Supporting Information

Exploratory machine-learned theoretical chemical shifts can closely predict metabolic mixture signals

Kengo Ito,[ab] Yuka Obuchi,[b] Eisuke Chikayama,[ac] Yasuhiro Date[ab] and Jun Kikuchi*[abd]

[a] RIKEN Center for Sustainable Resource Science, 1-7-22 Suehiro-cho, Tsurumi-ku, Yokohama, Kanagawa 230-0045, Japan.

[b] Graduate School of Medical Life Science, Yokohama City University, 1-7-29 Suehiro-cho, Tsurumi-ku, Yokohama, Kanagawa 230-0045, Japan.

[c] Department of Information Systems, Niigata University of International and Information Studies, 3-1-1 Mizukino, Nishi-ku, Niigata-shi, Niigata 950-2292, Japan.

[d] Graduate School of Bioagricultural Sciences, Nagoya University, 1 Furo-cho, Chikusa-ku, Nagoya, Aichi 464-0810, Japan.

* Correspondence and requests for materials should be addressed to J.K. (email: jun.kikuchi@riken.jp).

Table of Contents

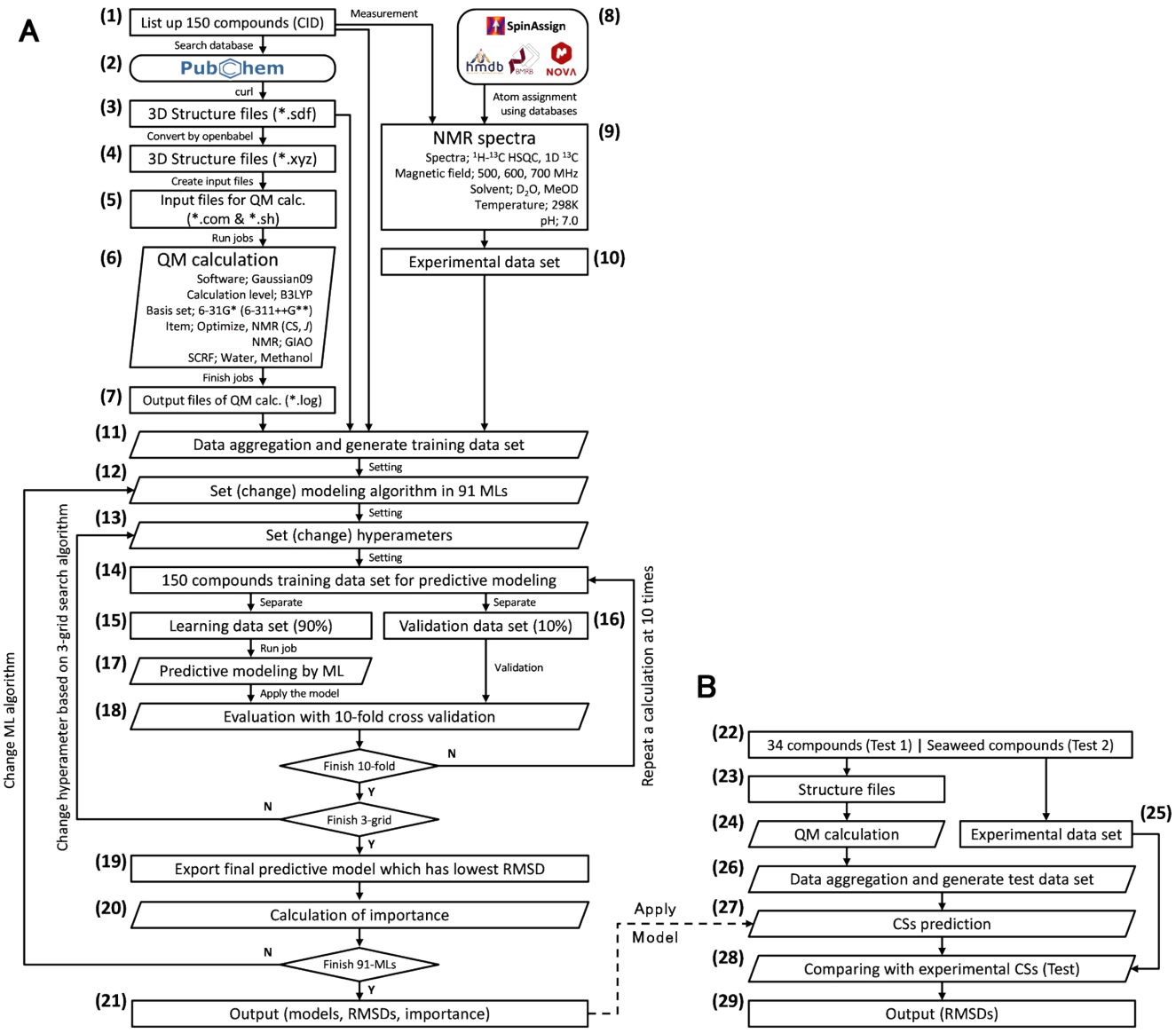


Fig. S1 Analytical flow chart from data collection to the evaluation of predictive modeling (**A**), and test of predictive model using test data set (**B**). First, PubChem compound identifications (CIDs) of 150 compounds that we used as standard substances (Table S3) were listed as "**metid_list.txt**" (**1**), and searched at the PubChem website (https://pubchem.ncbi.nlm.nih.gov/) (**2**) to obtain 3D structure files (**3**). The structure file format was converted from SDF (**3**) to XYZ (**4**) using openbabel software. To calculate theoretical NMR parameters, the XYZ file was converted to a Gaussian09 command file by shell script (**5-6**); a log file including theoretical CS and *J* value was also obtained at this time (**7**). Experimental NMR spectra of the 150 compounds were assigned by using databases (**8-9**). CID, atom number, solvent number, experimental CS, and theoretical shielding constant of each reference substance (e.g., tetramethylsilane) were saved as "**experiment_database.txt**" (**10**). Training data sets ("**metid_list.txt_H.txt**" and "**metid_list.txt_C.txt**") (Table S5 and S6) were generated by the Java program "**toolgaussianlearndata.jar**" from "**metid_list.txt**", "**experiment_database.txt**", "**CID.sdf**" and "**CID.log**" (**11**). In total, 91 ML algorithms (Table S7) (**12**) and their hyperparameters (Table S1) (**13**) were explored to identify the best predictive model. At this time, 10-fold CV was calculated to evaluate over-learning and over-fitting (**14-18**). After the 3-grid search, the final model with the lowest RMSD was exported (**19**), and the importance of explanatory variables was calculated. Lastly, the predictive models, RMSDs, and importance of 91 MLs were saved as Rdata (**21**). The steps of ML (**12-21**) were calculated automatically by using the R program "**Several_Predictive_Modeling_for_QM.R**", which depends on the caret library (https://topepo.github.io/caret/). A total of 34 compounds which does not include learning and k-fold validation (training) data sets of ML (Table S4), and seaweed components were used as the test data set (**22**). Collection of structure files (**23**), QM calculations (**24**), and collection of experimental CS (**25**) were performed in same way using steps (**2-10**). Test data sets were also generated (**26**) in the same way using step (**11**). CSs of test data sets were predicted by the learned predictive model using the R program "**Applying_Model.R**" (**27**). Finally, predicted CSs were compared with experimental CSs (**28**), and the predictive accuracy was determined by RMSDs (**29**). Example data and programs for generating the data set for ML from experimental/theoretical data, and the 91 MLs with the grid search CV approach used in this study are deposited on our website (http://dmar.riken.jp/Rscripts/).

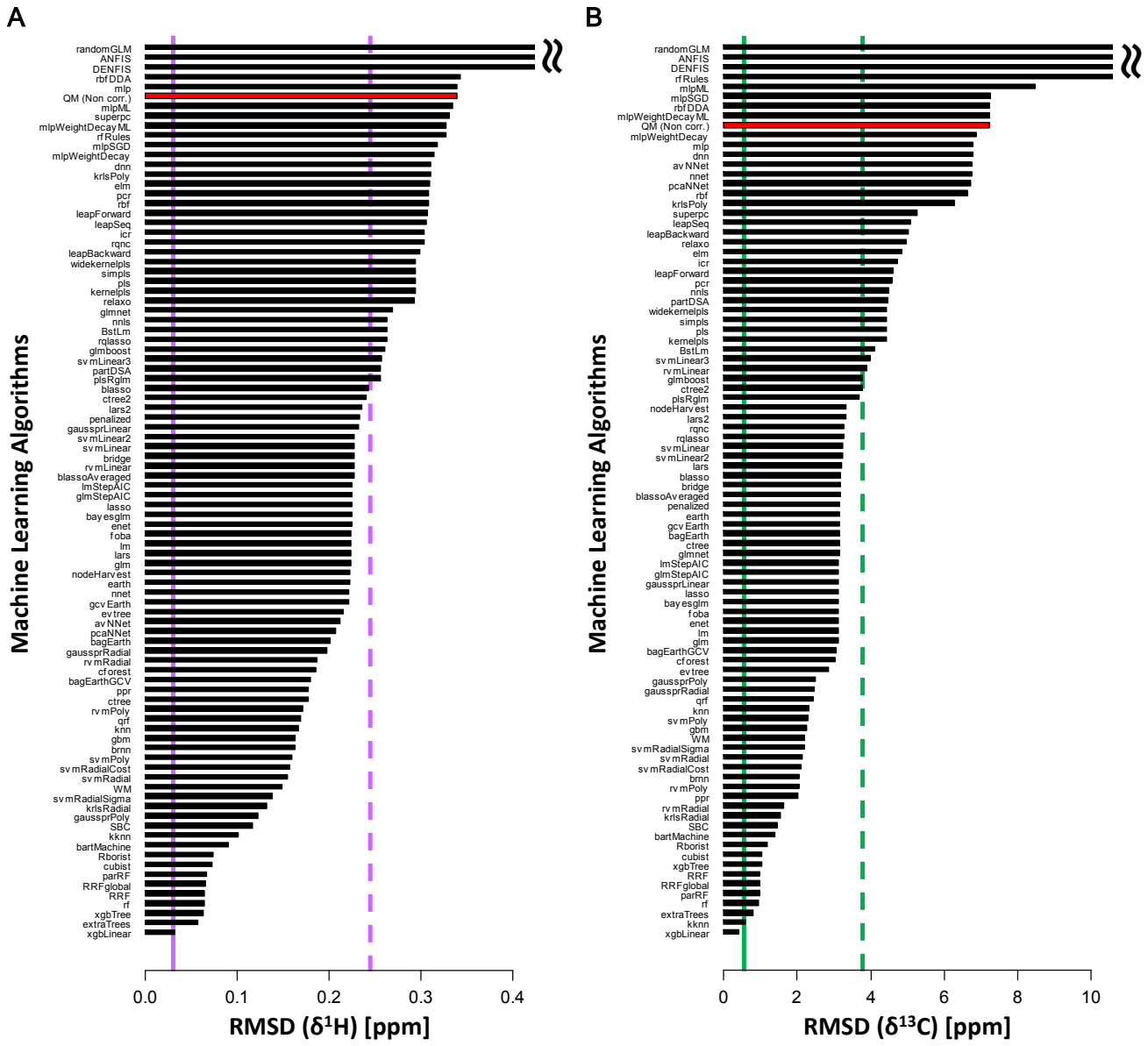


Fig. S2 RMSDs between experimental and theoretical (red)/predicted (black) CSs of 150 compounds as training data set. (**A**) δ^{1}H and (**B**) δ^{13}C were corrected by 91 ML algorithms after QM at the B3LYP/6-31G* level. These figures are expanded versions of Fig. 1B. These RMSDs indicate learning errors. The dotted line indicates the RMSD (δ^{1}H=0.2442 ppm, δ^{13}C=3.7513 ppm) of the predicted CSs of 150 compounds calculated by Mnova; the unbroken line indicates the recommended tolerances (δ^{1}H=0.03 ppm, δ^{13}C=0.53 ppm) for assignment from the SpinAssign tool.

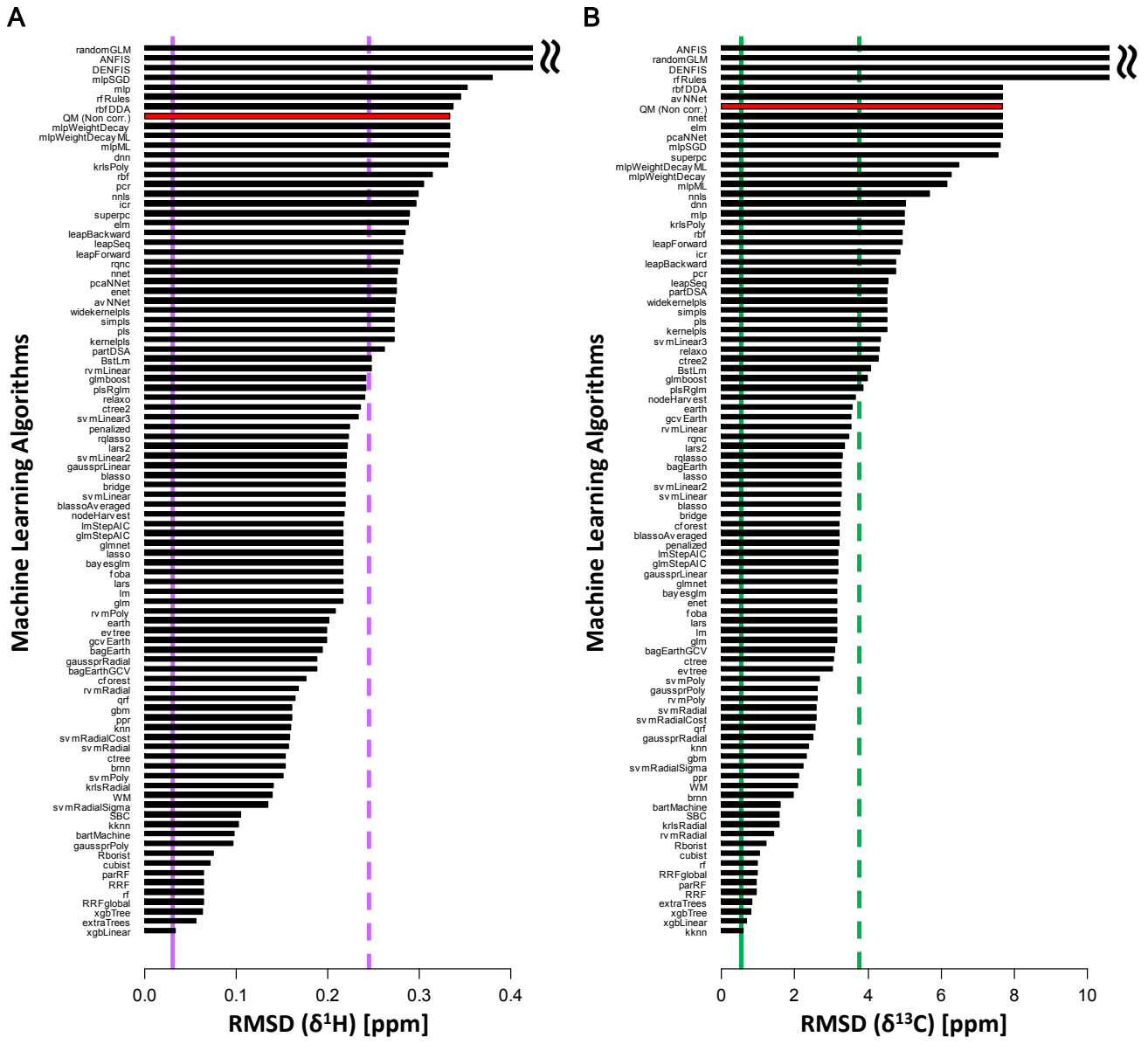


Fig. S3 RMSDs between experimental and theoretical (red)/predicted (black) CSs of 150 compounds as a training data set. (**A**) δ^{1}H and (**B**) δ^{13}C were corrected by 91 ML algorithms after QM at the B3LYP/6-311++G** level. These RMSDs indicate learning errors. The dotted line indicates the RMSDs (δ^{1}H=0.2442 ppm, δ^{13}C=3.7513 ppm) of predicted CSs of 150 compounds calculated by Mnova; the unbroken line indicates the recommended tolerances (δ^{1}H=0.03 ppm, δ^{13}C=0.53 ppm) for assignment from the SpinAssign tool.

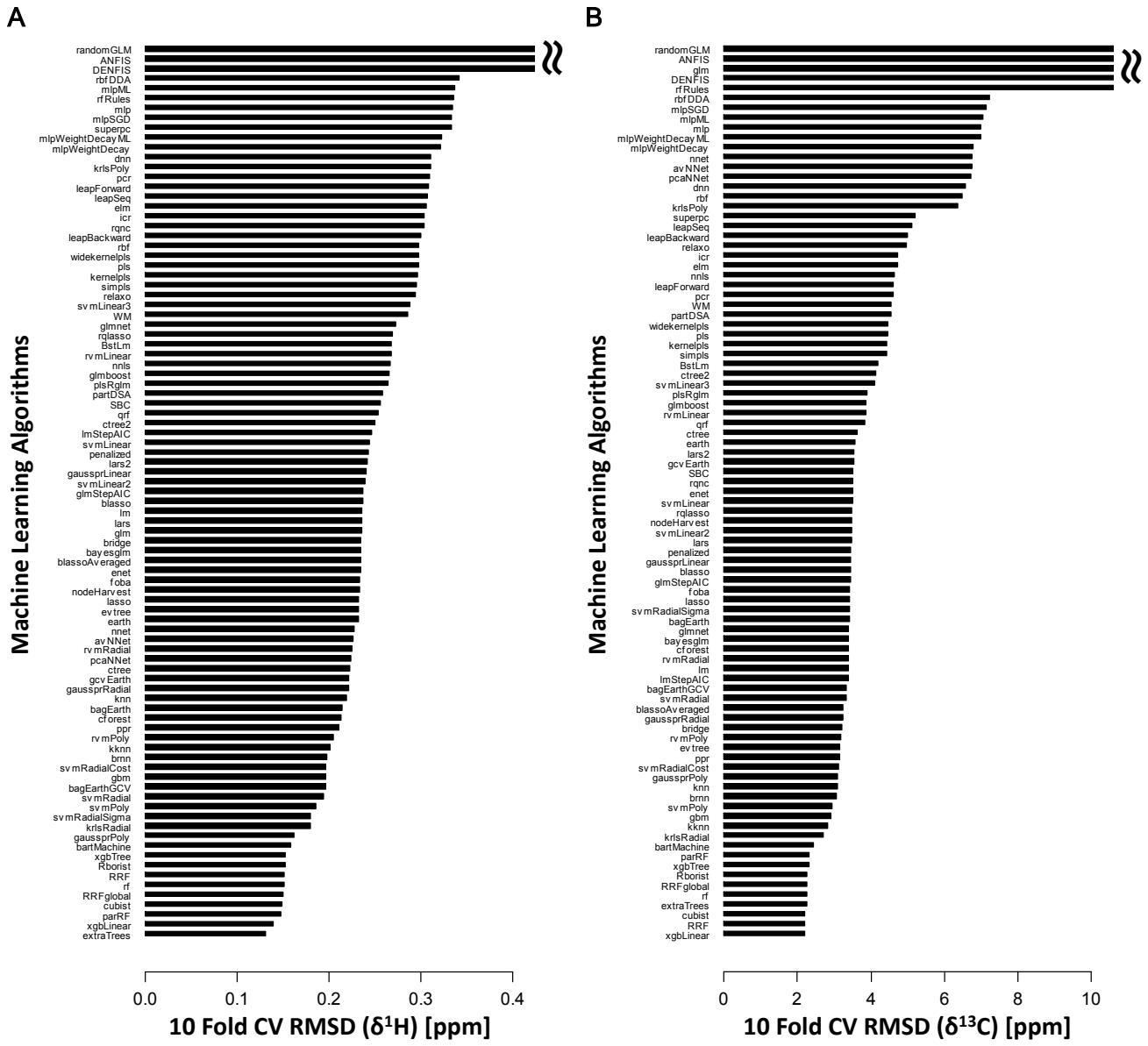


Fig. S4 10-fold CV RMSDs between experimental and predicted CSs of 150 compounds as a training data set. (**A**) δ^{1}H and (**B**) δ^{13}C were corrected by 91 ML algorithms after QM at the B3LYP/6-31G* level. These RMSDs indicate generalization errors.

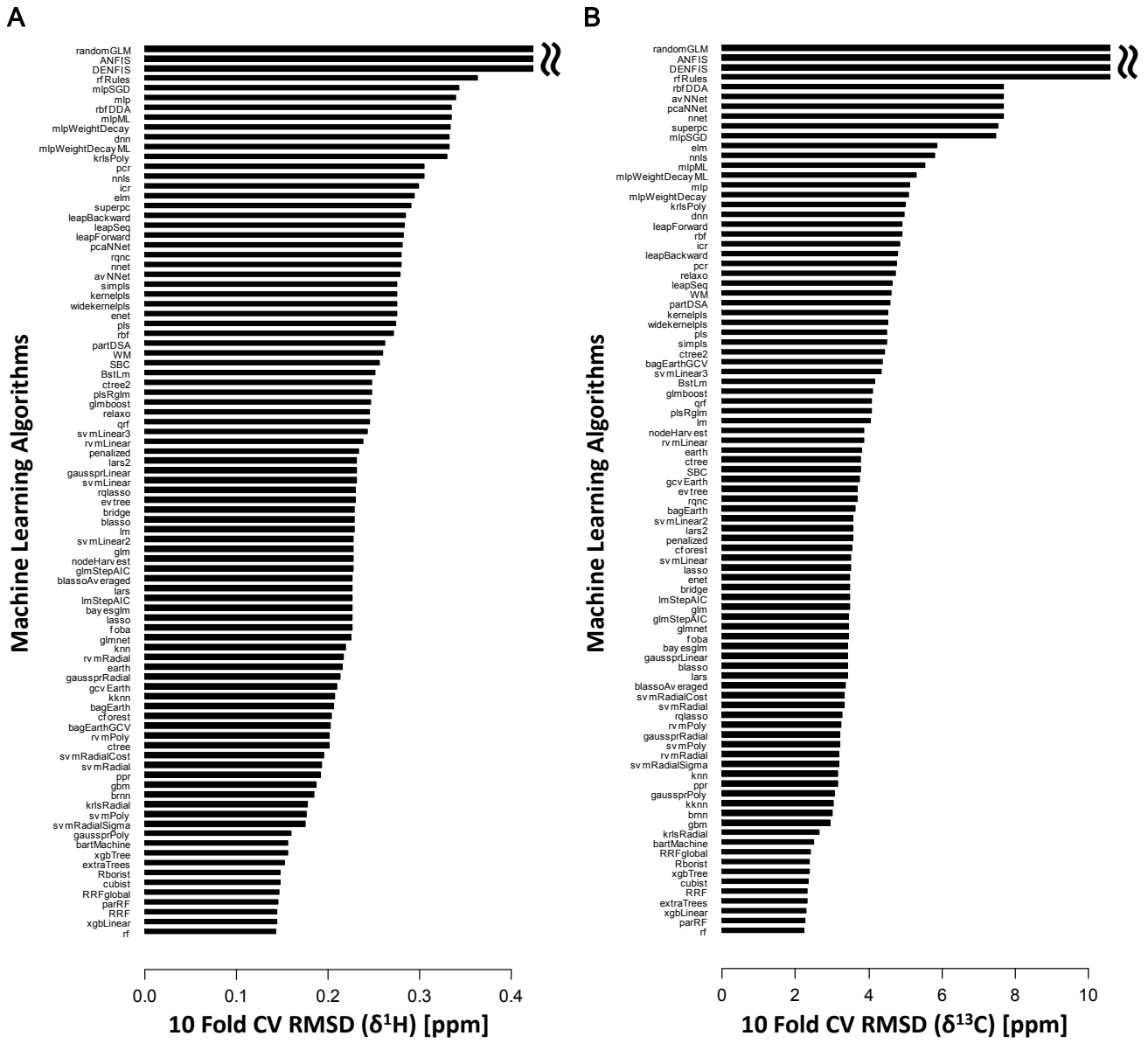
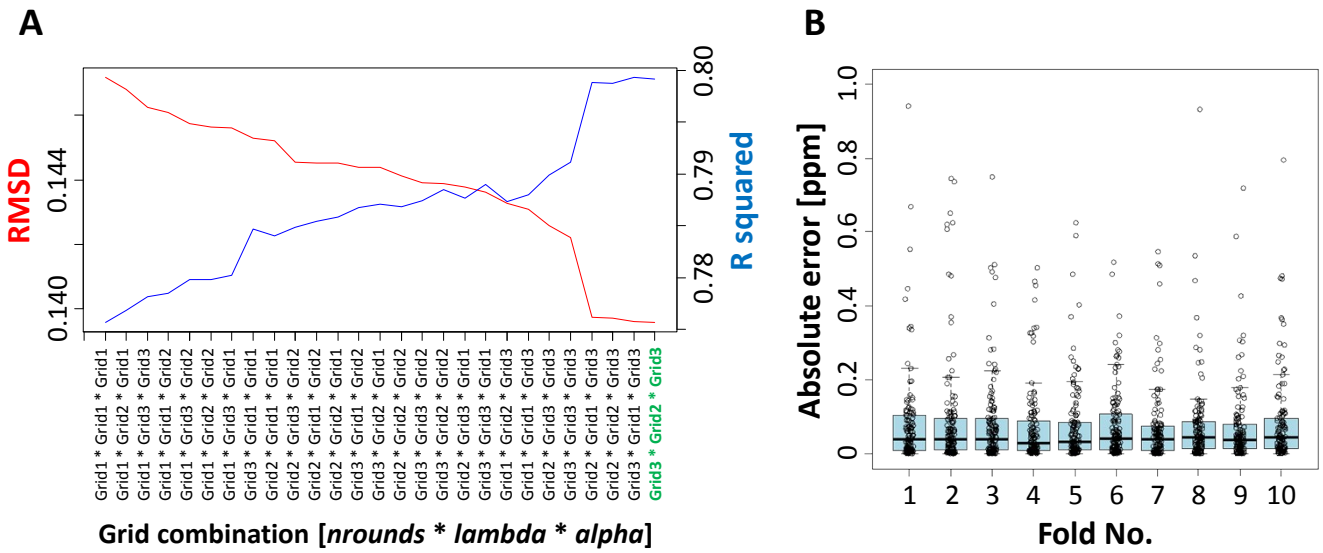


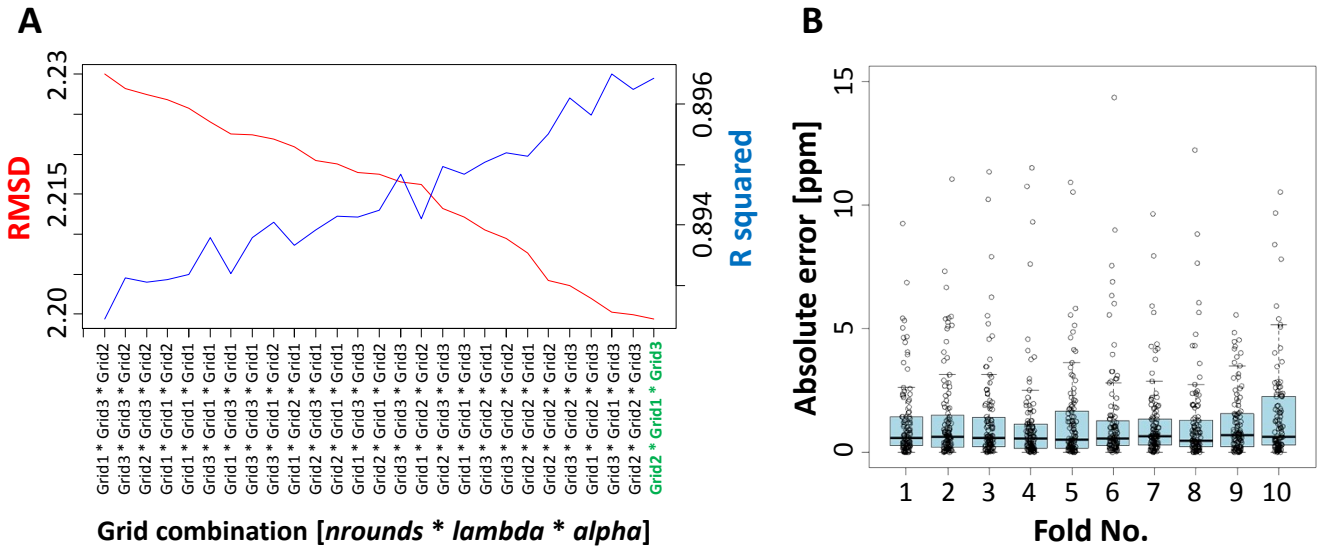
Fig. S5 10-fold CV RMSDs between experimental and predicted CSs of 150 compounds as a training data set. (**A**) δ^{1}H and (**B**) δ^{13}C were corrected by 91 ML algorithms after QM at the B3LYP/6-311++G** level. These RMSDs indicate generalization errors.



C

Fold No.	Abs. Min.	Abs. 1st Qu.	Abs. Median	Abs. Mean	Abs. 3rd Qu.	Abs. Max.	Variance
Fold 01	0.000454	0.009740	0.039180	0.081840	0.103800	0.940100	0.08876628
Fold 02	0.000160	0.012360	0.039170	0.093210	0.095120	0.744600	0.08045715
Fold 03	0.000090	0.011320	0.039480	0.083750	0.096810	0.749900	0.09273747
Fold 04	0.000244	0.008496	0.029740	0.070550	0.090130	0.502800	0.08376600
Fold 05	0.000168	0.011530	0.032990	0.075980	0.085930	0.624000	0.06856733
Fold 06	0.000421	0.011440	0.041270	0.077440	0.106700	0.518600	0.07560452
Fold 07	0.000076	0.008431	0.039100	0.069430	0.073400	0.547100	0.06987863
Fold 08	0.000296	0.013390	0.043630	0.075470	0.087010	0.932300	0.06642588
Fold 09	0.000882	0.013670	0.037990	0.070390	0.080540	0.718500	0.08310468
Fold 10	0.001049	0.012610	0.043010	0.083940	0.097080	0.793600	0.09211333

Fig. S6 Performance of the ML algorithm xgbLinear for δ^1H prediction. Theoretical CSs were calculated at the B3LYP/6-31G*//GIAO/B3LYP/6-31G* level. (**A**) Convergence curve of hyperparameters determined by the grid search for a combination of 3 hyperparameters (*nrounds*=50 [Grid1], 100 [Grid2], 150 [Grid3]; *lambda*=0 [Grid1], 0.0001 [Grid2], 0.1 [Grid3]; *alpha*=0 [Grid1], 0.0001 [Grid2], 0.1 [Grid3]). The right-most (green) grid combination is the best model, having the lowest RMSD (red) with highest R^2 (blue). The hyperparameters *nrounds*=150, *lambda*=0.0001, and *alpha*=0.1 were selected and used for the final CS prediction. Other hyperparameters are shown in Table S1. (**B**) Absolute errors between experimental CS and predicted CS for the validation set were calculated by using 10-fold CV to evaluate over-learning and over-fitting. Boxplots show absolute errors of 128 CSs (dot) in each fold. The 128 CSs in the validation set were calculated by using a predictive model, which was learned by using 1149 CSs as a learning set. The statistics (absolute minimum error, absolute 1st quarter error, absolute median error, absolute mean error, absolute 1st quarter error, absolute maximum error, and variance) are shown in (**C**).



C

Fold No.	Abs. Min.	Abs. 1st Qu.	Abs. Median	Abs. Mean	Abs. 3rd Qu.	Abs. Max.	Variance
Fold 01	0.001844	0.2538	0.5908	1.218	1.432	9.23	40.42158
Fold 02	0.001257	0.2222	0.6193	1.357	1.480	11.03	40.91832
Fold 03	0.000682	0.2324	0.5861	1.302	1.379	11.33	44.07397
Fold 04	0.007560	0.1869	0.5726	1.143	1.139	11.51	45.60710
Fold 05	0.014230	0.1671	0.5155	1.266	1.642	10.90	45.35547
Fold 06	0.017980	0.2731	0.5510	1.319	1.268	14.32	43.18276
Fold 07	0.002369	0.3146	0.6607	1.185	1.352	9.63	45.57893
Fold 08	0.003957	0.2208	0.4688	1.186	1.282	12.23	36.51587
Fold 09	0.002760	0.2201	0.7032	1.155	1.550	5.54	41.90630
Fold 10	0.006155	0.2991	0.6255	1.505	2.158	10.52	39.61191

Fig. S7 Performance of the ML algorithm xgbLinear for $\delta^{13}C$ prediction. Theoretical CSs were calculated at the B3LYP/6-31G*//GIAO/B3LYP/6-31G* level. (**A**) Convergence curve of hyperparameters determined by the grid search for a combination of 3 hyperparameters (*nrounds*=50 [Grid1], 100 [Grid2], 150 [Grid3]; *lambda*=0 [Grid1], 0.0001 [Grid2], 0.1 [Grid3]; *alpha*=0 [Grid1], 0.0001 [Grid2], 0.1 [Grid3]). The right-most grid combination (green) is the best model, having the lowest RMSD (red) with highest R^2 (blue). The hyperparameters *nrounds*=100, *lambda*=0, and *alpha*=0.1 were selected and used for the final CS prediction. Other hyperparameters are shown in Table S1. (**B**) Absolute errors between experimental CS and predicted CS for the validation set were calculated by using 10-fold cross validation to evaluate over-learning and over-fitting. Boxplots show absolute errors of 108 CSs (dot) in each fold. The 108 CSs in the validation set were calculated by using a predictive model, which was learned by using 970 CSs as a learning set. The statistics (absolute minimum error, absolute 1st quarter error, absolute median error, absolute mean error, absolute 1st quarter error, absolute maximum error, and variance) are shown in (**C**).

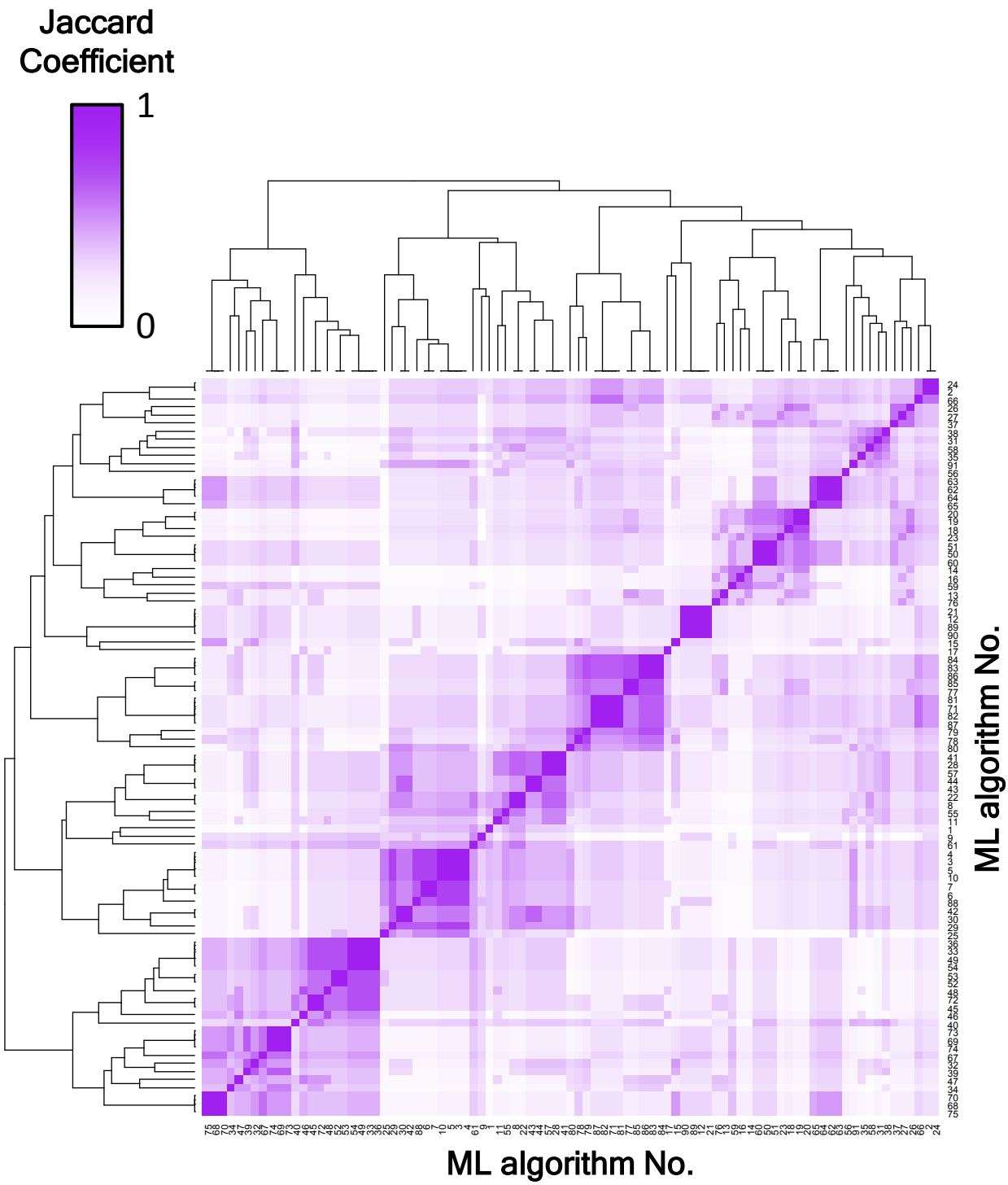


Fig. S8 Heat map showing the similarity among the 91 ML algorithms. The color indicates the Jaccard coefficient, with darker color indicating more similar models. The numbers correspond to the algorithms listed in Table S7.

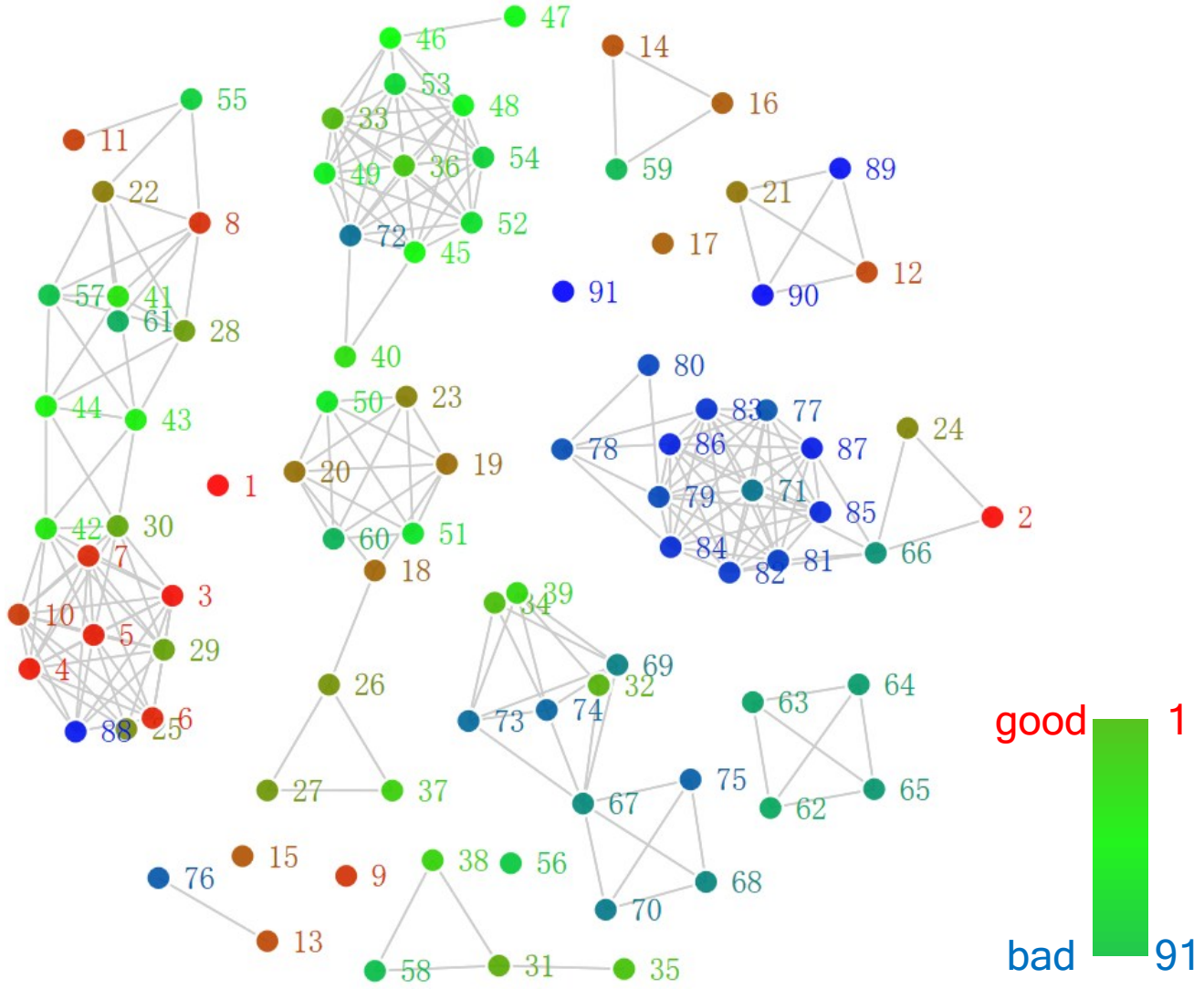


Fig. S9 Network diagram showing clusters of similar model types or relevant characteristics of the 91 ML algorithms. The numbers correspond to the algorithms listed in Table S7. The nodes are connected by Jaccard similarity (>0.56). Node colors indicate the performance of the $\delta^{13}C$ predictive model (Fig. S2B): good models show low RMSDs; poor models show high RMSDs.

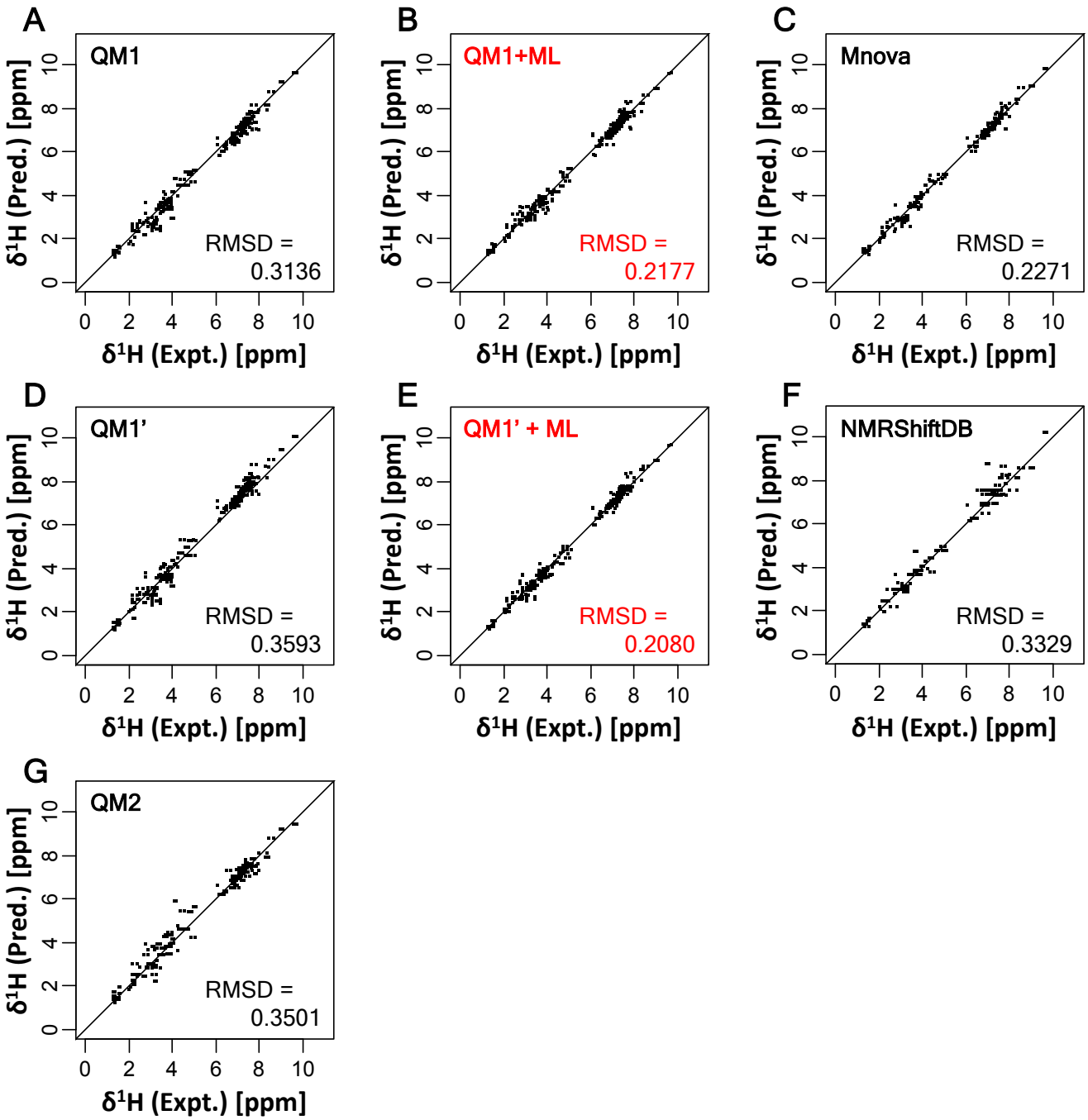


Fig. S10 Comparison of conventional prediction methods based on (**A, D, G**) quantum chemistry and (**C, F**) a data-driven approach with (**B, E**) this study's method. This figure is an expanded version of Fig. 2 with the addition of different calculation levels employed in Gaussian09 software (**D**), QM1' with ML predictive approach (**E**), NMRShiftDB (**F**), and Spartan (**G**) results. Experimental CSs are compared with the calculated δ^1H of 34 compounds in D_2O and MeOD solvent as test data set (Table 4). In total, 402 CSs of test data set were plotted for δ^1H. QM1 shows the theoretical CSs calculated at the B3LYP/6-31G*//GIAO/B3LYP/6-31G* level, and QM1' shows the theoretical CSs calculated at the B3LYP/6-311++G**//GIAO/B3LYP/6-311++G** level using Gaussian09 software. QM1+ML and QM1'+ML show the results of the predictive approach described in this study, in which the ML algorithm xgbLinear calculates an SF that is applied to QM1 and QM1'. QM2 shows the theoretical CSs calculated at the EDF2/6-31G* level using Spartan'14 software. The theoretical CSs of QM2 were corrected with the weighted average using a Boltzmann distribution after conformational analysis.

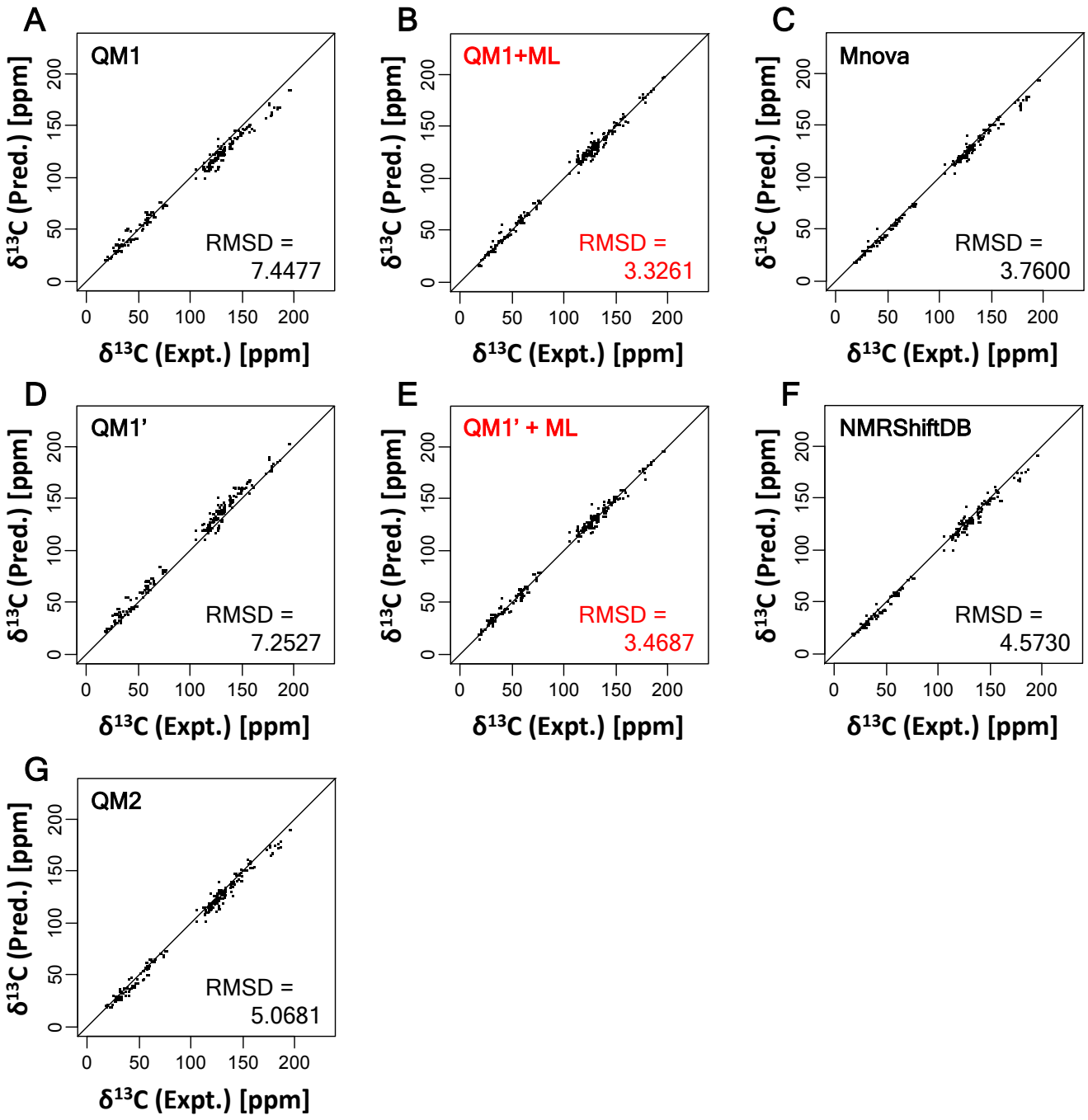


Fig. S11 Comparison of conventional prediction methods based on (**A, D, G**) quantum chemistry and (**C, F**) a data-driven approach with (**B, E**) this study's method. This figure is an expanded version of Fig. 2 with the addition of different calculation level using Gaussian09 software (**D**), QM1' with ML predictive approach (**E**), NMRShiftDB (**F**), and Spartan (**G**) results. Experimental CSs are compared with the calculated $\delta^{13}C$ of 34 compounds in D_2O and MeOD solvent as test data set (Table S4). In total, 376 CSs of test data set were plotted for $\delta^{13}C$. QM1 shows the theoretical CSs calculated at the B3LYP/6-31G*//GIAO/B3LYP/6-31G* level, and QM1' shows the theoretical CSs calculated at the B3LYP/6-311++G**//GIAO/B3LYP/6-311++G** level using Gaussian09 software. QM1+ML and QM1'+ML show the results of the predictive approach described in this study, in which the ML algorithm xgbLinear calculates an SF that is applied to QM1 and QM1'. QM2 shows the theoretical CSs calculated at the EDF2/6-31G* level using Spartan'14 software. The theoretical CSs of QM2 were corrected with the weighted average using a Boltzmann distribution after conformational analysis.

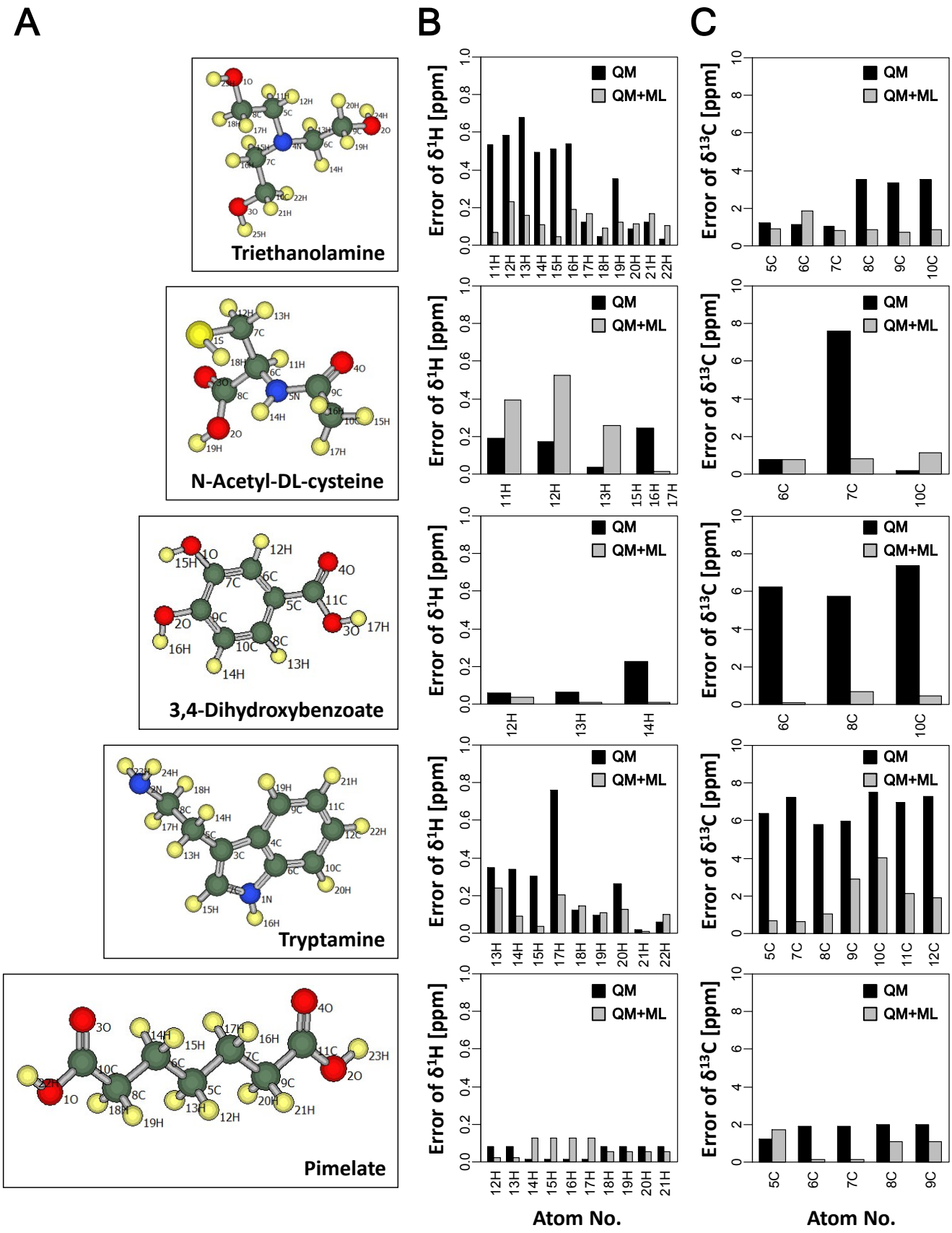


Fig. S12 Evaluation of correction effect for theoretical CSs of partial structure. Five compounds in test set and its atom number are shown as examples (**A**). The errors of δ^{1}H (**B**) and δ^{13}C (**C**) between experimental CS and predicted CS of each atoms are plotted. QM (black bar) shows the errors of theoretical CSs calculated at the B3LYP/6-31G*//GIAO/B3LYP/6-31G* level, and QM+ML (gray bar) shows the errors of predicted CSs using the predictive approach described in this study.

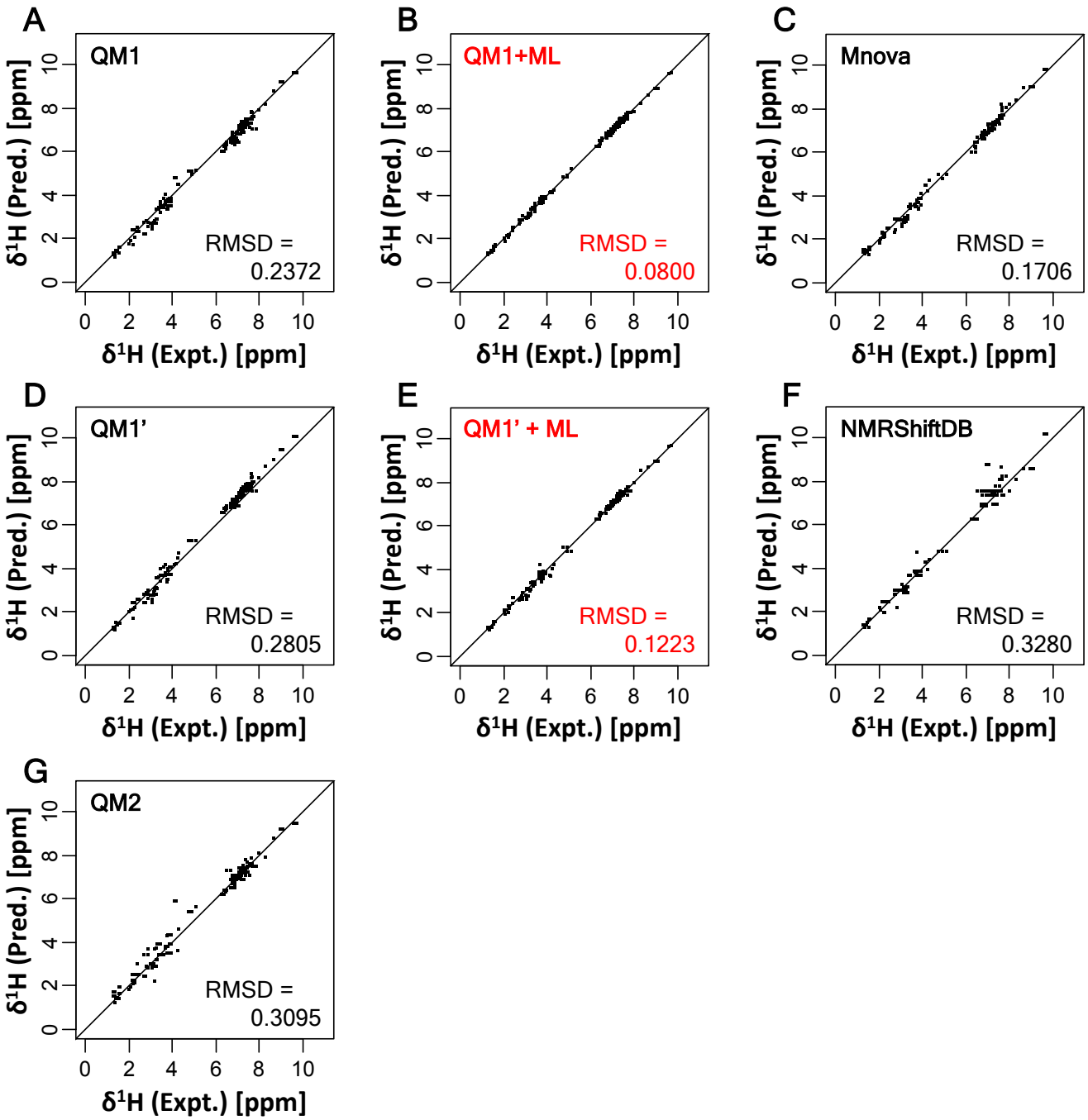


Fig. S13 Comparison of conventional prediction methods based on (**A, D, G**) quantum chemistry and (**C, F**) a data-driven approach with (**B, E**) this study's method. This figure is an expanded version of Fig. 3 with the addition of different calculation level using Gaussian09 software (**D**), QM1' with ML predictive approach (**E**), NMRShiftDB (**F**), and Spartan (**G**) results. Experimental CSs are compared with the calculated δ^1H of 34 compounds in D_2O and MeOD solvent as test data set (Table S4). In total, 256 CSs in 402 CSs of test data set were plotted for δ^1H. These CSs of partial structure were well learned. QM1 shows the theoretical CSs calculated at the B3LYP/6-31G*//GIAO/B3LYP/6-31G* level, and QM1' shows the theoretical CSs calculated at the B3LYP/6-311++G**//GIAO/B3LYP/6-311++G** level using Gaussian09 software. QM1+ML and QM1'+ML show the results of the predictive approach described in this study, in which the ML algorithm xgbLinear calculates an SF that is applied to QM1 and QM1'. QM2 shows the theoretical CSs calculated at the EDF2/6-31G* level using Spartan'14 software. The theoretical CSs of QM2 were corrected with the weighted average using a Boltzmann distribution after conformational analysis.

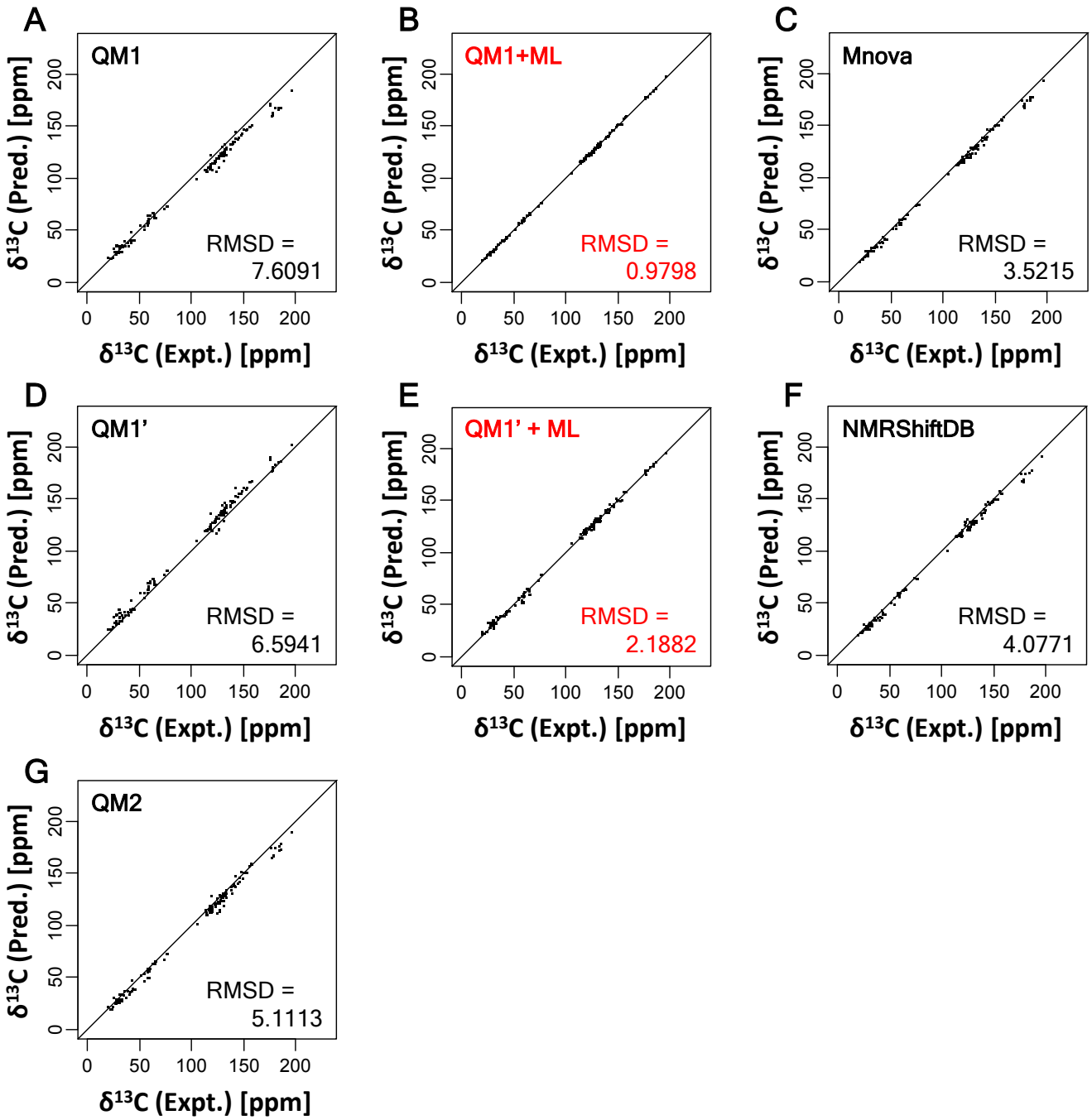


Fig. S14 Comparison of conventional prediction methods based on (**A, D, G**) quantum chemistry and (**C, F**) a data-driven approach with (**B, E**) this study's method. This figure is an expanded version of Fig. 3 with the addition of different calculation level using Gaussian09 software (**D**), QM1' with ML predictive approach (**E**), NMRShiftDB (**F**), and Spartan (**G**) results. Experimental CSs are compared with the calculated $\delta^{13}C$ of 34 compounds in D_2O and MeOD solvent as test data set (Table S4). In total, 216 CSs in 376 CSs of test data set were plotted for $\delta^{13}C$. These CSs of partial structure were well learned. QM1 shows the theoretical CSs calculated at the B3LYP/6-31G*//GIAO/B3LYP/6-31G* level, and QM1' shows the theoretical CSs calculated at the B3LYP/6-311++G**//GIAO/B3LYP/6-311++G** level using Gaussian09 software. QM1+ML and QM1'+ML show the results of the predictive approach described in this study, in which the ML algorithm xgbLinear calculates an SF that is applied to QM1 and QM1'. QM2 shows the theoretical CSs calculated at the EDF2/6-31G* level using Spartan'14 software. The theoretical CSs of QM2 were corrected with the weighted average using a Boltzmann distribution after conformational analysis.

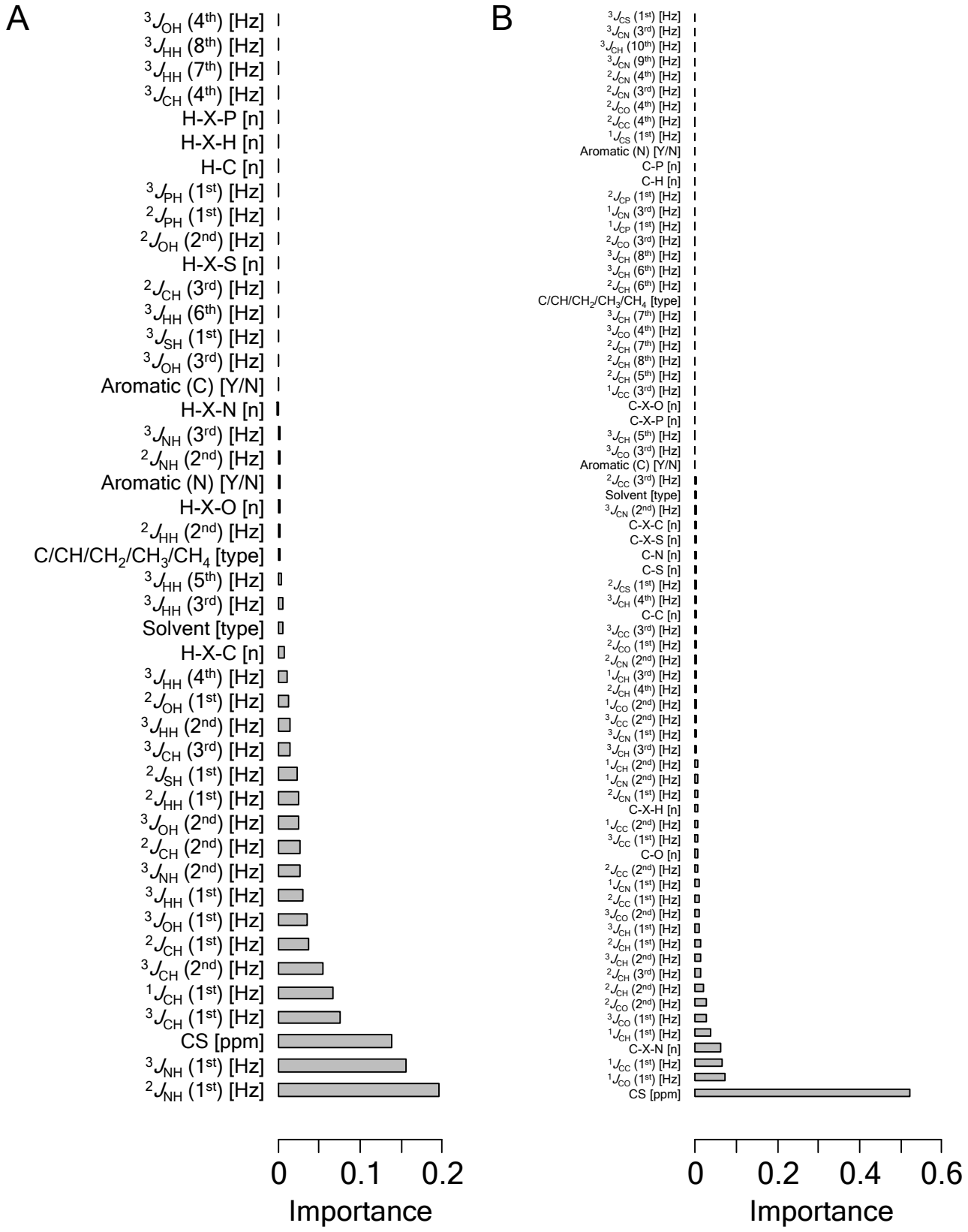


Fig. S15 The importance of explanatory variables in the predictive model based on xgbLinear. (**A**) δ^1H; (**B**) δ^{13}C (see also Fig. 4A and 4B). Interacting nuclides are shown in parentheses. The number in parentheses indicates which explanatory variable of each *J* value was used. The explanatory variables are described in detail in Table S5.

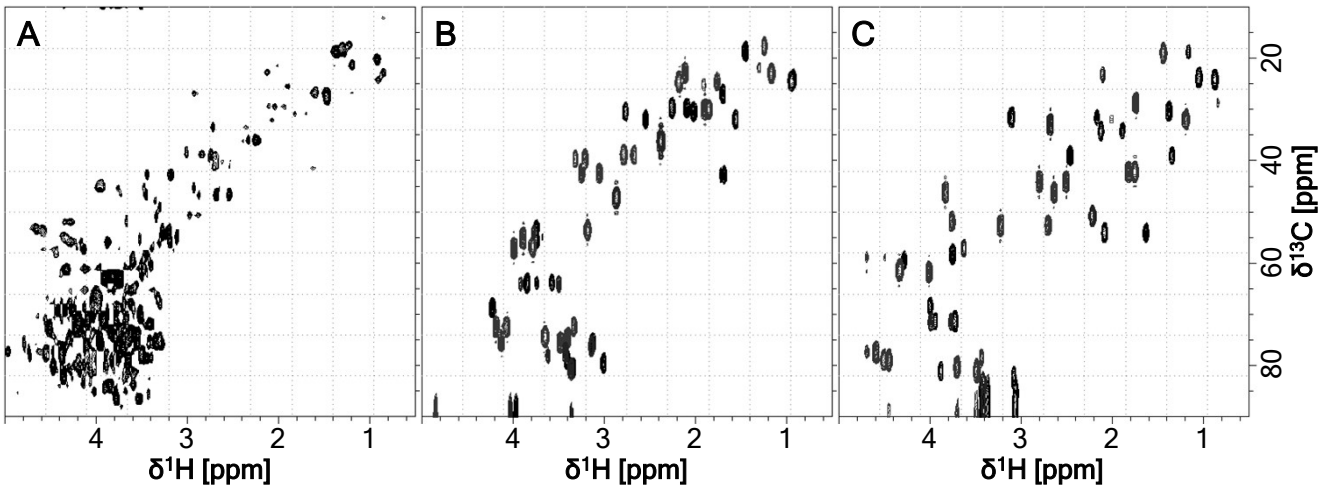


Fig. S16 Test of the predictive model by reproduction of the HSQC spectrum of *C. brachypus* extract (K. Ito et al., *ACS Chem. Biol.* 2016, **11**, 1030–1038). (**A**) Experimental spectrum, and (**B**) corrected and (**C**) uncorrected QM pseudo spectra. The RMSDs between the experimental and predicted CSs are given in Table S2.

Table S1 List of hyperparameters in each model determined by 10-fold CV with a grid search algorithm.

No.	ML Algorithm[1]	Required R library[2]	Hyperparameter[3]	Argument[4]	Optimized variable[5]	
					δ^{1}H pred.	δ^{13}C pred.
1	xgbLinear	xgboost	Boosting Iterations	*nrounds*	150	100
			L2 Regularization	*lambda*	0.0001	0
			L1 Regularization	*alpha*	0.1	0.1
			Learning Rate	*eta*	0.3	0.3
2	kknn	kknn	Max. Neighbors	*kmax*	5	9
			Distance	*distance*	2	2
			Kernel	*kernel*	optimal	optimal
3	extraTrees	extraTrees	Randomly Selected Predictors	*mtry*	23	37
			Random Cuts	*numRandomCuts*	2	2
4	rf	randomForest	Randomly Selected Predictors	*mtry*	23	37
5	parRF	e1071, randomForest, foreach, import	Randomly Selected Predictors	*mtry*	23	37
6	RRFglobal	RRF	Randomly Selected Predictors	*mtry*	23	37
			Regularization Value	*coefReg*	1	0.505
7	RRF	randomForest, RRF	Randomly Selected Predictors	*mtry*	23	37
			Regularization Value	*coefReg*	1	1
			Importance Coefficient	*coefImp*	0.5	0.5
8	xgbTree	xgboost, plyr	Boosting Iterations	*nrounds*	150	150
			Max Tree Depth	*max_depth*	3	3
			Shrinkage	*eta*	0.4	0.3
			Minimum Loss Reduction	*gamma*	0	0
			Subsample Ratio of Columns	*colsample_bytree*	0.8	0.8
			Minimum Sum of Instance Weight	*min_child_weight*	1	1
			Subsample Percentage	*subsample*	1	1
9	cubist	Cubist	Committees	*committees*	20	20
			Instances	*neighbors*	5	5
10	Rborist	Rborist	Randomly Selected Predictors	*predFixed*	23	37
11	bartMachine	bartMachine	Trees	*num_trees*	50	50
			Prior Boundary	*k*	2	2
			Base Terminal Node Hyperparameter	*alpha*	0.9	0.945
			Power Terminal Node Hyperparameter	*beta*	1	1
			Degrees of Freedom	*nu*	4	3
12	SBC	frbs	Radius	*r.a*	0.5	1
			Upper Threshold	*eps.high*	0.5	0.5
			Lower Threshold	*eps.low*	0	0
13	krlsRadial	KRLS, kernlab	Regularization Parameter	*lambda*	NA	NA
			Sigma	*sigma*	22.27361	41.80515
14	rvmRadial	kernlab	Sigma	*sigma*	0.006201054	0.01439434
15	ppr	stats	Terms	*nterms*	3	3

[1] The algorithm name was set to the method argument in the train function of the caret library in R.
[2] R libraries other than the caret library are called by the caret library in the background for using each ML algorithm.
[3] Hyperparameters available for tuning.
[4] Hyperparameter argument in the train function of the caret library in R.
[5] Up to three variables of each hyperparameter argument were used for tuning. These variables were generated automatically by the tuneLength argument in the train function of the caret library in R. The RMSD of 10-fold CV was calculated when each grid was combined, and the combined model that had the lowest RMSD (Fig. S4) was ultimately chosen as the optimal model. Hyperparameters of the ML models were optimized for QM at the B3LYP/6-31G* level.

Table S1 Continued.

No.	ML Algorithm[1]	Required R library[2]	Hyperparameter[3]	Argument[4]	Optimized variable[5]	
					δ^{1}H pred.	δ^{13}C pred.
16	rvmPoly	kernlab	Scale Polynomial Degree	*scale* *degree*	0.01 2	0.001 3
17	brnn	brnn	Neurons	*neurons*	3	3
18	svmRadialCost	kernlab	Cost	*C*	1	1
19	svmRadial	kernlab	Sigma Cost	*sigma* *C*	0.006440645 1	0.01386974 1
20	svmRadialSigma	kernlab	Sigma Cost	*sigma* *C*	0.00984396 1	0.01284332 1
21	WM	frbs	Fuzzy Terms Membership Function	*num.labels* *type.mf*	7 GAUSSIAN	7 GAUSSIAN
22	gbm	gbm, plyr	Boosting Iterations Max Tree Depth Shrinkage Min. Terminal Node Size	*n.trees* *interaction.depth* *shrinkage* *n.minobsinnode*	150 3 0.1 10	150 3 0.1 10
23	svmPoly	kernlab	Polynomial Degree Scale Cost	*degree* *scale* *C*	3 0.001 1	2 0.01 1
24	knn	class	Neighbors	*k*	5	5
25	qrf	quantregForest	Randomly Selected Predictors	*mtry*	23	73
26	gaussprRadial	kernlab	Sigma	*sigma*	0.005041228	0.01328753
27	gaussprPoly	kernlab	Polynomial Degree Scale	*degree* *scale*	2 0.1	2 0.01
28	evtree	evtree	Complexity Parameter	*alpha*	1	1
29	cforest	party	Randomly Selected Predictors	*mtry*	45	37
30	bagEarthGCV	earth	Product Degree	*degree*	1	1
31	glm	stats	-	-	-	-
32	lm	stats	-	-	-	-
33	enet	elasticnet	Fraction of Full Solution Weight Decay	*fraction* *lambda*	1 0.0001	1 0
34	foba	foba	Variables Retained L2 Penalty	*k* *lambda*	45 0.00001	73 0.00001
35	bayesglm	arm	-	-	-	-
36	lasso	elasticnet	Fraction of Full Solution	*fraction*	0.9	0.9
37	gaussprLinear	kernlab	-	-	-	-
38	glmStepAIC	MASS	-	-	-	-
39	lmStepAIC	MASS	-	-	-	-
40	glmnet	glmnet, Matrix	Mixing Percentage Regularization Parameter	*alpha* *lambda*	1 0.021535089	1 0.009389168
41	ctree	party	1 - P-Value Threshold	*mincriterion*	0.01	0.01
42	bagEarth	earth	Terms Product Degree	*nprune* *degree*	18 1	25 1
43	gcvEarth	earth	Product Degree	*degree*	1	1
44	earth	earth	Terms Product Degree	*nprune* *degree*	18 1	25 1

Table S1 Continued.

No.	ML Algorithm[1]	Required R library[2]	Hyperparameter[3]	Argument[4]	Optimized variable[5]	
					δ^{1}H pred.	δ^{13}C pred.
45	penalized	penalized	L1 Penalty L2 Penalty	*lambda1* *lambda2*	1 1	1 1
46	blassoAveraged	monomvn	-	-	-	-
47	bridge	monomvn	-	-	-	-
48	blasso	monomvn	Sparsity Threshold	*sparsity*	0.3	0.3
49	lars	lars	Fraction	*fraction*	1	0.525
50	svmLinear2	e1071	Cost	*cost*	1	1
51	svmLinear	kernlab	Cost	*C*	1	1
52	rqlasso	rqPen	L1 Penalty	*lambda*	0.0075	0.0001
53	rqnc	rqPen	L1 Penalty Penalty Type	*lambda* *penalty*	0.1 MCP	0.0001 MCP
54	lars2	lars	Steps	*step*	45	73
55	nodeHarvest	nodeHarvest	Maximum Interaction Depth Prediction Mode	*maxinter* *mode*	3 mean	3 mean
56	plsRglm	plsRglm	PLS Components p-Value threshold	*nt* *alpha.pvals.expli*	3 0.01	3 1
57	ctree2	party	Max Tree Depth 1 - P-Value Threshold	*maxdepth* *mincriterion*	3 0.01	3 0.99
58	glmboost	plyr, mboost	Boosting Iterations AIC Prune	*mstop* *prune*	150 no	150 no
59	rvmLinear	kernlab	-	-	-	-
60	svmLinear3	LiblineaR	Cost Loss Function	*cost* *Loss*	0.25 L2	0.25 L1
61	BstLm	bst, plyr	Boosting Iterations Shrinkage	*mstop* *nu*	150 0.1	150 0.1
62	simpls	pls	Components	*ncomp*	3	3
63	pls	pls	Components	*ncomp*	3	3
64	widekernelpls	pls	Components	*ncomp*	3	3
65	kernelpls	pls	Components	*ncomp*	3	3
66	partDSA	partDSA	Cut off growth	*cut.off.growth*	3	3
67	nnls	nnls	-	-	-	-
68	pcr	pls	Components	*ncomp*	3	3
69	leapForward	leaps	Maximum Number of Predictors	*nvmax*	4	4
70	icr	fastICA	Components	*n.comp*	3	3
71	elm	elmNN	Hidden Units Activation Function	*nhid* *actfun*	5 purelin	5 purelin
72	relaxo	relaxo, plyr	Penalty Parameter Relaxation Parameter	*lambda* *phi*	1.685195 0.1	145009.446 0.9
73	leapBackward	leaps	Maximum Number of Predictors	*nvmax*	4	4
74	leapSeq	leaps	Maximum Number of Predictors	*nvmax*	4	4
75	superpc	superpc	Threshold Components	*threshold* *n.components*	0.9 3	0.1 3

Table S1 Continued.

No.	ML Algorithm[1]	Required R library[2]	Hyperparameter[3]	Argument[4]	Optimized variable[5]	
					δ^{1}H pred.	δ^{13}C pred.
76	krlsPoly	KRLS	Regularization Parameter Polynomial Degree	*lambda* *degree*	NA 2	NA 1
77	rbf	RSNNS	Hidden Units	*size*	5	5
78	pcaNNet	nnet	Hidden Units Weight Decay	*size* *decay*	5 0.1	5 0.1
79	nnet	nnet	Hidden Units Weight Decay	*size* *decay*	5 0.1	5 0.1
80	avNNet	nnet	Hidden Units Weight Decay Bagging	*size* *decay* *bag*	5 0.1 FALSE	1 0.1 FALSE
81	dnn	deepnet	Hidden Layer 1 Hidden Layer 2 Hidden Layer 3 Hidden Dropouts Visible Dropout	*layer1* *layer2* *layer3* *hidden_dropout* *visible_dropout*	2 1 0 0 0	2 1 2 0 0
82	mlp	RSNNS	Hidden Units	*size*	3	1
83	mlpWeightDecay	RSNNS	Hidden Units Weight Decay	*size* *decay*	1 0.1	3 0.0001
84	mlpWeightDecayML	RSNNS	Hidden Units layer1 Hidden Units layer2 Hidden Units layer3 Weight Decay	*layer1* *layer2* *layer3* *decay*	3 0 0 0.0001	1 0 0 0.1
85	rbfDDA	RSNNS	Activation Limit for Conflicting Classes	*negativeThreshold*	0.001	0.001
86	mlpSGD	FCNN4R, plyr	Hidden Units L2 Regularization RMSE Gradient Scaling Learning Rate Momentum Learning Rate Decay Batch Size Models	*size* *l2reg* *lambda* *learn_rate* *momentum* *gamma* *minibatchsz* *repeats*	3 0 0 0.000002 0.9 0.001 425 1	5 0.0001 0 0.000002 0.9 0.001 359 1
87	mlpML	RSNNS	Hidden Units layer1 Hidden Units layer2 Hidden Units layer3	*layer1* *layer2* *layer3*	1 0 0	1 0 0
88	rfRules	randomForest, inTrees, plyr	Randomly Selected Predictors Maximum Rule Depth	*mtry* *maxdepth*	2 2	2 2
89	DENFIS	frbs	Threshold Max. Iterations	*Dthr* *max.iter*	0.1 100	0.3 100
90	ANFIS	frbs	Fuzzy Terms Max. Iterations	*num.labels* *max.iter*	7 10	3 10
91	randomGLM	randomGLM	Interaction Order	*maxInteractionOrder*	1	1

Table S2 List of RMSDs between the experimental and theoretical/predicted CSs of metabolites in *C. brachypus.* Reprinted with permission from our previous report (K. Ito et al., *ACS Chem. Biol.* 2016, **11**, 1030–1038). Copyright 2016 American Chemical Society.

Metabolites	Most stable structure*		Ionization structure*		Boltzmann distribution*		Regression*		This study's method	
	^{1}H	^{13}C	^{1}H	^{13}C	^{1}H	^{13}C	^{1}H	^{13}C	^{1}H	^{13}C
Citrulline	0.188	6.700	0.250	8.458	0.196	6.398	0.238	3.052	0.142	3.074
L-Alanine	0.121	5.237	0.157	6.666	0.120	5.252	0.186	1.632	0.080	1.164
L-Arginine	0.195	6.713	0.349	12.438	0.191	6.181	0.245	3.073	0.226	1.561
L-Aspartic acid	0.111	4.143	0.495	5.341	0.112	4.148	0.126	0.419	0.077	0.346
L-Glutamic acid	0.440	4.467	0.857	3.052	0.040	4.880	0.450	2.687	0.056	2.448
L-Leucine	0.170	4.332	0.127	5.080	0.167	4.351	0.202	2.573	0.075	1.086
L-Threonine	0.497	6.062	0.483	5.881	0.484	5.676	0.569	2.527	0.073	0.733
3-Phosphoglyceric acid	0.385	8.913	0.338	5.865	0.556	8.725	0.242	7.233	0.092	2.082
Acetic acid	0.234	2.786	0.133	2.524	0.235	2.741	0.184	6.687	0.010	0.892
Formic acid	0.322	1.279	0.882	4.238	0.269	1.293	0.612	6.185	0.651	9.298
Methylmalonic acid	0.408	5.132	0.202	6.176	0.383	4.044	0.345	6.559	0.055	1.923
Phosphoenolpyruvic acid	0.235	1.452	0.405	11.940	0.304	1.971	0.052	3.001	0.589	11.226
Succinate	0.316	4.053	0.170	9.286	0.339	2.401	0.243	8.018	0.012	0.520
α-D-Glucose	0.273	6.316	-	-	0.241	6.484	0.208	2.407	0.204	1.371
β-D-Glucose	0.196	5.003	-	-	0.368	6.727	0.117	0.975	0.285	2.927
β-D-Glucuronate	0.580	4.270	0.374	5.238	0.594	4.187	0.447	3.655	0.291	3.763
Methanol	0.418	5.558	-	-	-	-	0.302	1.426	0.440	2.326
Trimethylamine	0.617	3.503	-	-	-	-	0.672	0.587	0.001	0.160

* our previous report

Table S3 List of 150 compounds included in the training data set for predictive modeling by ML.

No.	CID	Compound	Formula	MW
1	6329	Methylamine	CH_5N	31.058
2	674	Dimethylamine	C_2H_7N	45.085
3	702	Ethanol	C_2H_6O	46.069
4	7855	Acrylonitrile	C_3H_3N	53.064
5	178	Acetamide	C_2H_5NO	59.068
6	1146	Trimethylamine	C_3H_9N	59.112
7	176	Acetate	$C_2H_4O_2$	60.052
8	700	Ethanolamine	C_2H_7NO	61.084
9	174	Ethylene glycol	$C_2H_6O_2$	62.068
10	1647	3-Aminopropiononitrile	$C_3H_6N_2$	70.095
11	6579	Acrylamide	C_3H_5NO	71.079
12	6581	Acrylic acid	$C_3H_4O_2$	72.063
13	6569	Methyl ethyl ketone	C_4H_8O	72.107
14	10111	Methylguanidine	$C_2H_7N_3$	73.099
15	760	Glyoxylate	$C_2H_2O_3$	74.035
16	1032	Propanoate	$C_3H_6O_2$	74.079
17	428	1,3-Diaminopropane	$C_3H_{10}N_2$	74.127
18	750	Glycine	$C_2H_5NO_2$	75.067
19	1145	Trimethylamine N-oxide	C_3H_9NO	75.111
20	439938	(R)-1-Aminopropan-2-ol	C_3H_9NO	75.111
21	757	Glycolate	$C_2H_4O_3$	76.051
22	1030	Propane-1,2-diol	$C_3H_8O_2$	76.095
23	10484	Thioacetate	C_2H_4OS	76.113
24	6058	Cysteamine	C_2H_7NS	77.145
25	9260	Pyrimidine	$C_4H_4N_2$	80.090
26	61020	3-Methyl-2-butenal	C_5H_8O	84.118
27	4837	Piperazine	$C_4H_{10}N_2$	86.138
28	1060	Pyruvate	$C_3H_4O_3$	88.062
29	264	Butanoic acid	$C_4H_8O_2$	88.106
30	6590	2-Methylpropanoate	$C_4H_8O_2$	88.106
31	1045	Putrescine	$C_4H_{12}N_2$	88.154
32	5950	L-Alanine	$C_3H_7NO_2$	89.094
33	239	β-Alanine	$C_3H_7NO_2$	89.094
34	1088	Sarcosine	$C_3H_7NO_2$	89.094
35	398	2-Nitropropane	$C_3H_7NO_2$	89.094
36	670	Glycerone	$C_3H_6O_3$	90.078
37	107689	(S)-Lactate	$C_3H_6O_3$	90.078
38	61503	(R)-Lactate	$C_3H_6O_3$	90.078
39	753	Glycerol	$C_3H_8O_3$	92.094
40	1133	Thioglycolate	$C_2H_4O_2S$	92.112
41	6115	Aniline	C_6H_7N	93.129
42	8871	2-Hydroxypyridine	C_5H_5NO	95.101
43	125468	Tiglic acid	$C_5H_8O_2$	100.117
44	5281167	3-Hexenol	$C_6H_{12}O$	100.161
45	535	1-Aminocyclopropane-1-carboxylate	$C_4H_7NO_2$	101.105
46	8102	Hexylamine	$C_6H_{15}N$	101.193
47	96	Acetoacetate	$C_4H_6O_3$	102.089
48	7991	Pentanoate	$C_5H_{10}O_2$	102.133
49	10430	3-Methylbutanoic acid	$C_5H_{10}O_2$	102.133
50	273	Cadaverine	$C_5H_{14}N_2$	102.181

Table S3 Continued.

No.	CID	Compound	Formula	MW
51	119	4-Aminobutanoate	$C_4H_9NO_2$	103.121
52	673	N,N-Dimethylglycine	$C_4H_9NO_2$	103.121
53	80283	(S)-2-Aminobutanoate	$C_4H_9NO_2$	103.121
54	5288725	N-Methyl-L-alanine	$C_4H_9NO_2$	103.121
55	439434	L-3-Aminoisobutanoate	$C_4H_9NO_2$	103.121
56	6119	2-Amino-2-methylpropanoate	$C_4H_9NO_2$	103.121
57	92135	(R)-3-Hydroxybutanoate	$C_4H_8O_3$	104.105
58	440864	2-Hydroxybutanoic acid	$C_4H_8O_3$	104.105
59	364	2,3-Diaminopropanoate	$C_3H_8N_2O_2$	104.109
60	305	Choline	$C_5H_{14}NO+$	104.173
61	5951	L-Serine	$C_3H_7NO_3$	105.093
62	8113	Diethanolamine	$C_4H_{11}NO_2$	105.137
63	439194	D-Glycerate	$C_3H_6O_4$	106.077
64	240	Aromatic aldehyde	C_7H_6O	106.124
65	2879	4-Cresol	C_7H_8O	108.140
66	403	4-Hydroxyaniline	C_6H_7NO	109.128
67	107812	Hypotaurine	$C_2H_7NO_2S$	109.143
68	289	Catechol	$C_6H_6O_2$	110.112
69	597	Cytosine	$C_4H_5N_3O$	111.104
70	774	Histamine	$C_5H_9N_3$	111.148
71	1174	Uracil	$C_4H_4N_2O_2$	112.088
72	588	Creatinine	$C_4H_7N_3O$	113.120
73	649	5,6-Dihydrouracil	$C_4H_6N_2O_2$	114.104
74	145742	L-Proline	$C_5H_9NO_2$	115.132
75	444972	Fumarate	$C_4H_4O_4$	116.072
76	444266	Maleic acid	$C_4H_4O_4$	116.072
77	49	3-Methyl-2-oxobutanoic acid	$C_5H_8O_3$	116.116
78	74563	2-Oxopentanoic acid	$C_5H_8O_3$	116.116
79	8892	Hexanoic acid	$C_6H_{12}O_2$	116.160
80	763	Guanidinoacetate	$C_3H_7N_3O_2$	117.108
81	6287	L-Valine	$C_5H_{11}NO_2$	117.148
82	138	5-Aminopentanoate	$C_5H_{11}NO_2$	117.148
83	798	Indole	C_8H_7N	117.151
84	1110	Succinate	$C_4H_6O_4$	118.088
85	487	Methylmalonate	$C_4H_6O_4$	118.088
86	248	Trimethyl glycine	$C_5H_{12}NO_2+$	118.156
87	6288	L-Threonine	$C_4H_9NO_3$	119.120
88	12647	L-Homoserine	$C_4H_9NO_3$	119.120
89	439656	2-Methylserine	$C_4H_9NO_3$	119.120
90	99289	L-Allothreonine	$C_4H_9NO_3$	119.120
91	2332	Benzamidine	$C_7H_8N_2$	120.155
92	5862	L-Cysteine	$C_3H_7NO_2S$	121.154
93	92851	D-Cysteine	$C_3H_7NO_2S$	121.154
94	8998	Butane-1,2,3,4-tetrol	$C_4H_{10}O_4$	122.120
95	243	Benzoate	$C_7H_6O_2$	122.123
96	126	4-Hydroxybenzaldehyde	$C_7H_6O_2$	122.123
97	936	Nicotinamide	$C_6H_6N_2O$	122.127
98	938	Nicotinate	$C_6H_5NO_2$	123.111
99	5922	Isonicotinic acid	$C_6H_5NO_2$	123.111
100	460	o-Methoxyphenol	$C_7H_8O_2$	124.139

Table S3 Continued.

No.	CID	Compound	Formula	MW
101	339	2-Aminoethylphosphonate	$C_2H_8NO_3P$	125.064
102	185992	D-(1-Aminoethyl)phosphonate	$C_2H_8NO_3P$	125.064
103	65040	5-Methylcytosine	$C_5H_7N_3O$	125.131
104	1123	Taurine	$C_2H_7NO_3S$	125.142
105	3614	N-Methylhistamine	$C_6H_{11}N_3$	125.175
106	1135	Thymine	$C_5H_6N_2O_2$	126.115
107	7405	Pidolic acid	$C_5H_7NO_3$	129.115
108	439227	L-Pipecolate	$C_6H_{11}NO_2$	129.159
109	2901	1-Aminocyclopentanecarboxylate	$C_6H_{11}NO_2$	129.159
110	4091	Metformin	$C_4H_{11}N_5$	129.167
111	811	Itaconate	$C_5H_6O_4$	130.099
112	638129	Mesaconate	$C_5H_6O_4$	130.099
113	643798	2-Methylmaleate	$C_5H_6O_4$	130.099
114	47	3-Methyl-2-oxopentanoate	$C_6H_{10}O_3$	130.143
115	199	Agmatine	$C_5H_{14}N_4$	130.195
116	137	5-Aminolevulinate	$C_5H_9NO_3$	131.131
117	5810	Hydroxyproline	$C_5H_9NO_3$	131.131
118	586	Creatine	$C_4H_9N_3O_2$	131.135
119	67701	3-Guanidinopropanoate	$C_4H_9N_3O_2$	131.135
120	6106	L-Leucine	$C_6H_{13}NO_2$	131.175
121	6306	L-Isoleucine	$C_6H_{13}NO_2$	131.175
122	21236	L-Norleucine	$C_6H_{13}NO_2$	131.175
123	564	6-Aminohexanoate	$C_6H_{13}NO_2$	131.175
124	439734	(s)-3-Amino-4-methylpentanoic acid	$C_6H_{13}NO_2$	131.175
125	94206	D-Alloisoleucine	$C_6H_{13}NO_2$	131.175
126	743	Glutarate	$C_5H_8O_4$	132.115
127	6267	L-Asparagine	$C_4H_8N_2O_3$	132.119
128	11163	Glycylglycine	$C_4H_8N_2O_3$	132.119
129	111	3-Ureidopropionate	$C_4H_8N_2O_3$	132.119
130	439960	(R)-2-Hydroxyisocaproate	$C_6H_{12}O_3$	132.159
131	637511	Cinnamaldehyde	C_9H_8O	132.162
132	6262	L-Ornithine	$C_5H_{12}N_2O_2$	132.163
133	5960	L-Aspartate	$C_4H_7NO_4$	133.103
134	222656	(S)-Malate	$C_4H_6O_5$	134.087
135	92824	(R)-Malate	$C_4H_6O_5$	134.087
136	439576	β-D-2-Deoxyribose	$C_5H_{10}O_4$	134.131
137	5280511	(Z)-Cinnamyl alcohol	$C_9H_{10}O$	134.178
138	190	Adenine	$C_5H_5N_5$	135.130
139	31229	Phenyl acetate	$C_8H_8O_2$	136.150
140	5353895	1-Methyl-2-(nitrosomethylidene)pyridine	$C_7H_8N_2O$	136.154
141	227	Anthranilate	$C_7H_7NO_2$	137.138
142	978	4-Aminobenzoate	$C_7H_7NO_2$	137.138
143	3767	Isoniazid	$C_6H_7N_3O$	137.142
144	457	1-Methylnicotinamide	$C_7H_9N_2O+$	137.162
145	5610	Tyramine	$C_8H_{11}NO$	137.182
146	135	4-Hydroxybenzoate	$C_7H_6O_3$	138.122
147	338	Salicylate	$C_7H_6O_3$	138.122
148	736715	Urocanate	$C_6H_6N_2O_2$	138.126
149	5571	N-Methylnicotinic acid	$C_7H_8NO_2+$	138.146
150	980	4-Nitrophenol	$C_6H_5NO_3$	139.110

Table S4 List of 34 compounds included in the test data set which does not include the learning and k-fold validation (training) data set of ML.

No.	CID	Compound	Formula	MW
1	441696	(S)-2-Methylmalate	$C_5H_8O_5$	148.114
2	444539	trans-Cinnamate	$C_9H_8O_2$	148.161
3	7618	Triethanolamine	$C_6H_{15}NO_3$	149.190
4	5852	Penicillamine	$C_5H_{11}NO_2S$	149.208
5	875	2,3-dihydroxybutanedioic acid	$C_4H_6O_6$	150.086
6	439655	(S,S)-Tartaric acid	$C_4H_6O_6$	150.086
7	827	Xylitol	$C_5H_{12}O_5$	152.146
8	127	4-Hydroxyphenylacetate	$C_8H_8O_3$	152.149
9	1183	4-Hydroxy-3-methoxy-benzaldehyde	$C_8H_8O_3$	152.149
10	11914	(R)-Mandelate	$C_8H_8O_3$	152.149
11	11970	2-Hydroxyphenylacetate	$C_8H_8O_3$	152.149
12	72	3,4-Dihydroxybenzoate	$C_7H_6O_4$	154.121
13	3469	2,5-Dihydroxybenzoate	$C_7H_6O_4$	154.121
14	6274	L-Histidine	$C_6H_9N_3O_2$	155.157
15	93	3-Oxoadipate	$C_6H_8O_5$	160.125
16	385	Pimelate	$C_7H_{12}O_4$	160.169
17	5460362	D-alanyl-D-alanine	$C_6H_{12}N_2O_3$	160.173
18	1150	Tryptamine	$C_{10}H_{12}N_2$	160.220
19	439389	O-Acetyl-L-homoserine	$C_6H_{11}NO_4$	161.157
20	581	N-Acetyl-DL-cysteine	$C_5H_9NO_3S$	163.191
21	997	Phenylpyruvate	$C_9H_8O_3$	164.160
22	637540	trans-2-Hydroxycinnamate	$C_9H_8O_3$	164.160
23	637541	trans-3-Hydroxycinnamate	$C_9H_8O_3$	164.160
24	637542	4-Coumarate	$C_9H_8O_3$	164.160
25	6140	L-Phenylalanine	$C_9H_{11}NO_2$	165.192
26	1066	Quinolinate	$C_7H_5NO_4$	167.120
27	6041	Phenylephrine	$C_9H_{13}NO_2$	167.208
28	547	3,4-Dihydroxyphenylacetate	$C_8H_8O_4$	168.148
29	1054	Pyridoxine	$C_8H_{11}NO_3$	169.180
30	64969	N(pi)-Methyl-L-histidine	$C_7H_{11}N_3O_2$	169.184
31	10457	Suberic acid	$C_8H_{14}O_4$	174.196
32	5281416	Esculetin	$C_9H_6O_4$	178.143
33	1318	1,10-Phenanthroline	$C_{12}H_8N_2$	180.210
34	6057	L-Tyrosine	$C_9H_{11}NO_3$	181.191

Table S5 List of the objective and explanatory variables used to explore the scaling factor calculated by the 91 ML algorithms. These variables were collected and were generated as a training data set automatically from log files Gaussian09 software by using a Java program. The example is shown in Table S6.

No.	Common Objective Variable (Descriptor)
0	Difference between the experimental and theoretical CS [ppm]

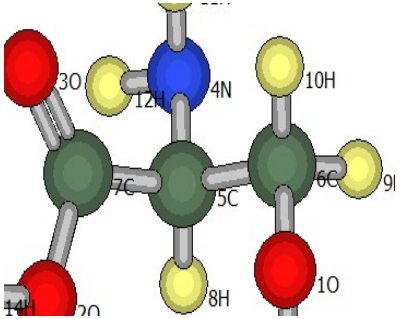
No.	Common Explanatory Variables (Descriptor)
1	Theoretical CS [ppm]
2	C–H bond type (C =1, CH = 2, CH_2 =3, CH_3 = 4, CH_4 = 5)
3-8	Number of the bonded atoms, like H-X or C-X (X = C, H, N, O, P, S) [n]
9-14	Number of the second atoms, like H-X-Y or C-X-Y (Y = C, H, N, O, P, S) [n]
15	Solvent (D_2O = 1, MeOD = 2)
16-18	Aromatic ring and include C or O or N (Yes = 1, No = 0)
19	Pyranose type (Yes = 1, No = 0)

No.	Explanatory Variables for δ^1H (Descriptor)
20	Theoretical $^1J_{HC}$ [Hz]
21-23	Theoretical $^2J_{HC}$ [Hz]
24-26	Theoretical $^2J_{HH}$ [Hz]
27-29	Theoretical $^2J_{HO}$ [Hz]
30-32	Theoretical $^2J_{HN}$ [Hz]
33-35	Theoretical $^2J_{HP}$ [Hz]
36-38	Theoretical $^2J_{HS}$ [Hz]
39-47	Theoretical $^3J_{HC}$ [Hz]
48-56	Theoretical $^3J_{HH}$ [Hz]
57-65	Theoretical $^3J_{HO}$ [Hz]
66-74	Theoretical $^3J_{HN}$ [Hz]
75-83	Theoretical $^3J_{HP}$ [Hz]
84-92	Theoretical $^3J_{HS}$ [Hz]

No.	Explanatory Variables for δ^{13}C (Descriptor)
20-23	Theoretical $^1J_{CC}$ [Hz]
24-27	Theoretical $^1J_{CH}$ [Hz]
28-31	Theoretical $^1J_{CO}$ [Hz]
32-35	Theoretical $^1J_{CN}$ [Hz]
36-39	Theoretical $^1J_{CP}$ [Hz]
40-43	Theoretical $^1J_{CS}$ [Hz]
44-55	Theoretical $^2J_{CC}$ [Hz]
56-67	Theoretical $^2J_{CH}$ [Hz]
68-79	Theoretical $^2J_{CO}$ [Hz]
80-91	Theoretical $^2J_{CN}$ [Hz]
92-101	Theoretical $^2J_{CP}$ [Hz]
102-111	Theoretical $^2J_{CS}$ [Hz]
112-131	Theoretical $^3J_{CC}$ [Hz]
132-151	Theoretical $^3J_{CH}$ [Hz]
152-171	Theoretical $^3J_{CO}$ [Hz]
172-191	Theoretical $^3J_{CN}$ [Hz]
192-201	Theoretical $^3J_{CP}$ [Hz]
202-211	Theoretical $^3J_{CS}$ [Hz]

* Theoretical *J* values were sorted in ascending order.

Table S6 Part of the data set for ML. This is an example of the prediction of δ^1H for serine



Variable No. →	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
No.	metid	Atom No.	Atom label	Chemical shift(expt)(ppm)	Diff(ppm)	Chemical shift(theor)(ppm)	C/CH/CH2/CH3/CH4	Bonded (C)	Bonded (H)	Bonded (O)	Bonded (N)	Bonded (P)	Bonded (S)	Bonded Bonded (C)	Bonded Bonded (H)	Bonded Bonded (O)	Bonded Bonded (N)	Bonded Bonded (P)	Bonded Bonded (S)	Solvent	Aromatic (C)	Aromatic (O)	Aromatic (N)	Pyranose	1J(C)_1
***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***
517	5951	8	H	3.8325	0.53401111	3.2984889	2	1	0	0	0	0	0	2	0	0	1	0	0	1	0	0	0	0	149.543
518	5951	9	H	3.9545	−0.01528889	3.9697889	3	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	147.381
519	5951	10	H	3.9545	0.51061111	3.4438889	3	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	147.36
***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***

21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
2J(C)_1	2J(C)_2	2J(C)_3	2J(H)_1	2J(H)_2	2J(H)_3	2J(O)_1	2J(O)_2	2J(O)_3	2J(N)_1	2J(N)_2	2J(N)_3	2J(P)_1	2J(P)_2	2J(P)_3	2J(S)_1	2J(S)_2	2J(S)_3	3J(C)_1	3J(C)_2	3J(C)_3	3J(C)_4	3J(C)_5	3J(C)_6	3J(C)_7	3J(C)_8	3J(C)_9	3J(H)_1
***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***
−0.855587	−3.28315	0	0	0	0	0	0	0	1.93036	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11.8145
2.59279	0	0	−13.0179	0	0	−2.49382	0	0	0	0	0	0	0	0	0	0	0	7.87132	0	0	0	0	0	0	0	0	5.34561
−1.32471	0	0	−13.0179	0	0	−16.2875	0	0	0	0	0	0	0	0	0	0	0	2.27639	0	0	0	0	0	0	0	0	13.0066
***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***

49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76
3J(H)_2	3J(H)_3	3J(H)_4	3J(H)_5	3J(H)_6	3J(H)_7	3J(H)_8	3J(H)_9	3J(O)_1	3J(O)_2	3J(O)_3	3J(O)_4	3J(O)_5	3J(O)_6	3J(O)_7	3J(O)_8	3J(O)_9	3J(N)_1	3J(N)_2	3J(N)_3	3J(N)_4	3J(N)_5	3J(N)_6	3J(N)_7	3J(N)_8	3J(N)_9	3J(P)_1	3J(P)_2
***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***
9.46186	5.34561	4.04369	0	0	0	0	0	0.343996	0.164684	−0.14637	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.91076	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.106445	0	0	0	0	0	0	0	0	0	0
9.46186	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.689154	0	0	0	0	0	0	0	0	0	0
***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***

77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92
3J(P)_3	3J(P)_4	3J(P)_5	3J(P)_6	3J(P)_7	3J(P)_8	3J(P)_9	3J(S)_1	3J(S)_2	3J(S)_3	3J(S)_4	3J(S)_5	3J(S)_6	3J(S)_7	3J(S)_8	3J(S)_9
***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***

Table S7 List of the 91 ML algorithms used to explore an effective predictive model. The ML definition used in the caret library is shown in parentheses.

No.	Machine Learning Algorithm
1	eXtreme Gradient Boosting (xgbLinear)
2	k-Nearest Neighbors (kknn)
3	Random Forest by Randomization (extraTrees)
4	Random Forest (rf)
5	Parallel Random Forest (parRF)
6	Regularized Random Forest (RRFglobal)
7	Regularized Random Forest (RRF)
8	eXtreme Gradient Boosting (xgbTree)
9	Cubist (cubist)
10	Random Forest (Rborist)
11	Bayesian Additive Regression Trees (bartMachine)
12	Subtractive Clustering and Fuzzy c-Means Rules (SBC)
13	Radial Basis Function Kernel Regularized Least Squares (krlsRadial)
14	Relevance Vector Machines with Radial Basis Function Kernel (rvmRadial)
15	Projection Pursuit Regression (ppr)
16	Relevance Vector Machines with Polynomial Kernel (rvmPoly)
17	Bayesian Regularized Neural Networks (brnn)
18	Support Vector Machines with Radial Basis Function Kernel (svmRadialCost)
19	Support Vector Machines with Radial Basis Function Kernel (svmRadial)
20	Support Vector Machines with Radial Basis Function Kernel (svmRadialSigma)
21	Wang and Mendel Fuzzy Rules (WM)
22	Stochastic Gradient Boosting (gbm)
23	Support Vector Machines with Polynomial Kernel (svmPoly)
24	k-Nearest Neighbors (knn)
25	Quantile Random Forest (qrf)
26	Gaussian Process with Radial Basis Function Kernel (gaussprRadial)
27	Gaussian Process with Polynomial Kernel (gaussprPoly)
28	Tree Models from Genetic Algorithms (evtree)
29	Conditional Inference Random Forest (cforest)
30	Bagged MARS using gCV Pruning (bagEarthGCV)
31	Generalized Linear Model (glm)
32	Linear Regression (lm)
33	Elasticnet (enet)
34	Ridge Regression with Variable Selection (foba)
35	Bayesian Generalized Linear Model (bayesglm)
36	The lasso (lasso)
37	Gaussian Process (gaussprLinear)
38	Generalized Linear Model with Stepwise Feature Selection (glmStepAIC)
39	Linear Regression with Stepwise Selection (lmStepAIC)
40	glmnet (glmnet)
41	Conditional Inference Tree (ctree)
42	Bagged MARS (bagEarth)
43	Multivariate Adaptive Regression Splines (gcvEarth)
44	Multivariate Adaptive Regression Spline (earth)
45	Penalized Linear Regression (penalized)

Table S7 Continued.

No.	Machine Learning Algorithm
46	Bayesian Ridge Regression (Model Averaged) (blassoAveraged)
47	Bayesian Ridge Regression (bridge)
48	The Bayesian lasso (blasso)
49	Least Angle Regression (lars)
50	Support Vector Machines with Linear Kernel (svmLinear2)
51	Support Vector Machines with Linear Kernel (svmLinear)
52	Quantile Regression with LASSO penalty (rqlasso)
53	Non-Convex Penalized Quantile Regression (rqnc)
54	Least Angle Regression (lars2)
55	Tree-Based Ensembles (nodeHarvest)
56	Partial Least Squares Generalized Linear Models (plsRglm)
57	Conditional Inference Tree (ctree2)
58	Boosted Generalized Linear Model (glmboost)
59	Relevance Vector Machines with Linear Kernel (rvmLinear)
60	L2 Regularized Support Vector Machine (dual) with Linear Kernel (svmLinear3)
61	Boosted Linear Model (BstLm)
62	Partial Least Squares (simpls)
63	Partial Least Squares (pls)
64	Partial Least Squares (widekernelpls)
65	Partial Least Squares (kernelpls)
66	partDSA (partDSA)
67	Non-Negative Least Squares (nnls)
68	Principal Component Analysis (pcr)
69	Linear Regression with Forward Selection (leapForward)
70	Independent Component Regression (icr)
71	Extreme Learning Machine (elm)
72	Relaxed Lasso (relaxo)
73	Linear Regression with Backwards Selection (leapBackward)
74	Linear Regression with Stepwise Selection (leapSeq)
75	Supervised Principal Component Analysis (superpc)
76	Polynomial Kernel Regularized Least Squares (krlsPoly)
77	Radial Basis Function Network (rbf)
78	Neural Networks with Feature Extraction (pcaNNet)
79	Neural Network (nnet)
80	Model Averaged Neural Network (avNNet)
81	Stacked AutoEncoder Deep Neural Network (dnn)
82	Multi-Layer Perceptron (mlp)
83	Multi-Layer Perceptron (mlpWeightDecay)
84	Multi-Layer Perceptron, multiple layers (mlpWeightDecayML)
85	Radial Basis Function Network (rbfDDA)
86	Multilayer Perceptron Network by Stochastic Gradient Descent (mlpSGD)
87	Multi-Layer Perceptron, with multiple layers (mlpML)
88	Random Forest Rule-Based Model (rfRules)
89	Dynamic Evolving Neural-Fuzzy Inference System (DENFIS)
90	Adaptive-Network-Based Fuzzy Inference System (ANFIS)
91	Ensembles of Generalized Linear Models (randomGLM)

Table S8 List of model types or relevant characteristics of ML algorithms defined in the caret library. These model types were used to calculate Jaccard similarity among the 91 ML algorithms.

No.	Model Type	No.	Model Type
1	Classification	31	Linear Regression Models
2	Regression	32	Logic Regression
3	Accepts Case Weights	33	Logistic Regression
4	Bagging	34	Mixture Model
5	Bayesian Model	35	Model Tree
6	Binary Predictors Only	36	Multivariate Adaptive Regression Splines
7	Boosting	37	Neural Network
8	Categorical Predictors Only	38	Oblique Tree
9	Cost Sensitive Learning	39	Ordinal Outcomes
10	Discriminant Analysis	40	Partial Least Squares
11	Discriminant Analysis Models	41	Patient Rule Induction Method
12	Distance Weighted Discrimination	42	Polynomial Model
13	Ensemble Model	43	Prototype Models
14	Feature Extraction	44	Quantile Regression
15	Feature Extraction Models	45	Radial Basis Function
16	Feature Selection Wrapper	46	Random Forest
17	Gaussian Process	47	Regularization
18	Generalized Additive Model	48	Relevance Vector Machines
19	Generalized Linear Model	49	Ridge Regression
20	Generalized Linear Models	50	Robust Methods
21	Handle Missing Predictor Data	51	Robust Model
22	Implicit Feature Selection	52	ROC Curves
23	Kernel Method	53	Rule-Based Model
24	L1 Regularization	54	Self-Organizing Maps
25	L1 Regularization Models	55	String Kernel
26	L2 Regularization	56	Support Vector Machines
27	L2 Regularization Models	57	Text Mining
28	Linear Classifier	58	Tree-Based Model
29	Linear Classifier Models	59	Two Class Only
30	Linear Regression	□	□