In Silico Prediction of Physicochemical Properties of Environmental Chemicals Using Molecular Fingerprints and Machine Learning

Qingda Zang, Kamel Mansouri, Antony J. Williams, Richard S. Judson, David G. Allen, Warren M. Casey, and Nicole C. Kleinstreuer

Qingda Zang

Integrated Laboratory Systems, Inc., Research Triangle Park, North Carolina 27709, USA
dan.zang@nih.gov

Kamel Mansouri

National Center for Computational Toxicology, Office of Research and Development, the U.S. Environmental Protection Agency, Research Triangle Park, North Carolina 27711, USA
mansouri.kamel@epa.gov

Antony J. Williams

National Center for Computational Toxicology, Office of Research and Development, the U.S. Environmental Protection Agency, Research Triangle Park, North Carolina 27711, USA
Williams.Antony@epa.gov

Richard S. Judson

National Center for Computational Toxicology, Office of Research and Development, the U.S. Environmental Protection Agency, Research Triangle Park, North Carolina 27711, USA
Judson.Richard@epa.gov

David G. Allen

Integrated Laboratory Systems, Inc., Research Triangle Park, North Carolina 27709, USA
dallen@ils-inc.com

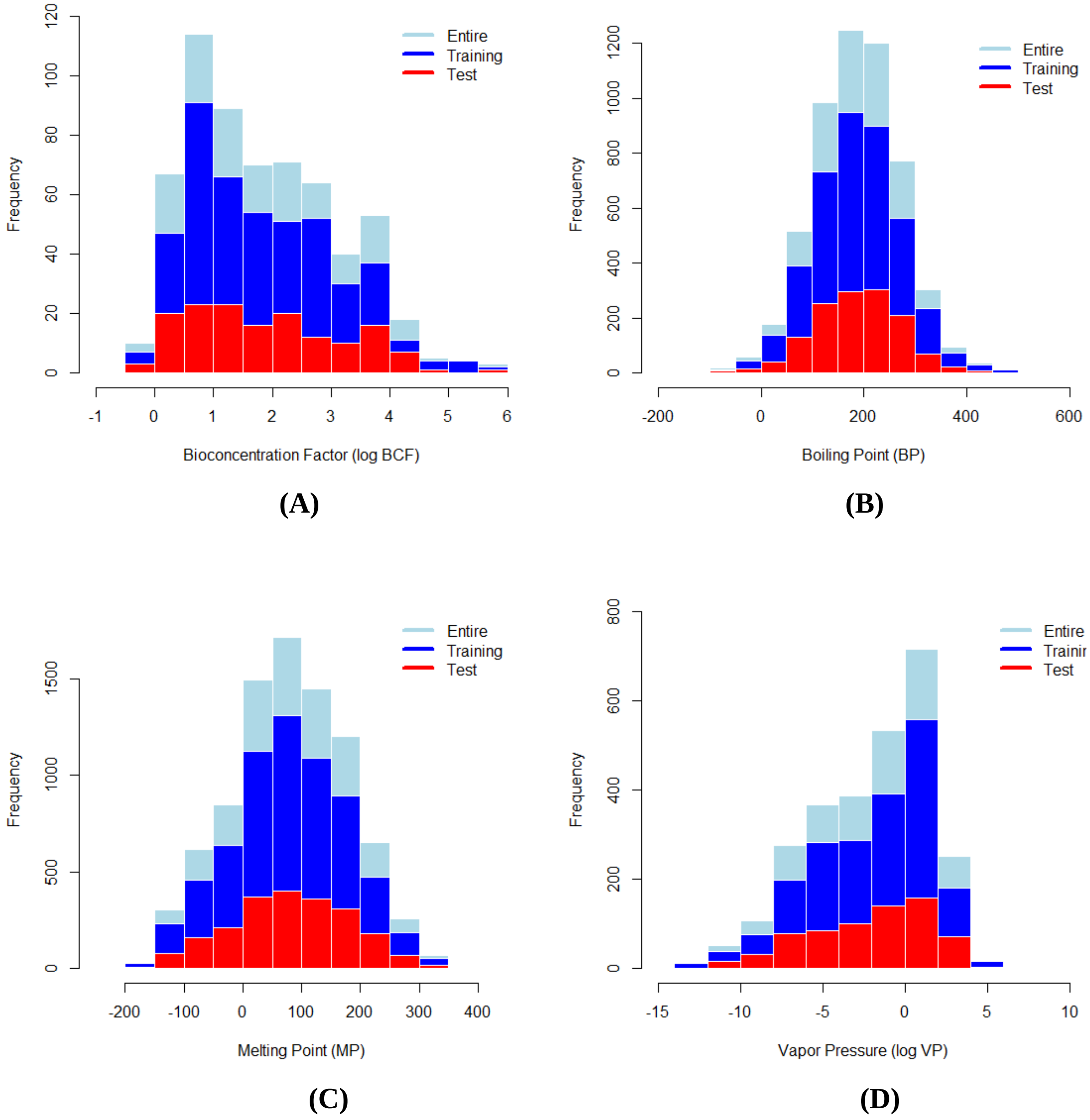
Warren M. Casey

National Toxicology Program, National Institute of Environmental Health Sciences, Research Triangle Park, North Carolina 27703, USA
warren.casey@nih.gov

Nicole C. Kleinstreuer,

National Toxicology Program, National Institute of Environmental Health Sciences, Research Triangle Park, North Carolina 27703, USA,
nicole.kleinstreuer@nih.gov

List of Contents



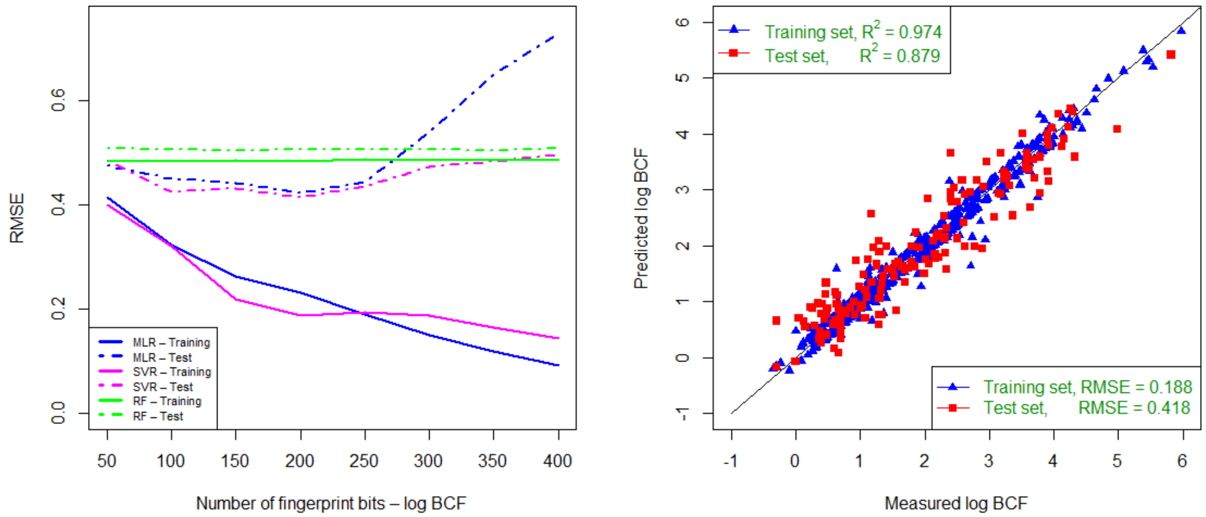
(A)

(B)

(C)

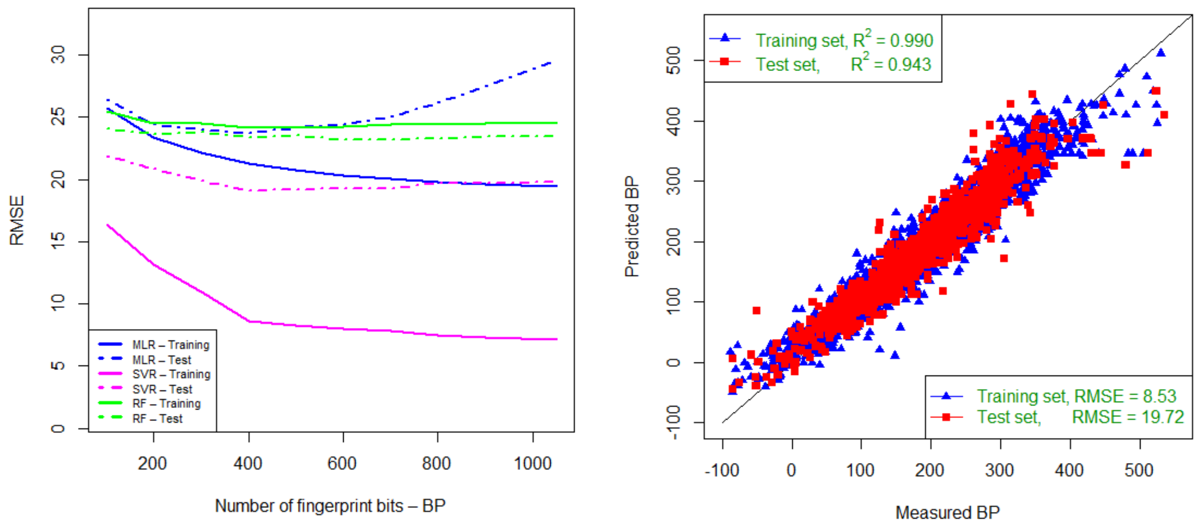
(D)

Figure S1. Data distribution of logBCF (A), BP (B), MP (C), and logVP (D).



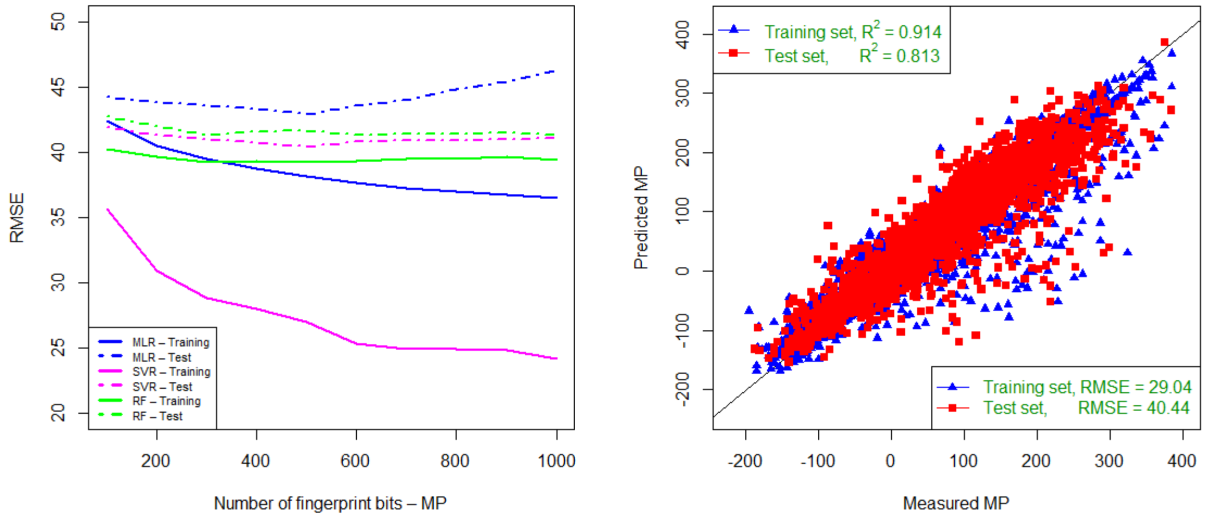
(A) **(B)**

Figure S2. Relationship between model complexity and prediction errors (RMSE) (A) and plot of experimental data versus estimated values by SVR using 200 fingerprint bits (B) for logBCF.



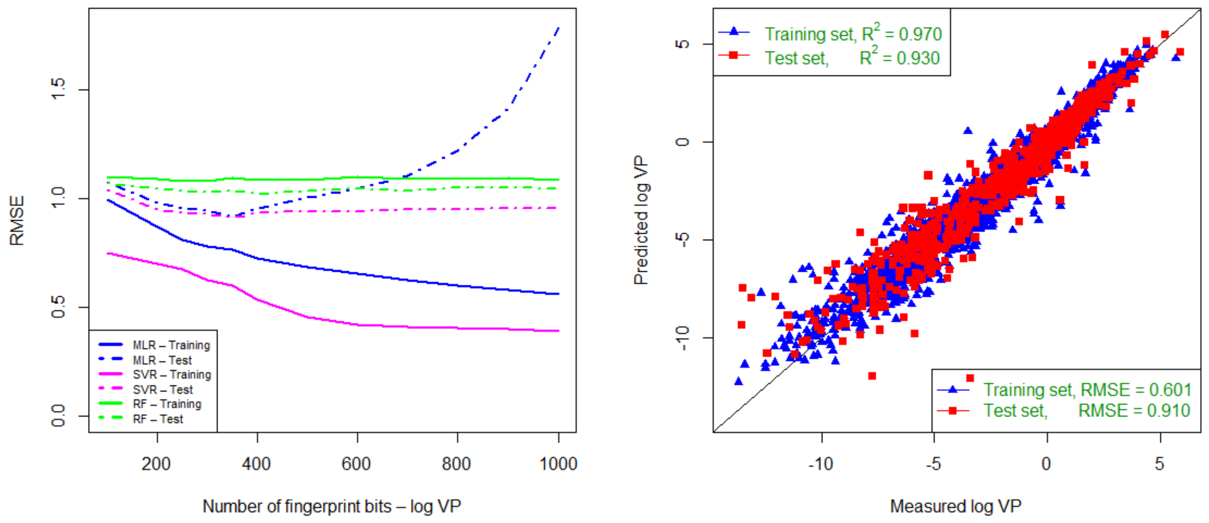
(A) **(B)**

Figure S3. Relationship between model complexity and prediction errors (RMSE) (A) and plot of experimental data versus estimated values by SVR using 400 fingerprint bits (B) for BP.



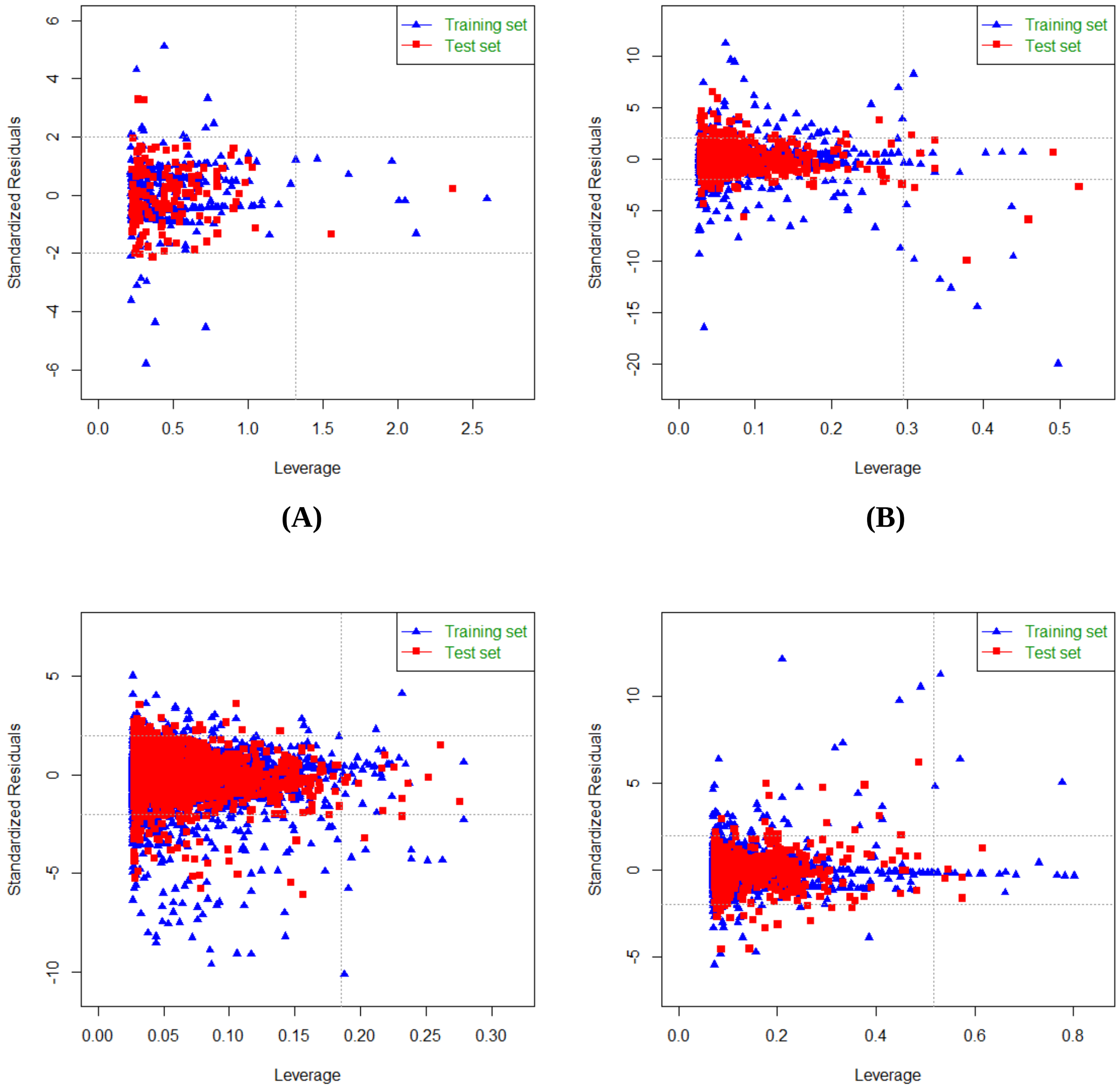
(A) **(B)**

Figure S4. Relationship between model complexity and prediction errors (RMSE) (A) and plot of experimental data versus estimated values by SVR using 500 fingerprint bits (B) for MP.



(A) **(B)**

Figure S5. Relationship between model complexity and prediction errors (RMSE) (A) and plot of experimental data versus estimated values by SVR using 350 fingerprint bits (B) for logVP.



(A) **(B)**

(C) **(D)**

Figure S6. Plots of leverage versus standardized residuals for logBCF (A), BP (B), MP (C) and logVP (D) models' training and test sets. The models were built by SVR using 200, 400, 500 and 350 fingerprint bits for logBCF, BP, MP and logVP, respectively. Vertical dashed line marks AD threshold based on the leverage value. Horizon dashed lines define a region where predictions were within two standardized residuals.

Table S1. Number of Chemicals in Training and Test Sets according to Product Classes

Class	logP		logS		logBCF		BP		MP		logVP	
	Training	Test	Training	Test	Training	Test	Training	Test	Training	Test	Training	Test
Antimicrobial	58	23	46	14	15	8	34	11	62	22	59	15
Industrial Use	866	215	523	180	199	64	766	260	1067	353	726	233
Chemical Warfare	13	0	6	2	2	0	20	4	15	8	20	2
Colorant	129	37	90	29	45	18	115	44	157	61	101	43
Consumer Use	464	121	348	110	112	38	563	180	601	198	483	157
Fertilizer	1	0	0	1	0	0	0	0	2	0	0	1
Flame Retardant	16	7	14	5	12	3	3	3	17	4	18	6
Food Additive	357	104	240	92	36	4	628	198	516	177	336	119
Fragrance	121	35	90	26	32	9	159	54	151	48	128	41
Herbicide	24	13	18	6	5	1	1	0	28	7	24	7
Inert Ingredient	246	69	180	58	51	13	239	97	329	109	245	84
Personal Care	195	57	128	49	34	6	152	66	257	83	150	53
Pesticide	707	160	405	121	125	54	191	73	697	208	561	178
Petrochemical	23	7	24	5	10	0	21	12	28	6	21	14
Pharmaceutical	1007	275	350	108	47	13	205	63	1207	406	186	63

The Chemical and Product Categories (CPCat) from EPA (https://www.epa.gov/chemical-research/chemical-and-product-categories-cpcat) were used to describe the chemical classes.

There are 15 chemical classes, which were retrieved from the Aggregated Computational Toxicology Resource (ACToR) database (https://actor.epa.gov/cpcat/faces/home.xhtml).

One chemical may have multiple product classes.

Table S2. Regression Statistics of LogBCF Using Subsets of Fingerprint Bits, MW and LogP

Variable	Model Statistics	Data Set	MLR	PLSR	RF	SVR
450 FP bits	R^2	Training	0.997	0.884	0.759	0.894
		Test	0.002	0.723	0.724	0.725
	RMSE	Training	0.063	0.384	0.496	0.381
		Test	47.33	0.577	0.509	0.556
200 FP bits	R^2	Training	0.965	0.965	0.777	0.974
		Test	0.863	0.864	0.734	0.879
	RMSE	Training	0.231	0.233	0.482	0.188
		Test	0.422	0.421	0.505	0.418
200 FP bits + MW	R^2	Training	0.966	0.967	0.786	0.975
		Test	0.868	0.869	0.749	0.883
	RMSE	Training	0.224	0.223	0.474	0.185
		Test	0.417	0.415	0.500	0.409
200 FP bits + MW + logP	R^2	Training	0.969	0.970	0.845	0.977
		Test	0.874	0.876	0.816	0.885
	RMSE	Training	0.219	0.217	0.437	0.182
		Test	0.413	0.412	0.469	0.405

Table S3. Regression Statistics of BP Using Subsets of Fingerprint Bits and MW

Variable	Model Statistics	Data Set	MLR	PLSR	RF	SVR
1050 FP bits	R^2	Training	0.945	0.932	0.901	0.991
		Test	0.873	0.890	0.908	0.940
	RMSE	Training	19.46	21.45	24.55	8.527
		Test	29.59	27.07	23.47	19.82
400 FP bits	R^2	Training	0.933	0.932	0.904	0.990
		Test	0.914	0.915	0.909	0.943
	RMSE	Training	21.30	21.40	24.18	8.595
		Test	23.74	23.69	23.37	19.72
400 FP bits + MW	R^2	Training	0.949	0.948	0.931	0.994
		Test	0.935	0.936	0.940	0.965
	RMSE	Training	18.78	18.90	20.74	6.556
		Test	20.71	20.62	19.27	15.63

Table S4. Regression Statistics of MP Using Subsets of Fingerprint Bits, MW and BP

Variable	Model Statistics	Data Set	MLR	PLSR	RF	SVR
1424 FP bits	R^2	Training	0.841	0.801	0.785	0.931
		Test	0.716	0.745	0.800	0.799
	RMSE	Training	36.00	39.29	39.56	24.41
		Test	51.15	45.17	41.18	41.27
500 FP bits	R^2	Training	0.816	0.815	0.787	0.914
		Test	0.780	0.781	0.802	0.813
	RMSE	Training	38.14	38.21	39.20	27.04
		Test	42.86	42.79	41.01	40.44
500 FP bits + MW	R^2	Training	0.823	0.823	0.798	0.923
		Test	0.788	0.789	0.807	0.822
	RMSE	Training	37.59	37.60	37.93	25.73
		Test	42.21	42.14	40.61	39.59
500 FP bits + MW + BP	R^2	Training	0.834	0.833	0.802	0.925
		Test	0.803	0.804	0.807	0.826
	RMSE	Training	36.67	37.68	37.85	25.56
		Test	41.02	40.98	40.59	39.14

Table S5. Regression Statistics of LogVP Using Subsets of Fingerprint Bits, MW and BP

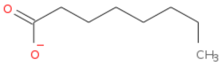
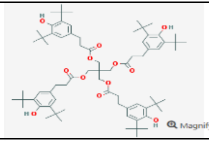
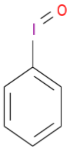
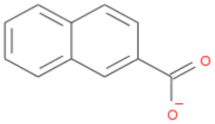
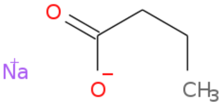
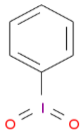
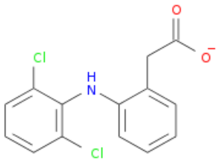
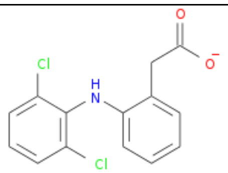
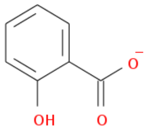
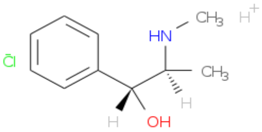
Variable	Model Statistics	Data Set	MLR	PLSR	RF	SVR
1145 FP bits	R^2	Training	0.978	0.929	0.886	0.988
		Test	0.635	0.878	0.896	0.921
	RMSE	Training	0.528	0.920	1.093	0.381
		Test	2.413	1.203	1.058	0.953
350 FP bits	R^2	Training	0.952	0.951	0.888	0.970
		Test	0.928	0.929	0.902	0.930
	RMSE	Training	0.763	0.765	1.090	0.601
		Test	0.914	0.913	1.033	0.910
350 FP bits + MW	R^2	Training	0.959	0.958	0.908	0.980
		Test	0.938	0.939	0.920	0.941
	RMSE	Training	0.708	0.711	1.002	0.495
		Test	0.859	0.858	0.949	0.843
350 FP bits + MW + BP	R^2	Training	0.963	0.963	0.922	0.982
		Test	0.945	0.946	0.941	0.946
	RMSE	Training	0.680	0.681	0.938	0.473
		Test	0.812	0.811	0.830	0.810

Table S6. Applicability Domain of logBCF, BP, MP and logVP Models: Test Set Evaluation[a]

Property	Measure	Chemicals outside AD	Chemicals inside AD	Experimental vs Predicted Test Chemicals inside AD	
				R^2	RMSE
logBCF	Leverage (I)	2	150	0.886	0.405
	Distance from centroid (II)	6	146	0.888	0.403
	Distance by kNN (III)	9	143	0.891	0.399
	I and II and III	1	151	0.885	0.405
	I or II or III	13	139	0.889	0.400
BP	Leverage (I)	9	1349	0.965	15.15
	Distance from centroid (II)	69	1289	0.967	14.81
	Distance by kNN (III)	67	1291	0.971	13.91
	I and II and III	4	1354	0.965	15.34
	I or II or III	111	1247	0.970	13.96
MP	Leverage (I)	14	2149	0.826	39.11
	Distance from centroid (II)	103	2060	0.827	39.08
	Distance by kNN (III)	119	2044	0.828	39.00
	I and II and III	9	2154	0.826	39.12
	I or II or III	182	1981	0.827	39.05
logVP	Leverage (I)	5	674	0.946	0.808
	Distance from centroid (II)	32	647	0.947	0.763
	Distance by kNN (III)	37	642	0.949	0.744
	I and II and III	2	677	0.946	0.809
	I or II or III	48	631	0.947	0.752

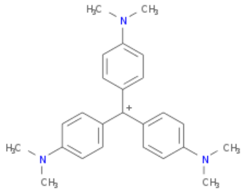
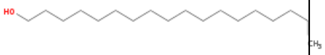
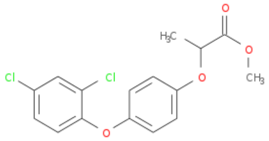
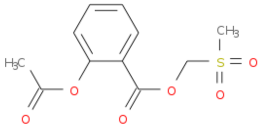
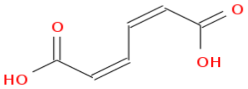
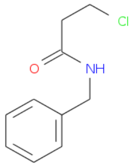
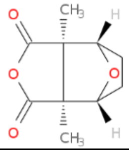
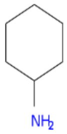
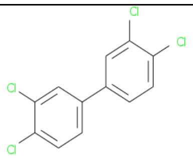
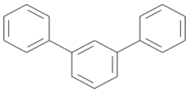
[a]: The models were built by SVR using 200 FP bits + MW + logP for logBCF, 400 FP bits + MW for BP, 500 FP bits + MW + BP for MP, and 350 FP bits + MW + BP for logVP.

Table S7. Top Ten Chemicals with Largest Prediction Residuals from LogP Models

Chemical Name	CASRN	Exp[a]	Pred[b]	Residuals	Training /Test	AD[c]	Structure
Sodium octanoate	1984-06-1	-1.38	3.30	-4.68	Test	In	
Irganox 1010	6683-19-8	1.36	5.42	-4.06	Training	Out	
Benzene, iodosyl-	536-80-1	-1.61	2.45	-4.06	Test	In	
2-naphthoic acid, sodium salt	17273-79-9	-1.07	2.93	-4.00	Training	In	
Sodium butyrate	156-54-7	-3.20	0.76	-3.96	Training	In	
Iodoxybenzene	696-33-3	-1.33	2.53	-3.86	Training	In	
Diclofenac potassium	15307-81-0	0.65	4.43	-3.78	Test	In	
Diclofenac sodium	15307-79-6	0.70	4.43	-3.73	Training	In	
Sodium salicylate	54-21-7	-1.43	2.05	-3.48	Training	In	
Ephedrine hydrochloride	50-98-6	-2.45	0.90	-3.35	Training	In	

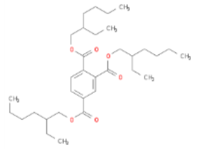
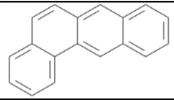
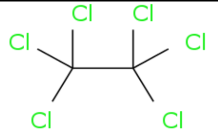
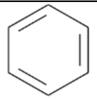
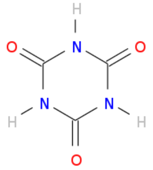
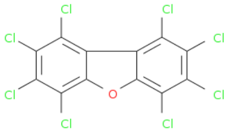
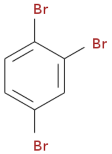
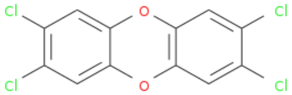
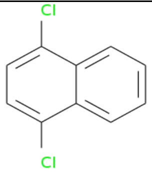
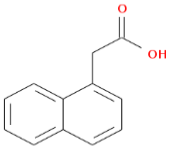
[a]: Experimental values; [b]: Predicted values; [c]: Applicability domain from kNN measure.

Table S8. Top Ten Chemicals with Largest Prediction Residuals from LogS Models

Chemical Name	CASRN	Exp[a]	Pred[b]	Residuals	Training /Test	AD[c]	Structure
Gentian Violet	548-62-9	-2.01	-4.41	2.40	Test	Out	
1-Octadecanol	112-92-5	-8.39	-6.28	-2.11	Test	In	
Diclofop-methyl	51338-27-3	-3.83	-5.89	2.05	Training	Out	
Methylsulfonyl methyl 2-acetyl oxybenzoate	76432-35-4	-3.39	-1.50	-1.89	Test	In	
Muconic acid	505-70-4	-2.85	-1.00	-1.85	Test	In	
Beclamide	501-68-8	-3.30	-1.47	-1.83	Training	In	
Cantharidin	56-25-7	-3.82	-2.01	-1.81	Test	In	
Cyclohexyl amine	108-91-8	1.00	-0.77	1.77	Test	In	
3,3',4,4'-Tetrachlorobin phenyl	32598-13-3	-8.71	-6.97	-1.74	Training	In	
1,1':3',1''-Terphenyl	92-06-8	-5.18	-6.90	1.72	Training	In	

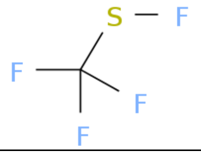
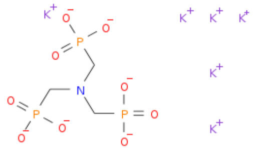
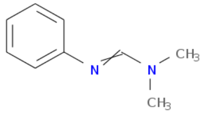
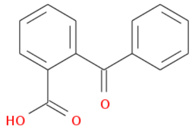
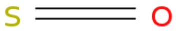
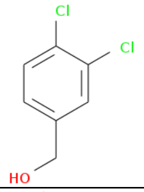
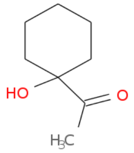
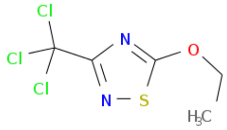
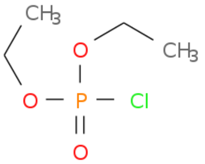
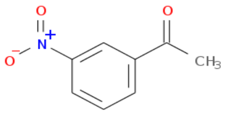
[a]: Experimental values; [b]: Predicted values; [c]: Applicability domain from kNN measure.

Table S9. Top Ten Chemicals with Largest Prediction Residuals from LogBCF Models

Chemical Name	CASRN	Exp[a]	Pred[b]	Residuals	Training /Test	AD[c]	Structure
Tris (2-ethylhexyl) trimellitate	3319-31-1	1.17	2.57	-1.40	Test	Out	
Benz(a) anthracene	56-55-3	2.41	3.65	-1.24	Test	In	
Hexachloro ethane	67-72-1	2.71	1.63	1.08	Training	In	
Benzene	71-43-2	0.63	1.58	-0.95	Training	In	
Cyanuric acid	108-80-5	-0.30	0.65	-0.95	Test	In	
Octachloro dibenzofuran	39001-02-0	2.89	1.94	0.95	Test	In	
1,2,4-Tribromo benzene	615-54-3	3.63	2.68	0.95	Test	In	
2,3,7,8-Tetra chlorodibenzo -p-dioxin	1746-01-6	4.99	4.08	0.91	Test	In	
Naphthalene, 1,4-dichloro-	1825-31-6	3.75	2.86	0.89	Training	In	
1-Naphthalene acetic acid	86-87-3	0.47	1.33	-0.86	Test	Out	

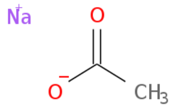
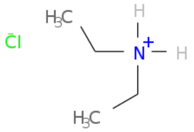
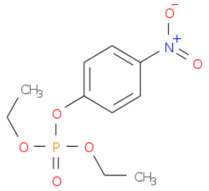
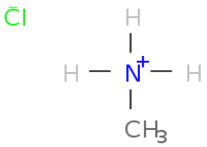
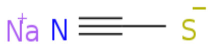
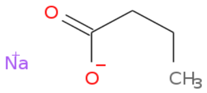
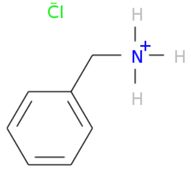
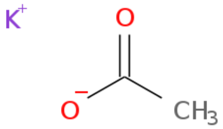
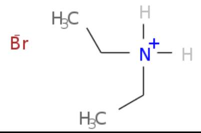
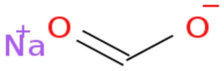
[a]: Experimental values; [b]: Predicted values; [c]: Applicability domain from kNN measure.

Table S10. Top Ten Chemicals with Largest Prediction Residuals from BP Models

Chemical Name	CASRN	Exp[a]	Pred[b]	Residuals	Training /Test	AD[c]	Structure
Methane sulfenyl fluoride, trifluoro-	17742-04-0	147.50	-33.95	181.45	Training	In	
Amino trimethylene phosphonic acid	6419-19-8	480.00	332.47	147.53	Test	Out	
Formamidine, N,N-dimethyl-N'-phenyl-	1783-25-1	127.00	244.89	-117.89	Test	Out	
Benzoic acid, 2-benzoyl-	85-52-9	261.00	367.30	-106.30	Test	Out	
Carbonyl sulfide	463-58-1	-50.00	45.06	-95.06	Test	In	
3,4-Dichloro benzyl alcohol	1805-32-9	149.50	244.35	-94.85	Training	In	
Ethanone, 1-(1-hydroxy cyclohexyl)-	1123-27-9	125.50	215.30	-89.80	Test	In	
Etridiazole	2593-15-9	188.00	274.29	-86.29	Test	Out	
Diethyl phosphoro chloridate	814-49-3	93.50	176.99	-83.49	Training	In	
3-Nitroaceto phenone	121-89-1	202.00	282.37	-80.37	Test	Out	

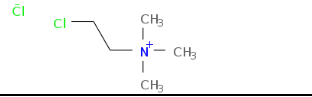
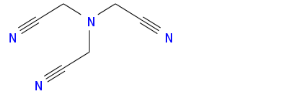
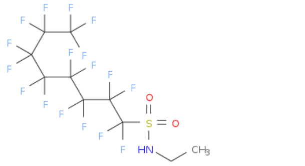
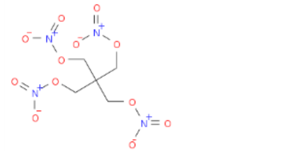
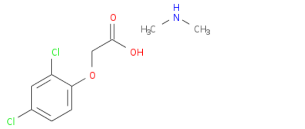
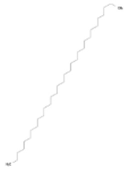
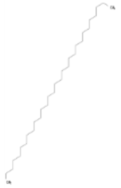
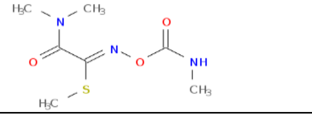
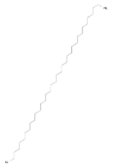
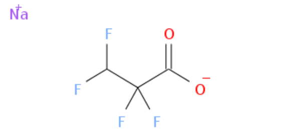
[a]: Experimental values; [b]: Predicted values; [c]: Applicability domain from kNN measure.

Table S11. Top Ten Chemicals with Largest Prediction Residuals from MP Models

Chemical Name	CASRN	Exp[a]	Pred[b]	Residuals	Training /Test	AD[c]	Structure
Sodium acetate	127-09-3	324.00	30.04	293.96	Training	In	
Diethylamine hydrochloride	660-68-4	228.50	-56.01	284.51	Training	In	
Paraoxon	311-45-5	300.00	24.19	275.81	Test	In	
Methylamine, hydrochloride	593-51-1	227.50	-37.46	264.96	Training	In	
Sodium thiocyanate	540-72-7	287.00	23.05	263.95	Training	In	
Sodium butyrate	156-54-7	251.00	-5.53	256.53	Training	In	
Benzylamine hydrochloride	3287-99-8	262.50	7.59	254.91	Training	In	
Potassium acetate	127-08-2	292.00	37.19	254.81	Test	In	
Diethylamine, hydrobromide	6274-12-0	219.00	-35.66	254.66	Test	In	
Sodium formate	141-53-7	253.00	-1.27	254.27	Test	In	

[a]: Experimental values; [b]: Predicted values; [c]: Applicability domain from kNN measure.

Table S12. Top Ten Chemicals with Largest Prediction Residuals from LogVP Models

Chemical Name	CASRN	Exp[a]	Pred[b]	Residuals	Training /Test	AD[c]	Structure
Chlormequat chloride	999-81-5	-7.12	-0.07	-7.05	Training	In	
Acetonitrile, 2,2',2"-nitrilotris-	7327-60-8	-6.50	-2.01	-4.49	Test	In	
Sulfluramid	4151-50-2	-6.37	-1.90	-4.47	Test	Out	
Pentaerythritol tetranitrate	78-11-5	-8.26	-4.00	-4.27	Test	In	
2,4-D, Dimethylamine salt	2008-39-1	-9.00	-4.91	-4.09	Training	In	
Hentriacontane	630-04-6	-10.85	-7.12	-3.73	Training	In	
Triacontane	638-68-6	-10.56	-6.90	-3.66	Test	In	
Oxamyl	23135-22-0	-3.64	-7.06	3.42	Test	In	
Pentatri acontane	630-07-9	-11.27	-7.99	-3.28	Training	In	
Fluopropanate-sodium	22898-01-7	-3.52	-0.28	-3.24	Training	In	

[a]: Experimental values; [b]: Predicted values; [c]: Applicability domain from kNN measure.

	QMRF identifier (JRC Inventory): **To be entered by JRC**
	QMRF Title: **QSARs for octanol-water partition coefficient (LogP), water solubility (LogS), melting point (MP), boiling point (BP), vapor pressure (LogVP) and bioconcentration factor (LogBCF)**
	Printing Date: **December 20, 2016**

1.QSAR identifier

1.1.QSAR identifier (title):

QSARs for octanol-water partition coefficient (LogP), water solubility (LogS), melting point (MP), boiling point (BP), vapor pressure (LogVP) and bioconcentration factor (LogBCF)

1.2.Other related models:

1.3.Software coding the model:

The R statistical computing environment for Windows

(version 3.2.1) http://cran.r-project.org/

2.General information

2.1.Date of QMRF:

December 20, 2016

2.2.QMRF author(s) and contact details:

[1]Qingda Zang, Integrated Laboratory Systems, Inc., dan.zang@nih.gov

[2]Nicole C. Kleinstreuer, National Toxicology Program, National Institute of Environmental Health Sciences, nicole.kleinstreuer@nih.gov

[3]Kamel Mansouri, National Center for Computational Toxicology, Office of Research and Development, the U.S. Environmental Protection Agency, mansouri.kamel@epa.gov

[4]Antony J. Williams, National Center for Computational Toxicology, Office of Research and Development, the U.S. Environmental Protection Agency, Williams.Antony@epa.gov

[5]Richard S. Judson, National Center for Computational Toxicology, Office of Research and Development, the U.S. Environmental Protection Agency, Judson.Richard@epa.gov

[6]David G. Allen, Integrated Laboratory Systems, Inc., dallen@ils-inc.com

[7]Warren M. Casey, National Toxicology Program, National Institute of Environmental Health Sciences, warren.casey@nih.gov

2.3.Date of QMRF update(s):

This is a new QMRF.

2.4.QMRF update(s):

NA

2.5.Model developer(s) and contact details:

[1]Qingda Zang, Integrated Laboratory Systems, Inc., dan.zang@nih.gov

[2]Nicole C. Kleinstreuer, National Toxicology Program, National Institute of Environmental Health Sciences, nicole.kleinstreuer@nih.gov

2.6.Date of model development and/or publication:

December 20, 2016

2.7.Reference(s) to main scientific papers and/or software package:

[1]Qingda Zang, Kamel Mansouri, Antony J. Williams, Richard S. Judson, David G. Allen, Warren Casey, and Nicole C. Kleinstreuer. In Silico Prediction of Physicochemical Properties of Environmental Chemicals Using Molecular Fingerprints and Machine Learning (Journal of Chemical Information and Modeling, http://dx.doi.org/10.1021/acs.jcim.6b00625)

[2]The R statistical computing environment for Windows (version 3.2.1) http://cran.r-project.org/

2.8.Availability of information about the model:

Algorithms are available.

Training and test sets are available.

2.9.Availability of another QMRF for exactly the same model:

NA

3.Defining the endpoint - OECD Principle 1

3.1.Species:

Not applicable

3.2.Endpoint:

[1]QMRF 1. Physical Chemical Properties QMRF 1. 1. Melting point

[2]QMRF 1. Physical Chemical Properties QMRF 1. 2. Boiling point

[3]QMRF 1. Physical Chemical Properties QMRF 1. 3. Water solubility

[4]QMRF 1. Physical Chemical Properties QMRF 1. 4. Vapour pressure

[5]QMRF 1. Physical Chemical Properties QMRF 1. 6. Octanol-water partition coefficient (Kow)

[6]QMRF 2. Environmental fate parameters QMRF 2. 4.a. Bioconcentration . BCF fish

3.3.Comment on endpoint:

The BCF for a particular chemical compound is defined as the equilibrium ratio of the concentration of a chemical inside an organism to the concentration in the surrounding environment. End point data was based on experimental measurements contained in the US EPA Estimation Program Interface (EPI) Suite database. http://esc.syrres.com/interkow/EPiSuiteData.htm

3.4.Endpoint units:

Melting point: C°

Boiling point: C°

Solubility: mol/L

Vapor pressure: mmHg

Partition coefficient: unitless

Bioconcentration factor: unitless

3.5.Dependent variable:

Octanol-water partition coefficient (LogP), water solubility (LogS), melting point (MP), boiling point (BP), vapor pressure (LogVP) and bioconcentration factor (LogBCF).

3.6.Experimental protocol:

NA

3.7.Endpoint data quality and variability:

The data set was retrieved from US EPA EPI Suite.

LogP	Max: 11.29;	Min: -5.40;	Mean: 2.07;	Deviation: 1.83.
LogS	Max: 1.58;	Min: -12.06;	Mean: -2.60;	Deviation: 2.19.
BP	Max: 548.00;	Min:-88.60;	Mean: 188.98;	Deviation: 85.07.
MP	Max: 385.00;	Min: -196.00;	Mean: 80.35;	Deviation: 99.12.
LogVP	Max: 5.67;	Min: -13.68;	Mean: -2.04;	Deviation: 3.57.
LogBCF	Max: 5.97;	Min: -0.35;	Mean: 1.88;	Deviation: 1.26.

4.Defining the algorithm - OECD Principle 2

4.1.Type of model:

QSAR

4.2.Explicit algorithm:

Support Vector Regression (SVR)

SVR can model both linear and non-linear relationships between the property and molecular descriptors by utilizing an appropriate kernel function to map the input variables from a lower dimensional space to a higher dimensional feature space and transform the non-linear relationship into a linear form. SVR with a Gaussian radial basis function (RBF) kernel was employed to explore the possible nonlinear dependency between molecular fingerprints and the property.

```
LogPmodel <- svm(LogP~., data=LogPdataTraining, cost = 150, epsilon
   = 0.05, gamma = 0.00014)
```

```
LogSmodel <- svm(LogS~., data=LogSdataTraining, cost = 260, epsilon
    = 0.145, gamma = 0.000031)

BPmodel <- svm(BP~., data=BPdataTraining, cost = 9, epsilon=0.012,
    gamma = 0.001)

 MPmodel <- svm(MP~., data=MPdataTraining, cost = 9, epsilon = 0.18,
    gamma = 0.00065)

LogVPmodel <- svm(LogVP~., data=VPdataTraining, cost = 115,
    epsilon = 0.105, gamma = 0.00011)

LogBCFmodel <- svm(LogBCF~., data=BCFdataTraining, cost = 5500,
    epsilon=0.113, gamma = 0.00004)
```

4.3.Descriptors in the model:

Molecular fingerprints: the chemicals were represented by fingerprints derived from their molecular structures. A total of 8097 binary bits were generated with 1 and 0 denoting the presence and absence of a specific structural fragment.

4.4.Descriptor selection:

To obtain reliable and robust regression models with high predictive performance, genetic algorithm (GA) was employed to select the most information-rich subset of fingerprint bits.

GA is an efficient stochastic optimization tool and randomized search technique, and can deal with a great number of descriptors and effectively select a subset from them.

4.5.Algorithm and descriptor generation:

A wide variety of fingerprints were calculated using publicly available SMARTS systems implemented in PADEL: Estate (79bits), Extended (1024 bits), Substructure (307 bits), Klekota Roth (4860 bits), PubChem (881 bits), Atom Pairs 2D (780 bits), and MACCS (166 bits).

Yap, C. W. PaDEL-descriptor: an open source software to calculate molecular descriptors and fingerprints. J. Comput. Chem. 2011, *32(7)*, 1466-1474.

4.6.Software name and version for descriptor generation:

Software: PaDEL-Descriptor; Version: 2.21.
http://www.yapcwsoft.com/dd/padeldescriptor/

4.7.Chemicals/Descriptors ratio:

LogP: 11370/600

LogS: 1507/350

BP: 4074/400

MP: 6485/500

LogVP: 2034/350

LogBCF: 456/200

5.Defining the applicability domain - OECD Principle 3

5.1.Description of the applicability domain of the model:

The QSAR models were developed using training sets and thus, their applicability to external chemicals depends on the structural similarity between the training chemicals and the external test chemicals. The models are presumed to provide more reliable predictions for chemicals that fall in the AD, as defined by the three distance measures below. In this study, only if the thresholds from all three distance measures are exceeded is a test chemical deemed to be outside the AD. Otherwise, if only one or two thresholds are exceeded, the chemical is considered to be potentially outside the AD.

5.2.Method used to assess the applicability domain:

Three distance-based measures (i.e., leverage, distance from centroid and k-nearest neighbors (kNN)), were applied to assess the applicability domain (AD) of each regression model. The distance of a test chemical from a defined point in the descriptor space of the training set was calculated and compared to a predefined threshold. The test chemical is located inside the AD if its distance is less than or equal to the threshold. Leverage is the diagonal element of the covariance matrix for a given dataset, and the leverage of a test chemical is proportional to Hotellings T^2 statistic and its Mahalanobis distance. The threshold was set to three times the average of the leverage (3 *m*/ *n*, with *m* being the number of variables and *n* the number of training chemicals). For the measure of distance from centroid, the distance of a test chemical from the training set centroid is compared with a threshold, which is determined as follows: (1) calculate the distances of training chemicals from their centroid; (2) sort the vector of distances in ascending order; (3) set the distance value corresponding to 95th percentile as the threshold. The kNN measure defines the model's AD based on the similarity between a test chemical and the training chemicals. The average distance of the test chemical from its five nearest neighbors in the training set is compared with a threshold, which is the 95th percentile of average distance of training chemicals from their five nearest neighbors.

5.3.Software name and version for applicability domain assessment:

The R statistical computing environment for Windows (version 3.2.1) http://cran.r-project.org/

5.4.Limits of applicability:

6.Internal validation - OECD Principle 4

6.1.Availability of the training set:

Yes

6.2.Available information for the training set:

CAS RN: Yes

Chemical Name: Y

Smiles: Yes

Formula: No

INChI: Yes

MOL file: No

6.3.Data for each descriptor variable for the training set:

All

6.4.Data for the dependent variable for the training set:

All

6.5.Other information about the training set:

NA

6.6Pre-processing of data before modelling:

Fingerprint bits with zero variance (i.e. uniform observations across the set) were removed. To obtain reliable models, sufficient occurrences of the fingerprint bits throughout the entire data sets are necessary, and thus bits with low occurrences were eliminated. Following the removal of highly correlated and sparsely occurring bits, finally 1681, 1061, 450, 1050, 1424 and 1145 bits corresponding to LogP, LogS, LogBCF, BP, MP and LogVP, respectively, were retained and employed to build the regression models.

6.7.Statistics for goodness-of-fit:

Coefficient of determination (R^2)

LogP: 0.987

LogS: 0.966

LogBCF: 0.977

BP: 0.994

MP: 0.925

LogVP: 0.982

6.8.Robustness - Statistics obtained by leave-one-out cross-validation:

NA

6.9.Robustness - Statistics obtained by leave-many-out cross-validation:

Coefficient of determination (R^2) (10-fold cross-validation)

LogP: 0.932

LogS: 0.928

LogBCF: 0.863

BP: 0.955

MP: 0.824

LogVP: 0.929

6.10.Robustness - Statistics obtained by Y-scrambling:

NA

6.11.Robustness - Statistics obtained by bootstrap:

NA

6.12.Robustness - Statistics obtained by other methods:

NA

7.External validation - OECD Principle 4

7.1.Availability of the external validation set:

Yes

7.2.Available information for the external validation set:

CAS RN: Yes

Chemical Name: Y

Smiles: Yes

Formula: No

INChI: Yes

MOL file: No

7.3.Data for each descriptor variable for the external validation set:

All

7.4.Data for the dependent variable for the external validation set:

All

7.5.Other information about the external validation set:

NA

7.6.Experimental design of test set:

The data sets were randomly partitioned into training sets (75% of the chemicals) and test sets (25% of the chemicals) to build the models and validate their predictive power, respectively. The distribution of the test set is very similar to that of the training set.

7.7.Predictivity - Statistics obtained by external validation:

Coefficient of determination (R^2)

LogP: 0.935

LogS: 0.939

LogBCF: 0.885

BP: 0.965

MP: 0.826

LogVP: 0.946

7.8.Predictivity - Assessment of the external validation set:

As shown in Section 7.7, the R^2 values are greater than 0.900 for LogP, LogS, BP and LogVP, and greater than 0.800 for MP and LogBCF. The predictivity is high.

7.9.Comments on the external validation of the model:

NA

8.Providing a mechanistic interpretation - OECD Principle 5

8.1.Mechanistic basis of the model:

Here it is not practical to make an interpretation linking each and every selected fingerprint bit to the modeled endpoints. However, we assume that the statistically selected fingerprint bits represent fragments that are relevant to the studied endpoints.

8.2.A priori or a posteriori mechanistic interpretation:

NA

8.3.Other information about the mechanistic interpretation:

NA

9.Miscellaneous information

9.1.Comments:

NA

9.2.Bibliography:

https://www.epa.gov/tsca-screening-tools

9.3.Supporting information:

Training / Test sets

10.Summary (JRC QSAR Model Database)

10.1.QMRF number:

To be entered by JRC

10.2.Publication date:

To be entered by JRC

10.3.Keywords:

To be entered by JRC

10.4.Comments:

To be entered by JRC

R Code for Regression Analysis

1. Feature selection by genetic algorithm for partition coefficient (logP)

# Set the working environment and save data files in the fold

```
> setwd("C:/PropertyRegression")
```

# Read in logP data

```
> LogPdata <- read.table("LogP-All-Fingerprint-Bits.txt", header=T, sep="\t", as.is=T )
```

# There are 14207 rows (chemicals) and 1682 columns (1681 fingerprint bits + logP)

```
> dim(LogPdata)
[1] 14207  1682
```

# There are 11370 training chemicals and 2837 test chemicals

```
> LogPdataTraining <-LogPdata[1:11370,]

> LogPdataTest <-LogPdata[11371:14207,]

> dim(LogPdataTraining)
[1] 11370  1682

> dim(LogPdataTest)
[1] 2837 1682
```

# Load package *subselect*, which is used to select subsets of variables

```
> library(subselect)
```

# Use the function *lmHmat* to produce a matrix as an input to the variable selection

# Use the training set to select the subset of variables. LogPdataTraining[, c(1:1681)] is the

# matrix of 1681 fingerprint bits and LogPdata[, 1682] is the vector of measured logP

```
> LogPHmat <- lmHmat(LogPdataTraining[, c(1:1681)], LogPdataTraining[, 1682])
```

# Use the function *genetic* to select a subset of variables (fingerprint bits)

# Set different numeric values for *kmax.* We can select a series of subsets

# Here is an example of selecting 100 variables

# The size of population *popsize* is set to two times of number of fingerprint bits, i.e., 3362

# The number of generations *nger* is set to 1000

# The mutation *mutate* is set to TRUE, and the mutation probability *mutprob* is set to 0.01

# The criterion for judging the quality of the subset is *CCR12*, which gives the coefficient of

# determination (R^2)

```
> LogPsubset<- genetic(LogPHmat$mat, kmax = 100, popsize = 3362, nger = 1000,
mutate = TRUE, mutprob = 0.01, crit="CCR12")
```

# The outputs include *bestvalues*, which indicate the best values of the criterion (R^2 for

# CCR12), and *bestsets*, which give the variable index for the selected variable set

# The output – best values (coefficient of determination, R^2)

```
> LogPsubset$bestvalues
Card.100
0.8223439
```

# The output – best subset with index of variables

```
> LogPsubset$bestsets

         Var.1 Var.2 Var.3 Var.4 Var.5 Var.6 Var.7 Var.8 Var.9 Var.10 Var.11
Card.100     6     9    14    25    31    40    53    57    62     90    103
         Var.12 Var.13 Var.14 Var.15 Var.16 Var.17 Var.18 Var.19 Var.20 Var.21
Card.100    171    228    231    246    260    292    305    310    316    325
         Var.22 Var.23 Var.24 Var.25 Var.26 Var.27 Var.28 Var.29 Var.30 Var.31
Card.100    333    391    414    420    425    458    460    492    515    517
         Var.32 Var.33 Var.34 Var.35 Var.36 Var.37 Var.38 Var.39 Var.40 Var.41
Card.100    532    537    551    590    601    607    616    645    648    670
         Var.42 Var.43 Var.44 Var.45 Var.46 Var.47 Var.48 Var.49 Var.50 Var.51
Card.100    690    698    746    841    848    850    945    948    991   1005
         Var.52 Var.53 Var.54 Var.55 Var.56 Var.57 Var.58 Var.59 Var.60 Var.61
Card.100   1028   1050   1080   1099   1104   1106   1133   1214   1218   1219
         Var.62 Var.63 Var.64 Var.65 Var.66 Var.67 Var.68 Var.69 Var.70 Var.71
Card.100   1229   1253   1259   1310   1314   1325   1326   1345   1351   1366
         Var.72 Var.73 Var.74 Var.75 Var.76 Var.77 Var.78 Var.79 Var.80 Var.81
Card.100   1375   1382   1384   1403   1408   1411   1431   1434   1441   1443
         Var.82 Var.83 Var.84 Var.85 Var.86 Var.87 Var.88 Var.89 Var.90 Var.91
Card.100   1473   1484   1486   1507   1509   1511   1513   1527   1536   1548
         Var.92 Var.93 Var.94 Var.95 Var.96 Var.97 Var.98 Var.99 Var.100
Card.100   1554   1585   1602   1610   1622   1657   1662   1670    1681
```

2. Regression analysis for logP

# Read in logP data with the optimal subset of 600 fingerprint bits

```
> LogPdata600Bits <- read.table("LogP-600-Fingerprint-Bits.txt", header=T, sep="\t",
as.is=T )
```

# There are 14207 rows (chemicals) and 601 columns (600 fingerprint bits + logP)

```
> dim(LogPdata600Bits)
[1] 14207  601
```

# There are 11370 training chemicals and 2837 test chemicals

```
> LogPdata600BitsTraining <-LogPdata600Bits[1:11370,]

> LogPdata600BitsTest <-LogPdata600Bits[11371:14207,]

> dim(LogPdata600BitsTraining)
[1] 11370  601

> dim(LogPdata600BitsTest)
[1] 2837 601
```

2.1 Multiple linear regression

# Use the function *lm()* to build the MLR model

```
> LogPMLR<-lm(LogP~., data= LogPdata600BitsTraining)
```

# Predict logP from the training set

```
> PredLogPtrainingMLR<-predict(LogPMLR, LogPdata600BitsTraining)
```

# Correlation between measured and predicted logP values for the training set

```
> MeasuredLogPTraining<-LogPdata600BitsTraining$LogP

> CorrLogPtrainingMLR<-lm(PredLogPtrainingMLR ~ MeasuredLogPTraining)

> summary(CorrLogPtrainingMLR)

Call:
lm(formula = PredLogPtrainingMLR ~ MeasuredLogPTraining)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.2929 -0.3304 -0.0069  0.3227  5.2871 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.204067   0.007725   26.42   <2e-16 ***
MeasuredLogPTraining 0.901290   0.002798  322.18   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.546 on 11368 degrees of freedom
Multiple R-squared:  0.9013,    Adjusted R-squared:  0.9013 
F-statistic: 1.038e+05 on 1 and 11368 DF,  p-value: < 2.2e-16
```

# Predict logP from the test set

```
> PredLogPtestMLR<-predict(LogPMLR, LogPdata600BitsTest)
```

# Correlation between measured and predicted logP values for the test set

```
> MeasuredLogPTest<-LogPdata600BitsTest$LogP

> CorrLogPtestMLR<-lm(PredLogPtestMLR ~ MeasuredLogPTest)

> summary(CorrLogPtestMLR)

Call:
lm(formula = PredLogPtestMLR ~ MeasuredLogPTest)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.3853 -0.3622 -0.0094  0.3383  4.6228 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)      0.227695   0.016338   13.94   <2e-16 ***
MeasuredLogPTest 0.894621   0.005951  150.34   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5758 on 2835 degrees of freedom
Multiple R-squared:  0.8884,    Adjusted R-squared:  0.8885 
F-statistic: 2.26e+04 on 1 and 2835 DF,  p-value: < 2.2e-16
```

2.2 Partial least squares regression

# Load package *pls* for PLSR modeling

```
> library(pls)
```

# Use the function *plsr()* to build the PLSR model with optimal principal components of 42

```
> LogPPLSR <- plsr(LogP ~ ., data = LogPdata600BitsTraining, ncomp =42, scale = 
TRUE)
```

# Predict logP from the training set

```
> PredLogPtrainingPLSR<-predict(LogPPLSR,  newdata = LogPdata600BitsTraining)
```

# Correlation between measured and predicted logP values for the training set

```
> CorrLogPtrainingPLSR<-lm(PredLogPtrainingPLSR[,,42] ~ MeasuredLogPTraining)

> summary(CorrLogPtrainingPLSR)
```

```
Call:
lm(formula = PredLogPtrainingPLSR[,,42] ~ MeasuredLogPTraining)

Residuals:
    Min      1Q  Median      3Q     Max
-5.7136 -0.3253 -0.0128  0.3175  4.2668

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.195060   0.007449   26.18   <2e-16 ***
MeasuredLogPTraining   0.900762   0.002731  329.88   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5657 on 11989 degrees of freedom
Multiple R-squared:  0.9008,    Adjusted R-squared:  0.9008
F-statistic: 1.088e+05 on 1 and 11989 DF,  p-value: < 2.2e-16
```

# Predict logP from the test set

```
> PredLogPtestPLSR<-predict(LogPPLS,  newdata = LogPdata600BitsTest)
```

# Correlation between measured and predicted logP values for the test set

```
> CorrLogPtestPLSR<-lm(PredLogPtestPLSR[,,42] ~ MeasuredLogPTest)

> summary(CorrLogPtestPLSR)

Call:
lm(formula = PredLogPtestPLSR[,,42] ~ MeasuredLogPTest)

Residuals:
    Min      1Q  Median      3Q     Max
-3.3811 -0.3619 -0.0107  0.3368  4.6244

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.227963   0.016332   13.96   <2e-16 ***
MeasuredLogPTest 0.894523   0.005948  150.38   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5755 on 2835 degrees of freedom
Multiple R-squared:  0.8886,    Adjusted R-squared:  0.8886
F-statistic: 2.262e+04 on 1 and 2835 DF,  p-value: < 2.2e-16
```

2.3 Random forest regression

# Load package *randomForest* for random forest modeling

```
> library(randomForest)
```

# Use the function *randomForest()* to build the random forest model. The number of trees # (*ntree*) is set to 500; the node size (*nodesize*) is set to 5; the number of variables randomly # sampled at each tree node (*mtry*) is set to 1/3 the number of 600 fingerprint bits, i.e., 200

```
> LogPRF <- randomForest(LogP~., data= LogPdata600BitsTraining, ntree=500,
nodesize=5, mtry=200, importance=TRUE, na.action=na.omit)
```

# Predict logP from the training set

```
> PredLogPtrainingRF<-predict(LogPRF, LogPdata600BitsTraining)
```

# Correlation between measured and predicted logP values for the training set

```
> CorrLogPtrainingRF<-lm(PredLogPtrainingRF ~ MeasuredLogPTraining)

> summary(CorrLogPtrainingRF)

Call:
lm(formula = PredLogPtrainingRF ~ MeasuredLogPTraining)

Residuals:
    Min      1Q  Median      3Q     Max
-3.5732 -0.3002 -0.0052  0.2836  3.7885

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          0.362581   0.007681    47.2   <2e-16 ***
MeasuredLogPTraining 0.821148   0.002782   295.2   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.543 on 11368 degrees of freedom
Multiple R-squared:  0.8846,    Adjusted R-squared:  0.8846
F-statistic: 8.714e+04 on 1 and 11368 DF,  p-value: < 2.2e-16
```

# Predict logP from the test set

```
> PredLogPtestRF<-predict(LogPRF, LogPdata600BitsTest)
```

# Correlation between measured and predicted logP values for the test set

```
> CorrLogPtestRF<-lm(PredLogPtestRF ~ MeasuredLogPTest)

> summary(CorrLogPtestRF)
```

```
Call:
lm(formula = PredLogPtestRF ~ MeasuredLogPTest)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.1905 -0.3171 -0.0141  0.2906  4.5415 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)   0.353720   0.015868   22.29   <2e-16 ***
MeasuredLogPTest 0.829898   0.005779  143.60   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5592 on 2835 degrees of freedom
Multiple R-squared:  0.8791,    Adjusted R-squared:  0.8791 
F-statistic: 2.062e+04 on 1 and 2835 DF,  p-value: < 2.2e-16
```

2.4 Support vector regression

# Load package *e1071* for SVM modeling

```
> library(e1071)
```

# Use the function *svm()* to build the SVM model

```
> LogPSVM <- svm(LogP~., data= LogPdata600BitsTraining, cost = 150, epsilon = 0.05, gamma = 0.00014)
```

# Predict logP from the training set

```
> PredLogPtrainingSVM<-predict(LogPSVM, LogPdata600BitsTraining)
```

# Correlation between measured and predicted logP values for the training set

```
> CorrLogPtrainingSVM<-lm(PredLogPtrainingSVM ~ MeasuredLogPTraining)

> summary(CorrLogPtrainingSVM)

Call:
lm(formula = PredLogPtrainingSVM ~ MeasuredLogPTraining)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.0559 -0.0914 -0.0066  0.0741  7.0762 

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)    
(Intercept)          0.057734   0.002992   19.29   <2e-16 ***
MeasuredLogPTraining 0.975384   0.001083  900.31   <2e-16 ***
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2114 on 11358 degrees of freedom
Multiple R-squared:  0.9862,    Adjusted R-squared:  0.9862
F-statistic: 8.106e+05 on 1 and 11358 DF,  p-value: < 2.2e-16
```

# Predict logP from the test set

```
> PredLogPtestSVM<-predict(LogPSVM, LogPdata600BitsTest)
```

# Correlation between measured and predicted logP values for the test set

```
> CorrLogPtestSVM<-lm(PredLogPtestSVM ~ MeasuredLogPTest)

> summary(CorrLogPtestSVM)

Call:
lm(formula = PredLogPtestSVM ~ MeasuredLogPTest)

Residuals:
    Min      1Q  Median      3Q     Max
-1.8368 -0.2226 -0.0140  0.1971  4.2957

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       0.14057    0.01280   10.98   <2e-16 ***
MeasuredLogPTest  0.93824    0.00466  201.32   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4509 on 2831 degrees of freedom
Multiple R-squared:  0.9347,    Adjusted R-squared:  0.9347
F-statistic: 4.053e+04 on 1 and 2831 DF,  p-value: < 2.2e-16
```

2.5 Ten-fold cross validation by SVR

# Apply the built-in cross validation feature. Set the argument *cross* to 10.

```
> LogPSVMcv <- svm(LogP~., data = LogPdata600BitsTraining, cross = 10, cost =
150, epsilon = 0.05, gamma = 0.00014)
```

# Summary for 10-fold cross validation.

```
> summary(LogPSVMcv)

Call:
svm(formula = LogP ~ ., data = LogPdata600BitsTraining, cross = 10, cost = 150,
    epsilon = 0.05, gamma = 0.00014)
```

```
Parameters:
        SVM-Type:  eps-regression
        SVM-Kernel:  radial
        cost:  150
        gamma:  0.00014
        epsilon:  0.05

Number of Support Vectors:  9865

10-fold cross-validation on training data:

Total Mean Squared Error: 0.2289345
Squared Correlation Coefficient: 0.9317448
Mean Squared Errors:
 0.1862202    0.2602721    0.2445092    0.1835111    0.3111851
0.2232628    0.2362995    0.229375     0.2121004    0.2026094
```

2.6 Calculation of *k*-nearest neighbors for Applicability Domain (AD)

# The chemical space is the 600 FP bits selected by GA

```
> Data<- LogPdata600BitsTraining[, 1:600]

> Query<- LogPdata600BitsTest[, 1:600]
```

# Load package *FNN* for calculating 5-nearest neighbors

```
> library(FNN)

> Dtraining<-knn.dist(Data, k=5, algorithm=c("kd_tree", "cover_tree", "CR", "brute"))
```

# The distance of the test chemical from its five nearest neighbors in the training set

```
> Dtest<-knnx.dist(Data, Query, k=5, algorithm=c("kd_tree", "cover_tree", "CR",
"brute"))
```

3. Regression analysis for other properties using SVR

3.1 Water solubility (logS)

# Read in logS data with the optimal subset of 350 fingerprint bits

```
> LogSdata350Bits <- read.table("LogS-350-Fingerprint-Bits.txt", header=T, sep="\t",
as.is=T )
```

# There are 2010 rows (chemicals) and 353 columns (350 fingerprint bits + MW + logP +

# logS where MW and logP are employed as two additional variables)

```
> dim(LogSdata350Bits)
[1] 2010  353
```

# There are 1507 training chemicals and 503 test chemicals

```
> LogSdata350BitsTraining <-LogSdata350Bits[1:1507,]

> LogSdata350BitsTest <-LogSdata350Bits[1508:2010,]

> dim(LogSdata350BitsTraining)
[1] 1507  353

> dim(LogSdata350BitsTest)
[1] 503  353
```

# Use the function *svm()* to build the SVM model

```
> LogSSVM <- svm(LogS~., data = LogSdata350BitsTraining, cost = 260, epsilon =
0.145, gamma = 0.000031)
```

# Predict logS from the training set

```
> PredLogStrainingSVM<-predict(LogSSVM, LogSdata350BitsTraining)
```

# Correlation between measured and predicted logS values for the training set

```
> MeasuredLogSTraining<-LogSdata350BitsTraining$LogS

> CorrLogStrainingSVM<-lm(PredLogStrainingSVM ~ MeasuredLogSTraining)

> summary(CorrLogStrainingSVM)

Call:
lm(formula = PredLogStrainingSVM ~ MeasuredLogSTraining)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.10753 -0.25038 -0.01368  0.23977  1.79653 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)          -0.115862   0.015550  -7.451 1.55e-13 ***
MeasuredLogSTraining  0.955511   0.004648 205.591  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3879 on 1505 degrees of freedom
Multiple R-squared:  0.9656,    Adjusted R-squared:  0.9656 
F-statistic: 4.227e+04 on 1 and 1505 DF,  p-value: < 2.2e-16
```

# Predict logS from the test set

```
> PredLogStestSVM<-predict(LogSSVM, LogSdata350BitsTest)
```

# Correlation between measured and predicted logS values for the test set

```
> MeasuredLogSTest<-LogSdata350BitsTest$LogS

> CorrLogStestSVM<-lm(PredLogStestSVM ~ MeasuredLogSTest)

> summary(CorrLogStestSVM)

Call:
lm(formula = PredLogStestSVM ~ MeasuredLogSTest)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.41155 -0.33184  0.00721  0.33235  1.78058 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)      -0.14334    0.03733  -3.839 0.000139 ***
MeasuredLogSTest  0.92366    0.01050  87.966  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5416 on 501 degrees of freedom
Multiple R-squared:  0.9392,	Adjusted R-squared:  0.9391 
F-statistic:  7738 on 1 and 501 DF,  p-value: < 2.2e-16
```

# 10-fold cross validation by SVR. Set the argument *cross* to 10

```
> LogSSVMcv <- svm(LogS~., data = LogSdata350BitsTraining, cross = 10, cost = 260,
epsilon = 0.145, gamma = 0.000031)
```

# Summary for 10-fold cross validation.

```
> summary(LogPSSVMcv)

Call:
svm(formula = LogS~., data = LogSdata350BitsTraining, cross = 10, cost = 260, epsilon
= 0.145, gamma = 3.1e-05)

Parameters:
          SVM-Type:  eps-regression 
          SVM-Kernel:  radial 
          cost:  260 
          gamma:  3.1e-05 
          epsilon:  0.145 

Number of Support Vectors:  864
```

```
10-fold cross-validation on training data:

Total Mean Squared Error: 0.3342568
Squared Correlation Coefficient: 0.927789
Mean Squared Errors:
 0.3349027    0.2479804    0.3873414    0.4430222    0.2367207
0.316574      0.331453     0.2591354    0.4141808    0.371964
```

3.2 Bioconcentration factor (logBCF)

# Read in logBCF data with the optimal subset of 200 fingerprint bits

```
> LogBCFdata200Bits <- read.table("LogBCF-200-Fingerprint-Bits.txt", header=T,
sep="\t", as.is=T )
```

# There are 608 rows (chemicals) and 203 columns (200 fingerprint bits + MW + logP +

# logBCF where MW and logP are employed as two additional variables)

```
> dim(LogBCFdata200Bits)
[1] 608  203
```

# There are 456 training chemicals and 152 test chemicals

```
> LogBCFdata200BitsTraining <-LogBCFdata200Bits[1:456,]

> LogBCFdata200BitsTest <-LogBCFdata200Bits[457:608,]

> dim(LogBCFdata200BitsTraining)
[1] 456  203

> dim(LogBCFdata200BitsTest)
[1] 152  203
```

# Use the function *svm()* to build the SVM model

```
> LogBCFSVM <- svm(LogBCF~., data = LogBCFdata200BitsTraining, cost = 5500,
epsilon = 0.113, gamma = 0.0000385)
```

# Predict logBCF from the training set

```
> PredLogBCFtrainingSVM<-predict(LogBCFSVM, LogBCFdata200BitsTraining)
```

# Correlation between measured and predicted logBCF values for the training set

```
> MeasuredLogBCFTraining<-LogBCFdata200BitsTraining$LogBCF

> CorrLogBCFtrainingSVM<-lm(PredLogBCFtrainingSVM ~ MeasuredLogBCFTraining)

> summary(CorrLogBCFtrainingSVM)
```

```
Call:
lm(formula = PredLogBCFtrainingSVM ~ MeasuredLogBCFTraining)

Residuals:
    Min      1Q  Median      3Q     Max
-1.0486 -0.1246 -0.0140  0.1288  0.9293

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          0.042024   0.015876   2.647   0.0084 **
BCFdataTraining$LogBCF 0.973225   0.007051 138.018   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1875 on 454 degrees of freedom
Multiple R-squared:  0.9767,    Adjusted R-squared:  0.9767
F-statistic: 1.905e+04 on 1 and 454 DF,  p-value: < 2.2e-16
```

# Predict logBCF from the test set

```
> PredLogBCFtestSVM<-predict(LogBCFSVM, LogBCFdata200BitsTest)
```

# Correlation between measured and predicted logBCF values for the test set

```
> MeasuredLogBCFTest<-LogBCFdata200BitsTest$LogBCF

> CorrLogBCFtestSVM<-lm(PredLogBCFtestSVM ~ MeasuredLogBCFTest)

> summary(CorrLogBCFtestSVM)

Call:
lm(formula = PredLogBCFtestSVM ~ MeasuredLogBCFTest)

Residuals:
     Min       1Q   Median       3Q      Max
-0.84597 -0.26530  0.00363  0.27760  1.28193

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)         0.28110    0.05864   4.793 3.91e-06 ***
BCFdataTest$LogBCF  0.86743    0.02554  33.961  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4048 on 150 degrees of freedom
Multiple R-squared:  0.8849,    Adjusted R-squared:  0.8841
F-statistic:  1153 on 1 and 150 DF,  p-value: < 2.2e-16
```

# 10-fold cross validation by SVR. Set the argument *cross* to 10

```
> LogBCFSVMcv <- svm(LogBCF~., data = LogBCFdata200BitsTraining, cross = 10,
cost = 5500, epsilon = 0.113, gamma = 0.0000385)
```

# Summary for 10-fold cross validation.

```
> summary(LogBCFSVMcv)

Call:
svm(formula = LogBCF~., data = LogBCFdata200BitsTraining, cross = 10, cost =
5500, epsilon = 0.113, gamma = 3.85e-05)


Parameters:
        SVM-Type:  eps-regression
        SVM-Kernel:  radial
        cost:  5500
        gamma:  3.85e-05
        epsilon:  0.113

Number of Support Vectors:  359

10-fold cross-validation on training data:

Total Mean Squared Error: 0.2158818
Squared Correlation Coefficient: 0.8628049
Mean Squared Errors:
 0.2170155    0.2382286    0.2398231    0.193501     0.2513598
0.2760607     0.1301483    0.2446163    0.1722472    0.1982961
```

3.3 Boiling point (BP)

# Read in BP data with the optimal subset of 400 fingerprint bits

```
> BPdata400Bits <- read.table("BP-400-Fingerprint-Bits.txt", header=T, sep="\t",
as.is=T )
```

# There are 5432 rows (chemicals) and 402 columns (400 fingerprint bits + MW + BP

# where MW is employed as an additional variable)

```
> dim(BPdata400Bits)
[1] 5432  402
```

# There are 4074 training chemicals and 1358 test chemicals

```
> BPdata400BitsTraining <-BPdata400Bits[1:4074,]

> BPdata400BitsTest <-BPdata400Bits[4075:5432,]
> dim(BPdata400BitsTraining)
[1] 4074  402

> dim(BPdata400BitsTest)
[1] 1358  402
```

# Use the function *svm()* to build the SVM model

```
> BPSVM <- svm(BP~., data = BPdata400BitsTraining, cost = 9, epsilon = 0.012,
gamma = 0.0010)
```

# Predict BP from the training set

```
> PredBPtrainingSVM<-predict(BPSVM, BPdata400BitsTraining)
```

# Correlation between measured and predicted BP values for the training set

```
> MeasuredBPTraining<-BPdata400BitsTraining$BP

> CorrBPtrainingSVM<-lm(PredBPtrainingSVM ~ MeasuredBPTraining)

> summary(CorrBPtrainingSVM)

Call:
lm(formula = PredBPtrainingSVM ~ MeasuredBPTraining)

Residuals:
    Min       1Q   Median       3Q      Max
-181.628   -0.855    0.151    1.178   94.687

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       1.300775   0.250063   5.202 2.07e-07 ***
MeasuredBPTraining 0.992417   0.001206 823.066  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.556 on 4072 degrees of freedom
Multiple R-squared:  0.994,     Adjusted R-squared:  0.994
F-statistic: 6.774e+05 on 1 and 4072 DF,  p-value: < 2.2e-16
```

# Predict BP from the test set

```
> PredBPtestSVM<-predict(BPSVM, BPdata400BitsTest)
```

# Correlation between measured and predicted BP values for the test set

```
> MeasuredBPTest<-BPdata400BitsTest$BP

> CorrBPtestSVM<-lm(PredBPtestSVM ~ MeasuredBPTest)
> summary(CorrBPtestSVM)

Call:
lm(formula = PredBPtestSVM ~ MeasuredBPTest)

Residuals:
    Min       1Q   Median       3Q      Max 
-137.150   -6.595   -0.694    6.281  115.429 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)   7.082007   1.035835   6.837 1.22e-11 ***
MeasuredBPTest 0.963613   0.005009 192.370  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.63 on 1356 degrees of freedom
Multiple R-squared:  0.9647,    Adjusted R-squared:  0.9646 
F-statistic: 3.701e+04 on 1 and 1356 DF,  p-value: < 2.2e-16
```

# 10-fold cross validation by SVR. Set the argument *cross* to 10.

```
> BPSVMcv <- svm(BP~., data = BPdata400BitsTraining, cross = 10, cost = 9, epsilon
= 0.012, gamma = 0.0010)
```

# Summary for 10-fold cross validation.

```
> summary(BPSVMcv)

Call:
svm(formula = BP ~ ., data = BPdata400BitsTraining, cross = 10, cost = 9, 
   epsilon = 0.012, gamma = 0.001)

Parameters:
          SVM-Type:  eps-regression 
          SVM-Kernel:  radial 
          cost:  9 
          gamma:  0.001 
          epsilon:  0.012 

Number of Support Vectors:  3712

10-fold cross-validation on training data:

Total Mean Squared Error: 323.265 
Squared Correlation Coefficient: 0.9554685 
Mean Squared Errors:
 204.5346       400.041        332.6983       300.2321       327.1128 
425.8148       285.4233       321.4959       282.1084       353.0868 
```

3.4 Melting point (MP)

# Read in MP data with the optimal subset of 500 fingerprint bits

```
> MPdata500Bits <- read.table("MP-500-Fingerprint-Bits.txt", header=T, sep="\t",
as.is=T )
```

# There are 8648 rows (chemicals) and 503 columns (500 fingerprint bits + MW + BP + MP

# where MW and BP are employed as two additional variables)

```
> dim(MPdata500Bits)
[1] 8648  503
```

# There are 6485 training chemicals and 2163 test chemicals

```
> MPdata500BitsTraining <-MPdata500Bits[1:6485,]

> MPdata500BitsTest <-MPdata500Bits[6486:8648,]

> dim(MPdata500BitsTraining)
[1] 6485  503

> dim(MPdata500BitsTest)
[1] 2163  503
```

# Use the function *svm()* to build the SVM model

```
> MPSVM <- svm(MP~., data = MPdata500BitsTraining, cost = 9, epsilon = 0.18,
gamma = 0.00065)
```

# Predict MP from the training set

```
> PredMPtrainingSVM<-predict(MPSVM, MPdata500BitsTraining)
```

# Correlation between measured and predicted MP values for the training set

```
> MeasuredMPTraining<-MPdata500BitsTraining$MP

> CorrMPtrainingSVM<-lm(PredMPtrainingSVM ~ MeasuredMPTraining)

> summary(CorrMPtrainingSVM)

Call:
lm(formula = PredMPtrainingSVM ~ MeasuredMPTraining)

Residuals:
    Min       1Q   Median       3Q      Max
-269.829  -10.918    1.482   15.866  135.373

Coefficients:
```

```
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       5.619337   0.408137   13.77   <2e-16 ***
MeasuredMPTraining 0.908189   0.003224  281.70   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 25.56 on 6483 degrees of freedom
Multiple R-squared:  0.9245,    Adjusted R-squared:  0.9245
F-statistic: 7.935e+04 on 1 and 6483 DF,  p-value: < 2.2e-16
```

# Predict MP from the test set

```
> PredMPtestSVM<-predict(MPSVM, MPdata500BitsTest)
```

# Correlation between measured and predicted MP values for the test set

```
> MeasuredMPTest<-MPdata500BitsTest$MP

> CorrMPtestSVM<-lm(PredMPtestSVM ~ MeasuredMPTest)

> summary(CorrMPtestSVM)

Call:
lm(formula = PredMPtestSVM ~ MeasuredMPTest)

Residuals:
    Min       1Q   Median       3Q      Max
-236.970  -19.434   -0.092   23.082  139.926

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.904217   1.086860   7.273 4.91e-13 ***
MeasuredMPTest 0.844176   0.008327 101.379  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 39.14 on 2161 degrees of freedom
Multiple R-squared:  0.8263,    Adjusted R-squared:  0.8262
F-statistic: 1.028e+04 on 1 and 2161 DF,  p-value: < 2.2e-16
```

# 10-fold cross validation by SVR. Set the argument *cross* to 10

```
> MPSVMcv <- svm(MP~., data = MPdata500BitsTraining, cross = 10, cost = 9,
epsilon = 0.18, gamma = 0.00065)
```

# Summary for 10-fold cross validation.

```
> summary(MPSVMcv)

Call:
svm(formula = MP ~ ., data = MPdata500BitsTraining, cross = 10, cost = 9,
    epsilon = 0.18, gamma = 0.00065)

Parameters:
          SVM-Type:  eps-regression
          SVM-Kernel:  radial
          cost:  9
          gamma:  0.00065
          epsilon:  0.18

Number of Support Vectors:  3718

10-fold cross-validation on training data:

Total Mean Squared Error: 1715.096
Squared Correlation Coefficient: 0.8236632
Mean Squared Errors:
 1671.525     1841.16      2051.508     1726.576     1771.585
1511.029      1472.101     1583.778     1678.249     1843.559
```

3.5 Vapor pressure (logVP)

# Read in logVP data with the optimal subset of 350 fingerprint bits

```
> LogVPdata350Bits <- read.table("LogVP-350-Fingerprint-Bits.txt", header=T,
sep="\t", as.is=T )
```

# There are 2713 rows (chemicals) and 353 columns (350 fingerprint bits + MW + BP +

# logVP where MW and BP are employed as two additional variables)

```
> dim(LogVPdata350Bits)
[1] 2713  353
```

# There are 2034 training chemicals and 679 test chemicals

```
> LogVPdata350BitsTraining <-LogVPdata350Bits[1:2034,]

> LogVPdata350BitsTest <-LogVPdata350Bits[2035:2713,]

> dim(LogVPdata350BitsTraining)
[1] 2034  353

> dim(LogVPdata350BitsTest)
[1] 679  353
```

# Use the function *svm()* to build the SVM model

```
> LogVPSVM <- svm(LogVP~., data = LogVPdata350BitsTraining, cost = 115, epsilon
= 0.105, gamma = 0.00011)
```

# Predict logVP from the training set

```
> PredLogVPtrainingSVM<-predict(LogVPSVM, LogVPdata350BitsTraining)
```

# Correlation between measured and predicted logVP values for the training set

```
> MeasuredLogVPTraining<-LogVPdata350BitsTraining$LogVP

> CorrLogVPtrainingSVM<-lm(PredLogVPtrainingSVM ~ MeasuredLogVPTraining)

> summary(CorrLogVPtrainingSVM)

Call:
lm(formula = PredLogVPtrainingSVM ~ MeasuredLogVPTraining)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.1738 -0.3054  0.0217  0.2409  6.8797 

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)    
(Intercept)           -0.039999   0.012578   -3.18  0.00149 ** 
MeasuredLogVPTraining  0.969786   0.003067  316.23  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4948 on 2032 degrees of freedom
Multiple R-squared:  0.9801,	Adjusted R-squared:  0.9801 
F-statistic: 1e+05 on 1 and 2032 DF,  p-value: < 2.2e-16
```

# Predict logVP from the test set

```
> PredLogVPtestSVM<-predict(LogVPSVM, LogVPdata350BitsTest)
```

# Correlation between measured and predicted logVP values for the test set

```
> MeasuredLogVPTest<-LogVPdata350BitsTest$LogVP

> CorrLogVPtestSVM<-lm(PredLogVPtestSVM ~ MeasuredLogVPTest)

> summary(CorrLogVPtestSVM)

Call:
lm(formula = PredLogVPtestSVM ~ MeasuredLogVPTest)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.5793 -0.3397  0.0439  0.3684  4.1752 
```

```
Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      -0.032247   0.036288  -0.889    0.375
MeasuredLogVPTest  0.946577   0.008738 108.324   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8096 on 677 degrees of freedom
Multiple R-squared:  0.9455,    Adjusted R-squared:  0.9454
F-statistic: 1.173e+04 on 1 and 677 DF,  p-value: < 2.2e-16
```

# 10-fold cross validation by SVR. Set the argument *cross* to 10

```
> LogVPSVMcv <- svm(LogVP~., data = LogVPdata350BitsTraining, cross = 10, cost =
115, epsilon = 0.105, gamma = 0.00011)
```

# Summary for 10-fold cross validation.

```
> summary(LogVPSVMcv)

Call:
svm(formula = LogVP ~ ., data = LogVPdata350BitsTraining, cross = 10, cost = 115,
    epsilon = 0.105, gamma = 0.00011)

Parameters:
          SVM-Type:  eps-regression
          SVM-Kernel:  radial
          cost:  115
          gamma:  0.00011
          epsilon:  0.105

Number of Support Vectors:  954

10-fold cross-validation on training data:

Total Mean Squared Error: 0.9116988
Squared Correlation Coefficient: 0.9288863
Mean Squared Errors:
 1.167059       0.8566831      0.9616998      0.9246135      1.095954
0.6489537       0.8176335      0.8023975      1.118331       0.7239725
```