QSAR Techniques & Applications in Discovery Studio

Introduction

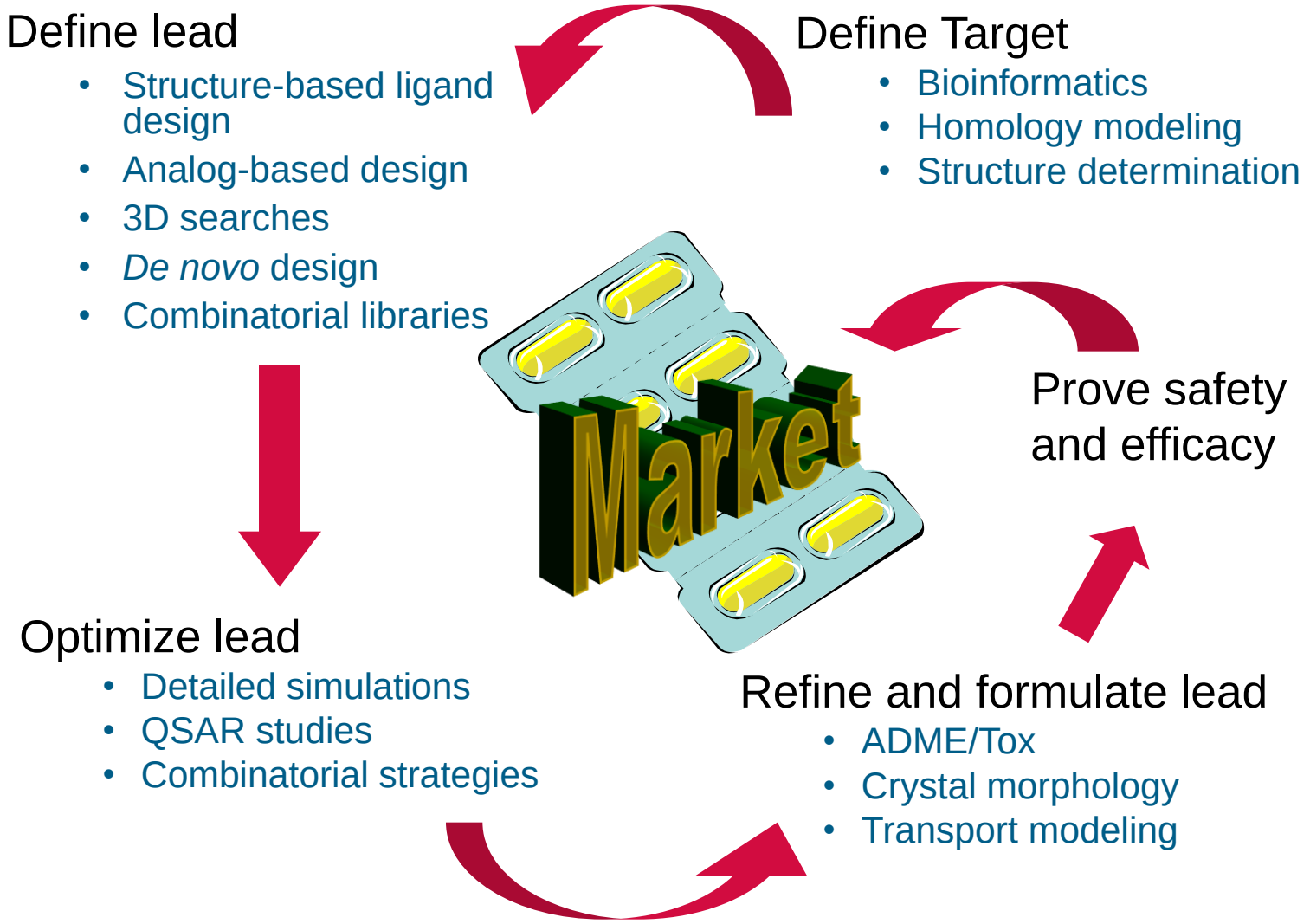
By
Anand Krishnamurthy
Senior Scientist-LS Modeling & Simulations

accelrys®

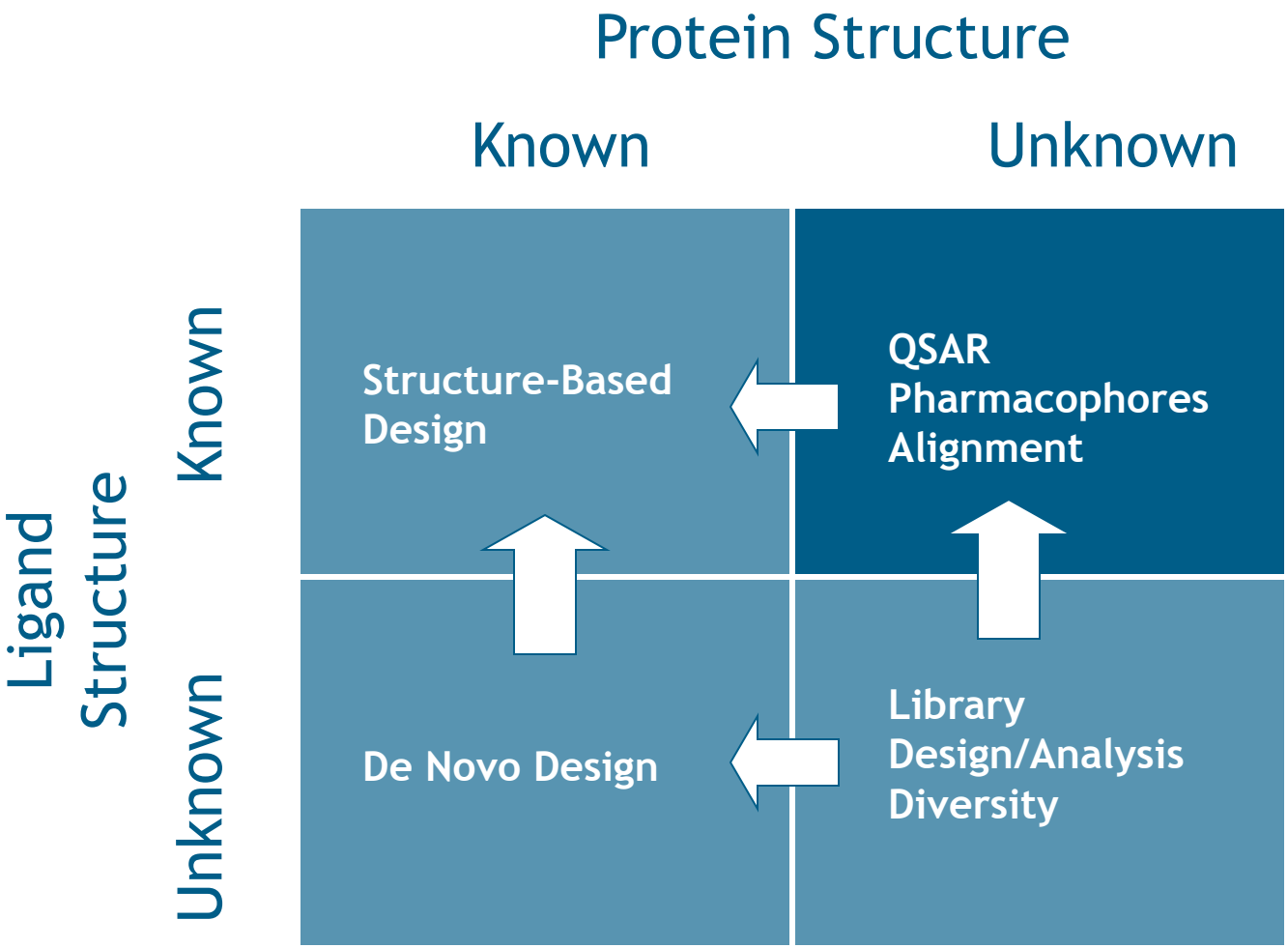
Agenda

- Introduction to QSAR
- Working with descriptors
  - available descriptors
  - how to calculate them
  - how to use them
- Regression techniques
  - MLR, PLS
- Modern methods
  - GFA, Bayesian model, Neural network, Recursive Partitioning
- Example workflow

Stages and Strategies of Drug Discovery



Scenarios


Protein Structure
Known
Unknown
Ligand Structure
Known
Unknown
Structure-Based Design
QSAR
Pharmacophores
Alignment
De Novo Design
Library
Design/Analysis
Diversity

Ligand Based Approaches

- Statistical analysis of the relationships between molecular structures and their descriptors to provide correlations for predicting biological activities (QSAR)
- Exploring common pharmacophore features amongst a set of active compounds (Pharmacophore modelling)
- Derive predictive models if SAR data is available (QSAR and Pharmacophore modelling)
- Searching for compounds with similar properties (Library analysis & Pharmacophore modelling)

What is QSAR?

- Quantitative Structure-Activity Relationships
- Addresses two questions:
  1. What features of a molecule affect its activity?
  2. What can be modified to enhance properties?
- Quantitative in that a mathematical model is used to account for the observed activity
- Helps to gain insight into underlying biological processes
- It alleviates the need to determine the experimental value of the property of hundreds of similar compounds that would take large amounts of resources to determine individually

History of QSAR

1868

Crum Brown and Frazer*

- Proposed that a“physiological action” was a function of “chemical constitution”…

$$\Phi = f(C)$$

- So how to describe *f(C)* ?

1950’s

Corwin Hansch’s work

1964

“Fragment and additive group contribution theory”

Hansch and Fujita

QSAR

Conditions: Y observations (Dependent variable)
X parameters (Independent variable)

Objective: Correlate Y with X1, X2 ...

Challenge: Variance is spread over X parameters
Find the QSAR signal ... in a huge field of variance!

- "Variations in X1, X2 ... that are correlated with changes in biological activity... and make sense!"

Considerations

Quality of Y:

"Strength of a QSAR model depends on the quality of the dependent variable"

Choice of X:

"Improper choice of independent variables often results in a poor QSAR model"

Overfitting:

"With enough parameters, you can correlate anything with anything!"

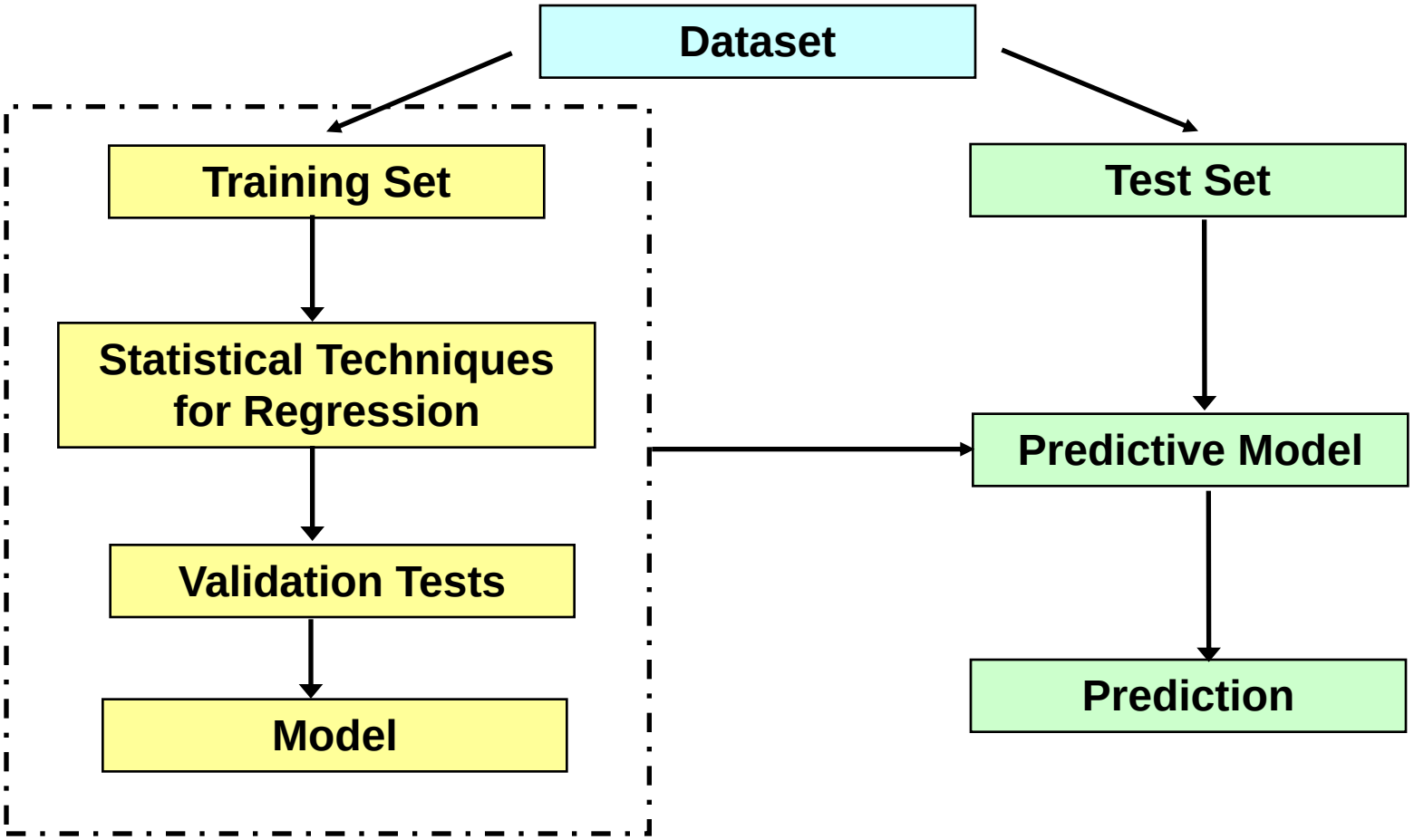
Training Set:

Quality of training set directly influences the quality of the final model

Applicability of QSAR

- QSAR models normally have local validity
  - can embrace only compounds with similar chemical data
- There is a trade-off between chemical diversity and applicability of the model
  - Local vs. global models
  - Local models aimed at particular series of compounds whereas global models can be valid for a more diverse set of data
- Caution: models are often specific to the same type of compounds as was used to derived them
  - Compare similarity of test set/external compound to training set compounds
  - Applicability domain

QSAR Flow Chart

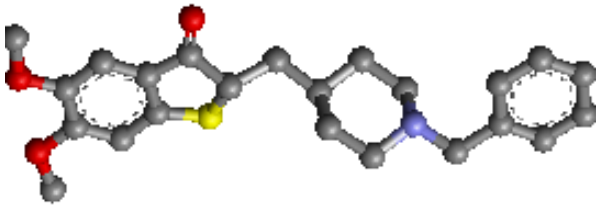


QSAR Workflow Steps

Data Set Preparation Biological activity 3D structures (if needed)	Import structures from various file formats Draw structures (sketching tools) Build congeneric series (Library Design protocols) Enter biological data via Data Table (Molecule Window) Copy and paste from text/spreadsheet file Generate Conformations protocol Minimization protocol
Descriptor Calculation 2D vs 3D	Calculate Molecular Properties protocol
Data Exploration Descriptors Training and Test sets	Charting tools, statistical analysis options (Calculate Molecular Properties protocol) Calculate Principal Components and Cluster Ligands (Library Analysis protocols)
QSAR Generation Appropriate methodology	Multiple Linear Regression, Partial Least Squares Back Propagation Neural Network, Bayesian, Genetic Function Approximation, Recursive Partitioning
Model Validation	Internal during model generation (cross validation) External on test set after model generation Outlier detection, Graphical analysis
Model Analysis	Charting tools to plot observed vs. predicted activities, Analysis of descriptors making up the model, Plotting estimated error
Model Prediction	Calculate Molecular Properties protocol to apply model

Application Example

- ARICEPT® for Alzheimer's disease – Eisai
  - Donepezil: Acetyl cholinesterase inhibitor developed via QSAR, molecular shape analysis, docking



  - Kawakami, Y. *et al*. *Bioorg. Med. Chem*. **1996**, *4*, 1429-1446

QSAR Reference Papers

- Rogers, D.; Hopfinger, A. J. 2Application of Genetic Function Approximation to Quantitative Structure-Activity Relationships and Quantitative Structure-Property Relationships", *J. Chem. Inf. Comput. Sci.*, **1994**, 34, 854

- Y. Fan, L. M. Shi, K. W. Kohn, Y. Pommier, and J. N. Weinstein. "Quantitative Structure-Antitumor Activity Relationships of Camptothecin Analogues: Cluster Analysis and Genetic Algorithm-Based Studies," *J. Med. Chem*., **2001**, 44, 3254-3263

- S. Weng, S. Sakamuri, I. J. Enyedy, A. P. Kozikowski, W. A. Zaman, K. M. Johnson. “Molecular Modeling, Structure-Activity Relationships and Functional Antagonism Studies of 4-Hydroxy-1-methyl-4-(4-methylphenyl)-3-piperidyl 4-Methylphenyl Ketones as a Novel Class of Dopamine Transporter Inhibitors,” *Bioorg. Med. Chem.*, **2001**, 9, 1753-1764

- Bureau R, Daveu C, Lancelot JC, Rault S. "Molecular Design Based on 3D-Pharmacophore. Application to 5-HT Subtypes Receptors,", *J Chem Inf Comput Sci.,* **2002**, 429-36

- Bhonsle JB, Bhattacharjee AK, Gupta RK. “Novel semi-automated methodology for developing highly predictive QSAR models: application for development of QSAR models for insect repellent amides”, J Mol Model. 2007 Jan;13(1):179-208. Epub 2006 Sep 20

QSAR Techniques & Applications in Discovery Studio

Working with Descriptors

accelrys®

Preparations

- Calculating descriptors
  - Examine correlations between descriptors
  - Examine data distribution
  - Any missing data?
  - Constants?
  - Explore Dataset visually in property space
- Identifying the training set
  - QSAR training set
    - Minimum of around 20-30 compounds
      - can use less if needed
    - Structurally diverse
      - Global vs. local model
    - Activity range
      - Extrapolation
      - Coverage of each activity range

Independent variables

- Physicochemical properties that describe some aspect of the chemical structure
- May be determined:
  - Experimentally
    - Melting point
    - NMR chemical shift
  - Calculated
    - logP
    - Electronic substituent constant
    - Molar refractivity
- Called ***descriptors***
- Choice of descriptors
  - Interpretability vs. predictability

Available descriptors in Discovery Studio

- AlogP
- Molecular Formats
- Element Counts
- Molecular Properties
- Molecular Property Counts
- Surface Area and Volume
- Topological Descriptors
- Fingerprints
  - Extended to SciTegic ECFP, FCFP, UserKeys
- Estate Keys
- Dipole
- Jurs Descriptors
- Principal Moments of Inertia
- Shadow Indices

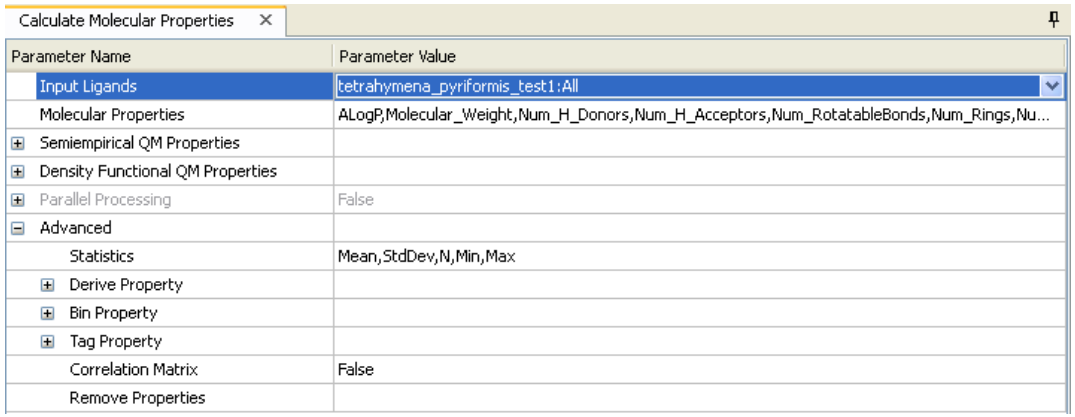
- Semiemperical QM
- Density Functional QM
- Other (includes user models & random number)

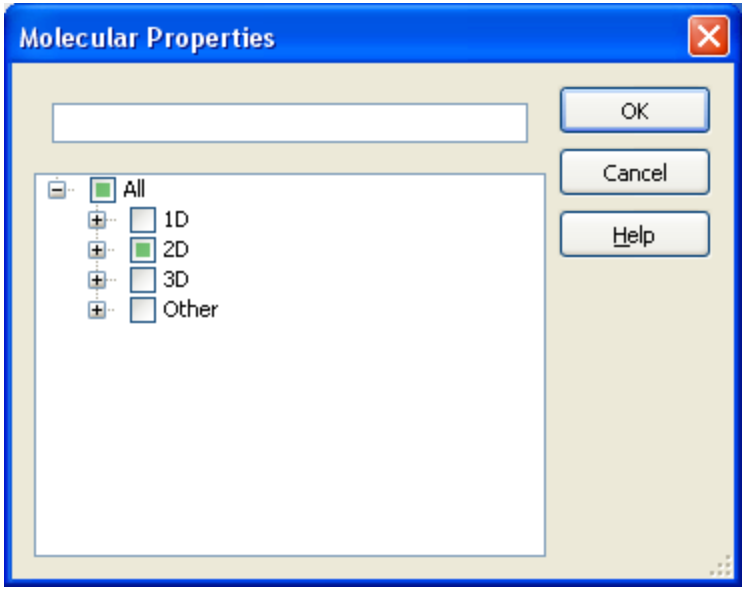
Full list and description is given under the General Purpose > Theory - General Purpose > Descriptors > Molecular properties

Section under the Help section

Protocol: Calculate Molecular Properties

- Central place for descriptor calculation
- Calculate many properties that can be used to create a QSAR model
  - traditional molecular descriptors, Semiempirical QM descriptors, Density Functional QM descriptors, and properties that can be calculated from user-built QSAR models
  - can be used to derive a new property based on other properties
  - can conduct basic statistics and correlations of all numeric properties





Topological descriptors

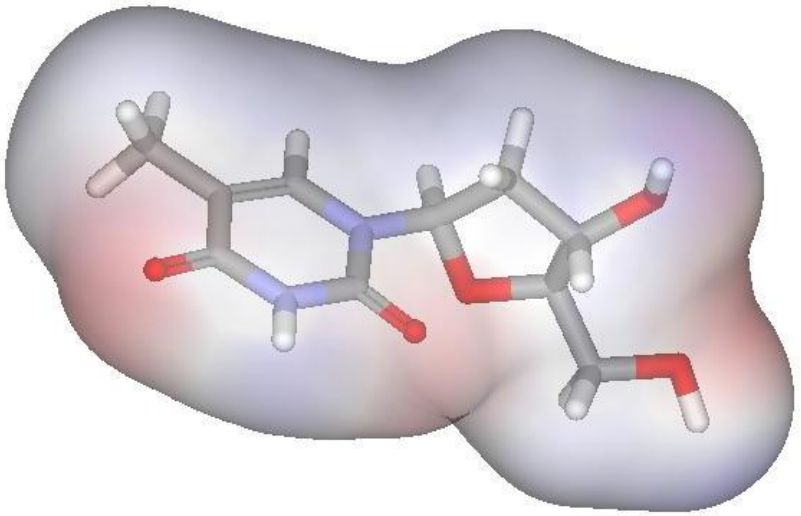
- 2D descriptors
- All calculations are derived from the two-dimensional topology of the molecule
- Relate to the molecule's size, overall shape and degree of branching
- Following options available in Discovery Studio
  - Balaban Indices
  - Wiener Index
  - Zagreb Index
  - Connectivity Indices
  - Graph-Theoretical InfoContent descriptors
  - Kappa Shape Indices
  - Subgraph Counts

Other 2D Descriptors

- AlogP
  - atom-based method published by Ghose and Crippen to calculate the octanol-water partition coefficient (LogP), and the molar refractivity (MR). LogP provides a measure of the hydrophobicity of the molecule, while MR contains information about molecular volume and polarizability
- Molecular Property Counts & Element counts
  - Number of hydrogen bond acceptor, donor, rotatable bonds, rings…
  - Number of particular element types
- Surface and Volume using 2D estimation
- Estate Keys
  - Calculates the sums of the Electrotopological State (E-state) values and/or the counts of each atom type

3D Descriptors

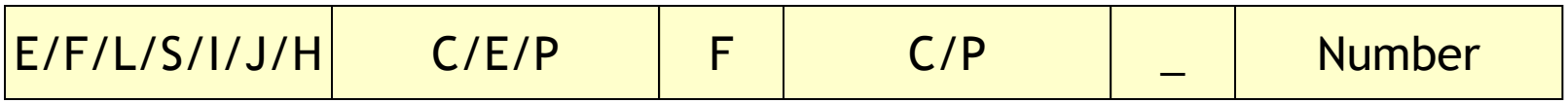
- Spatial descriptors
  - Jurs descriptors
  - Shadow Indicies
  - Inertia
  - Surface Area and Volume
- Electronic
  - Charges
  - Dipole
- Energy



Fingerprints

- Often used in diversity selection, library design & analysis work
- Can also be used in QSAR applications
  - Can be used **only** for qualitative model
  - Create Bayesian Model protocol only
- Quick to calculate and can be used to assess structural diversity in the dataset
- The following algorithms are available to calculate fingerprints:
  - SciTegic extended-connectivity fingerprints
  - Daylight-style path fingerprints
  - Atom Environment fingerprints
  - MDL public key fingerprints

Fingerprints – Naming Convention



Atom code:
E: Atom type
F: Functional class
L: ALogP types
S: Sybyl atom type
I: Prototype tautomer-independent Inchi-based atom type (analogous to E)
J: Prototype tautomer-independent Inchi-based functional class (analogous to F)
H: TypeChargeAndHybridization

Fingerprint type:
C: Extended connectivity
E: Atom Environment
P: Daylight Path-based

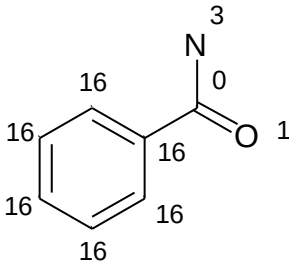
Output type:
C: Counts
(duplicates retained)
P: Fingerprints
(duplicates removed)

Maximum bond or path length

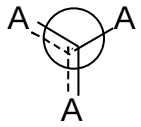
Note: Not all combinations are available

Other fingerprints available include MDLPublicKeys and UserKeys (fingerprints derived from substructures that you define (user key fingerprints).

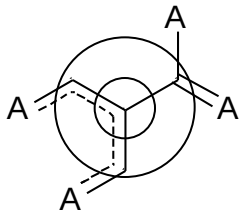
Fingerprints Generation Algorithm



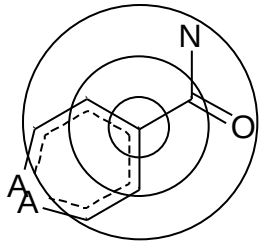
FCFP atom code bits from:
- 1: Has lone pairs
- 2: Is H-bond donor
- 4: Is negative ionizable
- 8: Is positive ionizable
- 16: Is aromatic
- 32: Is halogen



Iteration 0



Iteration 1



Iteration 2

Each iteration adds bits that represent larger and larger structures

- Process based on the Morgan algorithm

1. Each atom is given an initial atom code
2. Iterations are performed to generate higher-order features
3. Hashing

Common fingerprints – ECFP and FCFP

- Extended Connectivity Fingerprints (ECFPs)
  - Each bit represents the presence/absence of a structural feature
  - Do not depend on a predefined set of substructural features
  - Fast to calculate
  - 4 Billion possible different bits
  - Bits can be “interpreted”
  - Typical molecule generates ~100 bits
  - Typical library generates 100K - 10M different bits.

- Functional-Class Fingerprints (FCFPs)
  - Uses the role of an atom in the initial atom code rather than atom type
    - Hydrogen-bond acceptor / donor
    - Positively ionized or positively ionizable
    - Negatively ionized or negatively ionizable
    - Aromatic
    - Halogen

QM Properties

- Semiemperical (VAMP) and Density Functional (DMol3) QM descriptors

Semi-empirical QM Properties:

Total_Energy_VAMP,

Heat_of_Formation_VAMP,

HOMO_Eigenvalue_VAMP,

LUMO_Eigenvalue_VAMP,

Electronic_Energy_VAMP,

Molecular_Surface_Area_VAMP,

Molecular_Point_Group_VAMP,

Dipole_Mag_VAMP,

Dipole_Components_VAMP,

Quadrupole_Components_VAMP,

Octupole_Components_VAMP,

Mean_Polarizability_VAMP,

Propgen_Outputs_VAMP

Density Functional QM Properties:

Total_Energy_DMol3,

Binding_Energy_DMol3,

HOMO_Energy_DMol3,

LUMO_Energy_DMol3,

Band_Gap_Energy_DMol3,

Dipole_Mag_DMol3,

Dipole_Components_DMol3,

Dielectric_Energy_DMol3,

Solvation_Energy_DMol3,

Surface_Area_DMol3,

Cavity_Volume_DMol3

Other descriptors

- Descriptors generated from 3rd party applications
  - Can be imported as property fields in sd files
  - Copy/paste into new column directory from a text file
  - Customised protocol to join data from csv file & merge with sd file

- Descriptors from other methods
  - docking, pharmacophore
    - Fit value, LigScore, PLP, Ludi scores, number of hydrogen bond interactions…etc

- Must be calculated beforehand and also available for test set compounds

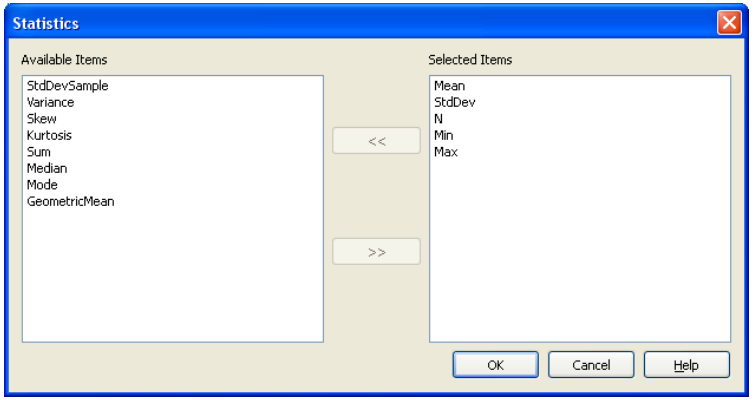
Exploring the calculated descriptors

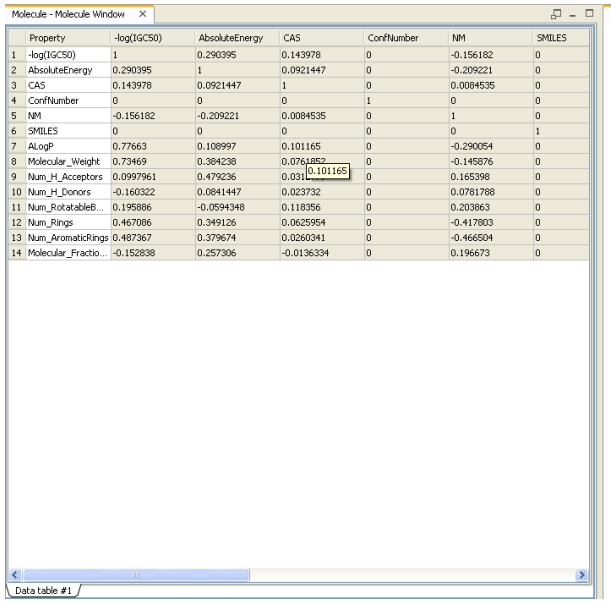
- Aim
  - Identify descriptors that correlate with biological activity (signal)
  - Leave the others behind (noise)
  - Compute and validate models that involve these descriptors

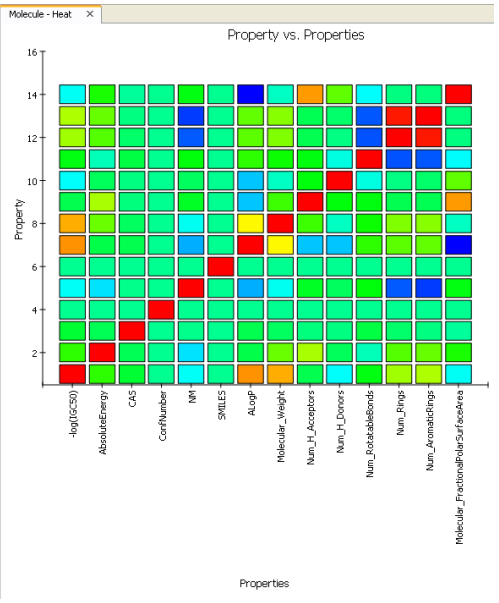
- QSAR methods
  - Some methods are more sensitive to descriptor selections
    - MLR, PLS
  - Methods such as GFA, Bayesian model, Neural Network or Recursive Partitioning less sensitive to these

Exploring the calculated descriptors

- Statistics can be calculated for each descriptors
  - Advanced section of the Calculate Molecular Protocols
  - Includes option to calculate correlation matrix
    - Shows inter-correlated descriptors
      - Can be used to remove redundant descriptors
- Charting tools
  - Plot correlation
    - ViewCorrelation.pl







Exploring the calculated descriptors

- Statistical summary
  - Variance, mean, Standard deviation, median, minimum, maximum, sum, kurtosis, skew, count
- Correlation Coefficients
  - Explains the covariance of a pair of variables
  - Positive if they change the same way
  - Negative if one gets smaller as the other gets larger
  - Values from 1 to -1
- Other tools
  - Histograms
    - Chart/Histogram
  - 2D and 3D Plot
    - Chart/Line Plot, Point Plot, 3D Plot
- Customised protocol
  - Remove zero variance descriptors

Preparations

- Calculating descriptors
  - Examine correlations between descriptors
  - Examine data distribution
  - Any missing data?
  - Constants?
  - Explore Dataset visually in property space

- Identifying the training set
  - QSAR training set
    - Minimum of around 20-30 compounds
      - can use less if needed
    - Structurally diverse
      - Global vs. local model
    - Activity range
      - Extrapolation
      - Coverage of each activity range

Training & Test Set Selection

- Generic analysis tools
  - Charting tools
  - Different viewing options on the data in the Table Browser
    - Group By
    - Represent By
    - Sort

- Protocols that can help in training & test set selection
  - Principal Component Analysis
    - Allows for visualisation of the data in property space

Training & Test Set Selection

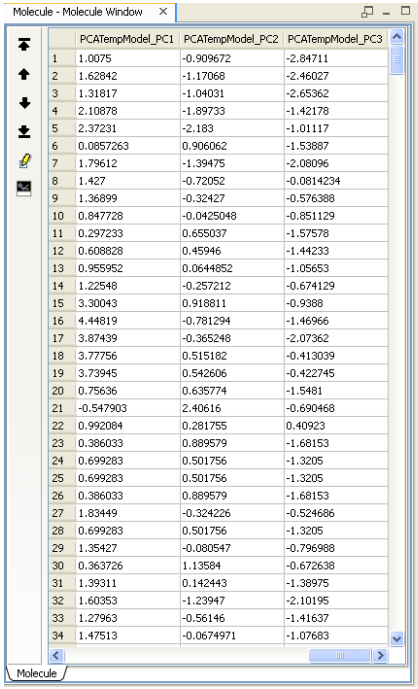
- Protocols that can help in training & test set selection (cont.)
  - Cluster Ligands (Library Analysis)
    - Clustering is based on the RMS difference of descriptor properties, or Tanimoto distance for fingerprints
    - relocation method based on maximal dissimilarity partitioning
  - Find Diverse Molecules (Library Analysis)
    - Selects a subset of ligands that are diverse with respect to the specified properties
    - The selection is based on a maximum dissimilarity method using the same algorithm as the Cluster Ligands protocol
  - Random selection
    - Customised protocol
      - Split data randomly
      - For Example: Select 30% of dataset randomly to use as test set

Training & Test Set Selection

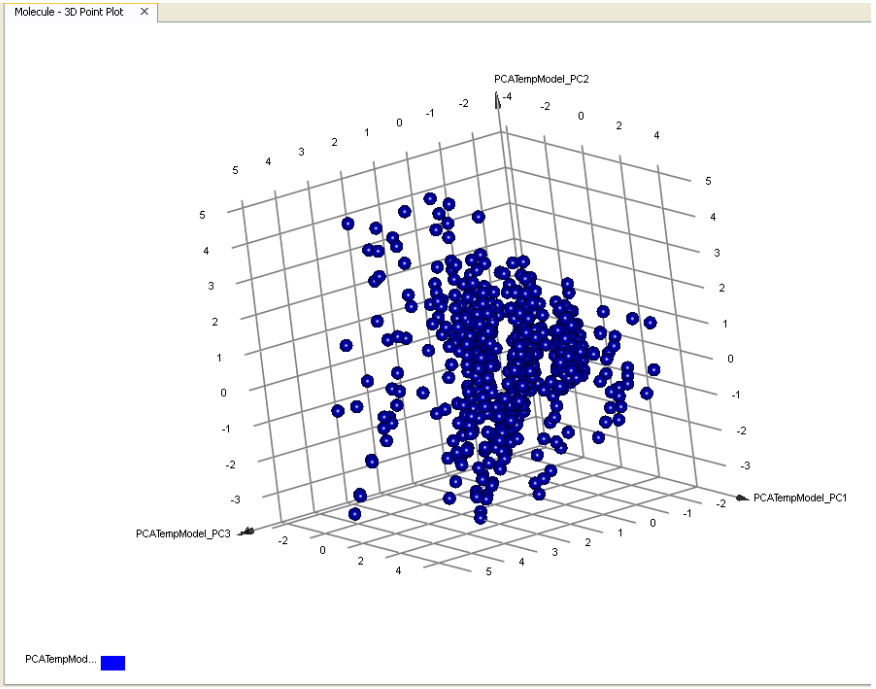
- Visualise dataset in property space
  - compress overall variance in the descriptors to a smaller number of components in order to visualise in 3D Space
    - Calculate Principal Component protocol (Library Analysis)
- Principal Component Analysis
  - Used to reduce dimensionality of full set of descriptors
  - Express several descriptors into a single principal component
  - PCs can be used as new variables in QSAR
  - Enables visualization of models in descriptor space
  - Removes redundancies due to correlation between descriptors
  - Can be run as a pre-cursor to multiple linear regression to perform a principal component regression analysis

Protocol: Calculate Principal Component

- User controllable:
  - Minimum variance explained
  - Minimum number of components
  - Scaling option used
- Descriptors to include
  - Pre-defined Set
    - Will calculate these during the run
  - User Defined
    - Selected from already calculated properties



	PCATempModel_PC1	PCATempModel_PC2	PCATempModel_PC3
1	1.0075	-0.909672	-2.84711
2	1.62842	-1.17068	-2.46027
3	1.31817	-1.04031	-2.65362
4	2.10878	-1.89733	-1.42178
5	2.37231	-2.183	-1.01117
6	0.0857263	0.906062	-1.53887
7	1.79612	-1.39475	-2.08096
8	1.427	-0.72052	-0.0814234
9	1.36899	-0.32427	-0.576388
10	0.847728	-0.0425048	-0.851129
11	0.297233	0.655037	-1.57578
12	0.608828	0.45946	-1.44233
13	0.955952	0.0644852	-1.05653
14	1.22548	-0.257212	-0.674129
15	3.30043	0.918811	-0.9388
16	4.44819	-0.781294	-1.46966
17	3.87439	-0.365248	-2.07362
18	3.77756	0.515182	-0.413039
19	3.73945	0.542606	-0.422745
20	0.75636	0.635774	-1.5481
21	-0.547903	2.40616	-0.690468
22	0.992084	0.281755	0.40923
23	0.386033	0.889579	-1.68153
24	0.699283	0.501756	-1.3205
25	0.699283	0.501756	-1.3205
26	0.386033	0.889579	-1.68153
27	1.83449	-0.324226	-0.524686
28	0.699283	0.501756	-1.3205
29	1.35427	-0.080547	-0.796988
30	0.363726	1.13584	-0.672638
31	1.39311	0.142443	-1.38975
32	1.60353	-1.23947	-2.10195
33	1.27963	-0.56146	-1.41637
34	1.47513	-0.0674971	-1.07683

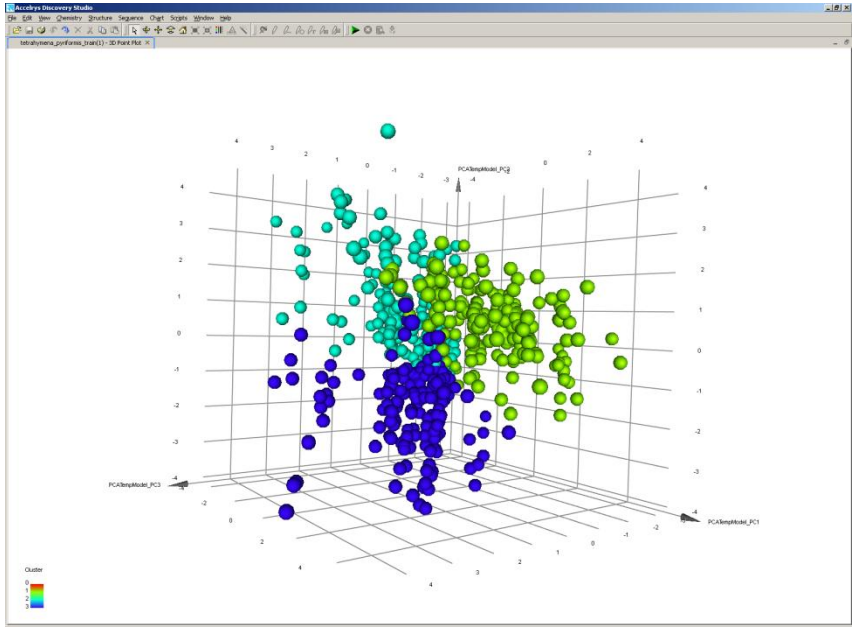


Cluster Analysis

- Objective:
  - to partition a dataset into classes or categories consisting of elements of comparable similarity
- Can be used for:
  - Diversity selection
  - Organizing large data sets
  - Visualization of variety in a set of data
- Approach
  - a maximum dissimilarity method is used
  - Some samples first chosen as cluster centers
  - Other samples then assigned to the nearest cluster center
  - Can use numeric data, fingerprints, or both together
  - CPU time scales as N x C (#samples x #cluster centers)

Protocol: Cluster Ligands

- Can be used to cluster data on the calculated PCs returned from the Calculate Pricipal Component protocol
- Can also be used on fingerprints or any other numerical data
- Visualise Clusters in descriptor space using 3D Plot
  - Cluster Ligands based on calculated PCs
  - Open resulting sd file
  - Sort on Distance to Cluster
    - First n compounds will be the n cluster centres returned
  - Generate 3D plot
    - Xyz axis = PC1, PC2, PC3
    - Color axis = Cluster



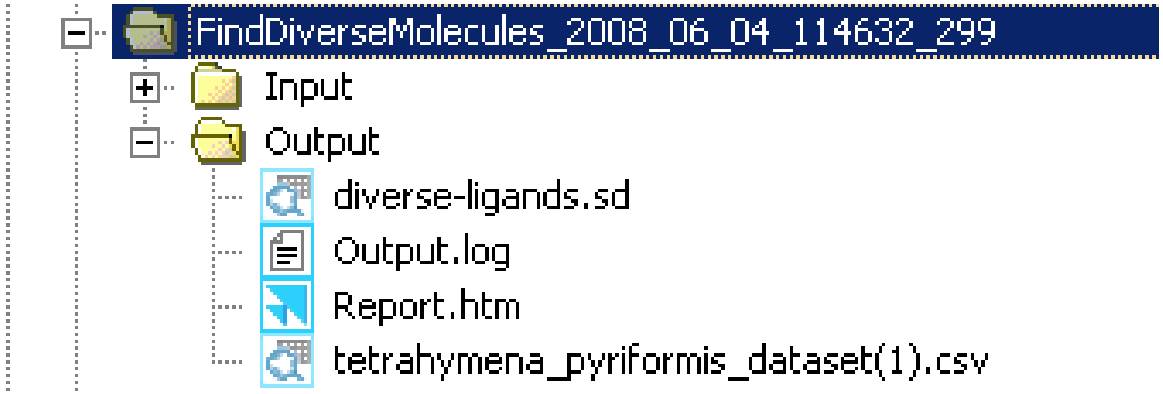
- Results
  - Cluster-Centers.sd
    - cluster-centers
  - Input_Ligands.sd
    - cluster assignment
      - DistanceToCenter

Diversity Selection

- Objective
  - To select a subset of diverse ligands from a library
- Can be used for
  - Selecting a representative test set
  - Selecting a dataset for other methods such as docking or pharmacophore modelling
- Approach
  - Diversity is measured with respect to the specified properties
  - The selection is based on a maximum dissimilarity method

Protocol: Find Diverse Molecules

- Can be used with PCs, numerical descriptors as well as with fingerprints
- Returns selected diverse subset as a separate sd file



Training & Test Set Selection: Example

Dataset

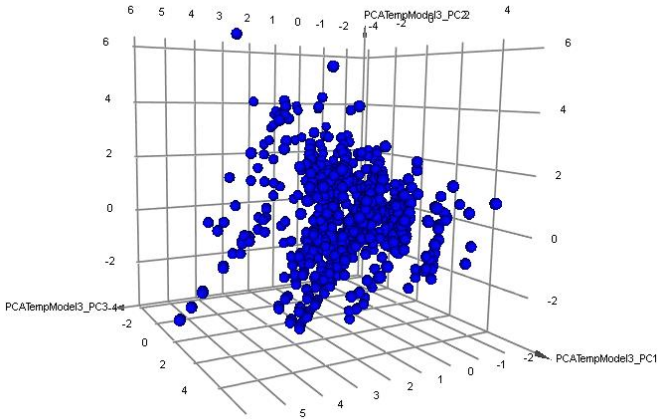
983 compounds in an sd file

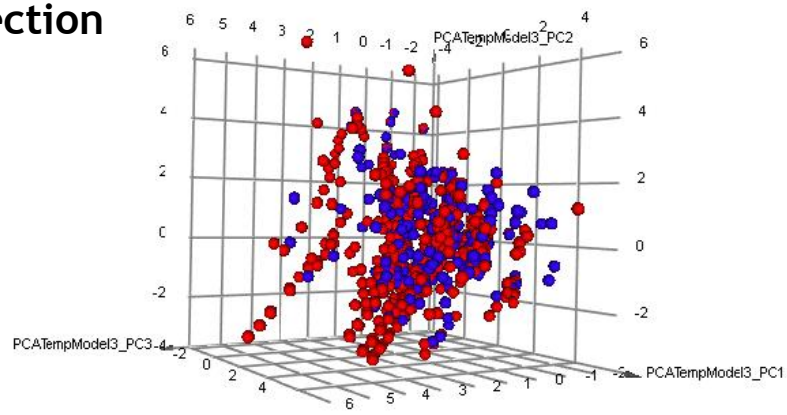
Training Set

783 compounds

Test Set

300 compounds
Based on diverse selection





Find Diverse Molecules

Parameter Name	Parameter Value
Input Ligands	tetrahymena_pyriformis_dataset(1)
Subset Size	300
Use Properties	PredefinedSet
Predefined Set	FCFP_4
User Set	

QSAR Techniques & Applications in Discovery Studio

Regression Methods

accelrys®

Regression techniques

- Under-Determined Dataset

Conditions: N molecules >> P descriptors

Example: Hansch type analysis

Method: Multiple Linear Regression

- Over-Determined Dataset

Conditions: P descriptors >> N molecules
Large number of correlated descriptors

Example: QSAR with high dimensionality descriptors (3D Fields)

Method: PLS, GFA

Multiple Linear Regression

- The mathematical model that relates the y values to the x values is assumed to be linear and of the form:

$$y = ax_1 + bx_2 + \ldots + \text{constant}$$

  - where the a,b... are the regression coefficients

- The values of the coefficients are found by the least-squares approach
  - Minimising the sum of the squared residuals

- The residual is the difference between the observed and predicted values for the ith observation:

$$e_i = Y_{(act)i} - Y_{(pred)i}$$

Multiple Linear Regression

- Single multiple-term linear equation is produced
- Equations maximise explanation of the correlation between dependent variable and independent variables
  - variance in independent variables (descriptors) ignored
  - regression coefficients calculated on basis of fit of *y* to *x* variables
- Method requires at least as many molecules as independent variables
  - To produce reliable results, typically need 5 times as many molecules as independent variables
  - Minimise the possibility of chance correlation
- Descriptor selection is much more critical than other methods such as PLS or GFA
  - Assumption is that the variables are independent
    - Not correlated

Multiple Linear Regression

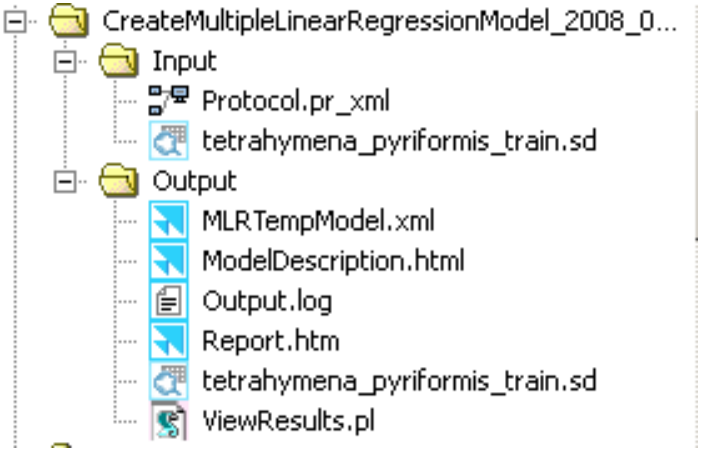
- **Advanced Options**
  - **OPS Analysis**
    - Performs a principal components analysis to determine the optimum prediction space (OPS) of the model in order to establish the model's applicability domain
    - Choosing the *modelname*_Applicability output when making predictions with the model on new data, the model output indicates whether each new sample is within or outside the training data range
  - **Fingerprint Tracking**
    - Tracks features of the fingerprint specified by *Model Domain Fingerprint* parameter and any other fingerprint properties used to build your model
    - Choosing the *modelname*_Applicability output when making predictions, you are warned when a sample contains any fingerprint feature not seen in the training data, or lacks any feature appearing in all of the training samples
  - **Save Training Properties**
    - **Through Pipeline Pilot, this allows** you to use New Model from Old to rebuild the model with new data added to the original data
  - **Encrypt Data**
    - If saving training properties, specifies that the data are to be encrypted to prevent direct access

Multiple Linear Regression

- Advantages
  - very quick
  - easy to interpret
- Disadvantages
  - does not work when the number of independent variables (molecular descriptors) is larger than (or even comparable to) the number of observations (molecules)
  - To produce reliable results, you typically need five times as many molecules as independent variables

Multiple Linear Regression - Results

- Ligands.sd
  - Output ligands with additional properties added by the model generation
    - MLRTempModel
    - MLRTempModel_Residual
- ViewResults.pl
  - Compares the predicted property with the modelled property



- ModelDescription.html
  - Contains important information about the model and training set
- MLRTempModel.xml
  - Contains the built model which can be shared with colleagues. This file name is set in the Model Name parameter in the protocol

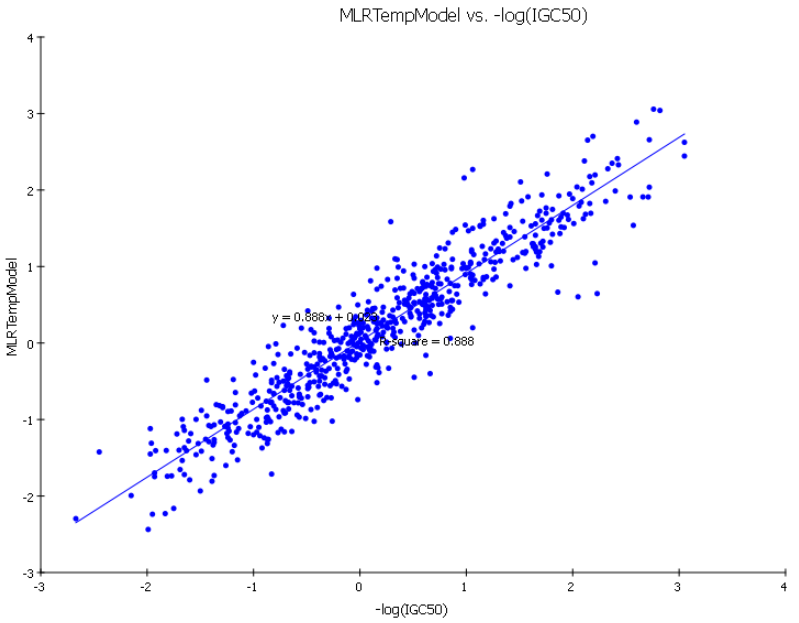
Multiple Linear Regression

- Training set of 644 compounds
- 8 descrtiptors

$R^2 = 0.747$

R^2 (prediction) = 0.738

Least-squared error 0.281704



- The created model is accessible for calculation under the Calculate Molecular Properties protocol

Measures of Regression Model Quality

- R-squared (R^2)
  - Commonly used (and misused)
  - Generally, larger values are preferred
    - But a value close to 1 often indicates **overfitting**
  - Available for any model
- Prediction R-squared (PRESS)
  - A better measure of predictive power than R-squared
  - Equivalent to Q-squared for leave-one-out cross-validation

Partial Least Squares

- Carries out regression using latent variables from the independent and dependent data that are along their axes of greatest variation and are most highly correlated
  - latent variables are maximally correlated with dependent variable (*y*)
  - contain linear combinations of correlated descriptors

- Typically applied when the independent variables are correlated or the number of independent variables exceeds the number of observations (rows)
  - under these conditions, it gives a more robust QSAR equation than multiple linear regression

Latent variables

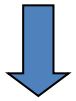
- First LV explains maximum variance in independent variable (x)
- Successive LV's explain successively smaller amounts of variance
- LV's conform to (1) and (2) with the provision that they are maximally correlated with the response
- Orthogonal to one another

$$LV_1 = b_{1,1}v_1 + b_{1,2}v_2 + \mathrm{K}\ b_{1,n}v_n$$

$$LV_2 = b_{2,1}v_1 + b_{2,2}v_2 + \mathrm{K}\ b_{2,n}v_n$$

$$\mathrm{M}$$

$$LV_q = b_{q,1}v_1 + b_{q,2}v_2 + \mathrm{K}\ b_{q,n}v_n$$



$$y = a_1 LV_{x1} + a_2 LV_{x2} + \ldots a_n LV_{xn}$$

Partial Least Squares

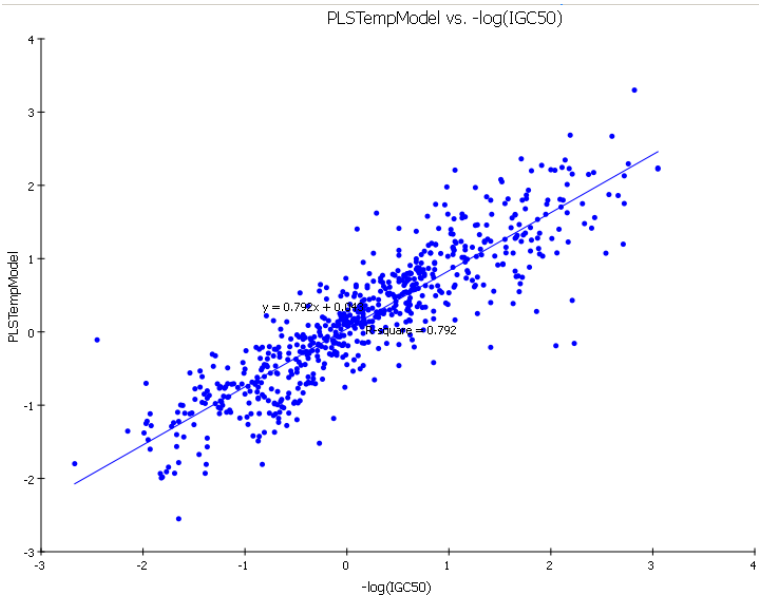
- Advanced Options
  - OPS Analysis
    - Performs a principal components analysis to determine the optimum prediction space (OPS) of the model in order to establish the model's applicability domain
    - Choosing the *modelname*_Applicability output when making predictions with the model on new data, the model output indicates whether each new sample is within or outside the training data range
  - Fignerprint Tracking
    - Tracks features of the fingerprint specified by *Model Domain Fingerprint* parameter and any other fingerprint properties used to build your model
    - Choosing the *modelname*_Applicability output when making predictions, you are warned when a sample contains any fingerprint feature not seen in the training data, or lacks any feature appearing in all of the training samples
  - Save Training Properites
    - Through Pipeline Pilot, this allows you to use New Model from Old to rebuild the model with new data added to the original data
  - Encrypt Data
    - If saving training properties, specifies that the data are to be encrypted to prevent direct access

Partial Least Squares

- 644 compounds
- 54 descriptors
- R^2= 0.792

  R^2 (adjusted) = 0.774

  Least-squared error = 0.231218

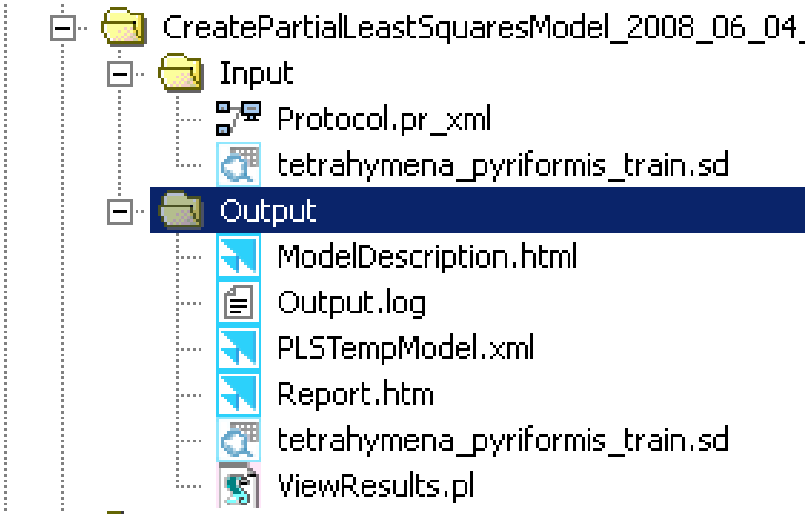


Create Partial Least Squares Model

Parameter Name	Parameter Value
Input Ligands	Molecule:All
Model Name	PLSTempModel
Dependent Property	
Independent Properties	This parameter is required.
Calculable Properties	ALogP,Molecular_Weight,Num_H_Donors,Num_H_Acceptors,Num_RotatableBonds,Num_Rings,Nu...
User Properties	
Advanced	
Learn Options	Perform OPS Analysis,Track Fingerprint Features
Model Domain Fingerprint	FCFP_2
Minimum Samples per Variable	SqrtEstimate
Number of Components	20

Partial Least Squares

- Ligands.sd
  - Output ligands with additional properties added by the model generation
    - PLSTempModel
    - PLSTempModel_Residual
- ViewResults.pl
  - Compares the predicted property with the modelled property
- ModelDescription.html
  - Contains important information about the model and training set
- PLSTempModel.xml
  - Contains the built model which can be shared with colleagues. This file name is set in the *Model Name* parameter



QSAR Techniques & Applications in Discovery Studio

Modern Regression Techniques

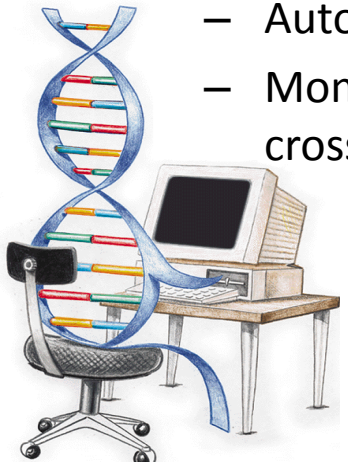
accelrys®

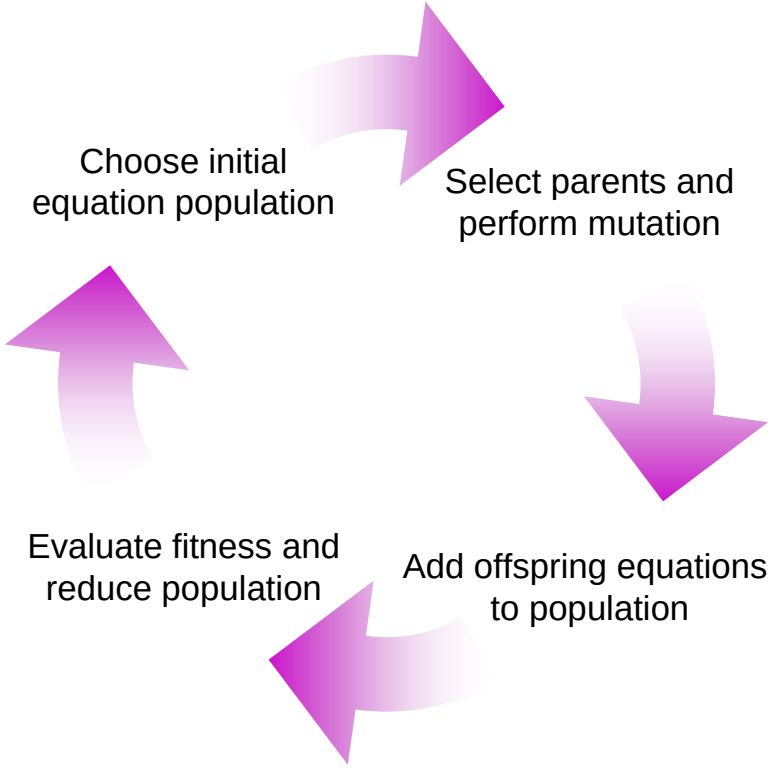
Modelling more complex data

- Analyse structure-activity relationships with:
  - Activity levels (1, 2 ...) or classes (A, B …)
  - Non-linearities
  - Thresholds effects
  - Variable interactions
  - Compounds acting by multiple mechanisms
  - Many thousands of compounds
- More modern regression methods are required:
  - GFA
  - Bayesian Model
  - Neural Network

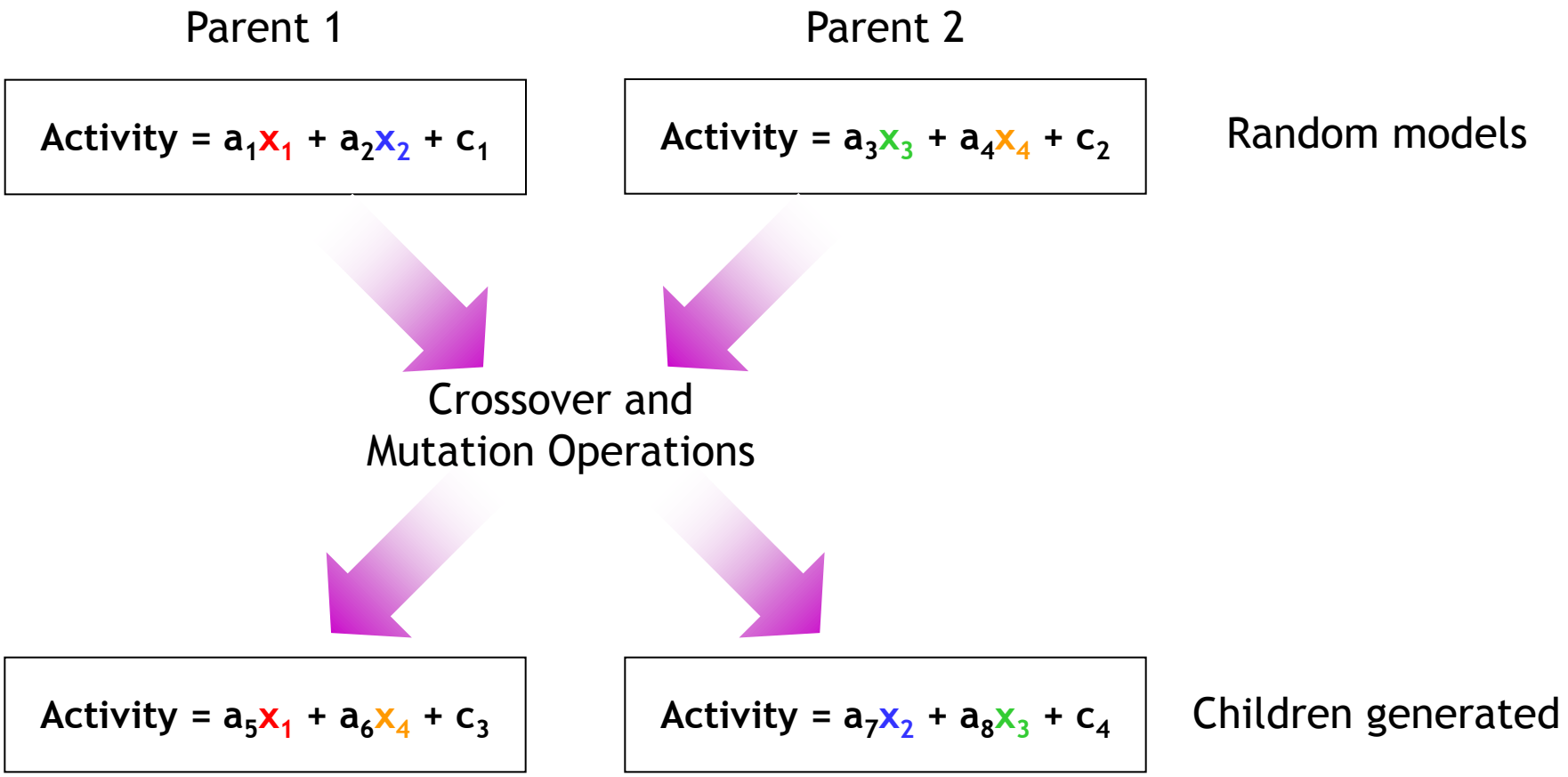
Genetic Function Approximation

- ‘Computer Simulation of Darwin's Theory’
- Advantages
  - Evolutionary algorithm (derives better models)
  - Multiple models provide different insights into system
  - Mechanisms to prevent over-fitting of data (LOF measure)
  - Linear and higher order polynomial terms available
  - Automatic outlier removal (splines)
  - Monitor descriptor usage by crossover plot





Equation Evolution



Genetic Function Approximation

- Independent properties (descriptors) need to be pre-calculated
  - Calculate Molecular Properties protocol (QSAR)

- Model names need to be unique
  - Identically named models overwritten
- Models generated
  - 10 (default)
  - Ensemble model
  - Transferable
    - GFATempModel.xml

Create Genetic Function Approximation Model

Parameter Name	Parameter Value
Input Ligands	Molecule:All
Model Name	GFATempModel
Dependent Property	
Independent Properties	
Calculable Properties	ALogP,Molecular_Weight,Num_H_Donors,Num_H_Acceptors,Num_RotatableBonds,Num_Rings,Nu...
User Properties	
Number of Models	10
Model Options	
Model Form	Linear
Additional Terms	
Initial Equation Length	5
Maximum Equation Length	10
Constant Equation Length	False
GFA Options	
Population Size	100
Maximum Generations	5000
Scoring Method	Single Score
Score Function	Friedman Lack-of-Fit
Additional Options	
Random Seed	12345
Learn Options	Perform OPS Analysis,Track Fingerprint Features
Minimum Samples Per Variable	SqrtEstimate
Maximum Correlation	0.99
GFA Progress Interval	10
Number of Variables in GFA Progress	20

Filter Algorithm

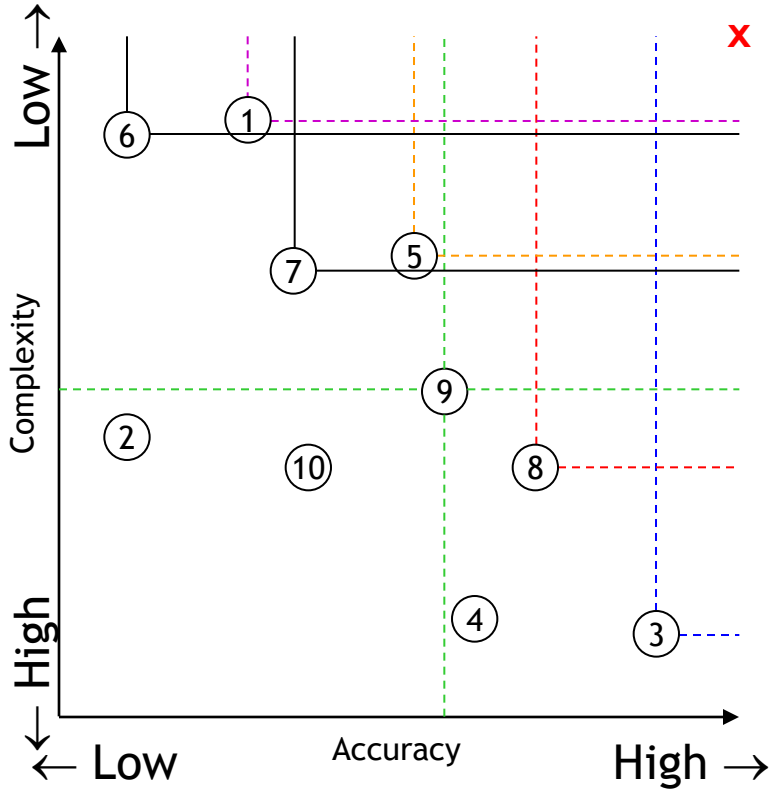
- Used to determine which equations enter the next generation with a fitness evaluation
  - Adding an extra term to an equation may increase the accuracy of its predictions, at the expense of making the equation more complex and increasing the risk of over-fitting
- Three algorithms available
  1. Original
     - Uses a single score which provides a trade off between accuracy and complexity
     - Equations with high scores are carried forward to the next generation
  - Pareto methods
     - Treat the maximization of accuracy and the minimization of complexity as an exercise in multiple objective optimization
       - NSGA-II (Non-dominated sorting genetic algorithm)

Pareto Method Filter Algorithms

- Pareto optimization
  - Each equation is scored on two criteria, which will be optimized
    - Accuracy (to be maximized)
    - Complexity (to be minimized)
  - Attempts to find the pareto-optimal models, that is, those with
    - The highest possible accuracy for a given complexity
    - The lowest complexity for a given accuracy
- Advantages of pareto-method over single score fitness measure
  - Final population will contain a range of pareto-optimal models
    - Simple models, which are less accurate
    - More complex models with greater accuracy
  - As population evolves, it will contain models of varying complexity and this tends to encourage the population to maintain greater genetic diversity than is the case for models that have all made the same trade-off between complexity and accuracy

Pareto Optimization (I)

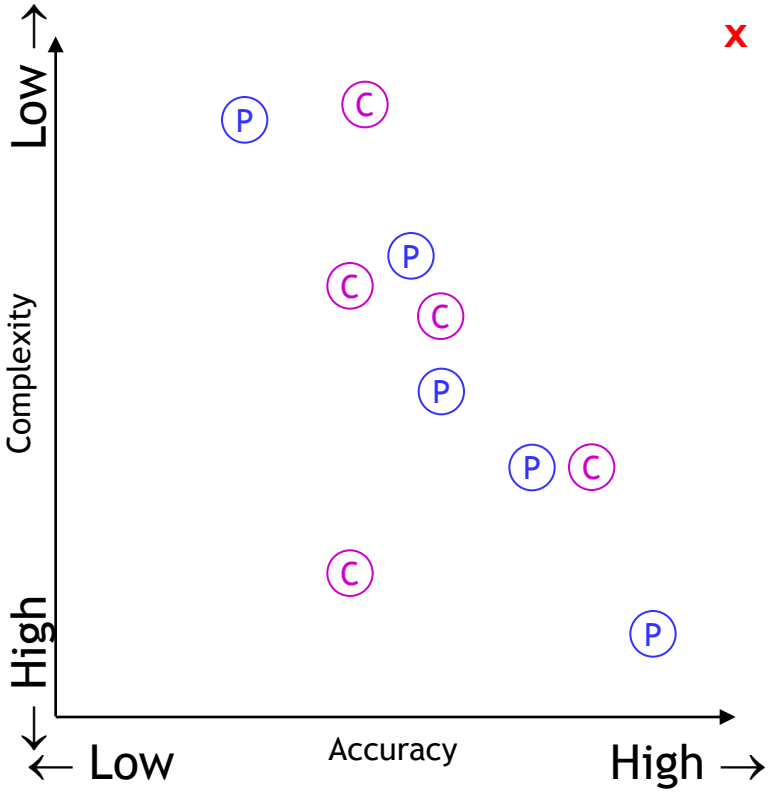
1. Determine the pareto front and each generation's survivors



- Equation 6 is not on the 1st front as equation 1 pareto dominates it
- Equation 1 is on the 1st front as
  - At least one of Eq. 1's properties (Accuracy) is better than the same property of Eq. 6
  - All other of Eq. 1's properties are no worse than the corresponding properties of Eq. 6
- Equation 9 is pareto dominant over equations 2 and 10
- All equations on the 1st pareto front are equal to each other
  - It is not possible to improve one property without degrading the other

Pareto Optimization (II)

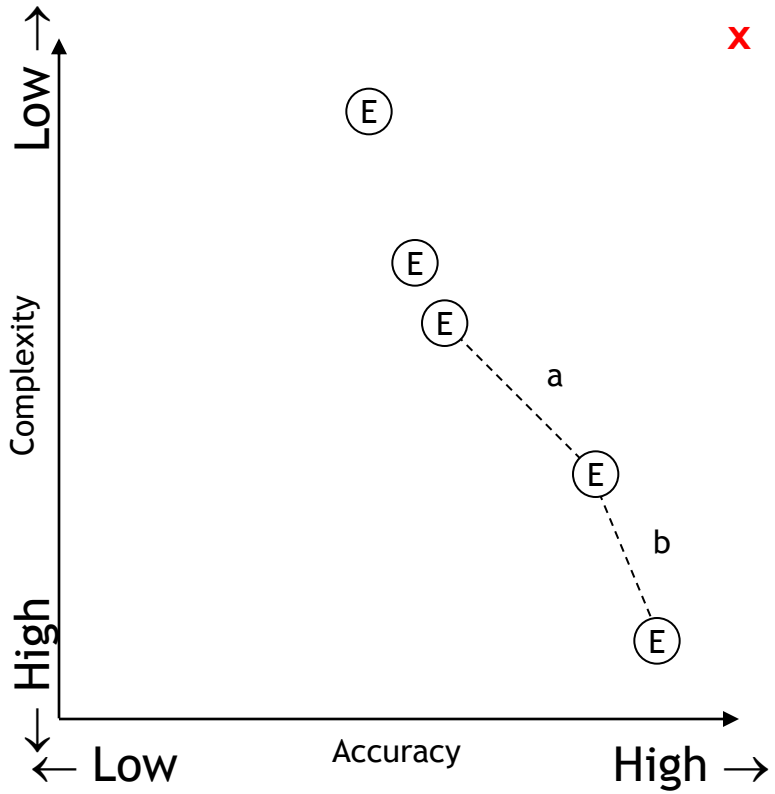
2. Increase population and determine survivors again



- Genetic evolutionary process
  - Select parents and perform crossover and mutation
  - Add offspring equations to population
  - Evaluate fitness and reduce population
- Repeat for maximum number of generations

Pareto Optimization (III)

3. Rank the equations



- Crowding Distance is measured
  - Measure of how near a point is to other points on the same front
  - Samples at the extremes of the front are assigned an "infinite" crowding distance (1.0e99)
  - Preference for larger crowding distances encourages a spread of points along the entire front

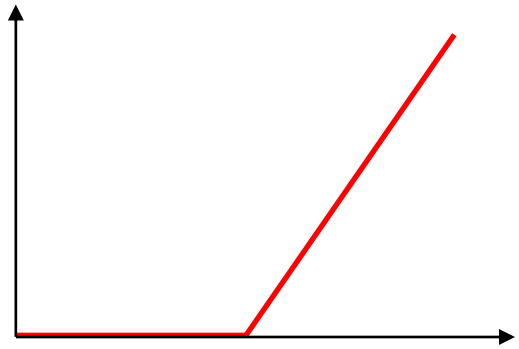
$$\text{Crowding Distance} = \frac{a+b}{2}$$

Pareto Method Filter Algorithms

- NSGA-II - Non-dominated sorting genetic algorithm
  - Determines which pareto front each model lies on
    - Pareto-optimal equations all lie on the first front
    - Second front consists of all equations that are dominated only by models that lie on the first front
  - Selection algorithm chooses equations from the fronts with the lowest indices

Equation and Additional Terms

- Linear - x
- Quadratic - x^2
- Offset quadratic - $(x-a)^2$
- Spline - <x-a>
- Quadratic spline - $<x-a>^2$



- Binary Interaction
  - A, B, C will create A x B, A x C and B x C as new terms
- Simple Quadratic
  - A, B, C will create A2, B2 and C2 as new terms
- Full Quadratic
  - A, B, C will create A^2, B^2, C^2, A x B, A x C and B x C as new terms
- Simple Cubic
  - A, B, C will create A^2, B^2, C^2, A^3, B^3 and C^3 as new terms
- Full Cubic
  - A, B, C will create A^3, B^3, C^3, A^2 x B, A^2 x C, A x B^2, A x C^2, B^2 x C, B x ^{C2}, A^2, B^2, C^2, A x B, A x C and B x C as new terms

Note: Take care when using additional terms, especially the Binary Interaction, Full Quadratic, and Full Cubic options. These can generate very large numbers of variables for analysis, requiring more memory and longer calculation times.

Additional Options

- Advanced Options for the Learner
  - OPS Analysis
    - Performs a principal components analysis to determine the optimum prediction space (OPS) of the model in order to establish the model's applicability domain
    - Choosing the *modelname*_Applicability output when making predictions with the model on new data, the model output indicates whether each new sample is within or outside the training data range
  - Fingerprint Tracking
    - Tracks features of the fingerprint specified by *Model Domain Fingerprint* parameter and any other fingerprint properties used to build your model
    - Choosing the *modelname*_Applicability output when making predictions, you are warned when a sample contains any fingerprint feature not seen in the training data, or lacks any feature appearing in all of the training samples
  - Save Training Properites
    - Through Pipeline Pilot, this allows you to use New Model from Old to rebuild the model with new data added to the original data
  - Encrypt Data
    - If saving training properties, specifies that the data are to be encrypted to prevent direct access

GFA Output (I)

Model Number	Equation
1	GFATempModel = −4.28186 − 0.0323964 * ES_Count_sCH3 * ES_Count_sCH3 + 0.068467 * ALogP * CHI_V_0 + 0.0913932 * ES_Sum_ddC * Kappa_2_AM − 1.70323 * ES_Count_ddC * ES_Count_dssC + 0.182708 * Num_AromaticRings * Num_H_Acceptors + 0.401191 * IAC_Mean * JX + 0.0732996 * <56.1354 − IAC_Total> + 0.0001044 * <Molecular_SASA − 280.444> ^2 − 0.00018961 * <Molecular_SAVol − 282.272> ^2 − 0.0500136 * <6.36624 − CHI_V_1> ^2
2	GFATempModel = −1.87231 + 2.23149 * Molecular_FractionalPolarSurfaceArea * Molecular_FractionalPolarSurfaceArea + 0.415071 * IAC_Mean * IAC + 0.0726437 * ES_Sum_ddC * − 1.82352 * ES_Count_ddC * + 0.102667 * Num_AromaticRi − 0.5002 * <5.0532 − ALogP> + 0.0808402 * <62.3231 − IAC + 8.83315e−005 * <Molecular_ − 0.000127346 * <Molecular_S − 0.0414198 * <9.57621 − CH
3	GFATempModel = −1.20384 + 2.28443 * ES_Sum_ddC * ES + 0.412948 * IAC_Mean * IAC − 4.97641 * ES_Sum_ddC * ES + 0.0824675 * Num_AromaticR − 0.493331 * <5.0532 − ALogl − 1.89609 * <0.598478 − Mole + 0.0932955 * <62.3231 − IAC + 0.000104166 * <Molecular_S − 0.000136547 * <Molecular_S

Summary Statistics

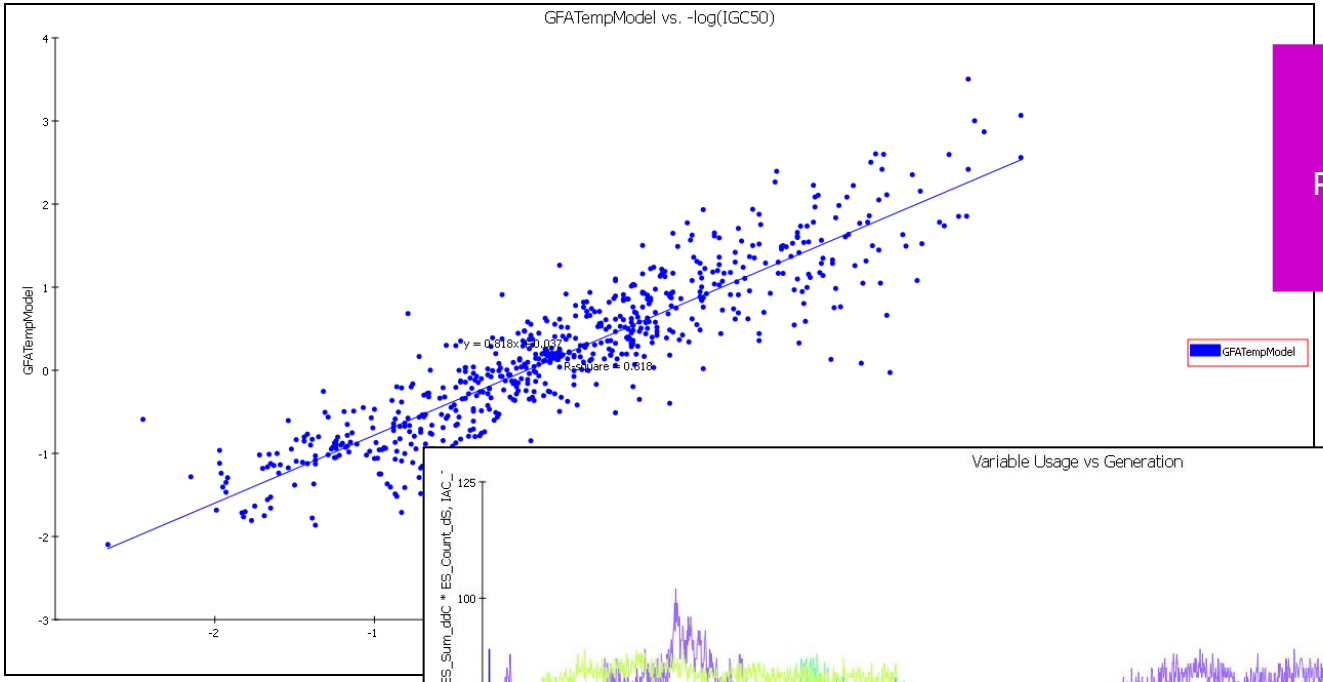
The following table contains summary statistics for the top 10 models b
Friedman L.O.F. is the Friedman lack-of-fit score; **S.O.R. p-value** is the

Model						
GFA_TP_1 = -3.6298 + 0.63307 * ALogP − 0.49144 * ES_Sum_ddsN + 0.32308 * Count<ECFP_6:1334400011> − 0.46122 * Count<ECFP_6:-659271057> + 1.7586 * Molecular_FractionalPolarSurfaceArea + 0.13975 * CHI_3_P + 0.70372 * CHI_V_1 − 0.45502 * CHI_V_3_P + 1.2112 * IAC_Mean − 0.060474 * IAC_Total	0.7887	0.7854	0.7780	0.4893	0.9516	3.365e-206
GFA_TP_2 = -3.4455 + 0.60164 * ALogP − 0.22265 * ES_Sum_ddsN + 0.34053 * Count<ECFP_6:1334400011> + 1.6592 * Molecular_FractionalPolarSurfaceArea + 0.14406 * CHI_3_P + 0.76493 * CHI_V_1 − 0.47571 * CHI_V_3_P + 1.1055 * IAC_Mean − 0.066492 * IAC_Total	0.7845	0.7815	0.7740	0.4938	0.8006	1.039e-204
GFA_TP_3 = -3.6402 + 0.63601 * ALogP + 0.2931 * Count<ECFP_6:1334400011> + 1.9288 * Molecular_FractionalPolarSurfaceArea + 0.15291 * CHI_3_P + 0.726 * CHI_V_1 − 0.47192 * CHI_V_3_P + 1.1813 * IAC_Mean − 0.063287 * IAC_Total	0.7824	0.7796	0.7726	0.4959	0.6783	1.353e-204
GFA_TP_4 = -3.5829 + 0.6465 * ALogP + 2.1061 * Molecular_FractionalPolarSurfaceArea + 0.17701 * CHI_3_P + 0.70358 * CHI_V_1 − 0.46344 * CHI_V_3_P + 1.1276 * IAC_Mean − 0.063922 * IAC_Total	0.7779	0.7755	0.7686	0.5006	0.5891	4.717e-203
GFA_TP_5 = -3.3999 + 0.63633 * ALogP + 1.843 * Molecular_FractionalPolarSurfaceArea + 0.1587 * CHI_1 + 0.32718 * CHI_V_1 + 1.0787 * IAC_Mean − 0.052118 * IAC_Total	0.7670	0.7648	0.7597	0.5123	0.5323	9.889e-198
GFA_TP_6 = -3.5028 + 0.69772 * ALogP + 2.1009 * Molecular_FractionalPolarSurfaceArea + 0.36309 * CHI_V_1 + 1.1307 * IAC_Mean − 0.035634 * IAC_Total	0.7589	0.7571	0.7520	0.5207	0.4791	2.233e-194
GFA_TP_7 = -3.754 + 0.86232 * ALogP − 0.39267 * Count<ECFP_6:-659271057> + 2.1458 * Molecular_FractionalPolarSurfaceArea + 1.2845 * IAC_Mean	0.7475	0.7459	0.7423	0.5325	0.4408	2.607e-189
GFA_TP_8 = -3.3445 + 0.82257 * ALogP + 1.7163 * Molecular_FractionalPolarSurfaceArea + 1.0902 * IAC_Mean	0.7412	0.7400	0.7370	0.5386	0.3998	2.435e-187
GFA_TP_9 = -3.3517 + 0.72686 * ALogP + 1.4982 * IAC_Mean	0.7233	0.7224	0.7201	0.5566	0.3811	1.519e-179
GFA_TP_10 = -1.1644 + 0.69703 * ALogP	0.5856	0.5849	0.5829	0.6806	0.5119	6.598e-125

GFATempModel.xml
Generated model which can be shared with colleagues

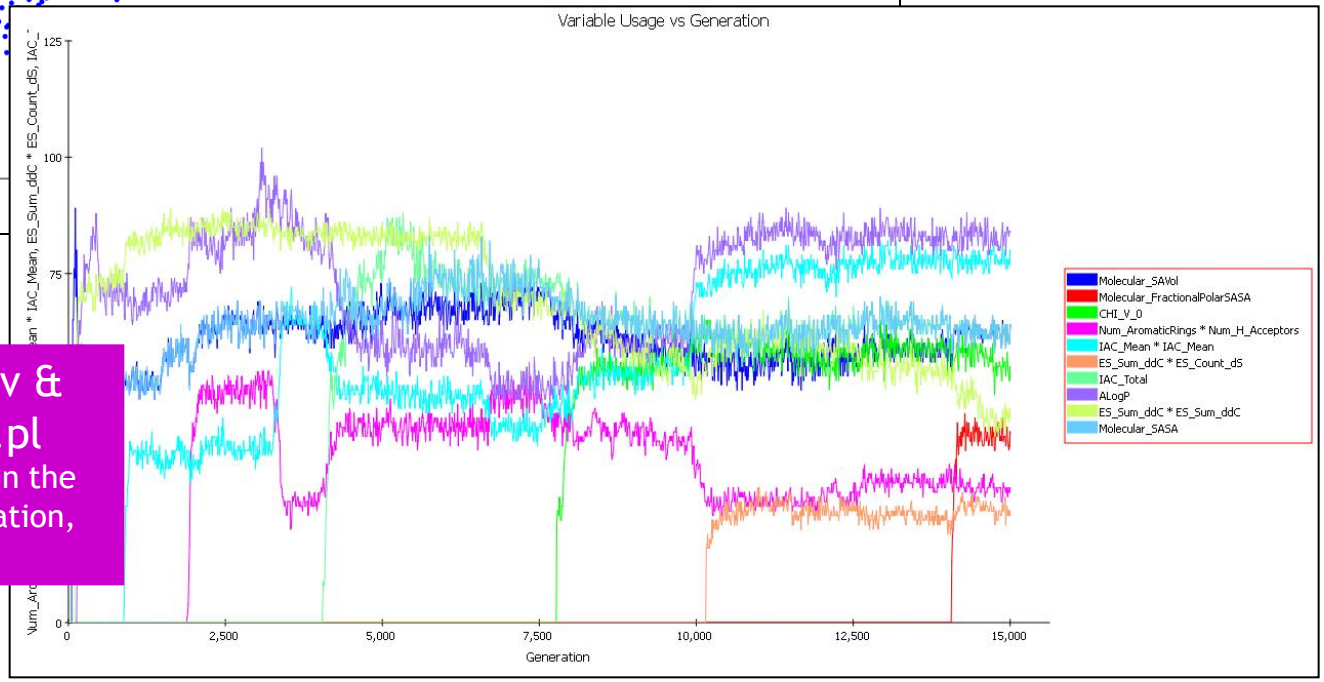
ModelDescription.html
Model and training set information, including model and input summaries, QSAR model equations, and model validation

GFA Output (II)



Input_Ligands.sd & ViewResults.pl

Prediction and estimated error for 1st model, plotted against actual activity



GFAVariableCounts.csv & ViewVariableCounts.pl

Occurrence of variables within the population against the generation, plotted

Statistical Output (I)

- Friedman lack-of-fit (LOF)
  - Automatically penalises models with too many features to prevent overfitting
  - Based on Least Squares Error (sum of squares of errors)
  - Reflects the equation size as well as the number of samples in the training set
  - The lower the LOF, the less likely it is that the model is overfitting the data

$$LOF = \frac{SSE}{\left(1 - \frac{c + df}{n}\right)^2}$$

SSE sum of squares of errors

c number of functions in equation (excluding constant term)

d smoothing parameter

f total number of features in all the functions (some equations may contain more than one feature, eg. X1X2 contains two features)

n total number of input ligands

$$SSE = \sum \left(y_{obs} - y_{pred}\right)^2$$

Statistical Output (II)

- R-squared (R^2)
  - Coefficient of determination
  - Square of correlation coefficient
  - The closer the value is to 1.0, the better the model explains the Y variable
  - The size of the model and the number of terms in the model are not accounted for
  - Alone does not indicate whether over-fitting has occurred
- Adjusted R-squared (Adj. R^2)
  - Modification of R^2 that adjusts for the number of explanatory terms in a model
  - Increases only if a new term improves the model more than would be expected by chance
  - Can be negative, and will always be less than or equal to R^2

$$R^2 = \frac{\sum (y_{pred} - y_{mean})^2}{\sum (y_{obs} - y_{mean})^2} = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

SSR sum of squares of regression

SST total sum of squares

SSE sum of squares of errors (residuals)

$$\text{Adj.}\, R^2 = 1 - \frac{n-1}{n-p}(1 - R^2) = 1 - \frac{\frac{SSE}{(n-p)}}{\frac{SST}{(n-1)}}$$

SSE sum of squares of errors (residuals)

SST total sum of squares

n total number of input ligands

p number of terms in model

Statistical Output (III)

- Cross validated R-squared (R^2)
  - Key measure of the predictive power of a model
    - The closer the value is to 1.0, the better the predictive power
  - For a good model, R^2(CV) should be fairly close to R^2
  - If R^2(CV) is much less than R^2, the model equation is probably over-fitting the data
- PRESS (Predicted sum of squares)
  - Sum over all ligands of the squared differences between the actual and predicted values for the independent variables:
  - The lower the value, the more reliable the equation

$$XV\,R^2 = 1 - \frac{PRESS}{SST}$$

PRESS predictive sum of squares

SST total sum of squares

$$SST = \sum (y_{obs} - y_{mean})^2$$

$$PRESS = \sum (y_{obs} - y_{pred})^2$$

Statistical Output (IV)

- Critical SOR F-value 95% (F_{cr})
  - Critical significance-of-regression (SOR) F value for probability 0.05 (at 95% confidence level)
  - Tabulated values for different values of n and p
- Significant regression (F-value)
  - Performed to assess whether or not the regression is statistically significant
  - Is the variance in the data which is explained by the regression much larger than the variance remaining due to errors?
  - Yes if $F > F_{cr}$, No if $F < F_{cr}$
  - The larger the value of F, the better the model

$$\text{F value} = \frac{\frac{SSR}{(p-1)}}{\frac{SSE}{(n-p)}}$$

SSR sum of squares of regression

SSE sum of squares of errors (residuals)

n total number of input ligands

p number of terms in model

$$SSR = \sum \left(y_{pred} - y_{mean} \right)^2$$

$$SSE = \sum \left(y_{obs} - y_{pred} \right)^2$$

Analysis of Models (I)

- Perform a critical evaluation of statistical models
  - Fit (R^2)
  - Predictivity (xv-R^2)
  - Analysis of residuals
  - Chemical significance
    - Interpretability
    - Applicability to design

- Analysis of residuals
  - Get insight into unexplained variance
    - High/low value(s) for 1-2 compounds that break down the analysis ('outliers')
    - Non-gaussian distribution with unexplained trend
    - Magnitude vs. experimental error

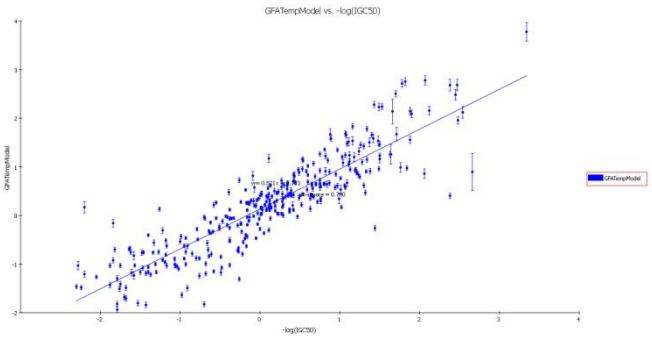
Analysis of Models (II)

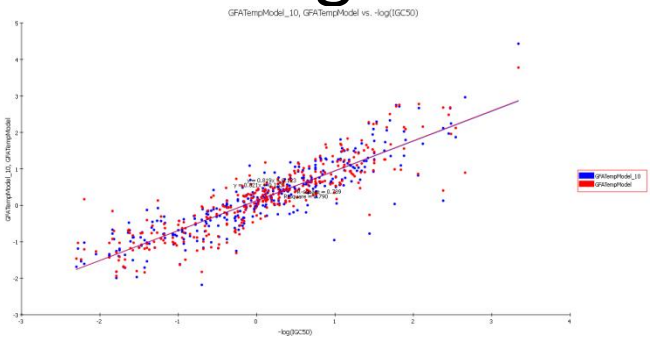
- Optimization of QSAR models
  - Look for similar models with additional benefits
    - Simpler model with less parameters
    - Involving specific descriptors
    - Substitute descriptors for others with added
      - Ease of computation
      - Chemical significance
      - Value for design of new ligands

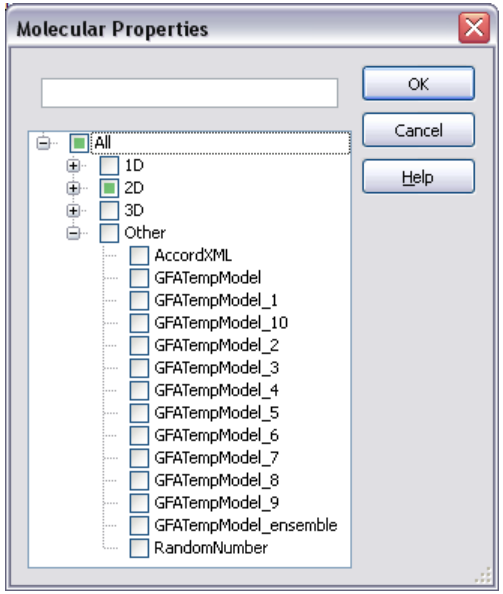
- Define criteria for terminating QSAR optimization process
  - Possible criteria
    - Good correlation and predictivity
    - Chemical significance and usefulness
    - No remaining trend to explain
    - Residuals compatible with experimental error

Prediction of New Ligands

- Calculate Molecular Properties protocol
  - No need to pre-calculate independent properties, unless they include
    - 3rd party or external properties (eg. LigScore2, FitValue etc.)
    - Semiempirical QM properties
    - Density Functional QM properties
  - Prediction
  - Estimated Error
- Plotting



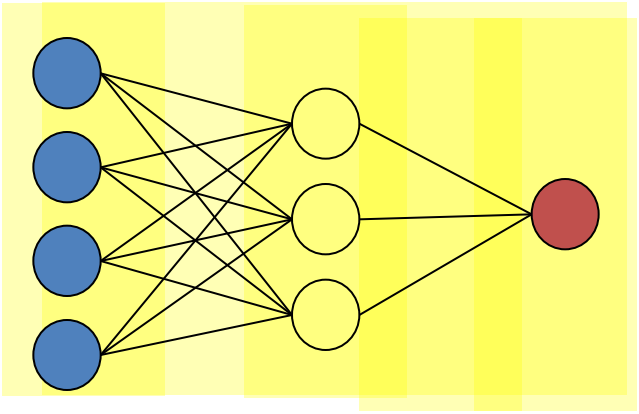




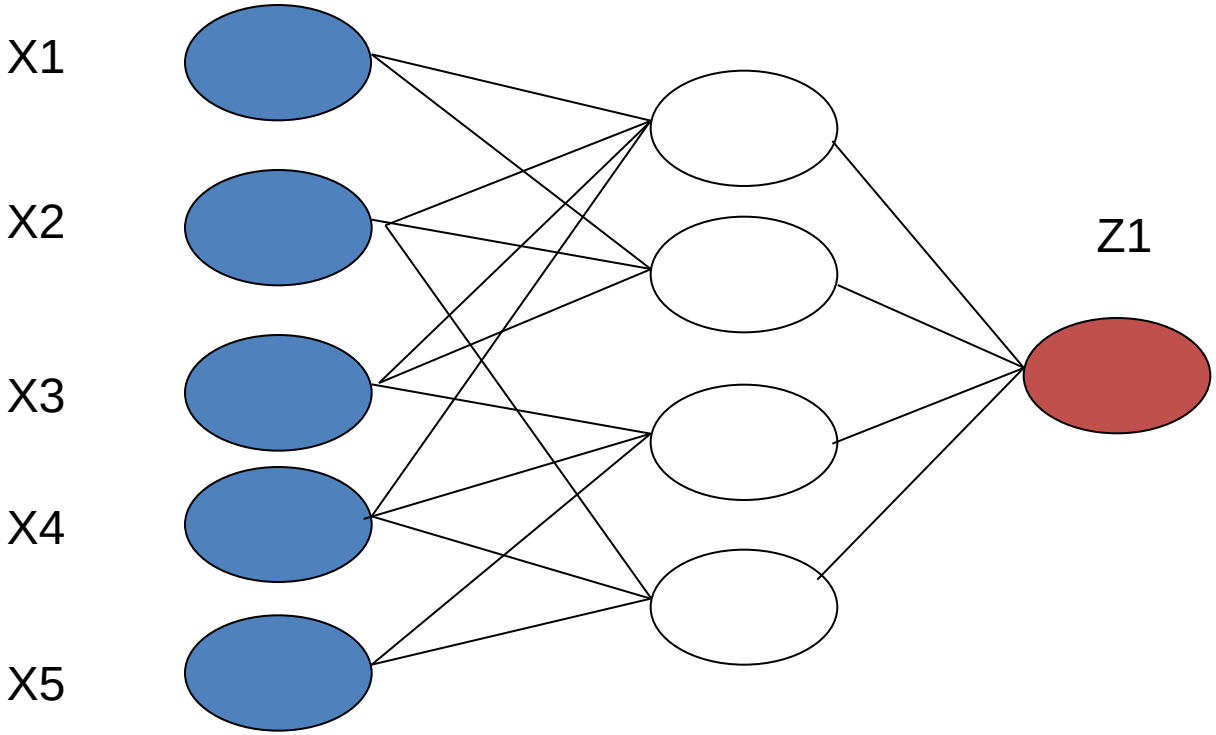
Back Propagation Neural Network

- Sophisticated model-building technique
- Capable of modelling data represented by non-linear functions
- Neural networks are inspired by the way the human brain works
  - The brain consists of billions of neurons, which are linked together into a complex network
  - A neuron communicates with another by sending an electrical signal along an axon, which is a long nerve fiber that connects to the second neuron at a synapse
  - Each neuron acts as an information processing element because the electrical signals sent out by one neuron depend on the strength of the incoming signals at its synapses

  - Neurons -> Nodes
  - Synapses -> Connections



Back Propagation Neural Network Architecture



Input Layer
used to introduce the input (**predictor**) variables to the network

Hidden layer

Output layer
nodes in this layer represent the predictions made by the network (**response**)

Back propagation Neural Network

- **The structure of a neural network**
  - Nodes:
    - An artificial neural network consists of an interconnected network of processing elements
  - Assumption
    - nodes are arranged in layers and that the input connections to each node come only from nodes in the layer directly below it

- **Overall steps**
  - Multiply Predictor by connection weight
  - Apply nonlinear transformation
  - Multiply by connection weight
  - Compares Response to Output
  - Adjust and repeat

Nodes and Training

- Each node (other than those in the input layer), takes as its input a transformed linear combination of the outputs from the nodes in the layer below it

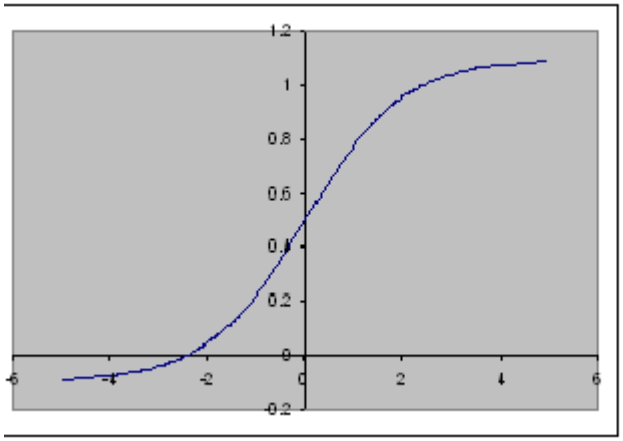
$$I_i = \sum_j w_{ij} X_j + \theta_i$$

where I_i is the input to the i^{th} node, X_j is the output of the j^{th} node in the previous layer, j is summed over all of the nodes in the previous layer, w_{ij} is the connection weight between the nodes, and θ_i is a parameter known as the bias

- This input is then passed through a transfer function to calculate the output of the node

Nodes and Training

- Transfer function is an s-shaped sigmoid function



- Transfer function: $y(x) = 1.2/(1 + e^{-x}) - 0.1$
- smooth and easily differentiable features that help the algorithm that is used to train the network
    - The sigmoid function can take inputs in any range, but produces an output within a narrow range
        - Scaling of input data is important

Nodes and Training

- Training
  - connection weights and biases are set so as to minimize the prediction error for the network
    - For a particular set of weights and biases, each of the training cases are introduced to the network and an error function is used to determine how well the calculated outputs match the expected output values
- Iterative method
  - Each training cycle the following steps are performed
    - the value and gradient of the error function are calculated, and the weights and biases are adjusted according to the algorithm being used
    - Can use either the BFGS (Broyden-Fletcher-Goldfarb-Shanno) method or a steepest descent algorithm as the Training Minimization Method to find the minimum
    - algorithm may find a local minimum of the error function rather than the global minimum
      - You can vary the random seed number in the Advanced section of the protocol to try and find a lower minimum

Parameters & Settings

Create Back Propagation Neural Network Model

Parameter Name	Parameter Value
Input Ligands	Molecule:All
Model Name	BNNTempModel
Dependent Properties	
⊞ Independent Properties	
⊟ Basic	
Nodes in Hidden Layers	*
Maximum Number of Cycles	4000
Maximum Cycles Without Improve...	300
Additional Terms	None
Data Standardization	Data Range
⊟ Advanced	
Cost Weightings for Outputs	
Training Minimization Method	BFGS
Random Seed	9999
Tolerance	0.000001
⊟ Missing Data	
Allow Missing Data	True
Randomization Scale Factor	0.5
Missing Value Penalty Factor	0.02
⊟ Weight Limits	
RMS Weight Soft Minimum	0.0001
RMS Weight Hard Minimum	0.00001
RMS Weight Soft Maximum	100
RMS Weight Hard Maximum	100000
Initial Penalty Fraction	0.3
Minimum Initial Weight	-0.5
Maximum Initial Weight	0.5
⊟ Validation	
Use Test Set	True
Test Set Fraction	0.1
Cross Validation Method	Omit Groups of Rows
Cross Validation Groups	3
Existing Predictor Component	

→ Single hidden layer is used by default
Number of nodes determined automatically

Parameters affecting the training cycles

→ Controls treatment of missing data

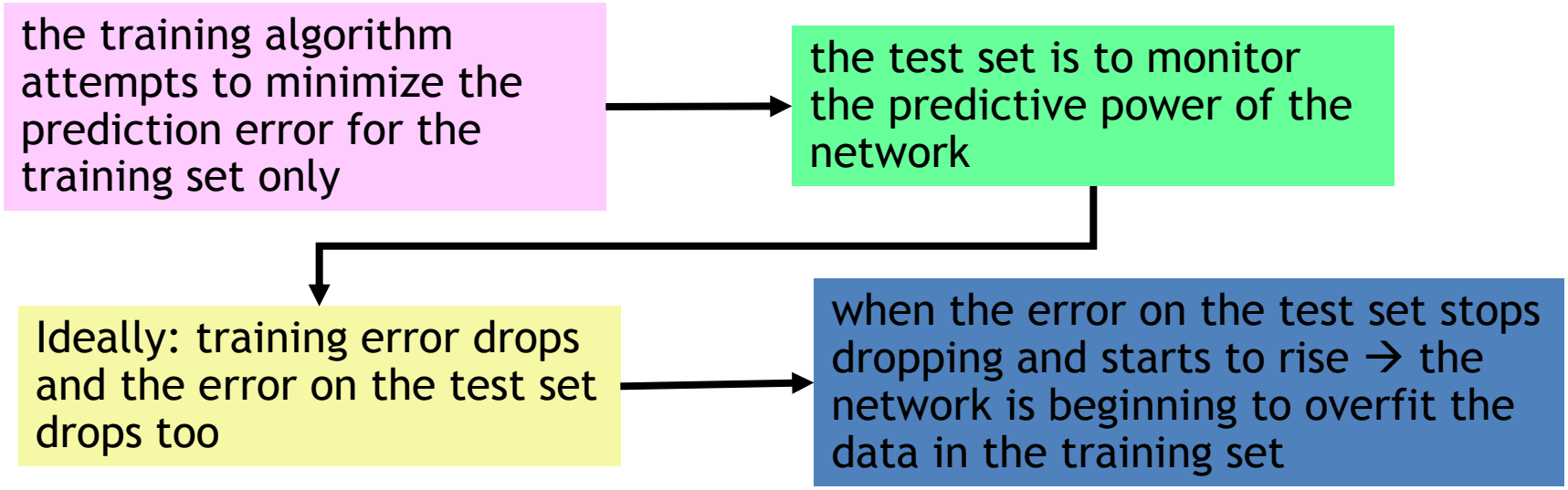
Control what happens if the network weights either all tend towards zero or all gradually increase beyond what was expected during the training process

Parameters & Settings

- Connection weight and node bias
  - parameter that can be adjusted during network training
  - Each connection and node corresponds to one degree of freedom of the model
  - Recommended to have at least twice as many observations as there are degrees of freedom
  - too many nodes in the hidden layer(s) leads to overfitting
  - too few nodes leads to weak model
- Missing Data
  - increases the number of degrees of freedom of the neural network model
  - This effect is not considered when QSAR automatically determines the optimal number of nodes to use
    - If handling a substantial number of missing data points use slightly fewer hidden layer nodes

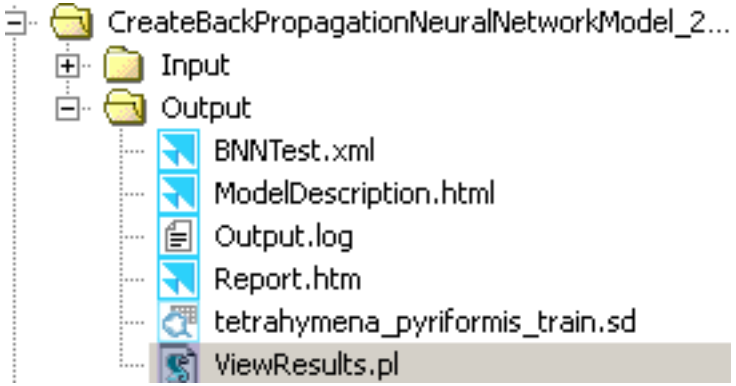
Overfitting

- The use of large number of connection weights can lead to overfitting
- To avoid this the data set is split into a training set and a test set
- Training algorithm attempts to minimize the prediction error for the training set only and the test set is to monitor the predictive power of the network
- Overfitting indicated by rising error on the test set

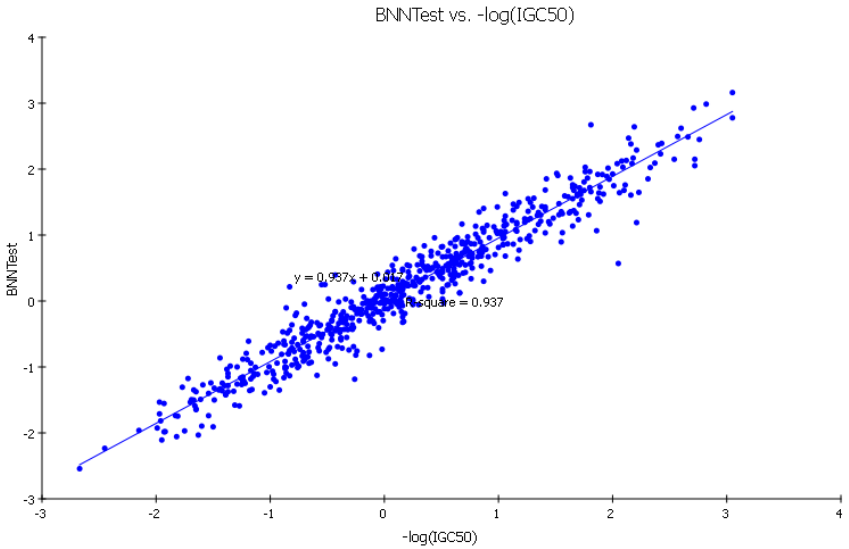


Results

- Ligands.sd
  - Output ligands with additional properties added by the model generation
    - BNNTempModel
- ViewResults.pl
  - Compares the predicted property with the modelled property



- ModelDescription.html
  - Contains important information about the model and training set
- BNNTempModel.xml
  - Contains the built model which can be shared with colleagues. This file name is set in the *Model Name* parameter



Neural Networks

- Advantages
  - Advanced model building tool
  - Can build a single model to predict multiple properties
  - Allows for incomplete data sets by using weighted “random” numbers
  - Allows building of non-linear models
  - Most resistant method to noisy data sets
- Disadvantage
  - More “black box”, difficult to interpret models

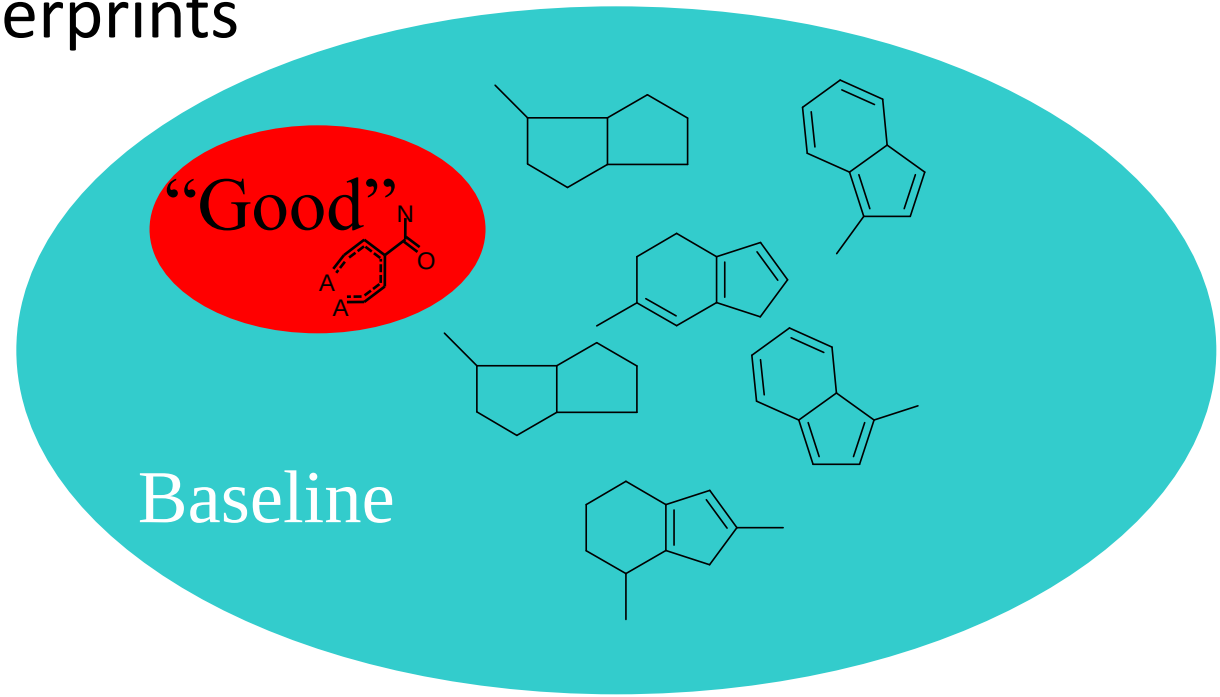
Working with Categorical Data

- How can we learn good from bad
- examine what distinguishes “good” from “bad” data records
  - Malignant versus benign tumors
  - Spam versus non-spam emails
  - One species of iris from others
- “Good” and “bad” are arbitrary labels for two categories of data

- Only more modern methods can handle categorical data
  - Bayesian Model
  - Recursive Partitioning

Working with Categorical Data

- Learn Good Molecules
- examines what distinguishes “good” from “baseline” compounds
  - Molecular properties (molecular weight, AlogP, etc.)
  - Molecular fingerprints



Protocol: Create Bayesian Model

- Uses Pipeline Pilot's proprietary Bayesian Modeling methodology
- provides unsupervised learning for large data collections
- ideal way to rapidly prioritize compound acquisitions or high throughput screening efforts
- A Bayesian model can capture a simple two-class relationship (such as active or inactive) and can learn multiple classes or multiple end-points in a single model
- Uses Bayesian categorization to build a model that distinguishes between "good" and "bad" ligands

Bayesian Categorisation

- Utilizes probabilities to classify objects into one or some categories
- Solves classification problems of learning good from bad
- Chemically aware
  - Handles fingerprints
- Applicable to HTS data
  - Large volume, few hits, many classes

Bayesian Model

- Based on Bayes' Theorem:

Guard against over fitting (Complex model has a smaller probability)

$$p(h \mid d) = \frac{P(d \mid h)P(h)}{P(d)}$$

where
- *h* is the hypothesis or model
- *d* is the observed data
- *p(h)* is the prior belief (probability of hypothesis *h* before observing any data)
- *p(d)* is the data evidence (marginal probability of the data)
- *p(d|h)* is the likelihood (probability of data *d* if hypothesis *h* is true)
- *p(h|d)* is the posterior probability (probability of hypothesis *h* being true given the observed data *d*)

Bayesian Model

- Considers the likelihood of a model but also takes into consideration the complexity of the model
  - the most simple model is preferred
  - prevents overfitting
- learn-by-example paradigm
  - the user marks the sample data that is of interest (good), and then the system learns to distinguish them from background data
  - No tuning parameters are required beyond the selection of the input descriptors from which to learn
- Advantages of Bayesian categorization
  - Can handle large amounts of data
  - learns fast
  - tolerant of random noise
  - Not prone to overfitting

Math Details: Bayes's Theorem

$$P(A|B) = P(B|A)\,P(A) / P(B)$$

where

$P(A|B)$ = probability A is true given that B is true

$P(A)$ = probability of A in absence of other info ("prior")

E.g.,

A = "This molecule is active."

B = "This molecule contains –OH"

For model-building, generalize to:

$$P(A|B_1,B_2,\ldots,B_n) = P(B_1,B_2,\ldots,B_n|A)\,P(A) / P(B_1,B_2,\ldots,B_n)$$

Learning process

1. Generates a large set of Boolean features from the input descriptors
2. Collects the frequency of occurrence of each feature in the good subset and in all data samples
3. Applies the model to a particular sample using the following steps:
   - the features of the sample are generated
   - weight is calculated for each feature using a Laplacian-adjusted probability estimate
   - The weights are summed to provide a probability estimate, which is a relative predictor of the likelihood of that sample being from the good subset
   - Laplacian-corrected estimator is used to adjust the uncorrected probability estimate of a feature to account for the different sampling frequencies of different features

Additional Options

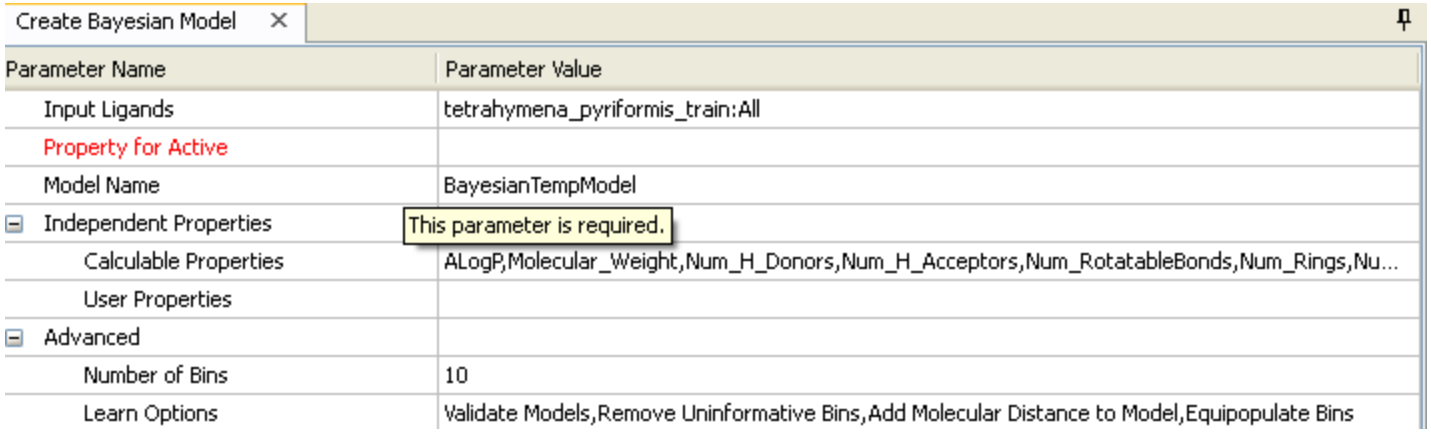
- **Validate Models**
  - Add leave-one-out cross-validation information to the model help
- **Remove Uninformative Bins**
  - Remove bins that contribute close to zero from the model for efficiency
- **Ignore Uninformative Bins**
  - Leave bins that contribute close to zero in the model, but ignore them when predicting
- **Add Molecular Distance to Model**
  - Save fingerprint of each molecule to allow run-time determination of the closeness of a test sample to the nearest training sample
- **Equipopulate Bins**
  - Subdivide continuous variables so that bins have about the same number of samples
- **Save Training Properites**
  - Through Pipeline Pilot, this allows you to use New Model from Old to rebuild the model with new data added to the original data
- **Encrypt Data**
  - If saving training properties, specifies that the data are to be encrypted to prevent direct access

Preparing the dataset

- Ligands that have a value of 1 for the property set by *Property for Active* are considered the "good" ligands
- Calculate Molecular Properties protocol can be used to tag data
- Decide on choice of descriptors
  - Good starting points are:
    - ALogP, Molecular_Weight Num_H_Donors, Num_H_Acceptors, Num_RotatableBonds, Molecular_FractionalPolarSurfaceArea, ECFP_6

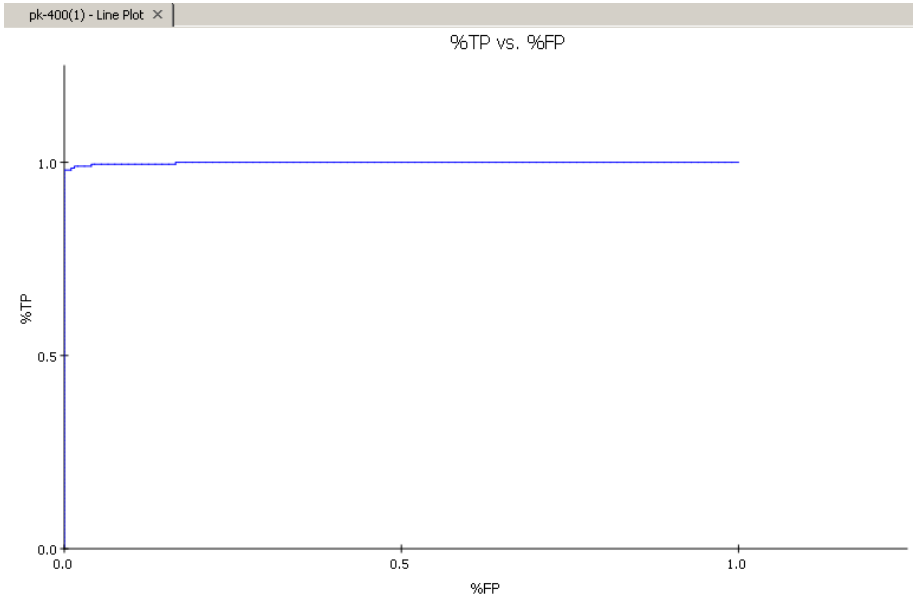
Setup Create Bayesian Model

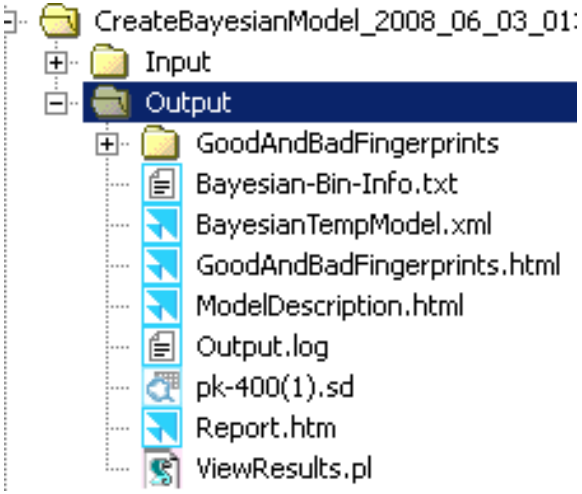
- Specify input ligands
  - must include property which mark active ligands as 1
- Specify property for active compounds
- Specify independent properties
- Advanced setting
  - Define number of bins
    - Determines the number of initial bins used to subdivide continuous variables
      - Default value = 10



Results

- Ligands.sd:
  - Output ligands with additional properties added by the model generation
- ViewResults.pl:
  - Receiver Operating Characteristic (ROC) plot of the input ligands
- ModelDescription.html:
  - Contains important information about the model and training set including leave-one-out cross-validation, ROC plots, true/false positives, and true/false and negatives.
- GoodAndBadFingerprints.html:
  - This file is generated if the model used FCFP or ECFP fingerprints. The file contains images of the top 20 fingerprints that made the most positive contribution to the model as well as images of the top 20 that made the most negative contribution to the model
- Bayesian-Bin-Info.txt:
  - Contains information of the feature statistics including which bins provided the most positive and negative information.
- BayesianTempModel.xml:





Evaluating the model: ModelDescription.html

- In-built cross validation
  - Leave-one-out Cross-Validation
    - Each sample was left out one at a time, and a model built using the results of the samples, and that model used to predict the left-out sample
- Test set validation
    - Split data into training and test sets
    - Build model using training set
    - Sort test set using model value
    - Plot how rapidly hits are found in sorted list

Evaluating the model: ModelDescription.html

- **Test Set Validation**
  - **Best Split** was calculated by picking the split that minimized the sum of the percent misclassified for category members and for category nonmembers, using the cross-validated score for each sample
  - Using that split, a contingency table is constructed, containing the number of true positives (**TP**), false negatives (**FN**), false positives (**FP**), and true negatives (**TN**)

Output	XV ROC AUC	Best Split	TP/FN FP/TN	# in Category
Bayesian TempModel	0.896	-1.243	164/36 31/169	200

Evaluation of the model: ModelDescription.html

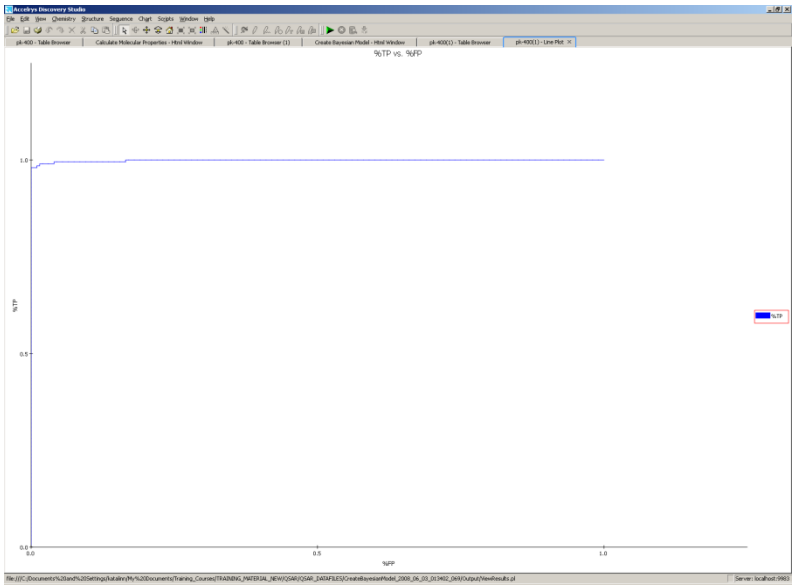
- Enrichment Plot
  - Once all the samples had predictions, an enrichment plot was generated, and the percentage of true category members captured at a particular percentage cutoff.
  - Table returned showing the output name, the percentage of samples that are in that particular category, the number of category members, and the percentage of true members found. Percentages that are less than 100% are in **bold**.

Output	Category %	1%	5%	10%	25%	50%	75%	90%	95%	99%
BayesianTemp Model	50%	**2.50%**	**10.50 %**	**20.50 %**	**48%**	**82.50 %**	**94%**	**98%**	**99%**	100%

Evaluating the Model: ModelDescription.html

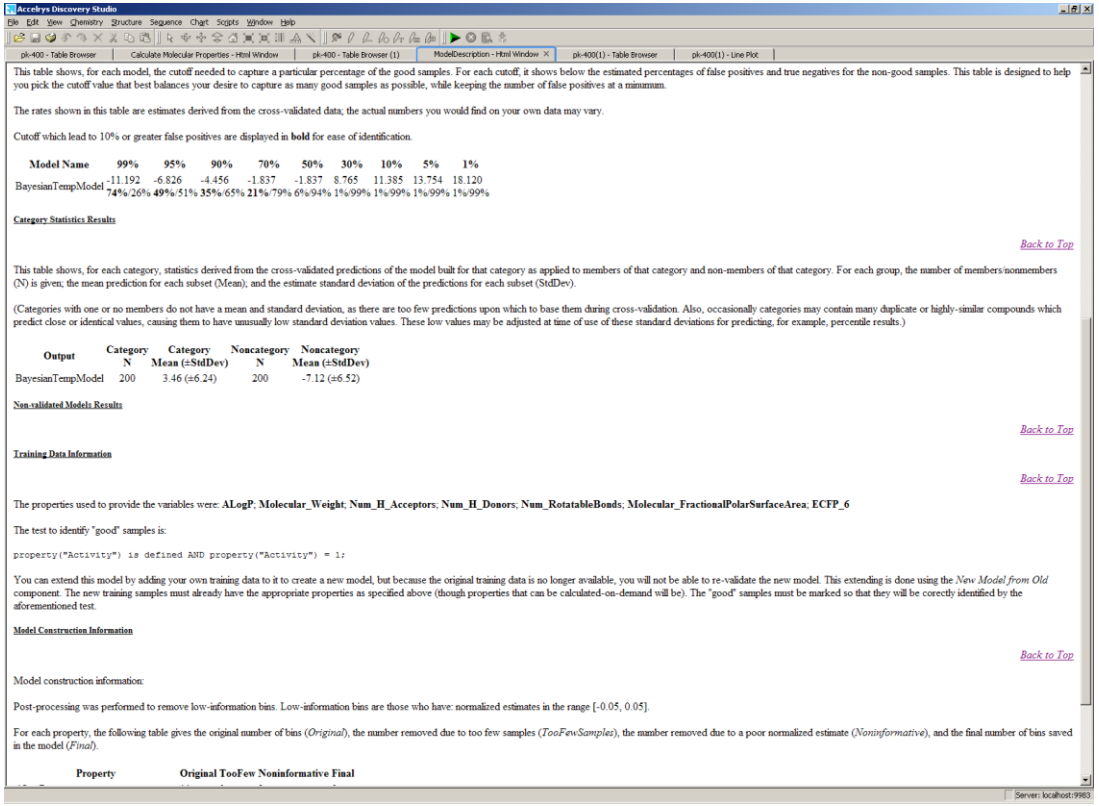
- ROC plot
  - area under the curve (XV ROC) found in ModelDescription.html file
  - The value indicates how often the model correctly identifies the true positive and the true negative
  - It can be used to classify the accuracy of the model used to generate an ROC curve:

XV ROC value	Model rating
0.90-1	excellent (A)
0.80-0.90	good (B)
0.70-0.80	fair (C)
0.60-0.70	poor (D)
0.50-0.60	fail (F)

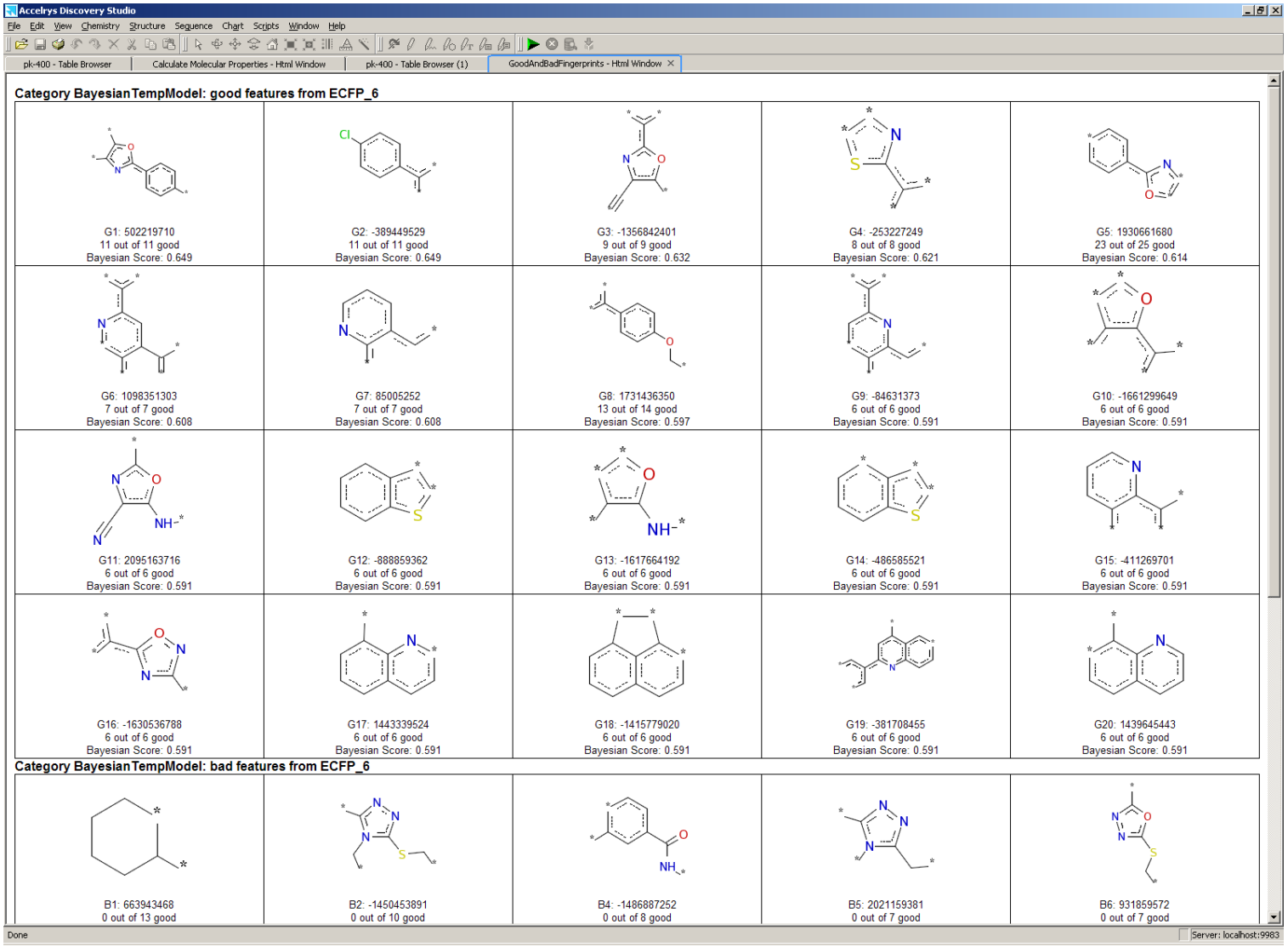


Evaluating Model: ModelDescription.html

- Additional information
  - Percentile Results
  - Category Statistics Results
  - Training Data Information
  - Model Construction Information



GoodAndBadFingerprints.html

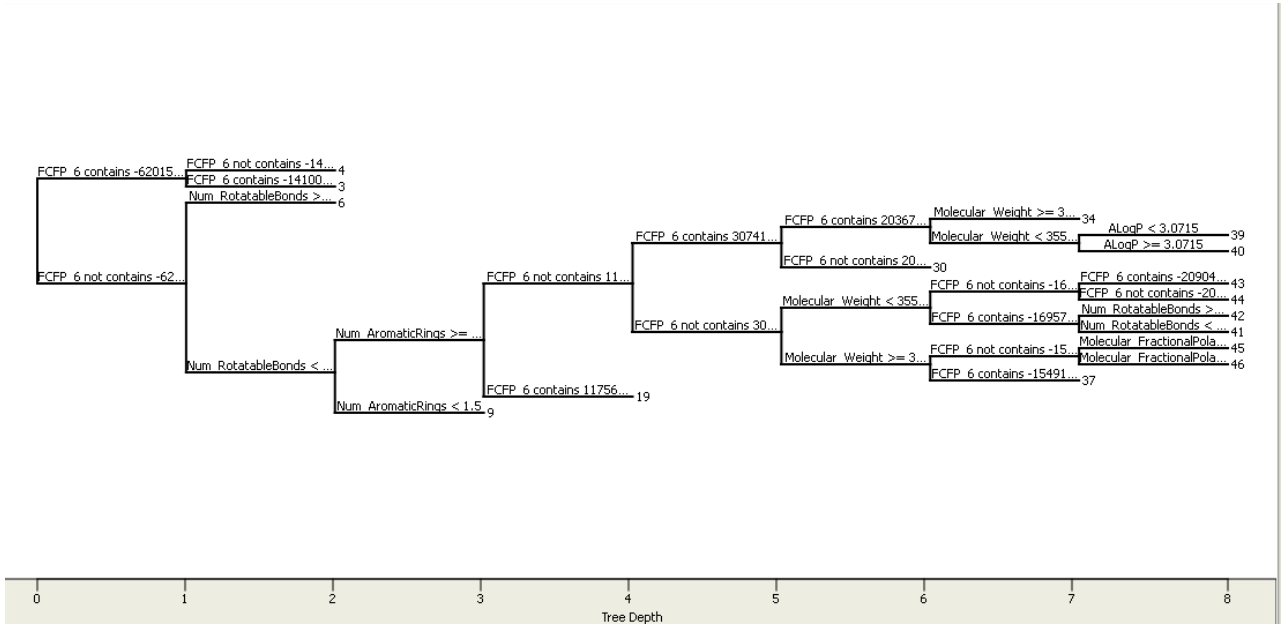


Recursive Partitioning

- Data classification method used to uncover the relationship between a dependent property (Y variable) to a set of independent properties (X variables)
- Analyse structure-activity relationships with:
  - Activity levels (1, 2 ...) or classes (A, B …)
  - Non-linearities
  - Thresholds effects
  - Variable interactions
  - Compounds acting by multiple mechanisms
  - Many thousands of compounds
- Advantages
  - Easy to interpet
  - Fast to calculate
  - Limited sensitivity to outliers
  - Applicable to all types of descriptors

What is Recursive Partitioning (RP)?

- A method for building classification models
- The splitting of a data set into smaller and smaller subsets based on yes/no questions
  - "Is MeanPixelIntensity < 86.5?"
  - "Is FCFP_6 feature 8137712 present?"
  - Finds questions that best separate different classes within the data
- Provides *predictive accuracy* and *interpretability*



Types of RP Models

- Conventional Single-Tree Model
- Lookahead Model
- Multi-Y Model
- Multi-Tree Forest Model
  - Small *Bagged* Forest
  - Large Forest of Random Trees

Conventional Single-Tree Model

- To build: Starting with the entire data set at the root node, find question that best splits data at each node into separate classes
- Continue until no split gives improvement

- Process of building tree is deterministic (nonrandom)
- Initially built tree tends to overfit data; must be *pruned*
- Use cross-validation (CV) or test set to determine best pruned tree

Measuring Node Purity

- Different measures are available
- Specified by *Split Method* parameter
- Default: Gini impurity index
  - $G = 1 - \sum P_i^2$
  - P_i = (weighted) fraction of node members in class i
  - Sum is over all classes
  - For pure node (all one class), G = 0

Making Predictions

- To make a prediction for a new sample with an unknown class:
  - Start at the root node
  - Given the split question at a node, determine which branch should be taken
  - Continue until reaching a leaf node
  - The predicted class is the class with the greatest value of P_i for the leaf node (as determined from training data)

Controlling Initial Tree Size

- *Minimum Samples Per Node:* Specifies minimum number of samples that must be present in each child node in order for a split to occur
- *Maximum Tree Depth:* Specifies maximum number of levels from the top (root) node to which a tree can be grown
- Suggestions
  - For data sets of at least a few hundred samples, default settings give reasonable results
  - If in doubt, set minimum samples small and maximum depth large. Pruning will take care of overgrown trees.
  - When data set is large with many members of each class, increasing minimum samples or decreasing maximum tree depth can reduce training time while maintaining predictive performance

Look-Ahead Tree Model

- Tree with ability to optimize a split based on its effect on subsequent splits down the branches
- Final model looks just like conventional tree model
- Like other single-tree models, must be pruned
- Unlike other tree models, more-pruned tree not necessarily a subtree of less-pruned tree

Multi-Y Tree Model (continued)

- Generic nodes “contain” data for all Ys
- Specific nodes “contain” data for a single Y
- Generic nodes
  - are split into generic nodes when
    - node depth is less than *Maximum Generic Depth* and
    - a split question can be found that improves class separation for all Ys
  - otherwise are split into specific nodes
- Specific nodes
  - are split only into specific nodes

Multi-Class Tree Model

- Tree model of a dependent (Y) variable with more than two possible values (e.g.
- Not a special algorithm; just a generalization from binary Y case
- Produces a matrix of partial ROC scores
  - ROC_{AB} = ROC score for distinguishing classes “A” and “B” based on predicted P(A)
  - ROC_{BA} = ROC score for distinguishing classes “A” and “B” based on predicted P(B)

Multi-Tree Forest Models

- Basic idea: resample the data and build multiple, independent trees
- Take consensus prediction of trees as overall prediction
- Some trees might get stuck with “bad” splits, but not all trees
- Any given tree in forest may overfit data and be a worse predictor than a single-tree model, but…

- **Many weakly predictive models averaged together yield a more strongly predictive model as long as their errors are uncorrelated**

Bagged Forest

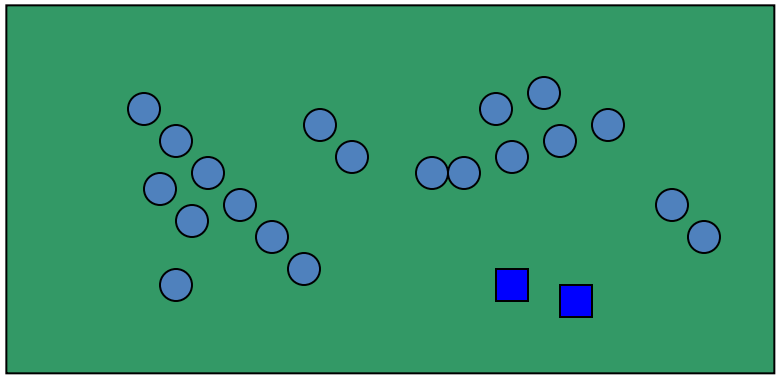
- Multiple trees built on bootstrap samples of data
- Descriptors are *not* resampled
- Typically 10 to 20 trees in forest is enough
- Better predictive performance than single-tree models, but less directly interpretable

Forest of Random Trees

- Multiple trees built on bootstrap samples of data
- Descriptors *are* resampled
  - For each node in each tree, consider a randomly-selected subset of descriptors for splits
  - Typical subset size for D descriptors is sqrt(D)
- Typically 500+ trees in forest to adequately sample both data and descriptors
- Better predictive performance than single-tree models, but less directly interpretable
- Outputs statistical measures of descriptor importance

Dealing with Imbalanced Data

- Example: data set in which there are many inactive compounds, but only a few active ones
- *Weighting Method* parameter
  - Uniform: each sample is weighted the same
  - By Class: minority class samples are up-weighted such that the total weight of all samples in each class is the same
  - Weights are used to determine splits and to make predictions
- *Equalize Class Sizes* parameter (forest models only)
  - Ensures all classes contain equal number of samples for each tree



Which type to select?

Model Type	Typical Number of Trees	Interpretability	Predictive Performance	Speed of Learning
Single-tree	1	Very Good	Good	Fast
Bagged Forest	10	Average	Very Good	Average
Forest of Random Trees	500	Direct: Poor Indirect: Good	Very Good	Slow
Balanced Forest of Random Trees	1000	Direct: Poor Indirect: Good	Very Good	Fairly Slow

- Direct" interpretability refers to the ease of interpretation by direct inspection of the tree structures
- "Indirect" interpretability refers to the ease of interpreting the descriptor importance measures produced by the learning process

RP versus Bayesian Learning

- Both Methods:
  - Can solve classification problems with high accuracy
  - Are chemically aware (e.g., handle fingerprints)
  - Can handle any type of descriptor
    - Categorical
    - Binary / Fingerprint
    - Continuous
  - Can provide measures of descriptor importance
  - Require no preprocessing or data normalization
  - Can handle large datasets
    - Large numbers of samples
    - Large numbers of descriptors
    - HTS-type data (large volume, few hits, highly imbalanced)

RP versus Bayesian Learning

- RP Trees:
  - Easier to interpret — readable tree display
  - Can account for interaction effects between descriptors
  - Directly supports multi-class and multi-Y models

- Bayesian Learning:
  - Simpler to use — fewer parameters
  - Assumes effects of descriptors are independent
  - Faster

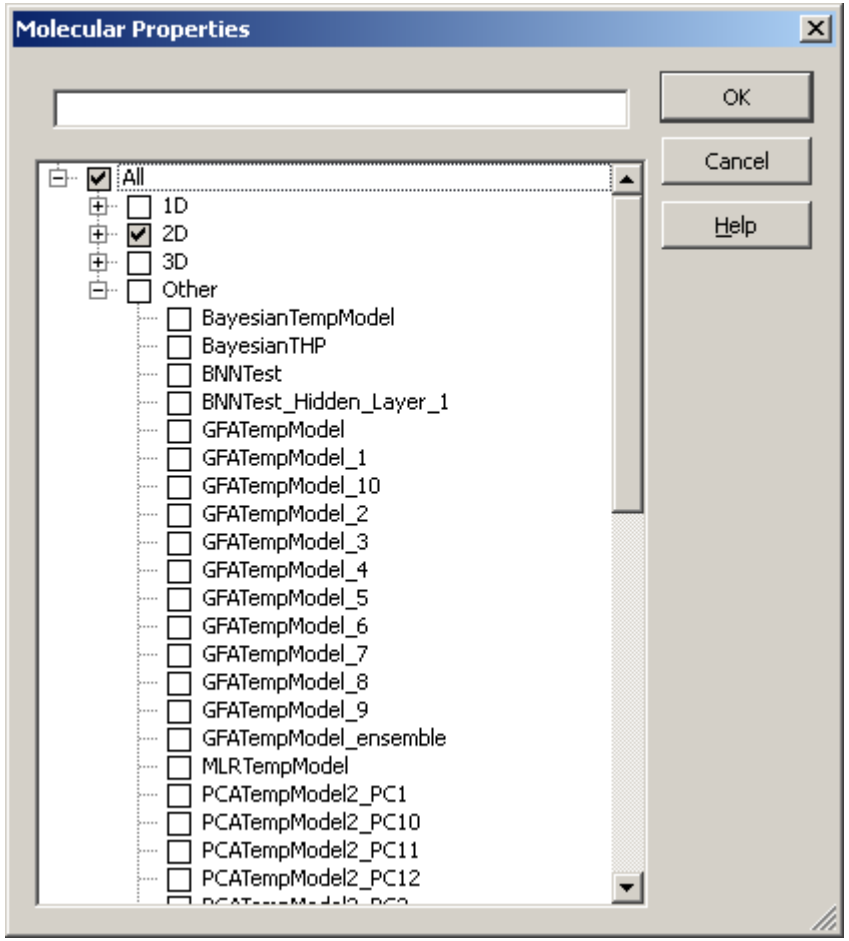
Application of both methods can add insight

Comparison of methods

	Dependent Variable	Independent Variable	Dataset	Comments
Multiple Linear Regression	Continous data only	Numerical data; fingerprints not allowed	Under-determined	Very simple, very basic implementation
Partial Least Squares	Continous data only	Numerical data; fingerprints not allowed	Over-determined	Harder to interpret (latent variables vs. descriptors)
Genetic Functional Approximation	Continous data only	Numerical data; can handle fingerprints	Over-determined	Multiple solutions & ensemble model; various scoring functions
Back Propagation Neural Network	Continous data only	Numerical data; fingerprints or categorical data not allowed	Applicable to HTS data, can handle very large data	Able to handle missing data; more of a black box solution; can handle multi-Y
Bayesian Model	Categorical data only	Numerical data & fingerprints allowed	Applicable to HTS data, can handle very large data	handles categorical data, harder to interpret but powerful
Recursive Partitioning	Categorical data only	Numerical data & fingerprints allowed	Applicable to HTS data, can handle very large data	Handles categorical data, easier to interpert, can handle multiple Y variables

Applying & Sharing Models

- Any model created can be applied on a dataset in the Calculate Molecular Properties protocol
- Xml file of the model from the Output folder can be emailed/sent to any other DS users
  - Import under user's own protocol folder
  - After import it will be available in the Calculate Molecular protocols automatically



Accelrys Support

- Technical Exploration
  Dr.Anand Krishnamurthy – akrishnamurthy@accelrys.com

- Procurement
  Mr. Krishnakumar- kmuralidheeran@accelrys.com

  - Accelrys Web Site www.accelrys.com
  - Accelrys Community www.accelrys.org
  - Accelrys Advantage customer.accelrys.com
  - Accelrys Training training@accelrys.com
  - Pipeline Pilot Support support@accelrys.com

- Contacts
  Tel: +91-80-41102242 ; +91-80-41102242 ;+91-80-41102243 ;
  Fax: +91-80-41102247
  Email - support@accelrys.com