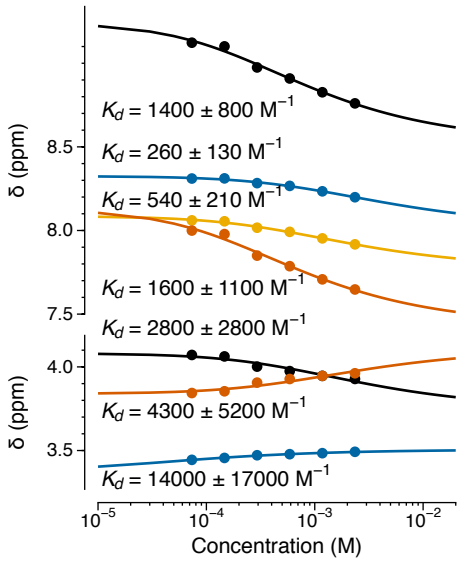

Figure S4. Concentration dependence of experimental NMR chemical shifts of **Tp(mP)$_2$Tp** in toluene-d_8.

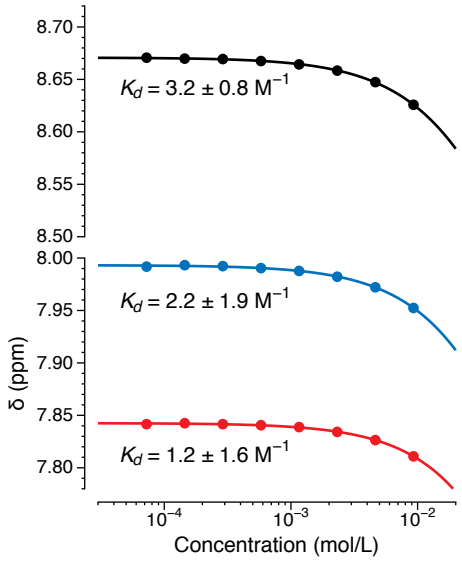

Figure S5. Concentration dependence of experimental NMR chemical shifts of **Tp(mP)$_2$** in $CDCl_3$.

Calculations of NMR Spectra

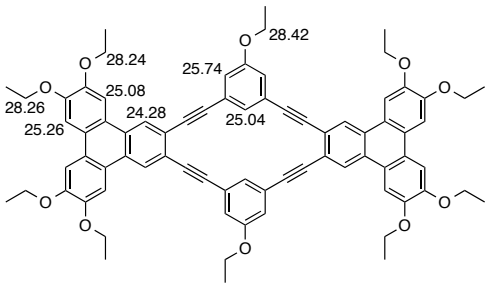

Figure S6. Calculated isotropic shieldings (PCM/WP04/6-31G(d)//B3LYP/6-31G(d)) for **Tp$_2$(mP)$_2$Tp′**.

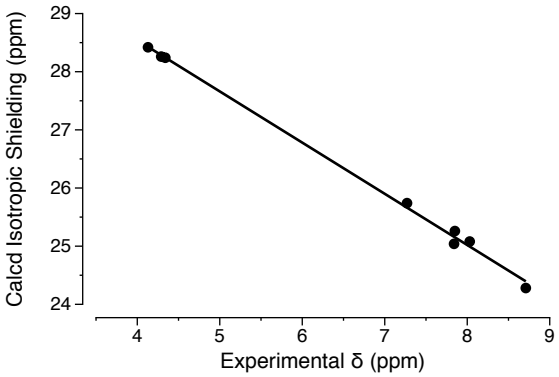

Figure S7. Calculated isotropic shieldings of **Tp$_2$(mP)$_2$Tp′** vs experimental δ_m of **Tp(mP)$_2$Tp**.

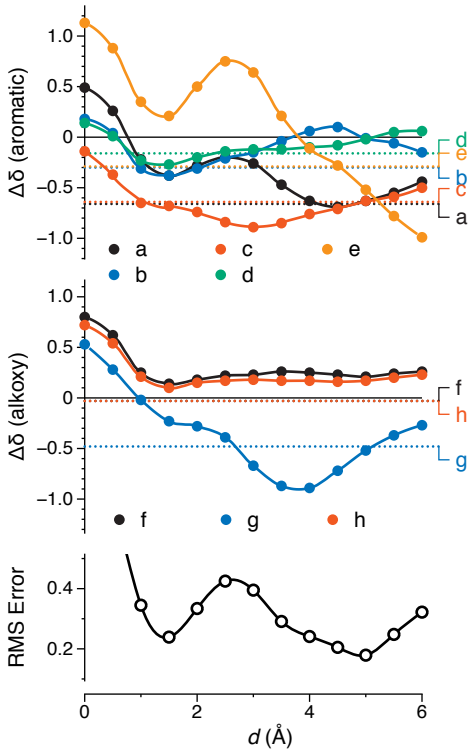

Figure S8. Comparison of $\Delta\delta_{calc}$ with d_l for an interarene separation of 3.2 Å.

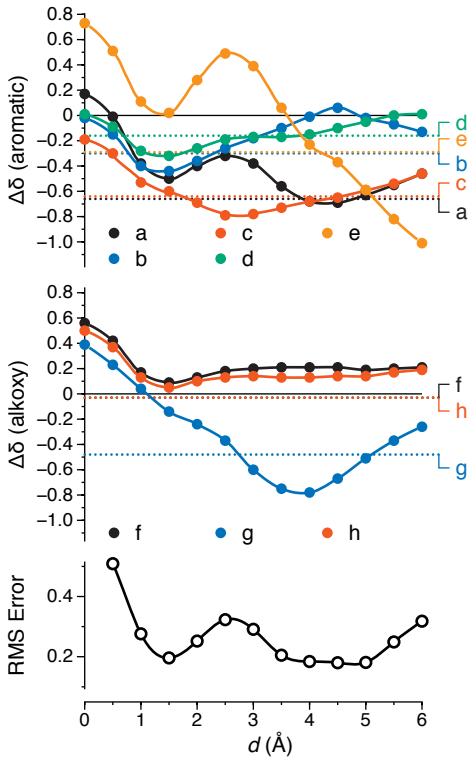

Figure S9. Comparison of $\Delta\delta_{calc}$ with d_l for an interarene separation of 3.3 Å.

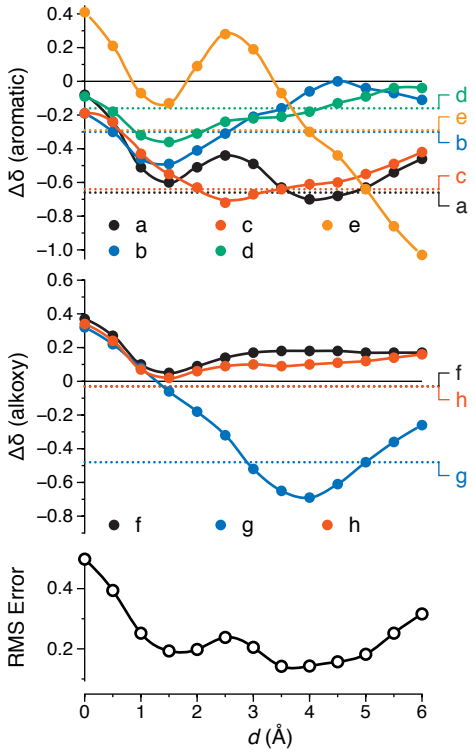

Figure S10. Comparison of $\Delta\delta_{calc}$ with d_l for an interarene separation of 3.4 Å.

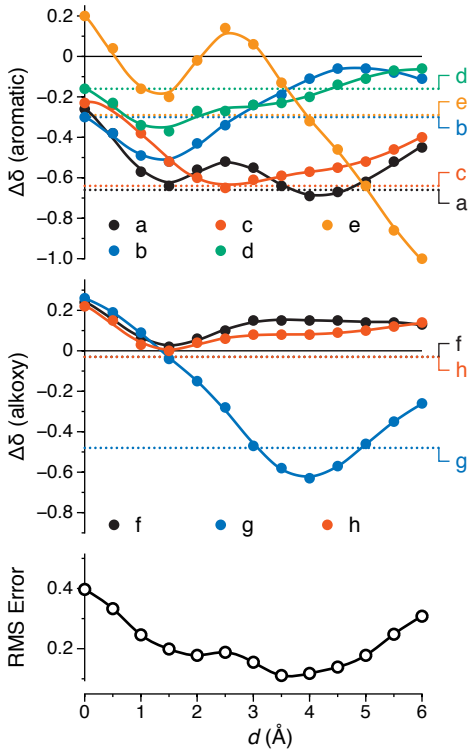

Figure S11. Comparison of $\Delta\delta_{calc}$ with d_l for an interarene separation of 3.5 Å.

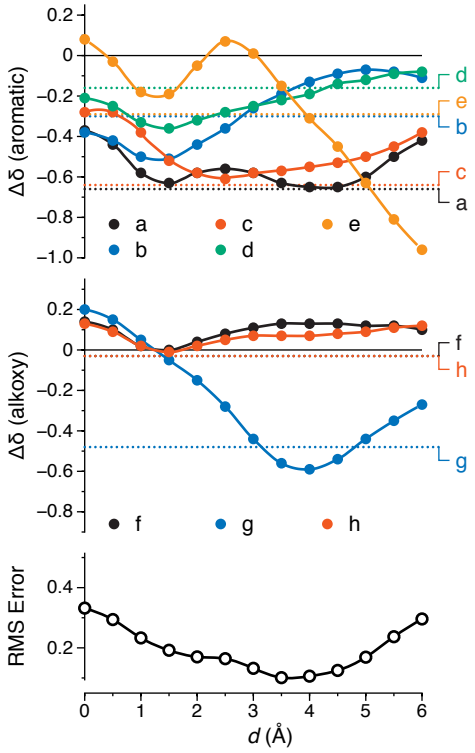

Figure S12. Comparison of $\Delta\delta_{calc}$ with d_l for an interarene separation of 3.6 Å.

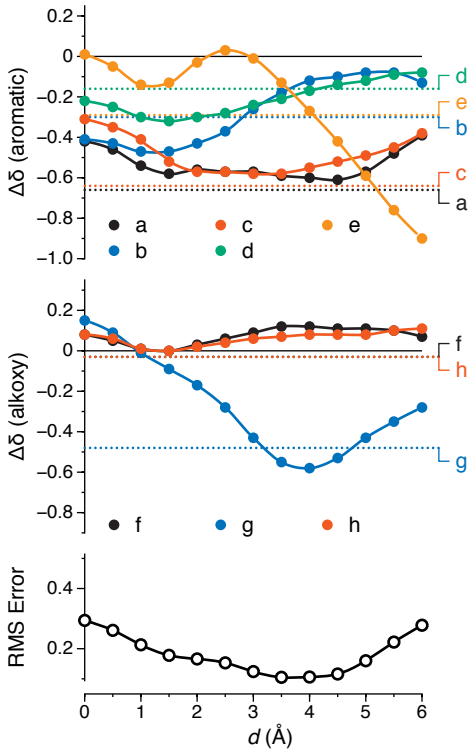

Figure S13. Comparison of $\Delta\delta_{calc}$ with d_l for an interarene separation of 3.7 Å.

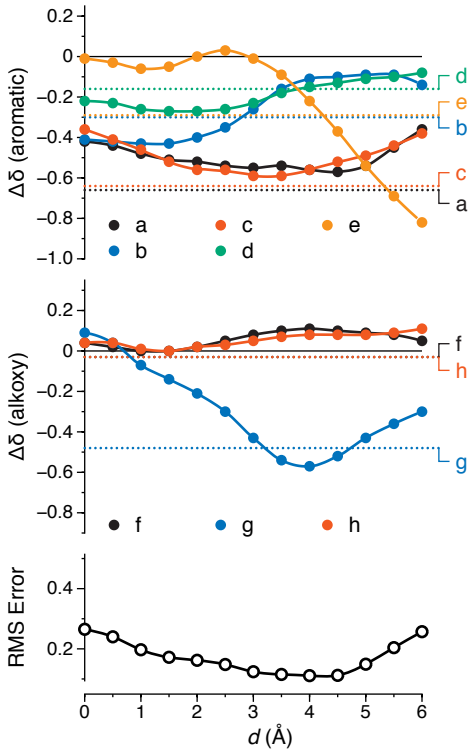

Figure S14. Comparison of $\Delta\delta_{calc}$ with d_l for an interarene separation of 3.8 Å.

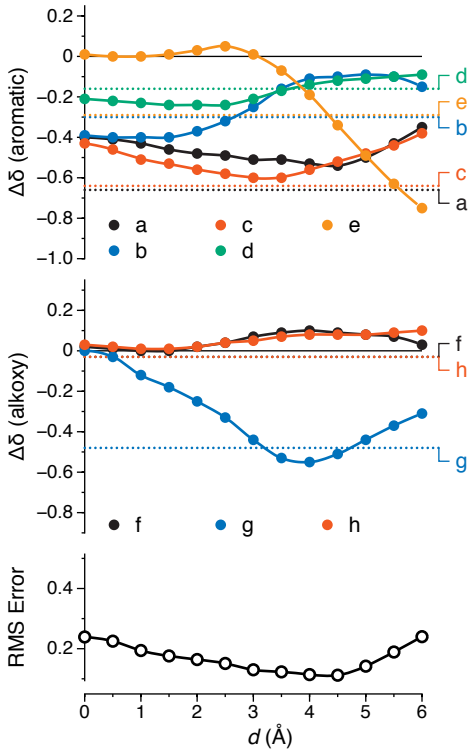

Figure S15. Comparison of $\Delta\delta_{calc}$ with d_l for an interarene separation of 3.9 Å.

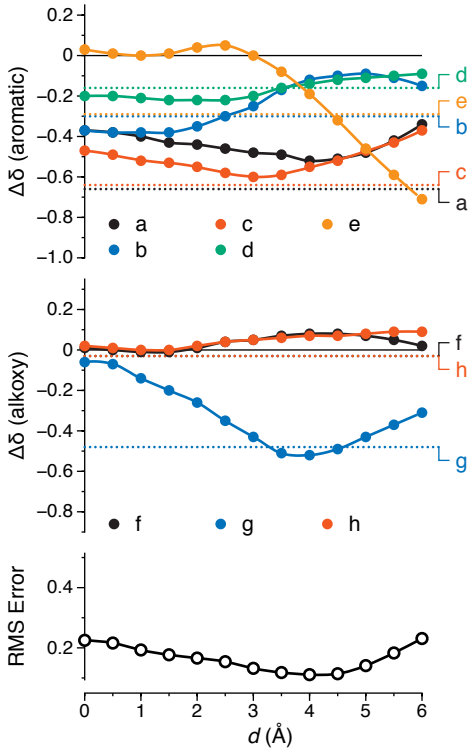

Figure S16. Comparison of $\Delta\delta_{calc}$ with d_l for an interarene separation of 4.0 Å.

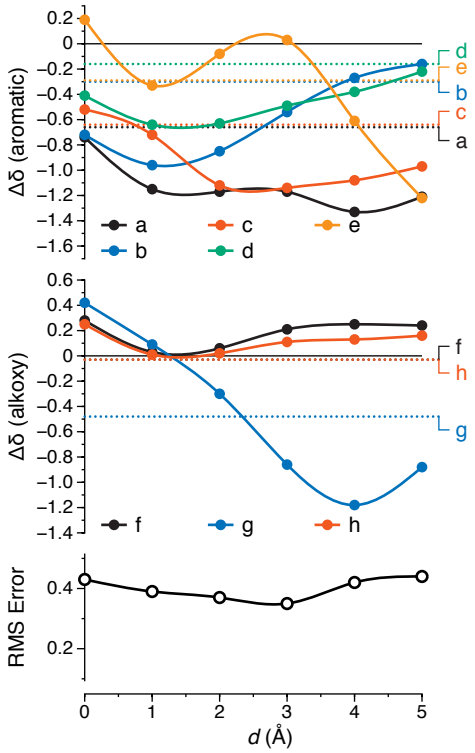

Figure S17. Comparison of $\Delta\delta_{calc}$ with d_l for a stacked trimer with interarene separations of 3.6 Å.

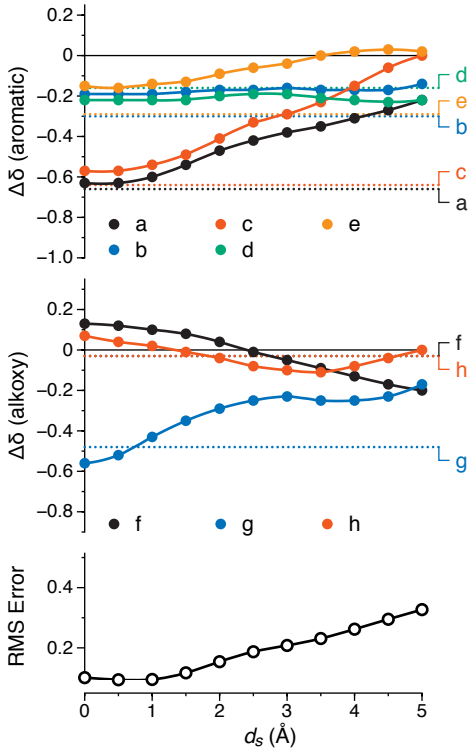

Figure S18. Comparison of $\Delta\delta_{calc}$ with d_s for a stacked dimer with an interarene separation of 3.6 Å and long axis displacement of d_l = 3.5 Å.

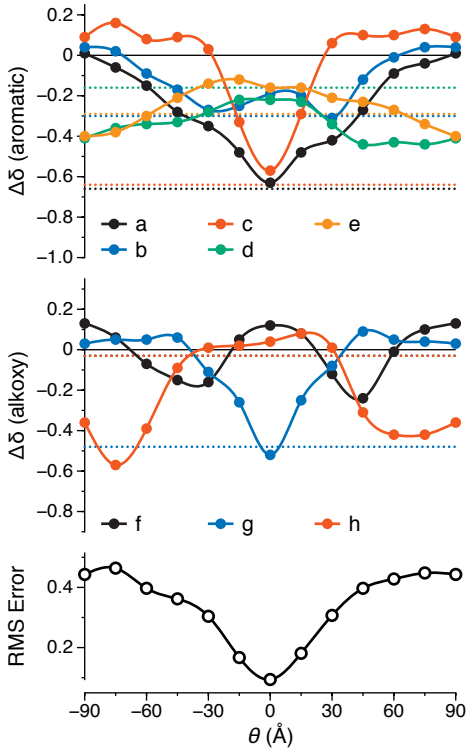

Figure S19. Comparison of $\Delta\delta_{calc}$ with θ for a stacked dimer with an interarene separation of 3.6 Å, long axis displacement of d_l = 3.5 Å, and short-axis displacement of d_s = 0.5 Å.

NMR Spectra of Synthesized Compounds

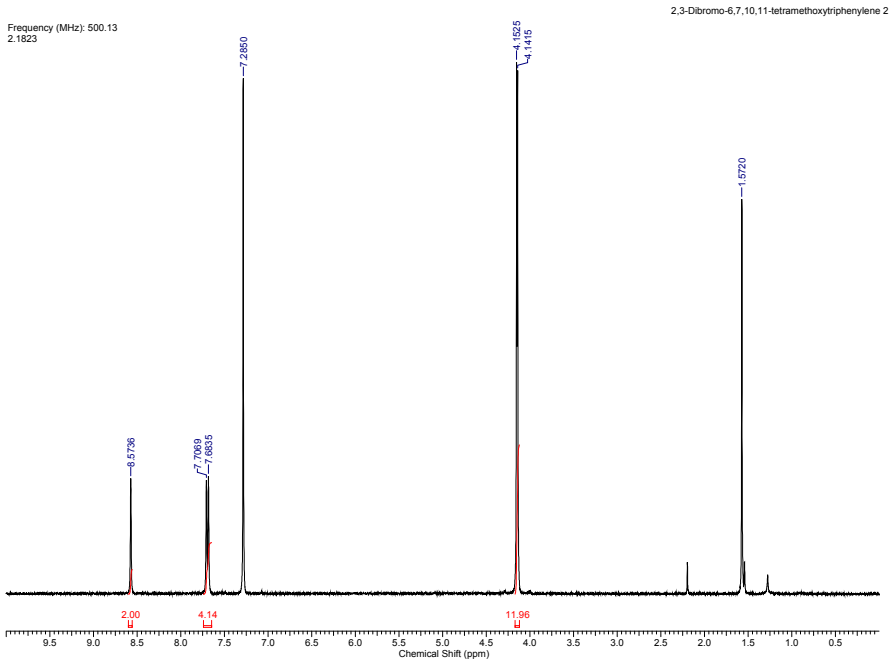

Figure S20. ^{1}H NMR spectrum of compound **2** ($CDCl_3$, 500 MHz).

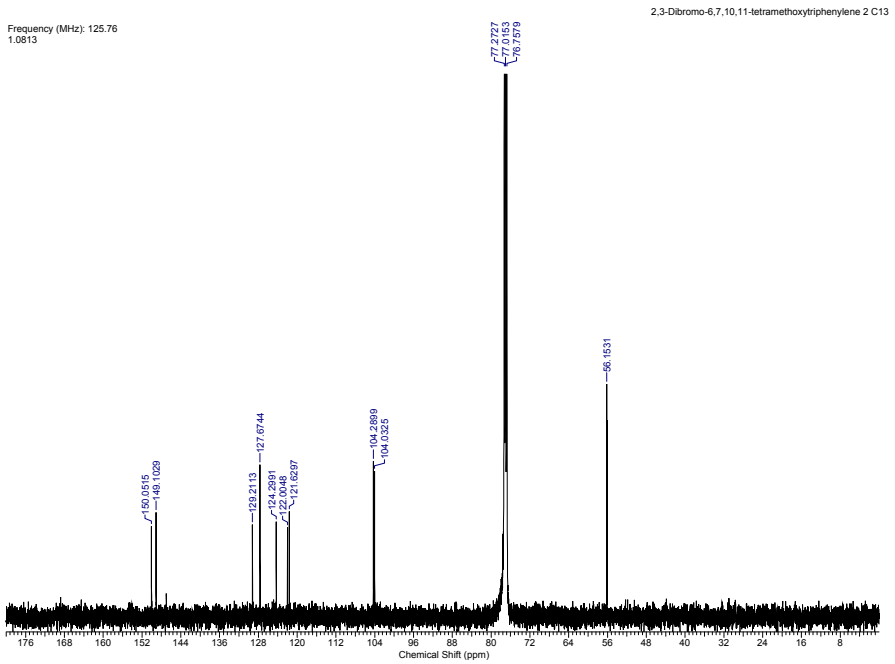

Figure S21. ^{13}C NMR spectrum of compound **2** ($CDCl_3$, 125 MHz).

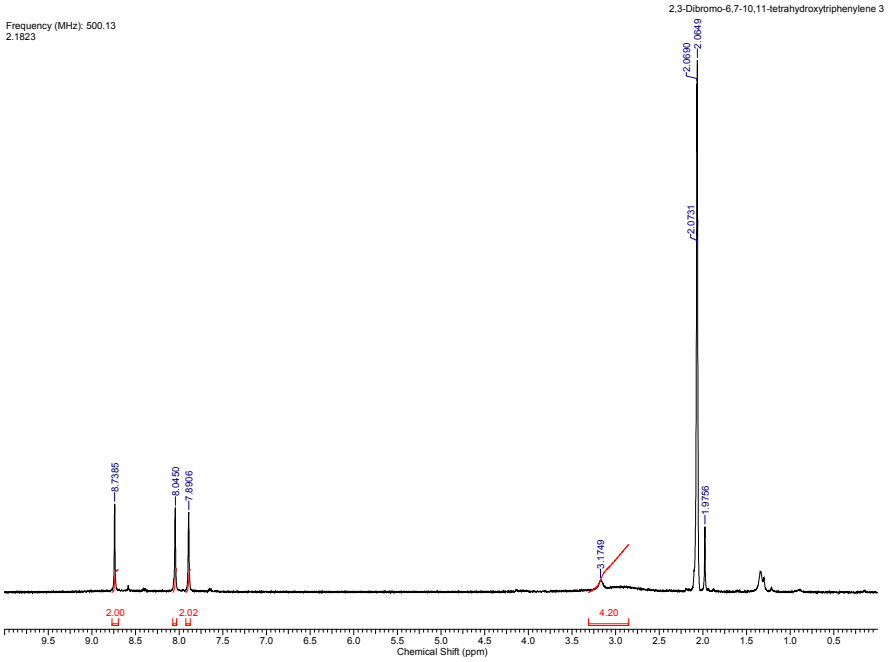

Figure S22. ^{1}H NMR spectrum of compound **3** (acetone-d_6, 500 MHz).

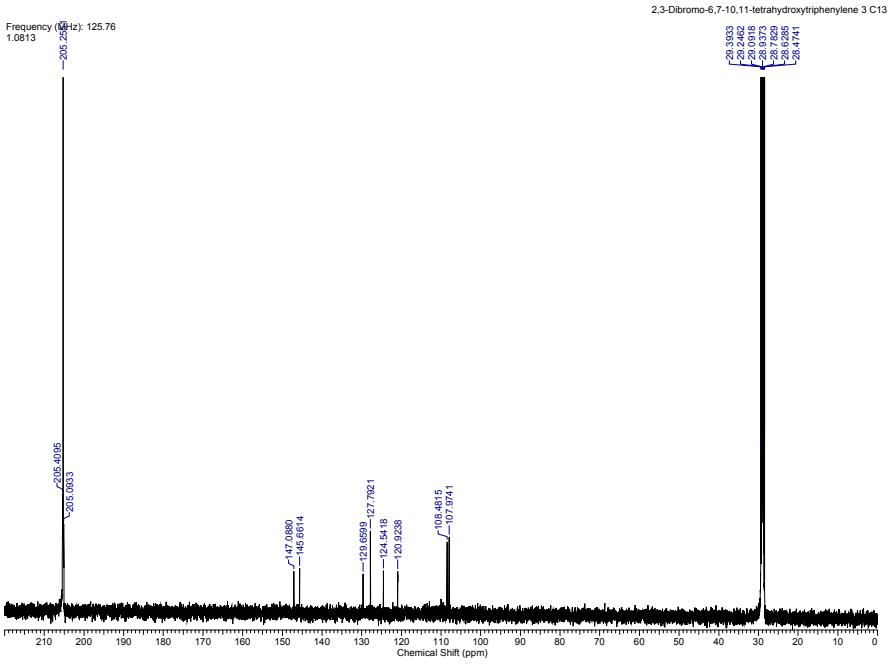

Figure S23. ^{13}C NMR spectrum of compound **3** (acetone-d_6, 125 MHz).

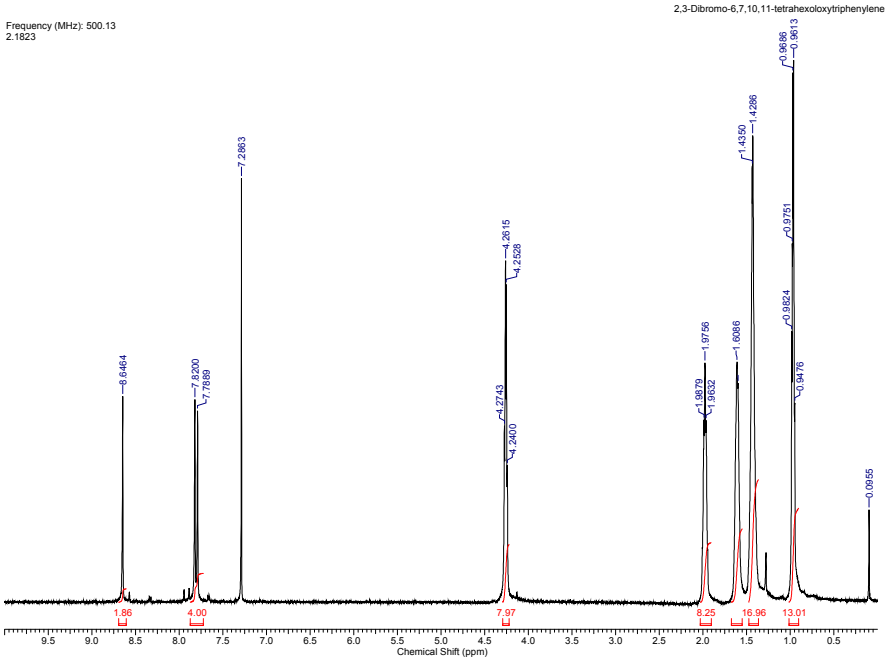

Figure S24. ^{1}H NMR spectrum of compound **4** ($CDCl_3$, 500 MHz). This spectrum is consistent with that previously reported (Cammidge, A.; Gopee, H. *J. Mater. Chem.* **2001**, *11*, 2773–2783).

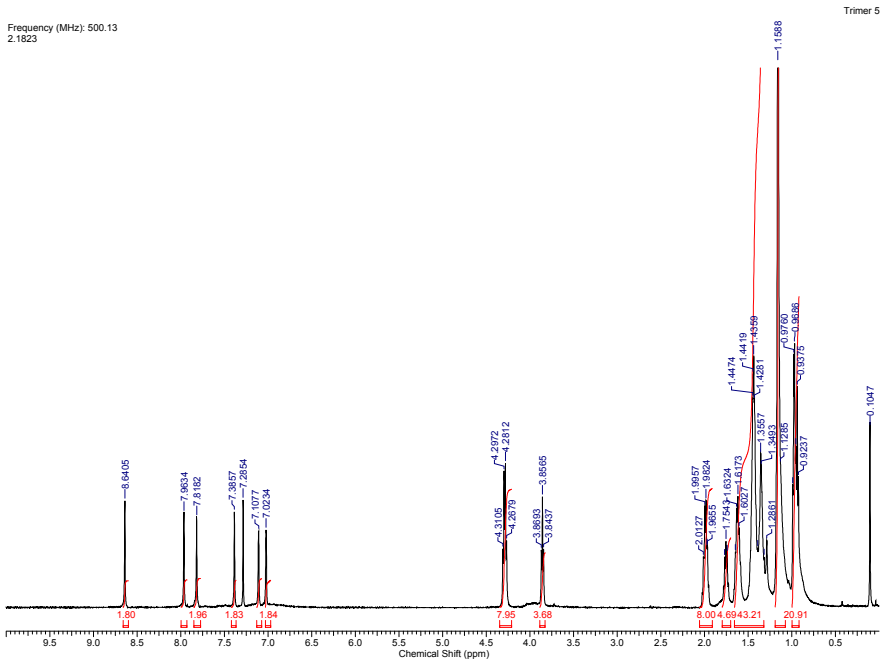

Figure S25. ^{1}H NMR spectrum of compound **5** ($CDCl_3$, 500 MHz).

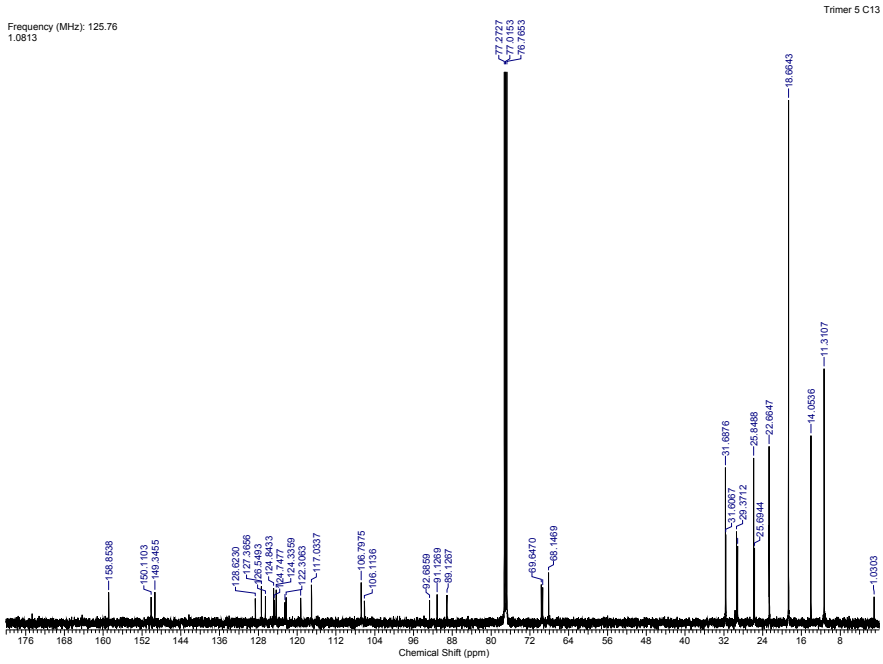

Figure S26. ^{13}C NMR spectrum of compound **5** ($CDCl_3$, 125 MHz).

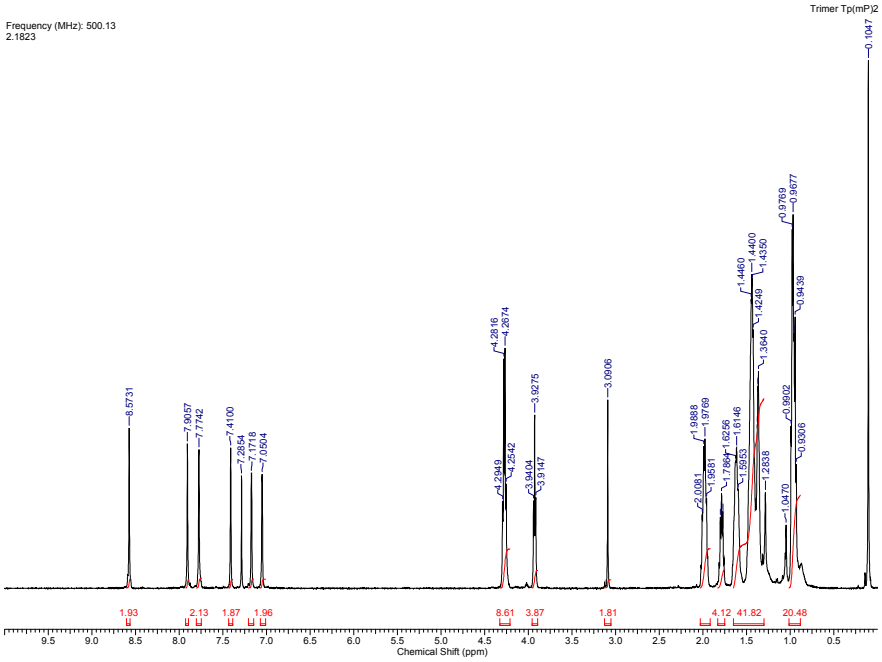

Figure S27. ^{1}H NMR spectrum of compound **Tp(mP)$_2$** ($CDCl_3$, 500 MHz).

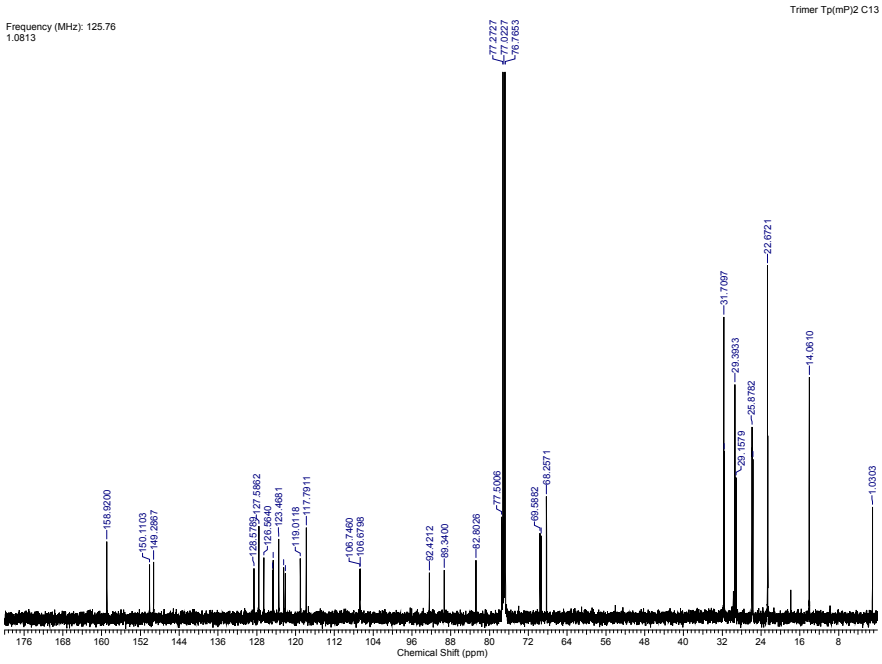

Figure S28. ^{13}C NMR spectrum of compound **Tp(mP)$_2$** ($CDCl_3$, 125 MHz).

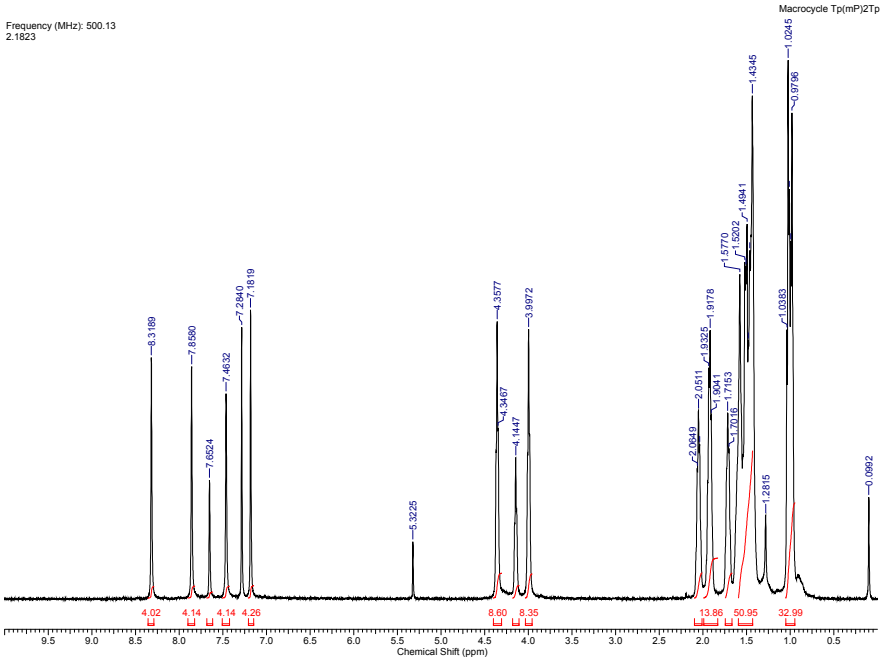

Figure S29. ^{1}H NMR spectrum of compound **Tp(mP)$_2$Tp** ($CDCl_3$, 500 MHz).

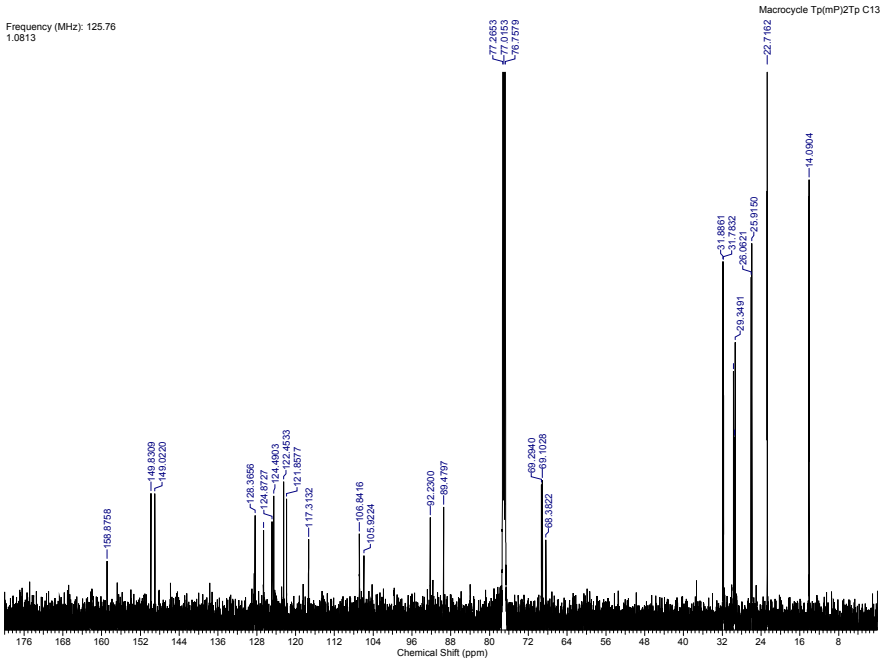

Figure S30. ^{13}C NMR spectrum of compound **Tp(mP)$_2$Tp** ($CDCl_3$, 125 MHz).

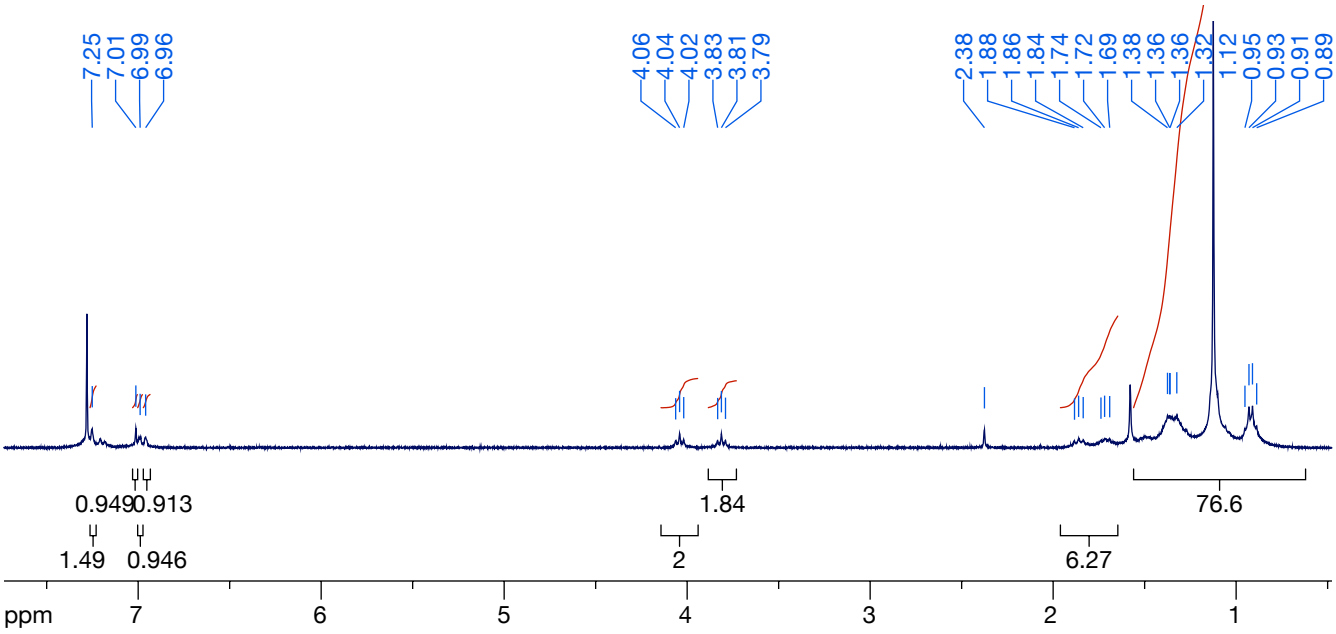

Figure S31. ^{1}H NMR spectrum of compound **9** ($CDCl_3$, 300 MHz).

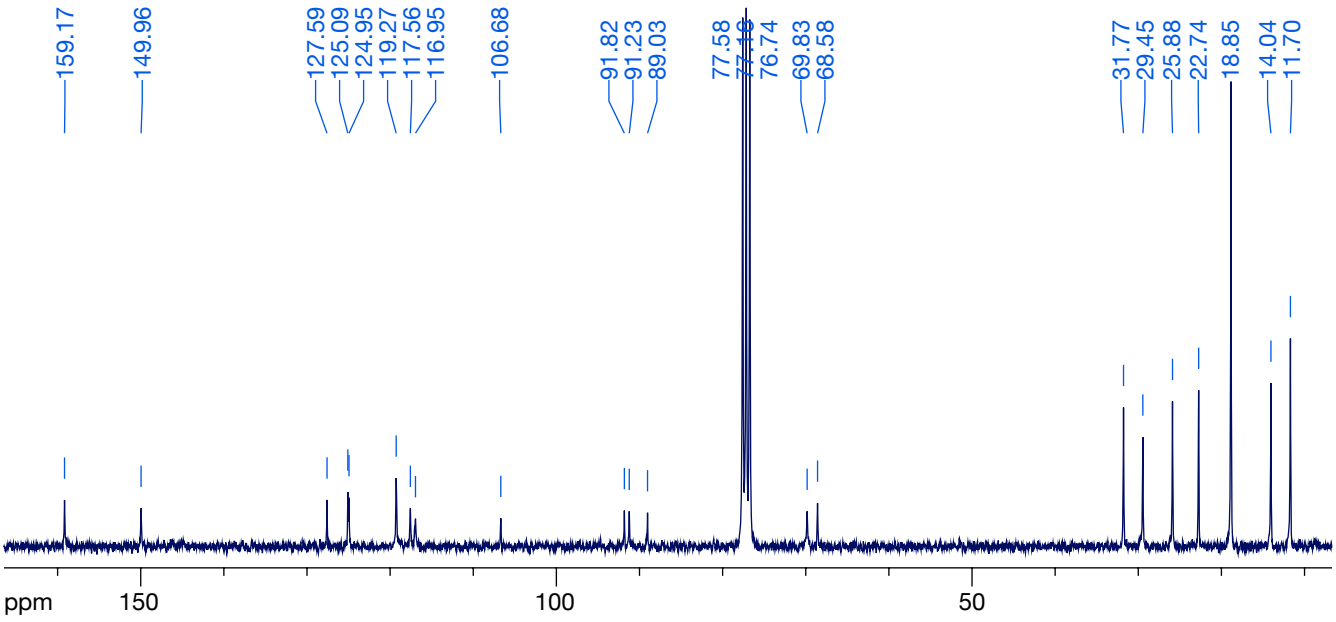

Figure S32. ^{13}C NMR spectrum of compound **9** ($CDCl_3$, 75 MHz).

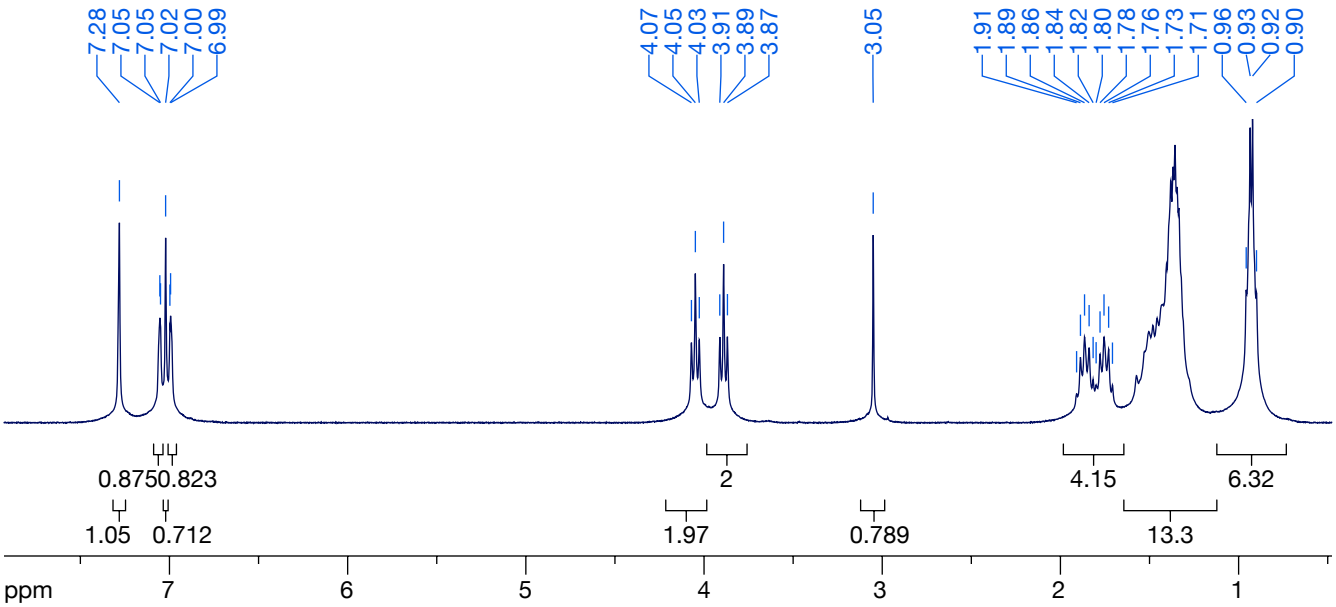

Figure S33. ^{1}H NMR spectrum of compound **B(mP)$_2$** ($CDCl_3$, 300 MHz).

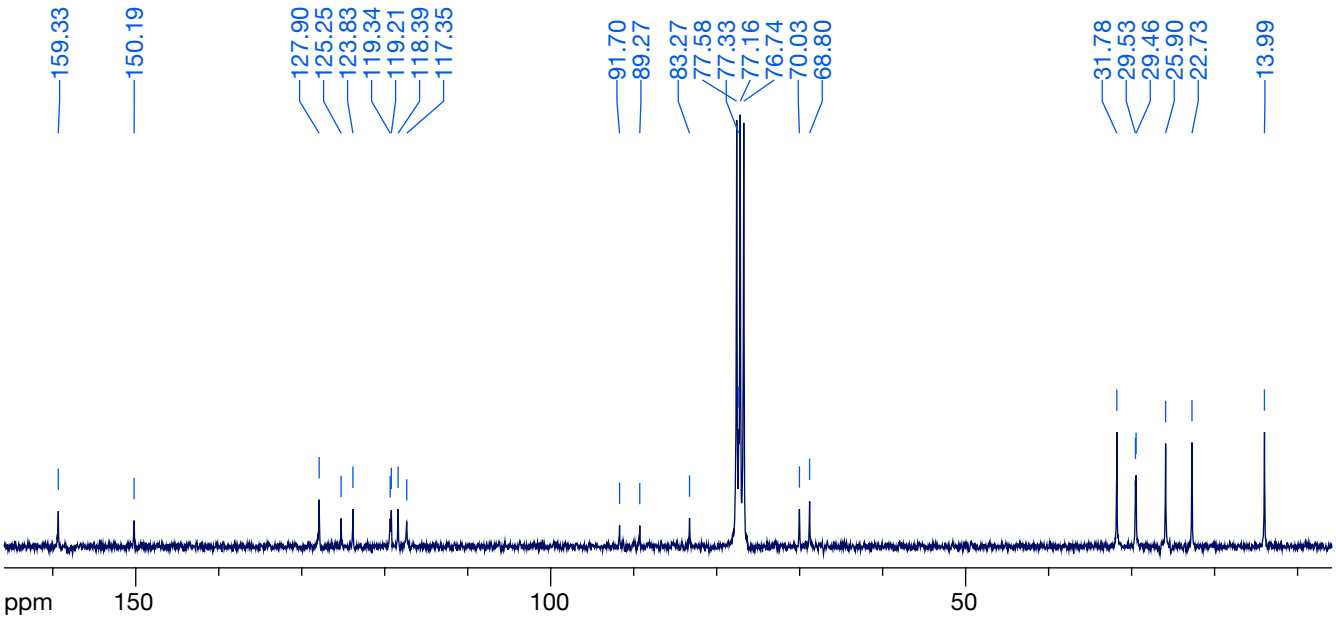

Figure S34. ^{13}C NMR spectrum of compound **B(mP)$_2$** ($CDCl_3$, 75 MHz).

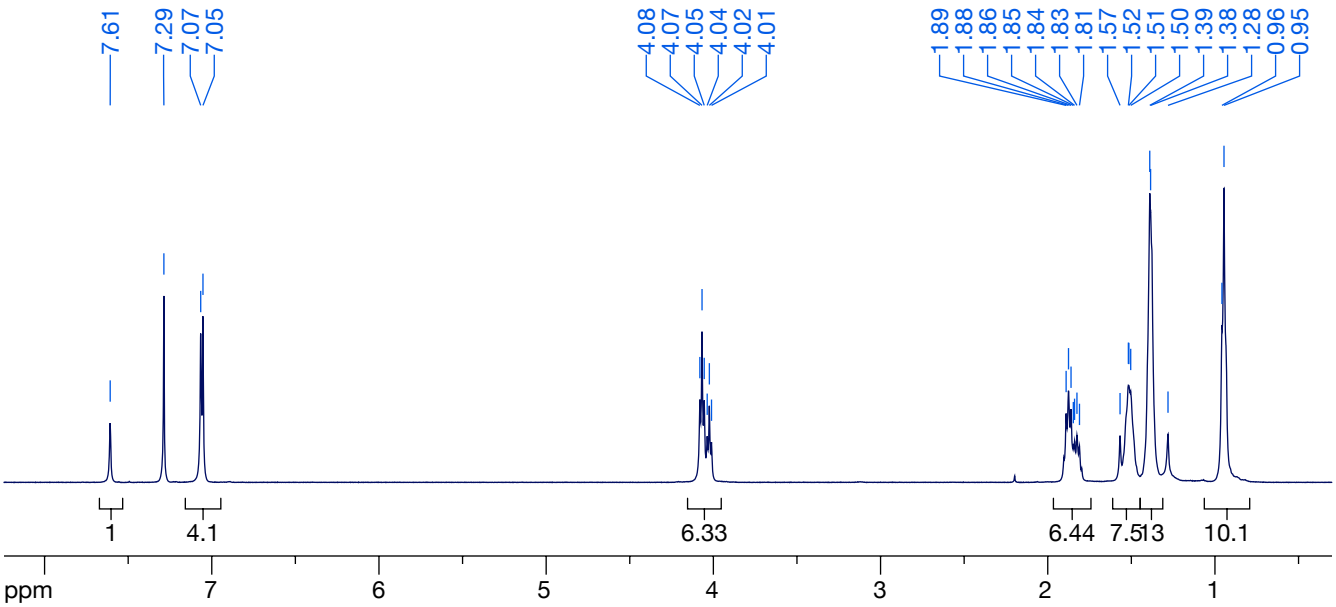

Figure S35. ^{1}H NMR spectrum of compound **B(mP)$_2$B** ($CDCl_3$, 500 MHz).

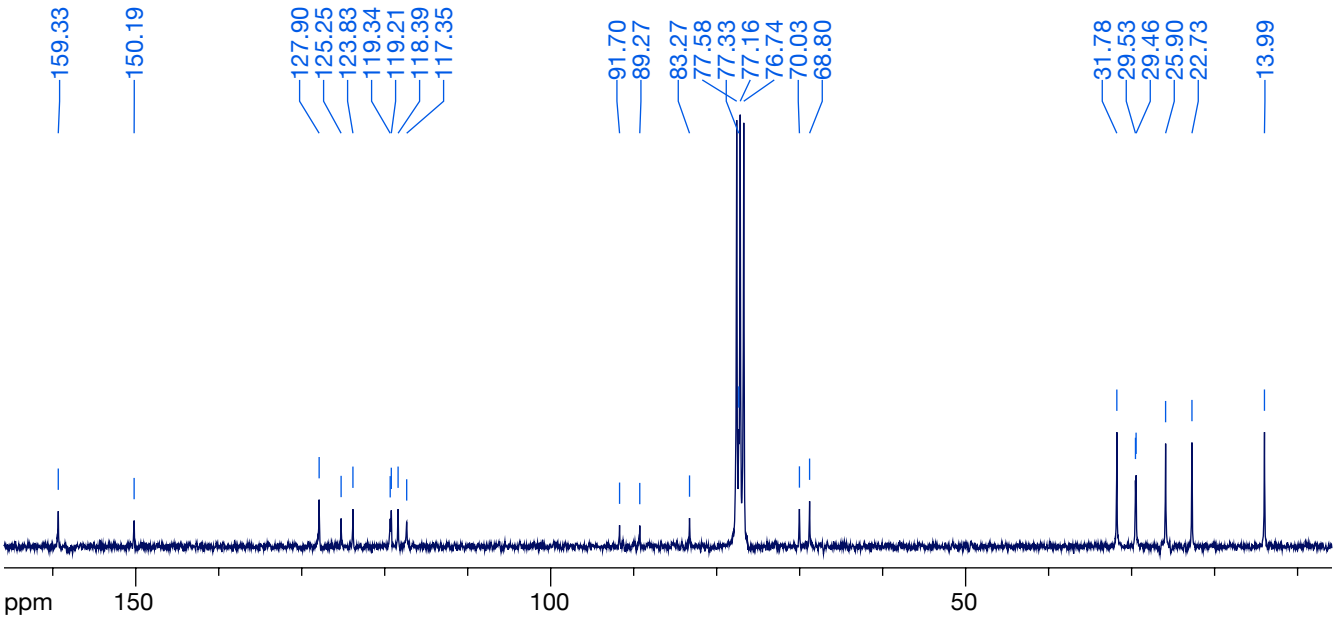

Figure S36. ^{13}C NMR spectrum of compound **B(mP)$_2$B** ($CDCl_3$, 125 MHz).

MS Isotopic Distributions of Synthesized Compounds

A number of the novel compounds reported here are missing high resolution mass spectrometry data, and instead report only low resolution MALDI data. The complete set of compounds was submitted to two of the primary mass spectrometry facilities of the Ohio Mass Spectrometry Consortium. In both cases, the facilities were readily able to obtain high-quality MALDI mass spectra. Unfortunately, however, they were unable to observe the molecular ions by ESI for high resolution analysis (many of these compounds have masses in excess of 1000 amu and so would be difficult to analyze by other MS methods). The co-director of one facility suspected that the compounds (like some others he had seen the past) tend to neutralize the added ESI cationization agents. As an alternative, we include here the experimental (MALDI) and calculated isotopic distributions for the molecular ion peaks, which

are in excellent agreement. It is highly unlikely, especially given that these compounds are synthesized using well-established reactions (Sonogashira couplings) from known compounds, that we would obtain materials with the correct unit mass and isotopic distribution but the wrong molecular formula. Also, the NMR characterization of the key target compound **Tp(mP)$_2$Tp** is of course supplemented by DFT calculations of the ^{1}H isotopic shieldings.

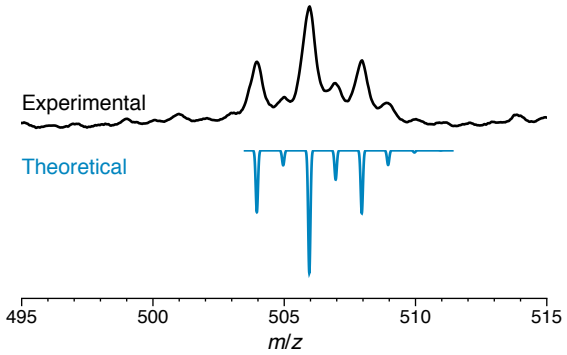

Figure S37. MALDI MS isotopic distribution of compound **2** ($C_{22}H_{18}Br_2O_4$).

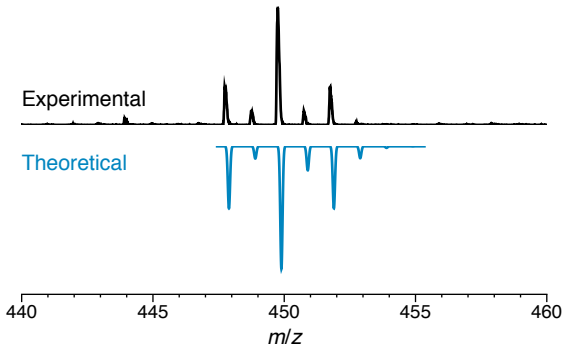

Figure S38. MALDI MS isotopic distribution of compound **3** ($C_{18}H_{10}Br_2O_4$).

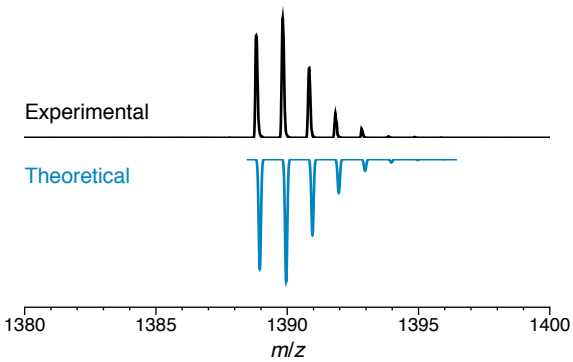

Figure S39. MALDI MS isotopic distribution of compound **5** ($C_{92}H_{132}O_6Si_2$).

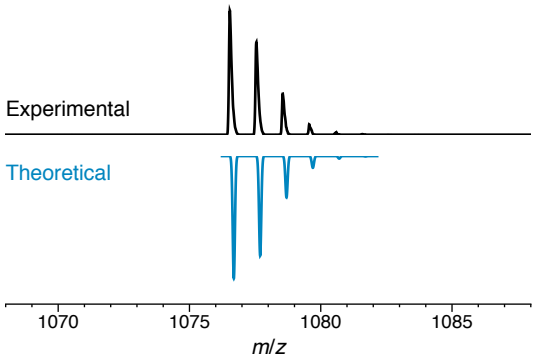

Figure S40. MALDI MS isotopic distribution of compound **Tp(mP)$_2$** ($C_{74}H_{92}O_6$).

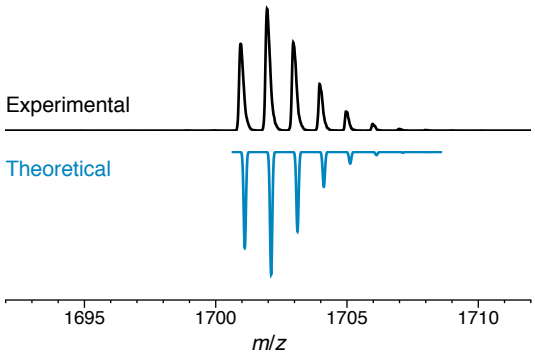

Figure S41. MALDI MS isotopic distribution of compound **Tp(mP)$_2$Tp** ($C_{116}H_{148}O_{10}$).

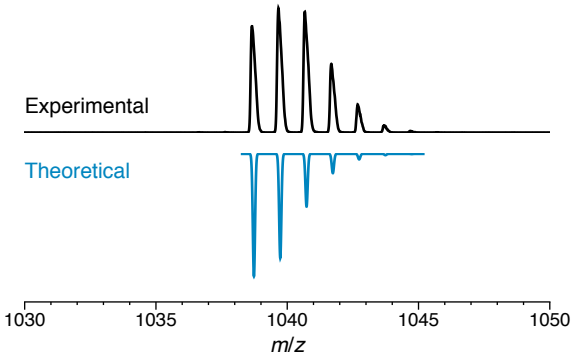

Figure S42. MALDI MS isotopic distribution of compound **9** ($C_{68}H_{102}O_4Si_2$).

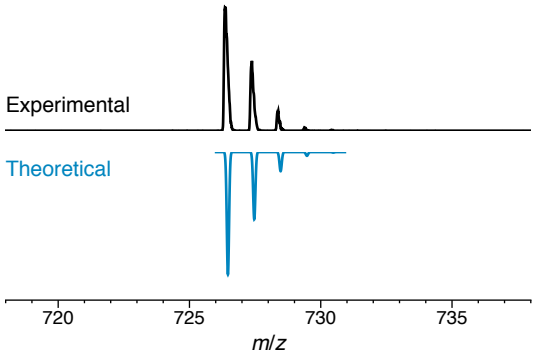

Figure S43. MALDI MS isotopic distribution of compound **B(mP)$_2$** ($C_{50}H_{63}O_4$, $M+H^+$).

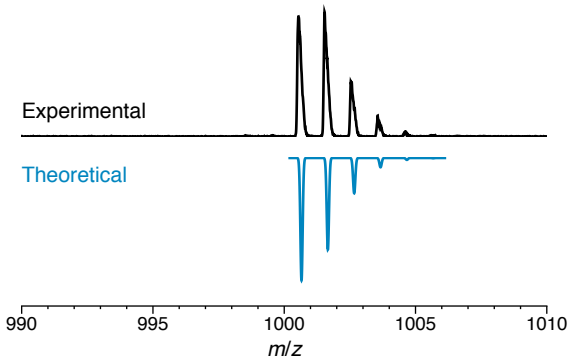

Figure S44. MALDI MS isotopic distribution of compound **B(mP)$_2$B** ($C_{68}H_{88}O_6$).