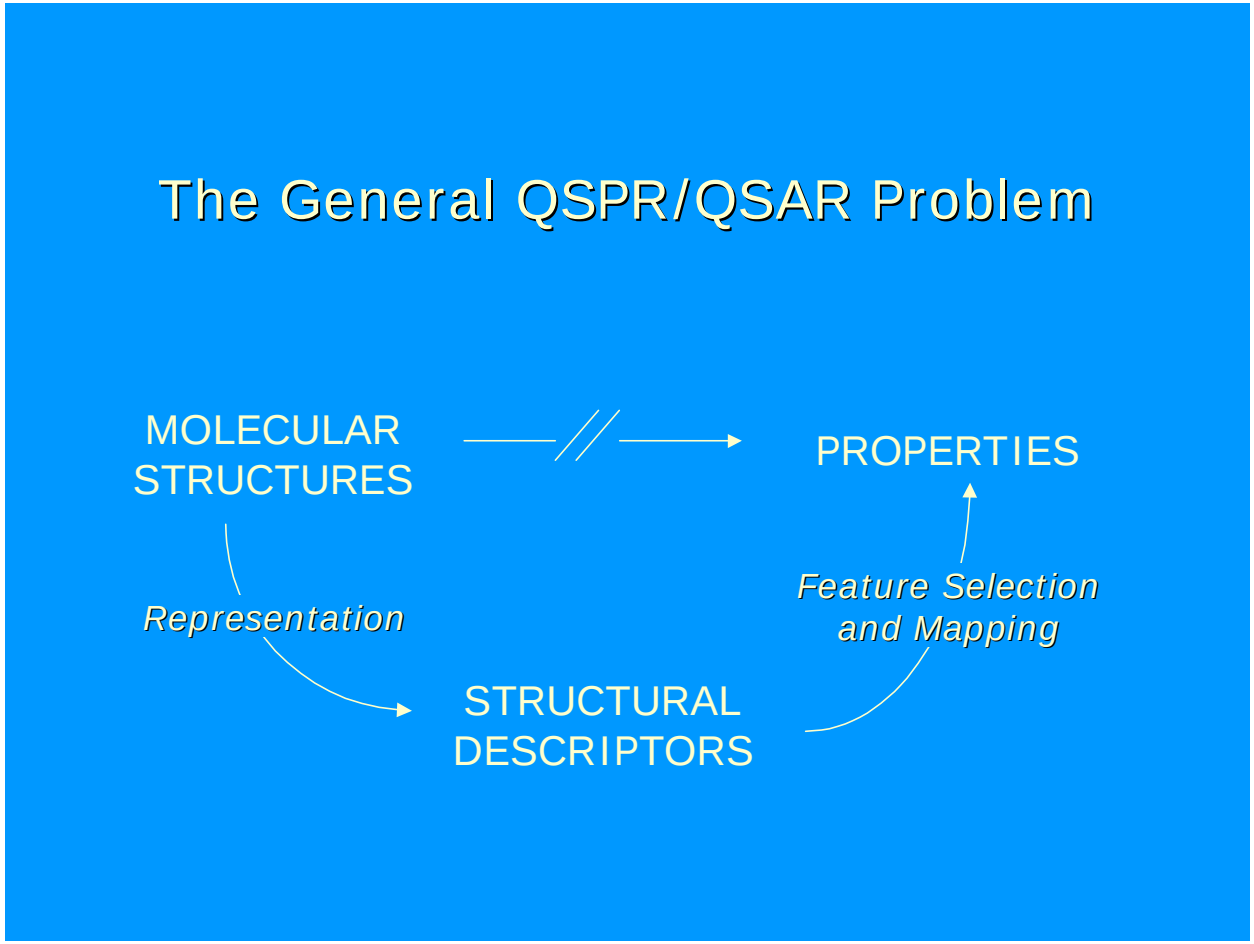
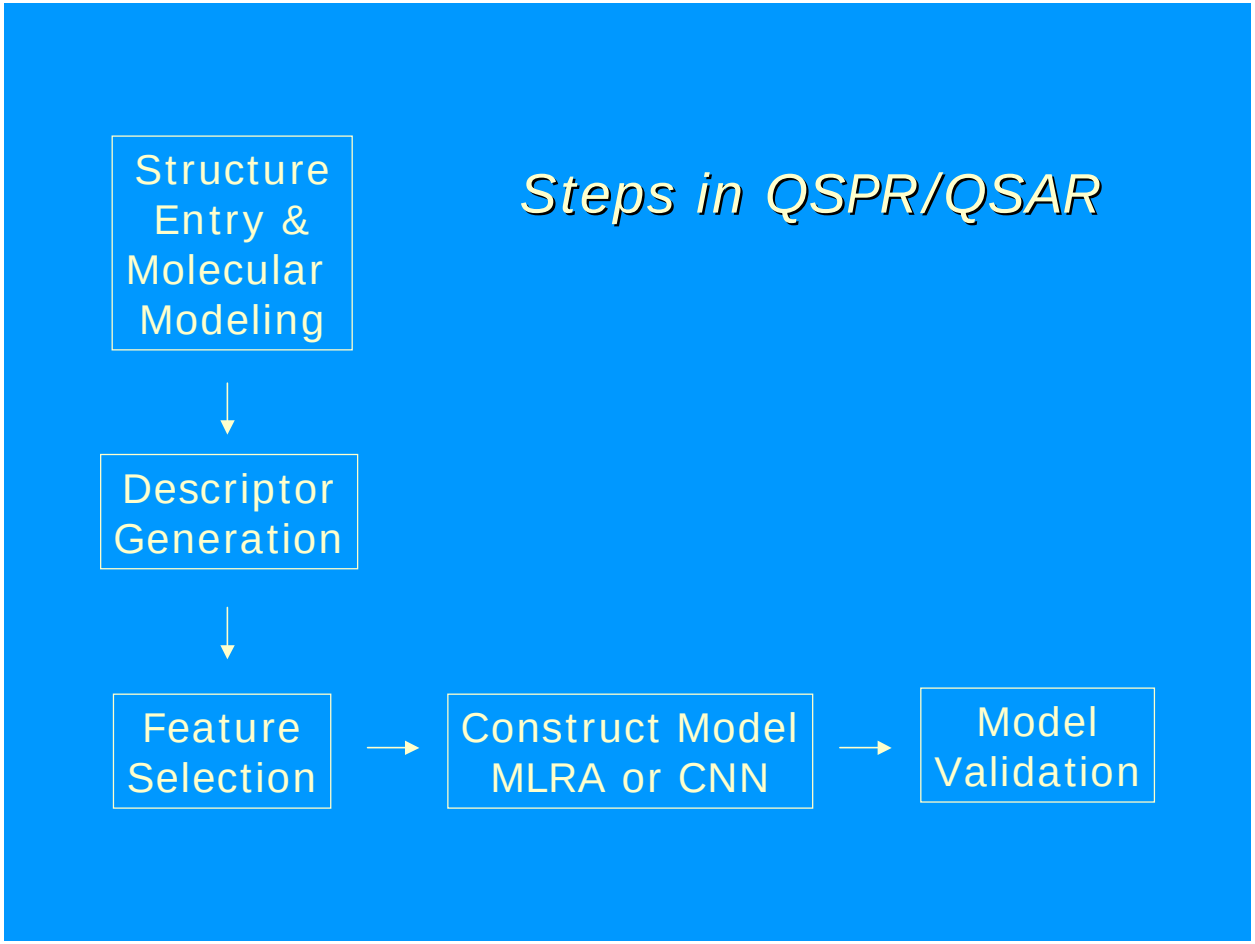
# Prediction of Chemical and Biological Properties of Organic Compounds from Molecular Structure

**Peter C. Jurs**
**Chemistry Department**
**Penn State University**
**University Park, PA 16802**
**pcj@psu.edu**

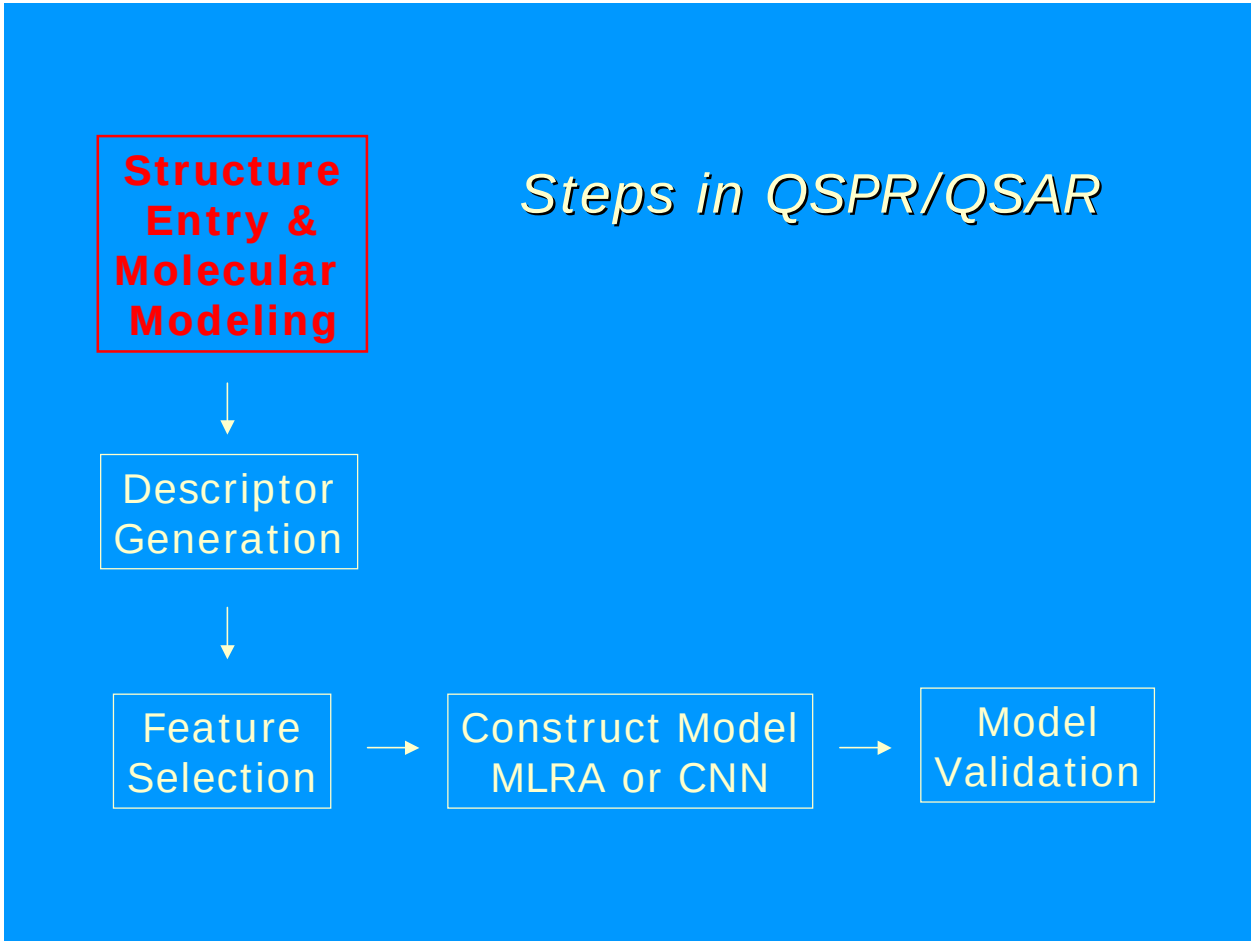
This set of slides and accompanying notes presents an introduction to the use of the ADAPT software package to develop **Q**uantitative **S**tructure-**P**roperty **R**elationships (QSPRs) and **Q**uantitative **S**tructure-**A**ctivity **R**elationships (QSARs). This approach deals with organic compounds of intermediate size, but not biopolymers or proteins. The models are developed using calculated numerical descriptors to encode information about each of the molecular structures. These descriptors are used to build statistical or computational neural network models to predict the property or activity of interest.




The molecular structure of an organic compound determines its properties. However, the arrow representing the direct link between them is broken to indicate that *a priori* methods are usually not available for direct predictions. Therefore, an indirect approach is used which consists of two main parts: (a) representing each compound's molecular structure with calculated structural descriptors, and (b) choosing subsets of the descriptors and building good models that predict the property or activity of interest. The models can be statistical models or computational neural network models. The method is inductive, that is, it depends on having a set of compounds with known properties or activities. This set of known compounds is used to develop the model. The approach should be applicable to any problem for which the property or activity of interest is dependent upon the molecular structure.

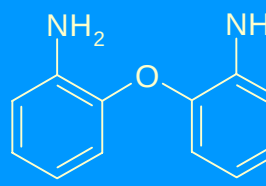



The main components of a QSPR/QSAR study are presented here. The molecular structure of each compound is entered and stored in a topological representation. Each structure is submitted to conformational analysis to generate a good, low-energy conformation. The topological and geometrical representation of the structures are used to calculate molecular structure descriptors. The descriptor set is submitted to feature selection, in which the best subsets of descriptors are sought. Models based on statistical methods or computational neural networks are built with the subsets of descriptors. The models are validated with an external prediction set.

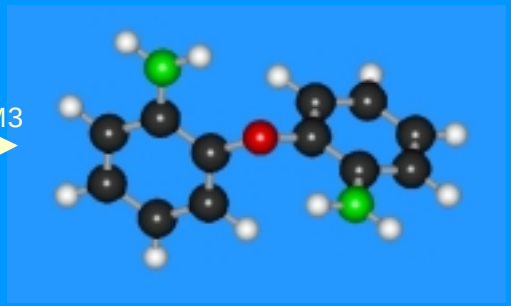



The first major step in a QSPR/QSAR study is the entry of the molecular structures and generation of the 3-D models.

# Structure Entry and Molecular Modeling



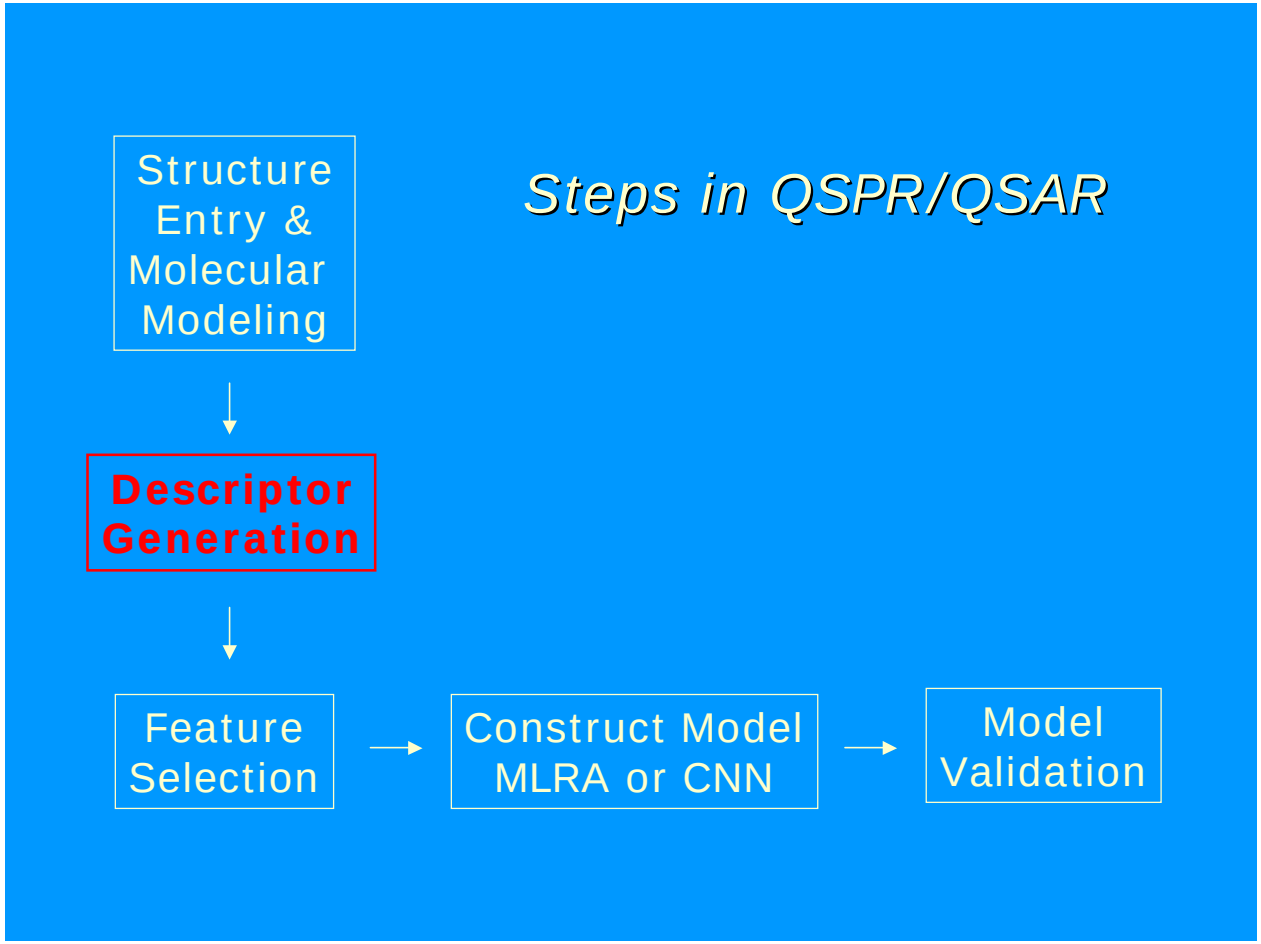

MOPAC / PM3



2-D Sketch

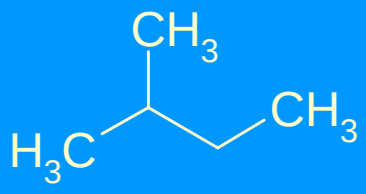
3-D Conformation

The structures are entered by sketching with HyperChem. The 2-D sketch is used to generate a good, low-energy conformation with mopac using the PM3 Hamiltonian. The 3-D molecular models are needed for geometric descriptor calculation.

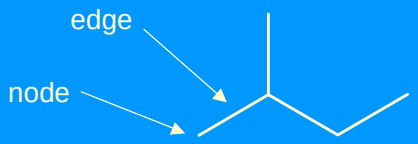



The second major step in a QSPR/QSAR study is the generation of the molecular structure descriptors.

# Topological Descriptors




Isopentane

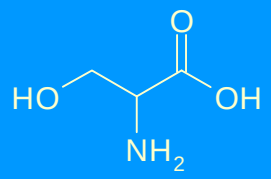



Graph Representation

The graph consists of nodes, edges, adjacencies
Use graph representation to derive descriptors

The structures of organic compounds can be represented as graphs, a field of mathematics. Shown here is the example molecule isopentane and its graphical representation consisting of five nodes, four edges, and the adjacency relationships implicit in the structure. Once the organic compounds are considered as graphs, the theorems of graph theory can then be applied to generating graph invariants, which in the context of chemistry are called topological descriptors.

# Topological Descriptors




Examples

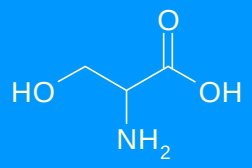
- Atom counts
- Molecular connectivity indices
- Substructure counts
- Molecular weight
- Weighted paths
- Molecular distance edges
- Kappa indices
- Electrotopological state indices

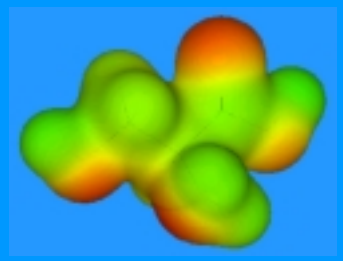
Examples of topological descriptors include the following: atom counts, ring counts, molecular connectivity indices, substructure counts, molecular weights, weighted paths, molecular distance edge descriptors, kappa indices, electrotopological state indices, and many other graph invariants.

# Electronic Descriptors

Examples

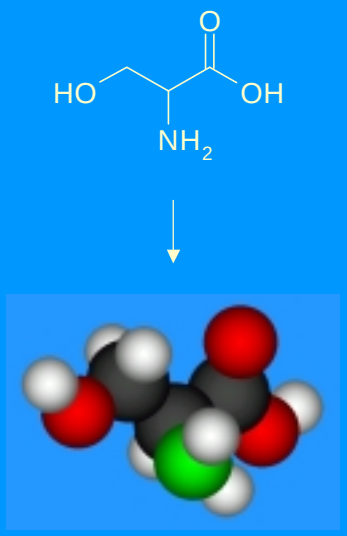
- Most positive or negative charge
- LUMO energy
- Partial atomic charges
- Dipole moment



Electronic descriptors are calculated to encode aspects of the structures that are related to the electrons. Examples of electronic descriptors include the following: partial atomic charges, HOMO or LUMO energies, dipole moment.

# Geometric Descriptors

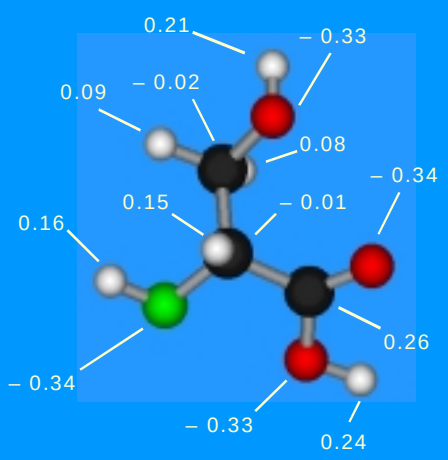



Examples

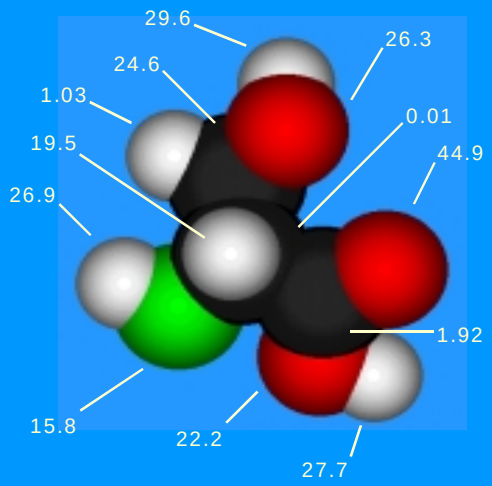
- Moments of inertia
- Accessible surface area & volume
- Shadow areas
- Length-breadth ratios

Geometric descriptors are calculated to encode the 3-D aspects of the structures and include such descriptors as moments of inertia, solvent-accessible surface area, length-to-breadth ratios, shadow areas, gravitational index.

# CPSA Descriptors for Serine




**AM1 Partial Atomic Charges**




**Partial Surface Areas**

SASA = 278 A²

A class of hybrid descriptors called charged partial surface area descriptors encode the propensity of compounds to engage in polar interactions. The set of cpsa descriptors is based on the partial atomic charges and the partial surface area of each atom. These two lists of attributes are mixed and matched with various weighting schemes to generate a set of approximately 25 cpsa descriptors. Examples of cpsa descriptors include the following: fractional positive surface area, charged-weighted negative surface area.

# CPSA Descriptors for Serine

| Atom No. | Atom Type | Charge | Surface Area ($A^2$) |
|---|---|---|---|
| 1 | N | -0.34 | 15.8 |
| 2 | C | -0.01 | 0.01 |
| 3 | C | -0.02 | 1.03 |
| 4 | O | -0.33 | 26.3 |
| 5 | C | 0.26 | 1.92 |
| 6 | O | -0.34 | 44.9 |
| 7 | O | -0.33 | 22.2 |
| 8 | H | 0.16 | 26.9 |
| 9 | H | 0.16 | 20.4 |
| 10 | H | 0.15 | 19.5 |
| 11 | H | 0.09 | 24.6 |
| 12 | H | 0.08 | 17.5 |
| 13 | H | 0.21 | 29.6 |
| 14 | H | 0.24 | 27.7 |

The partial charges and the partial solvent accessible surface areas for each atom in serine are listed. These values are used to compute the cpsa descriptors for serine.

# Example CPSA Descriptors for Serine

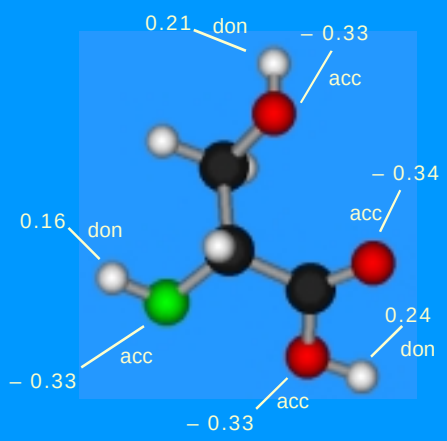
Partial Positive SA (PPSA)
$\Sigma(+SA) = 168\ A^2$

Partial Negative SA (PNSA)
$\Sigma(-SA) = 110\ A^2$

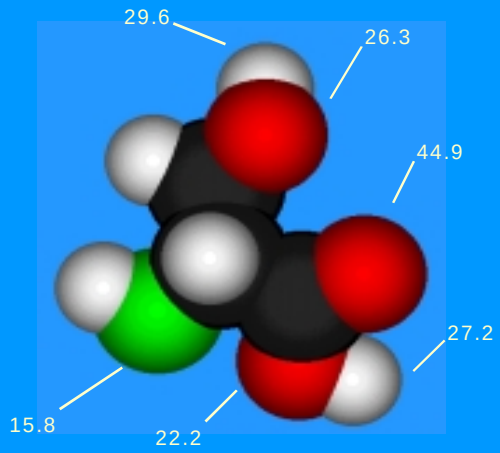
Fractional Positive SA (FPSA)
PPSA / SASA = 0.60

Three specific examples of cpsa descriptors are shown for serine. The actual cpsa routine computes 27 cpsa descriptors, including the three shown here.

# H-bonding Donors and Acceptors for Serine




**AM1 Partial Atomic Charges**




**Partial Surface Areas**

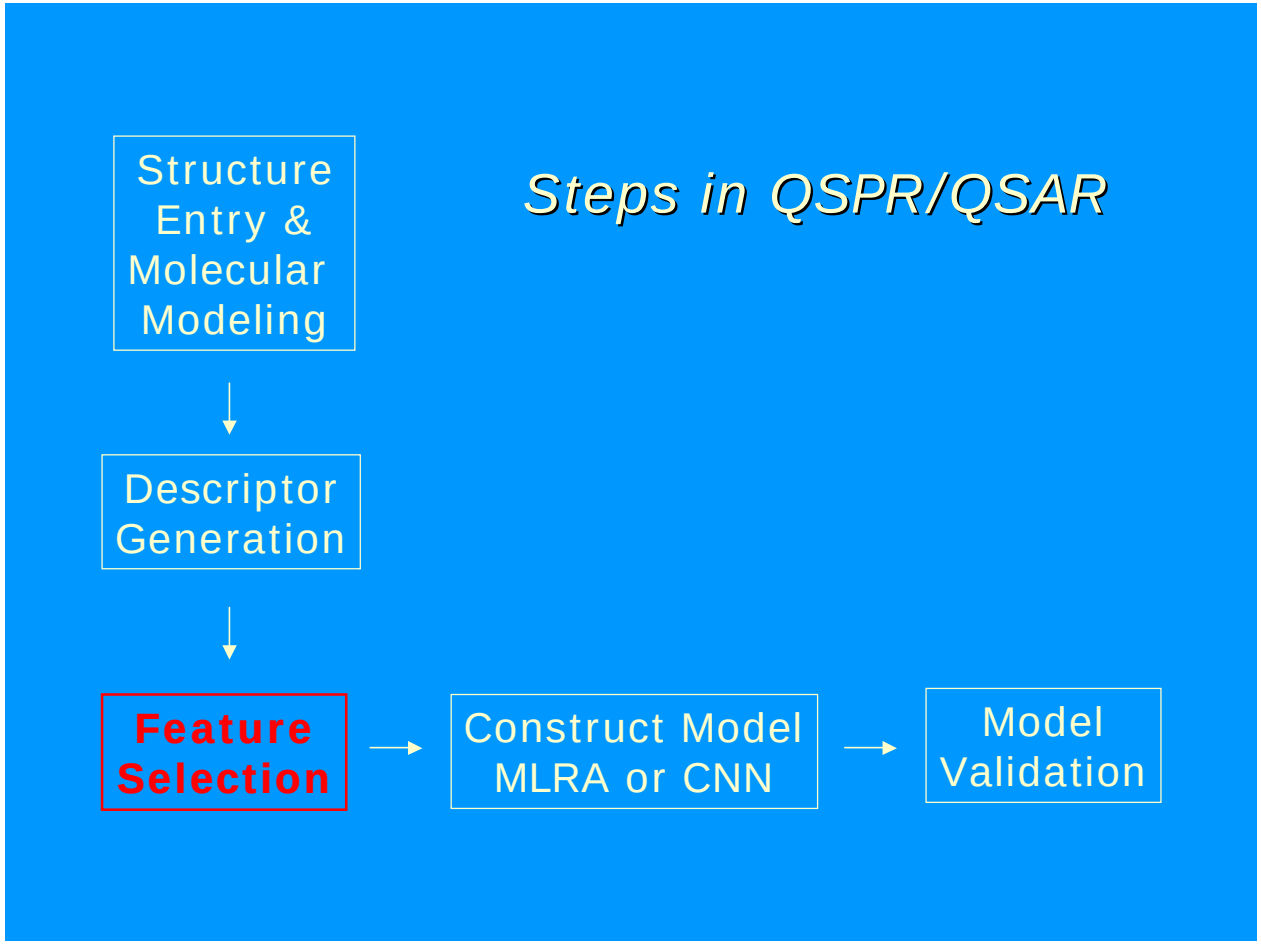
SASA = 278 A²

If one extends the cpsa concept to only those atoms that can act as hydrogen-bond donors and acceptors, a series of descriptors can be computed that encode the molecule's ability to engage in H-bonding. Here, the seven donor and acceptor atoms are highlighted.

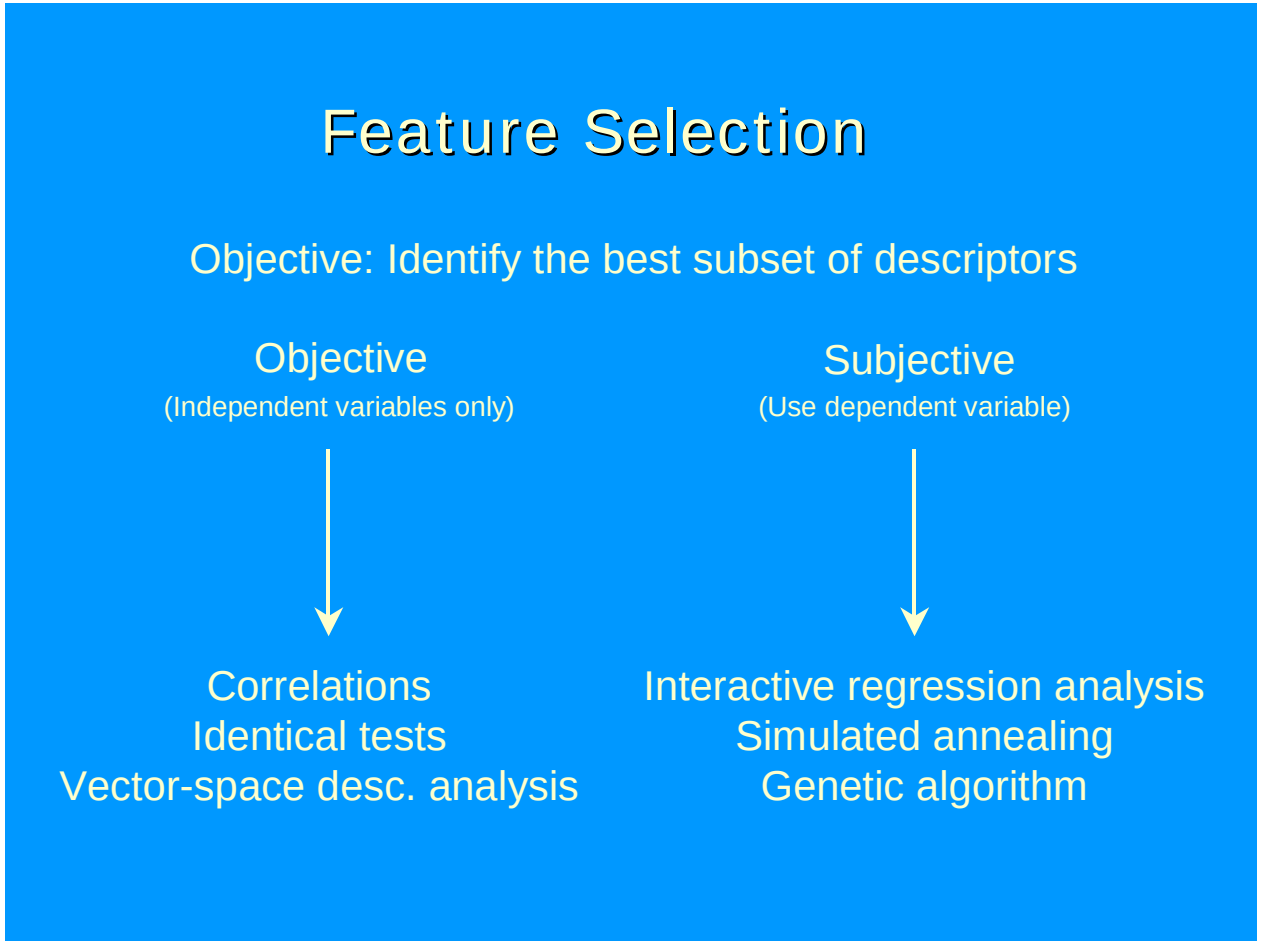
## Serine: H-bond Donors and Acceptors

| Atom No. | Atom Type | Charge | Surface Area ($A^2$) |
|---|---|---|---|
| 1 | N | -0.34 | 15.8 |
| 2 | C | -0.01 | 0.01 |
| 3 | C | -0.02 | 1.03 |
| 4 | O | -0.33 | 26.3 |
| 5 | C | 0.26 | 1.92 |
| 6 | O | -0.34 | 44.9 |
| 7 | O | -0.33 | 22.2 |
| 8 | H | 0.16 | 26.9 |
| 9 | H | 0.16 | 20.4 |
| 10 | H | 0.15 | 19.5 |
| 11 | H | 0.09 | 24.6 |
| 12 | H | 0.08 | 17.5 |
| 13 | H | 0.21 | 29.6 |
| 14 | H | 0.24 | 27.7 |

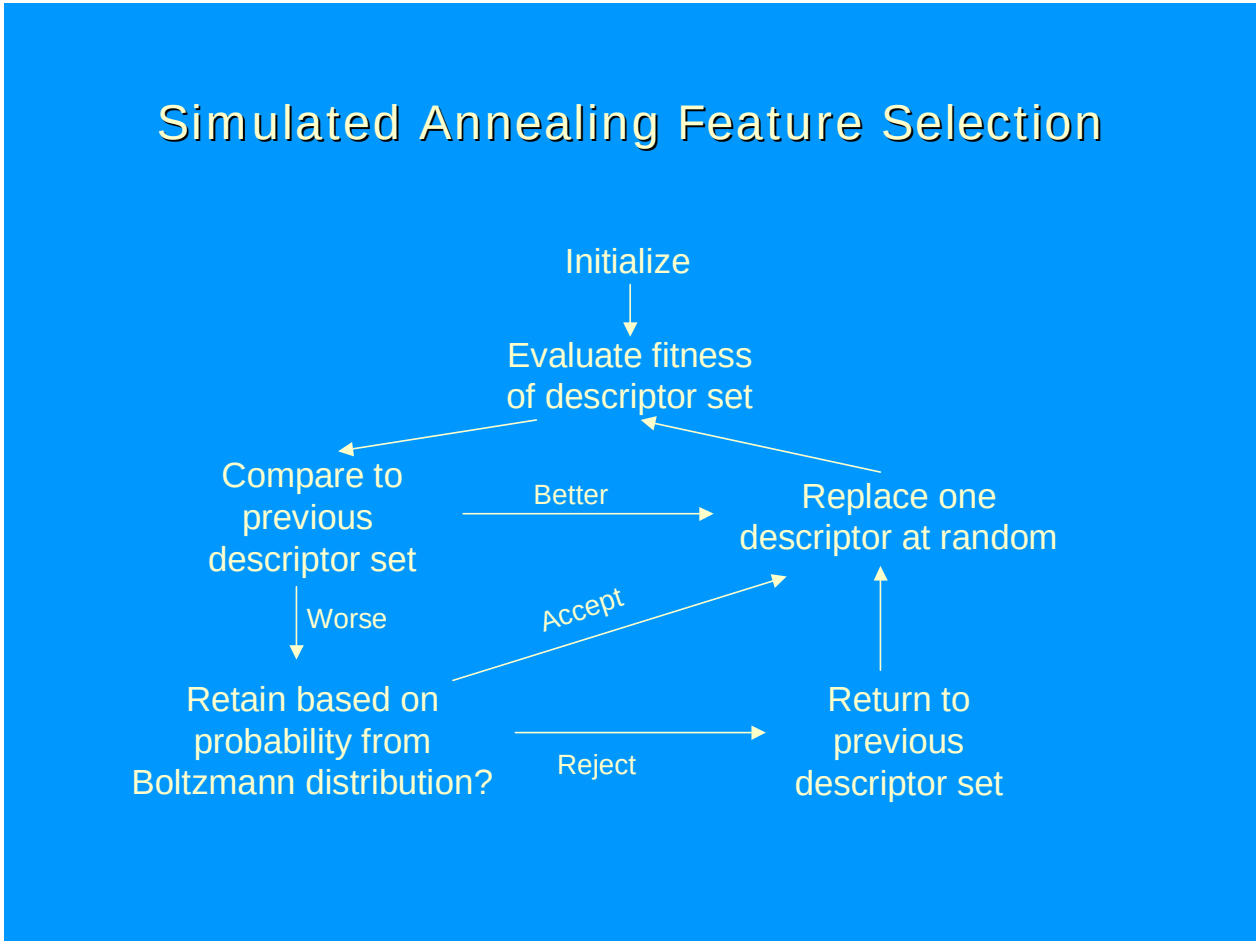
The donor and acceptor atoms are highlighted in color here. When these atoms are used to compute a set of descriptors, then the propensity of the compound to engage in H-bonding is encoded.



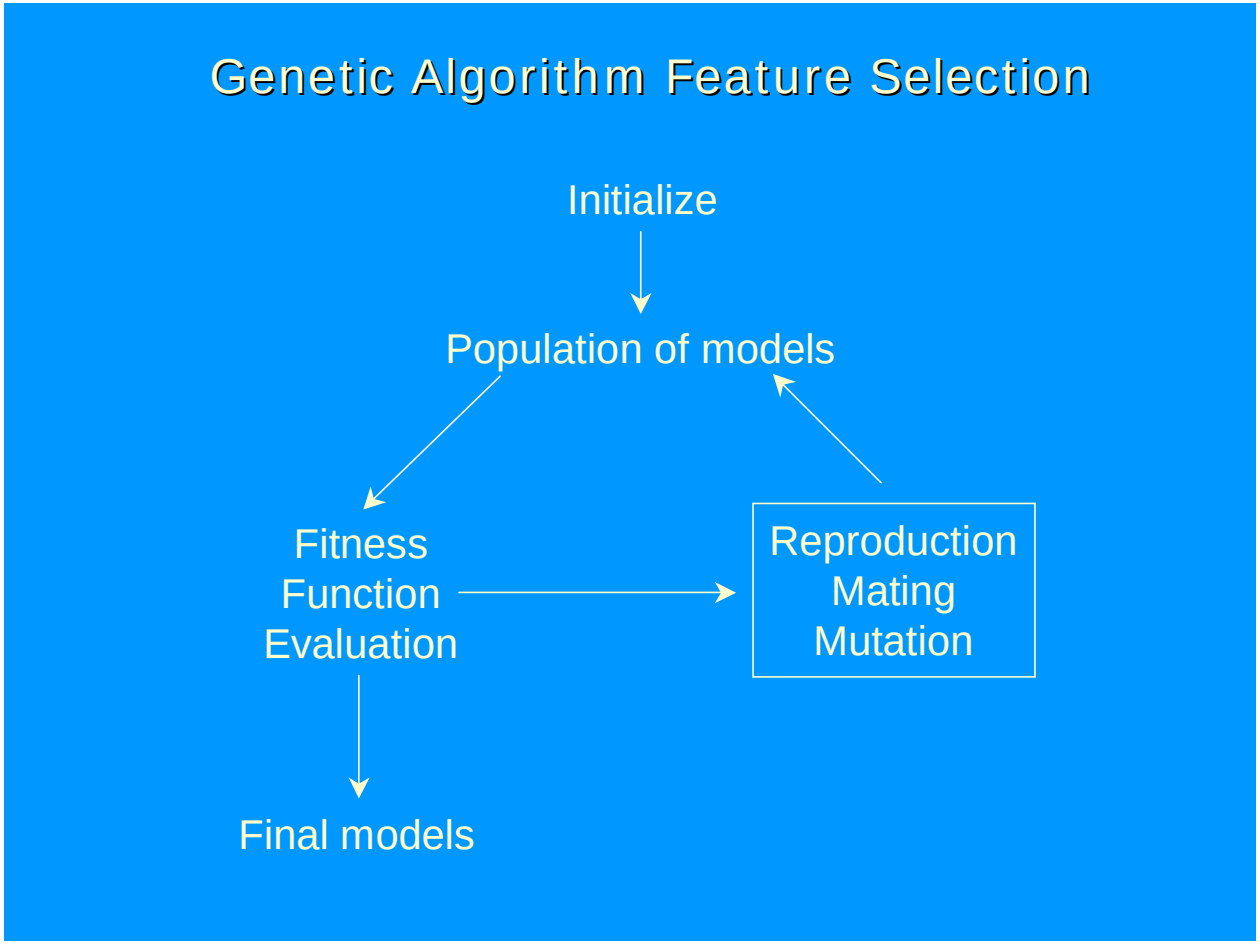

The third major step in a QSPR/QSAR study is the selection of the most important descriptors using feature selection methods.



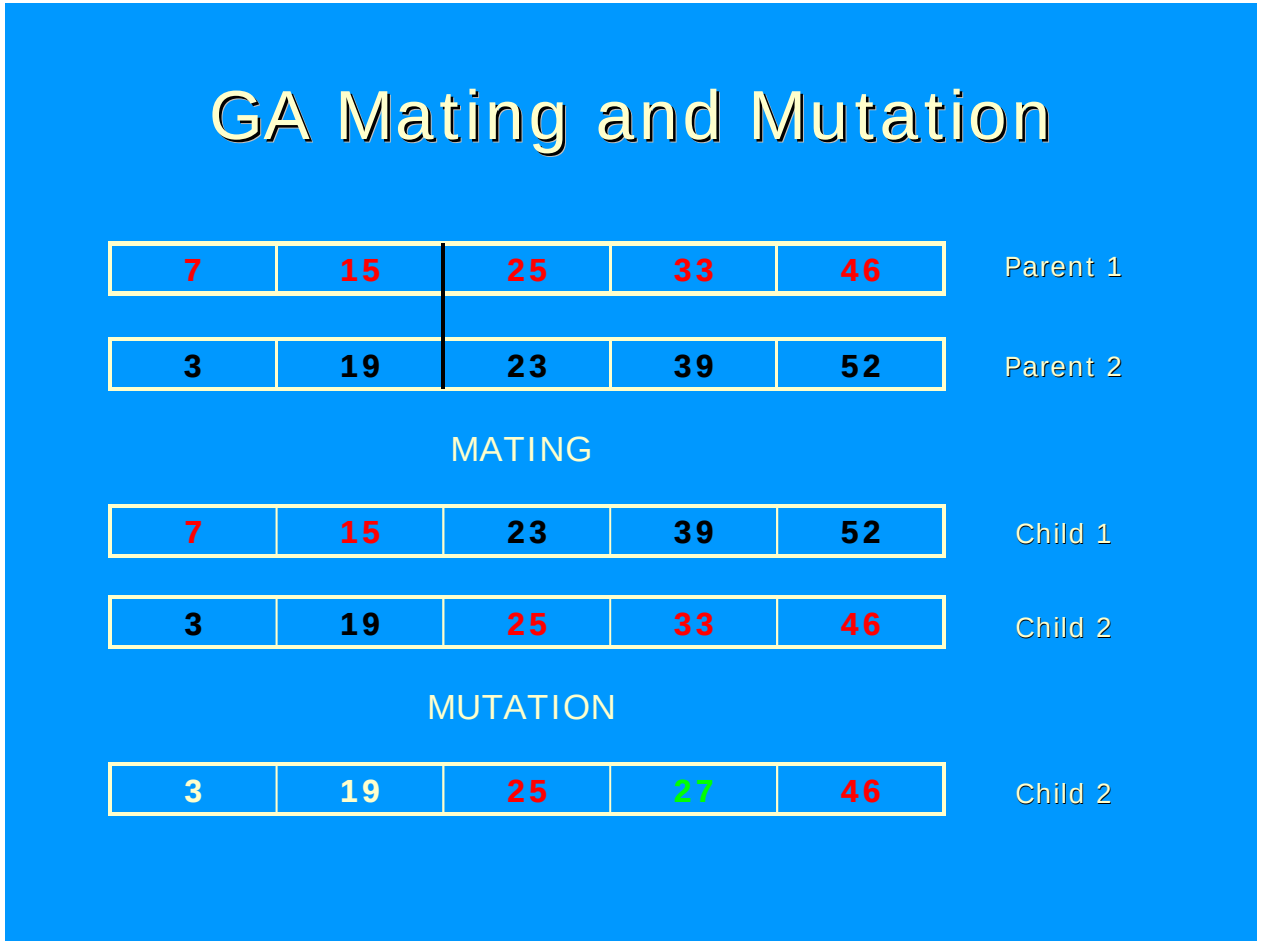

After descriptors have been calculated for each compound, this set must be reduced to a set of descriptors which is as information rich but as small as possible. Objective feature selection is done first. Objective feature selection uses only the independent variables, and descriptors to discard are identified by pairwise correlations, tests of identical values, and a vector space descriptor analysis which looks for descriptors that are orthogonal to one another. A reduced descriptor pool results. Secondly, subjective feature selection, which does use the dependent variable values, is applied to further reduce the descriptor set.



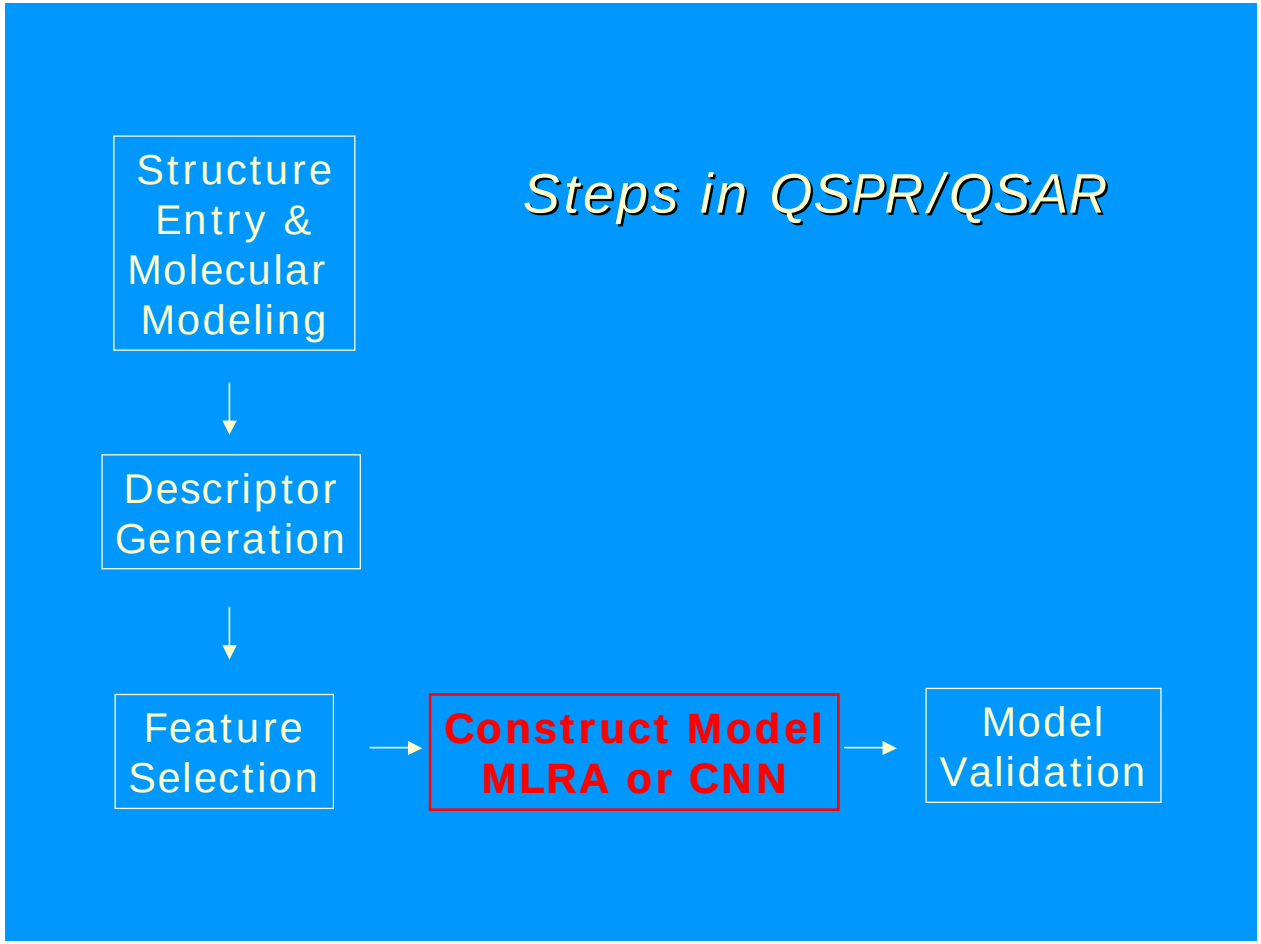

Simulated annealing searches the descriptor space for optimal subsets one string at a time. It begins with an initially random string of descriptors and replaces one or more of the descriptors with new descriptors from the reduced pool. Each new subset is evaluated by an appropriate cost function – typically an error minimization. If the cost function of the new subset is better than the previous subset, then the new subset is appropriately stored in the list of best models. If the cost function is worse, then a probability function is used to determine if the algorithm should take a detrimental step – that is, proceed with a mutation of the new subset – or revert back to the previous subset of descriptors and attempt a new mutation. The ability to take many more detrimental steps early in the optimization reduces the risk of converging in a local error minimum, thus as the optimization proceeds, detrimental steps become more difficult to take.




The genetic algorithm differs from simulated annealing in that a population of 40-50 strings are examined each iteration rather than a single one. Once an initial population of strings is created, EACH subset is evaluated by a cost function and subsequently ranked from best to worst. The best 50% of the models then undergo two processes called mating and mutation to generate a new population of "children" strings – that is to say, each population of strings serves as "parents" to the subsequent population of "children" strings.




To illustrate the process of mating, let's say we have two subset strings called parent 1 and parent 2. The algorithm determines a fixed split point to perform a cross-over mating process whereby the first two descriptors of parent 1 and the last three descriptors of parent 2 are combined to form child 1. The remaining descriptors from these two subsets are combined to form a second child. In addition, a low-probability single-descriptor mutation can occur in approximately 5% of the children strings to prevent premature convergence in local minima. Typically the mating and mutation process is repeated for 1000 iterations and the best models are ranked.




The fourth major step in a QSPR/QSAR study is the generation of the QSPR/QSAR models using the descriptor sets. Models can be statistical or can be computational neural networks.

# Multiple Linear Regression

Goal: Estimate coefficients in

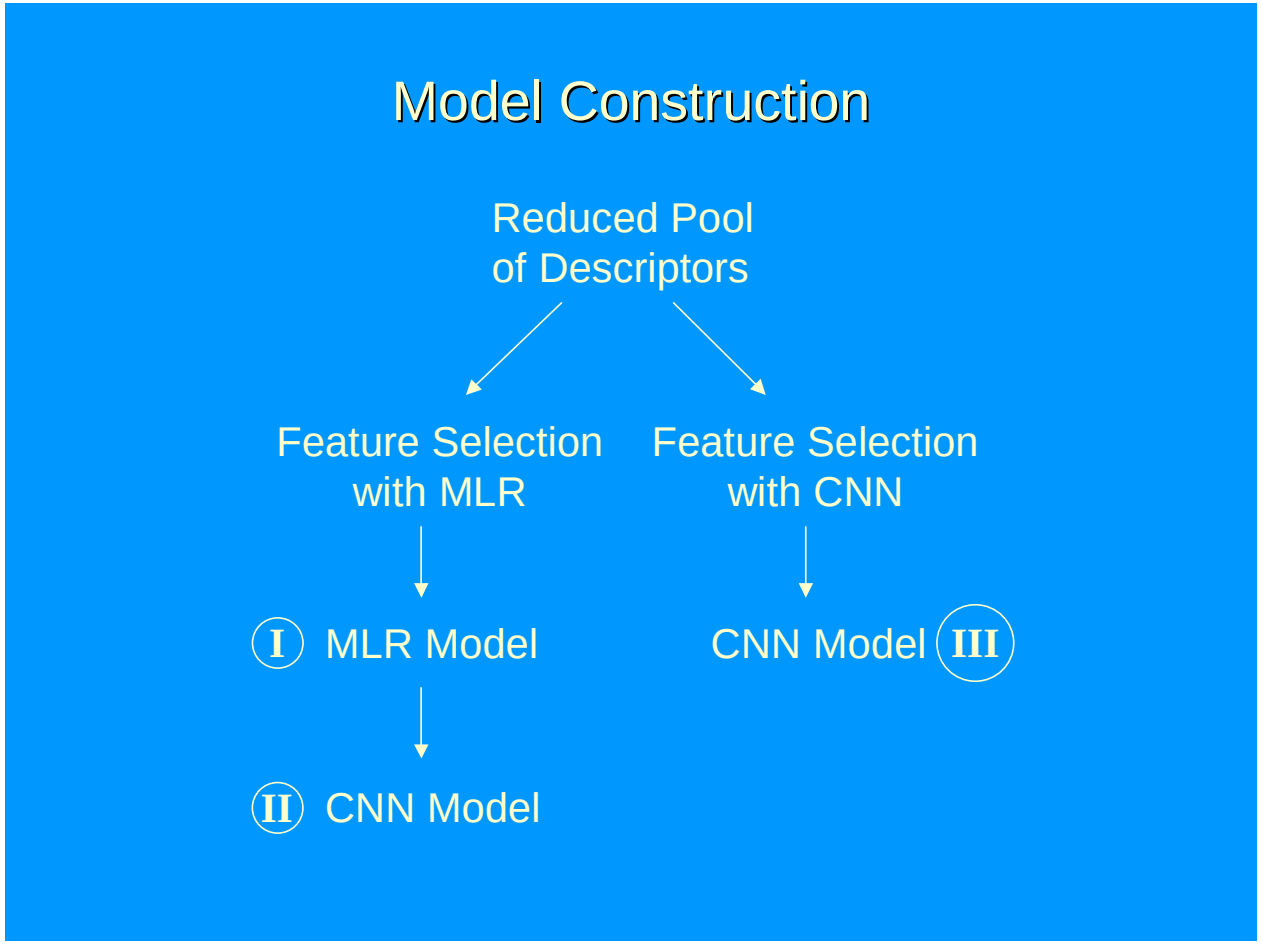
$$P = b_0 + b_1 X_1 + b_2 X_2 + \cdots + b_d X_d$$

Method: least squares error minimization

Evaluation: n, d, $R^2$ , s, rms error

Validation: plots of fitted vs. observed
residual plots
examine collinearities
internal validation
external validation

Multiple linear regression finds a correlation between molecular structures and their corresponding property through a linear combination of structural descriptors – as shown in this equation. Typically, the number of descriptors considered range between 3 to 12. During the subjective feature selection stage, the best subsets chosen for further analysis as potential models are initially based on the number of descriptors in the subsets, smaller subsets being preferred over larger ones, and the root mean square error of the training set compounds. Subsets satisfying these criteria are then evaluated by several statistical measures to assess the robustness and strength of each subset in regards to model coefficients, descriptor multicollinearities, and compound outliers. The final test of each linear regression model is the ability to generalize to external compounds contained in the prediction set. These models are termed Type I models.

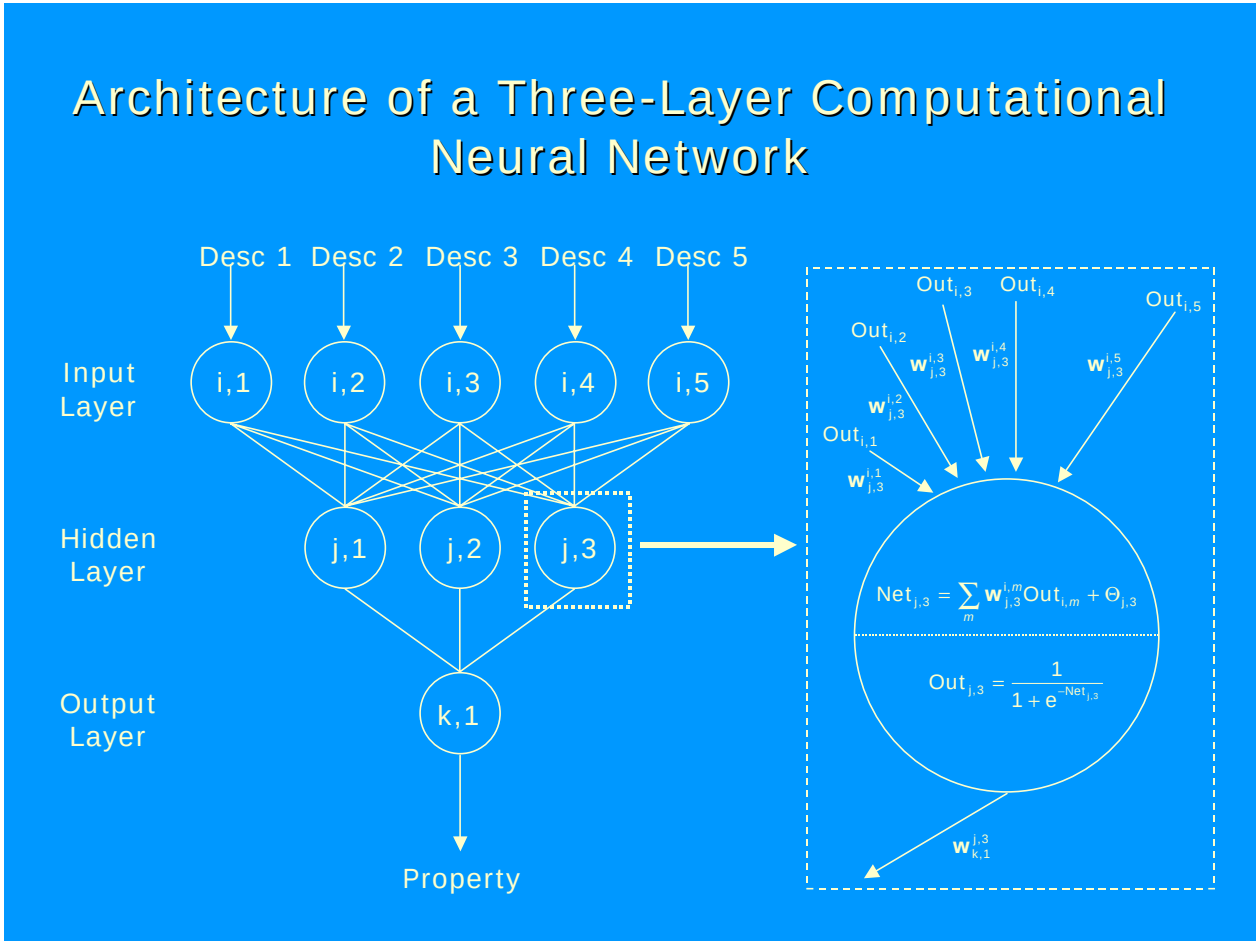



Models of three types are routinely generated for QSPR studies. A Type I model results when subsets of descriptors are chosen from the reduced descriptor pool by a genetic algorithm that uses a multiple linear regression analysis routine to assess model fitness. A Type I model is wholly linear. A Type II model results when the descriptors found to support a Type I model are then submitted to a computational neural network to develop a non-linear model. A Type II model is a hybrid of linear feature selection and non-linear model construction. A Type III model results when subsets of descriptors are chosen from the reduced descriptor pool by a genetic algorithm that uses a computational neural network routine to assess model fitness. A Type III model is wholly non-linear.

# Characteristics of the Three Model Types

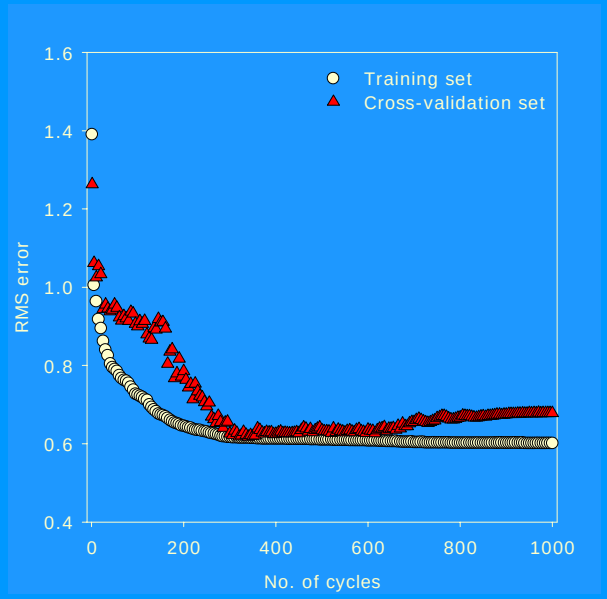
| Model Type | Feature Selection | Model |
|---|---|---|
| Type I | Linear | Linear |
| Type II | Linear | Non-linear |
| Type III | Non-linear | Non-linear |

Type I models are wholly linear. Type II models are hybrids where the descriptors were found with a linear process but the model was built with a non-linear CNN. Type III models are wholly non-linear. Type III models typically perform the best on QSPR problems.




A three-layer, fully-connected, feed-forward computational neural network (CNN) is shown. Its purpose is to build an association between the set of descriptors that encode each compound and the compound's property or activity value. In the illustration, five descriptors form the representation of a compound. These five values are fed to the first level neurons, where scaling occurs. The output from each of the five first-level neurons is passed to each of the three hidden-layer neurons. Each of the 15 connections has an adjustable weight. Each hidden-layer neuron sums its inputs and then puts this sum through a sigmoid forcing function that imparts non-linearity to the system. The outputs from the three hidden-layer neurons are passed, with weighting, to the output neuron in the third layer, which sums and transforms the values and produces the property value estimate. The learning of the neural network is done by adjusting the weights of the connections within the network. This is done by feedback whereby errors committed by the network are minimized by adjusting the weights of the network with back propagation. Second-order training using quasi-Newton methods is even faster and produces better networks at less computational cost.

# Behavior of Training and Cross-validation set RMS Errors

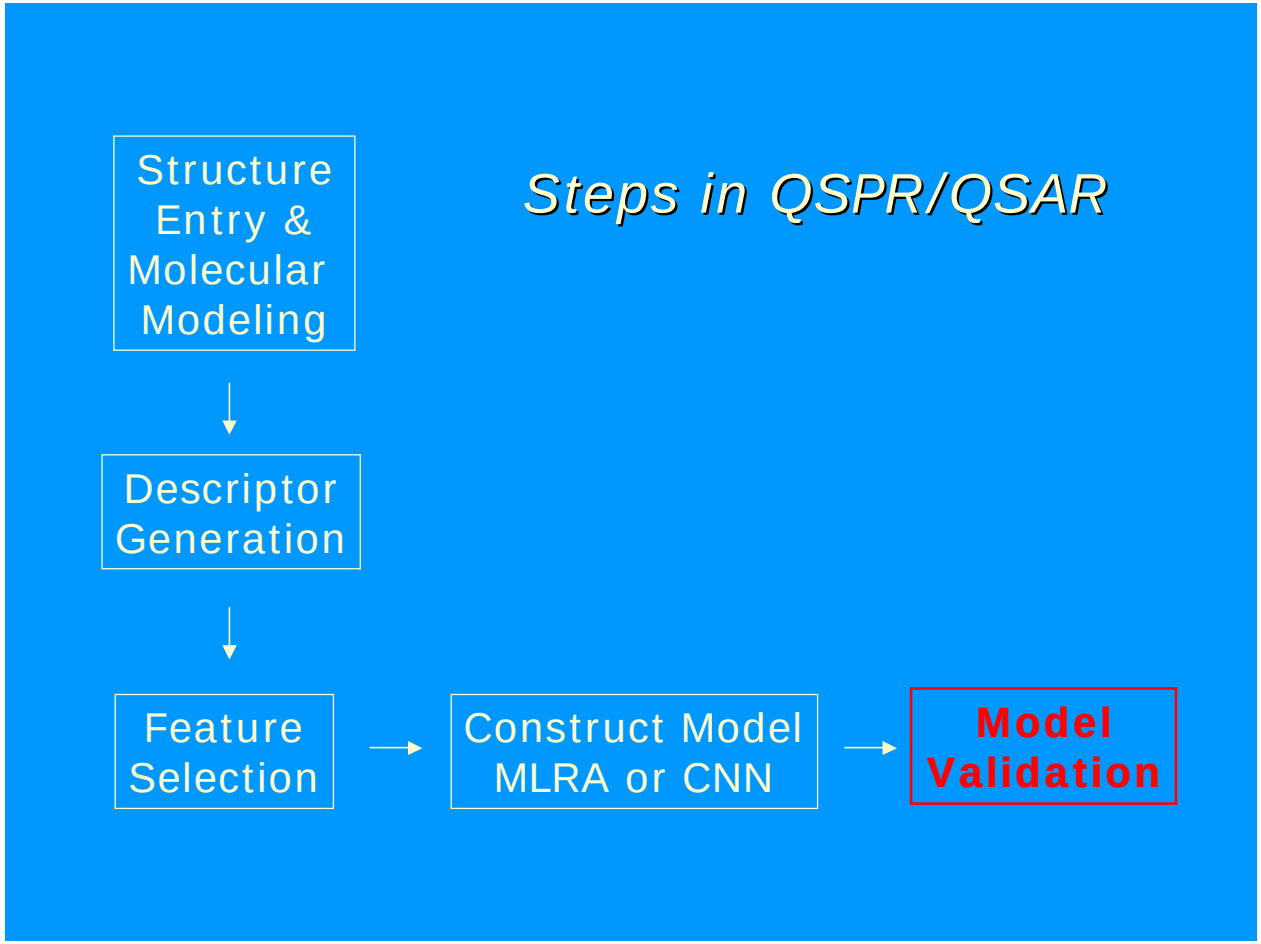



In training CNNs, it is possible to over-train the network. Training of a CNN occurs in two stages: first, the network learns the general features of the training set, which ought to be very useful in predicting the property for new compounds; secondly, at a later stage of training the network learns the individual characteristics of the individual compounds of the training set, that is, memorizes the training set members, which is not useful in predicting the property for new compounds. This is called over-training. One effective way to avoid over-training is to use a cross-validation set in addition to the training set of compounds. Periodically during training, the property is estimated for the members of the cvset, and a running error is kept. When this cvset error reaches a minimum and starts to rise, the best stopping place for training has been found.

# CNN Considerations

- Split data into tset, cvset, pset
- Cross-validation to prevent overtraining
- Ratio of observations in tset to total adjustable parameters should be >2
- Determine number of hidden layer neurons empirically -- start small
- Cross-validation set error should vary smoothly
- Assign starting weights and biases randomly

There are a number of considerations to take into account when using CNNs for QSAR models. Proper use of a cross-validation set can prevent over-training. The architecture of the CNN should be such that the number of adjustable parameters should be less than one-half the number of compounds in the training set. The number of neurons in the hidden layer must be determined empirically; usually this means starting with a small number and working up.




The fifth major step in a QSPR/QSAR study is the validation of the model by predicting the property or activity of compounds in the external prediction set. The errors produced in the predictions should be comparable to those achieved for the training set and cross-validation set.

# Model Validation

Two Parts:

- Prediction of new compounds in external prediction set
- Testing for chance correlation
  Could the results obtained have been due to chance?

Validation of a model involves two steps: (1) demonstration of predictive ability by predicting the property of interest for compounds not used during the generation of the model, that is, an external prediction set of compounds, and (2) Monte Carlo randomization testing to look for chance correlations.

# Testing for Chance Correlations

- Use same set of independent variables
- Randomize the dependent variable

| | | | | |
|---|---|---|---|---|
| Compound 1 | Value 1 | ⇒ | Compound 1 | Value 5 |
| Compound 2 | Value 2 | | Compound 2 | Value 4 |
| Compound 3 | Value 3 | | Compound 3 | Value 2 |
| Compound 4 | Value 4 | | Compound 4 | Value 1 |
| Compound 5 | Value 5 | | Compound 5 | Value 3 |

- Build best models with MLRA or CNNs
- Compare results to real models

Part of validating the models is to check for the possibility of chance correlations. This can be done by performing the entire sequence of computations over but with the dependent variables scrambled. This scrambling destroys any relationship between the descriptors and the dependent variable. No model that exceeds chance performance should be found. The results obtained are compared to the results achieved with the actual computations to demonstrate that the actual results were achieved by finding relationships rather than by finding chance correlations.

## Recent QSPR & QSAR Studies

| | |
|---|---|
| 1999 | Liquid crystal clearing temperature |
| | $CH_3$ and OH radical addition rate |
| | Organopesticide toxicity |
| | Fathead minnow acute toxicity |
| 2000 | $C_{60}$ solubility |
| | Vapor pressure |
| | Antiporter inhibition |
| | ACAT inhibition |
| | Multidrug resistance reversal agents |
| 2001 | Aqueous solubility |
| | Organic solvent properties |
| | *Tetrahymena* toxicity |
| | Peptide ion mobility |
| | Glass transition temperature |
| | Carbonic anhydrase inhibitors |
| | Selective COX-2 enzyme inhibitors |
| | 5-α-reductase inhibitors |

The methods described here have been applied to a number of QSPR and QSAR studies in the Jurs group. Listed here are a few of the specific topics we have pursued recently.

# Conclusions

- Develop quantitative predictive models using regression analysis or neural networks with errors comparable to experiment
- Encode structures successfully with calculated structural descriptors
- Develop predictive ability for properties of new compounds
- Focus on important structural features

In summary, QSPR/QSAR methods can be used to build models that can predict properties or activities for organic compounds. To do so requires an effective way to encode the structures with calculated molecular structure descriptors. Once the models have been generated, they have predictive ability for new compounds not in the training set. The descriptors that are incorporated into the models provide an opportunity to focus on the features of the compounds that account for the property or activity of interest.