Genetic Programming to Combine Machine Learning Classifiers in Drug Discovery

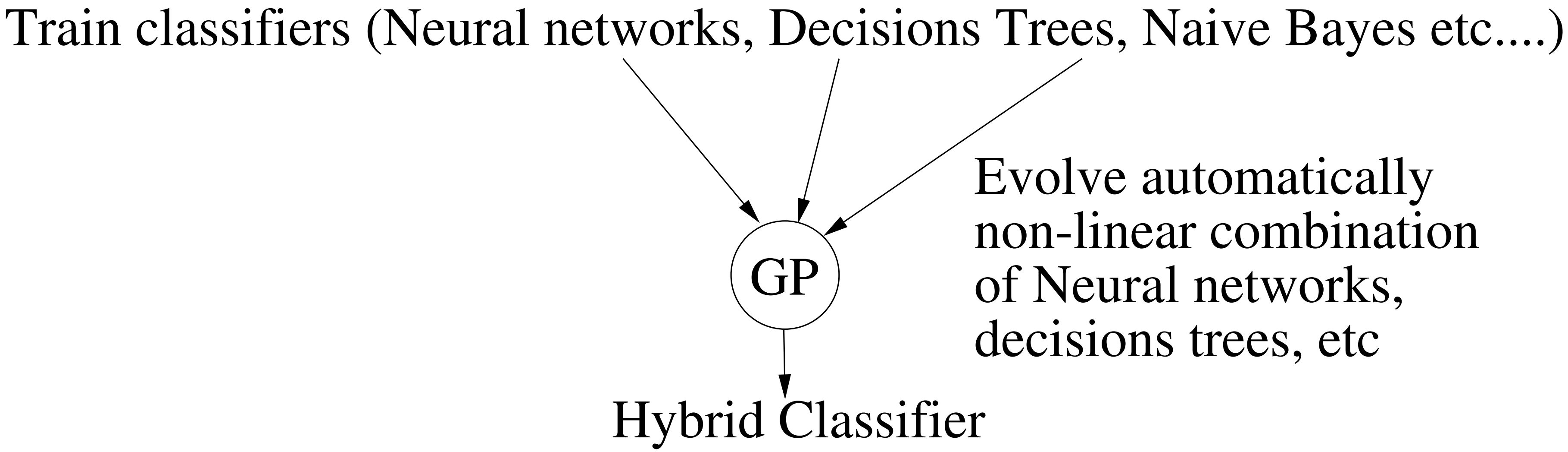
W. B. Langdon

Computer Science, University College, London
W.Langdon@cs.ucl.ac.uk
http://www.cs.ucl.ac.uk/staff/W.Langdon

Overview

Genetic programming can form non-linear combinations of diverse classifiers to yield a better ensemble classifier.

It has been demonstrated on artificial and real-world benchmarks. Combining classifier of the same type and classifiers of different types (e.g. C4.5 with neural networks with Naive Bayes). Classifiers trained on the same dataset, and classifiers train on different data.


Train classifiers (Neural networks, Decisions Trees, Naive Bayes etc....)
GP
Evolve automatically non-linear combination of Neural networks, decisions trees, etc
Hybrid Classifier

Summary

Application of *identical* GP technique for drug discovery

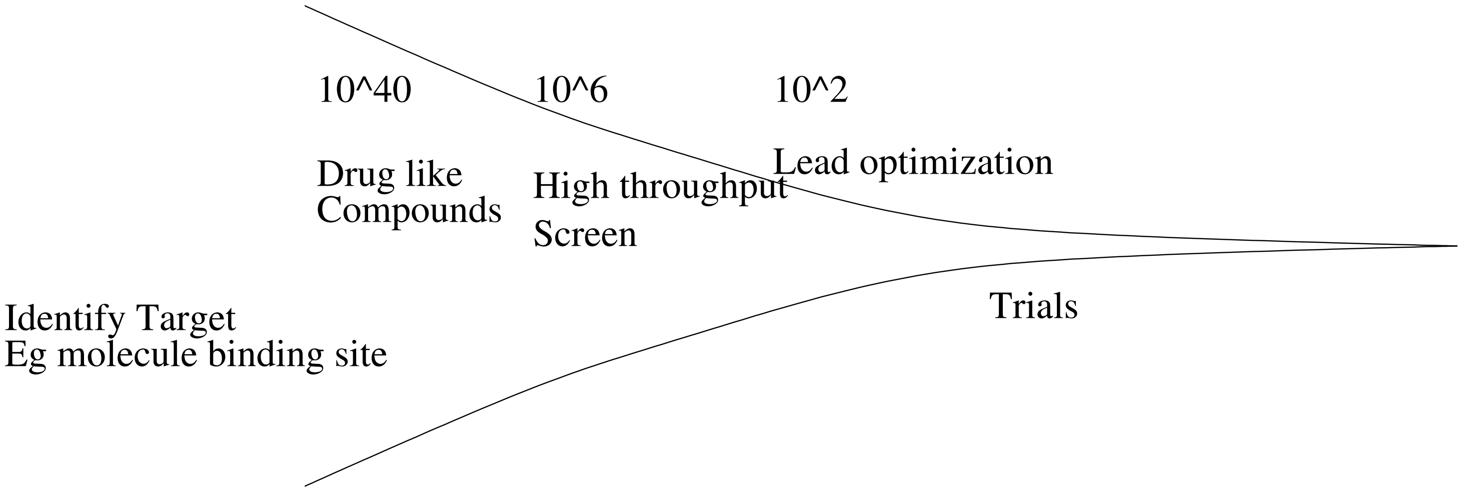
- Drug discovery. P450 in pilot study.
- Clementine used to train 75 diverse neural network classifiers
- Evolving a hybrid classifier:

  Representing lower level classifiers

  Multi-tree

  Fitness = Area Under ROC
- Results on holdout and extrapolation sets.

Discovery of a Drug to Treat a Disease



Discovery of a Drug to Treat a Disease

Medical research on a disease may discover the disease's life cycle.

There may be a critical point in the life cycle that might be disrupted by a drug. E.g. a point where a compound might bind and prevent the diseases action.

Search space of potential drugs is huge. Estimated 10^{40} drug like compounds.

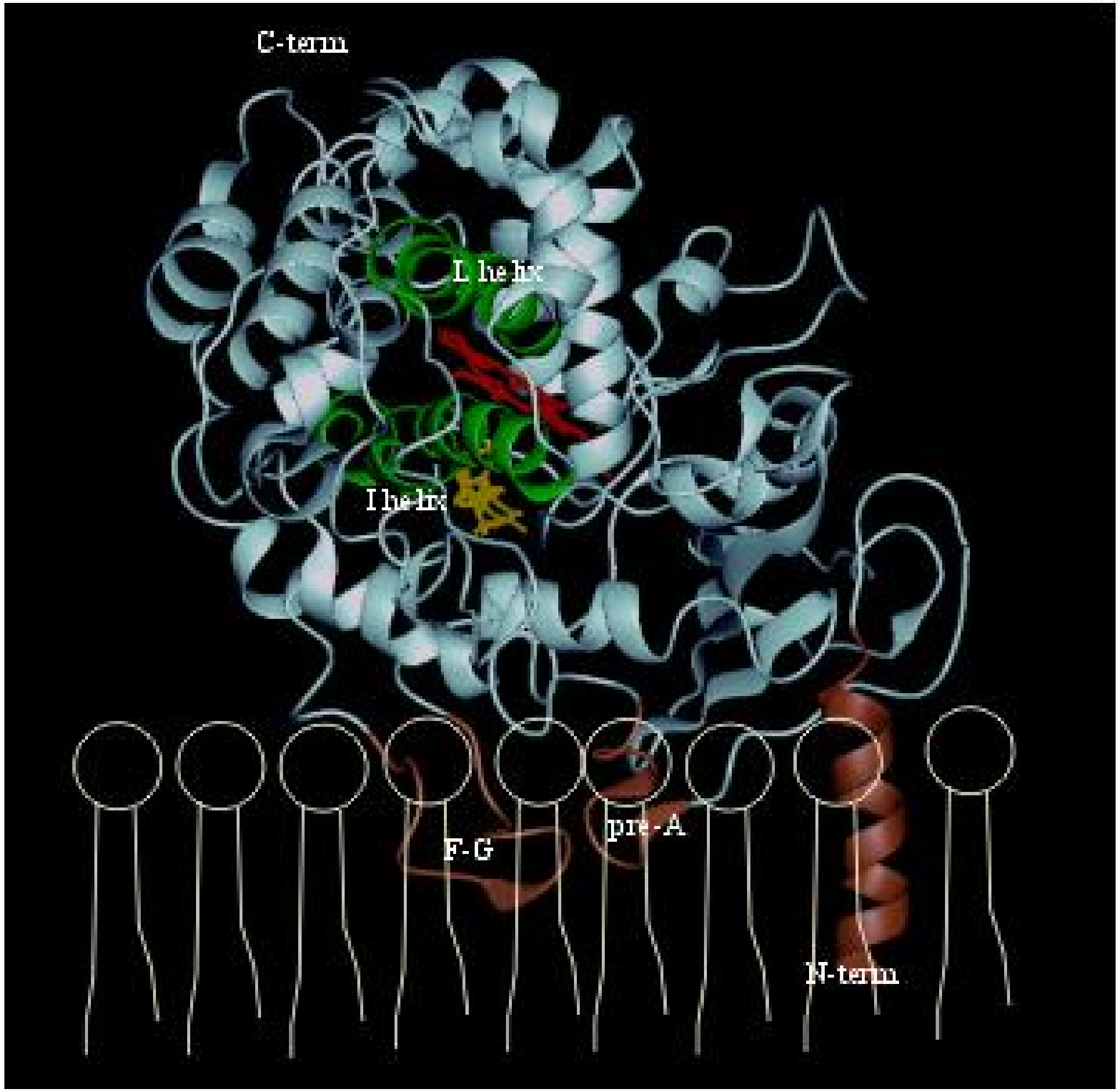
High throughput screening (HTS) in the region of 10^{6} can be measured.

Computer models are used at many stages of the discovery "funnel".

Computer models can extend scope of existing measurements. Can screen "virtual" chemicals (i.e. chemicals that could be manufactured, if the computer suggested they might be interesting).

Model of rat P450 2B1,

showing mode of membrane attachment



Why P450?

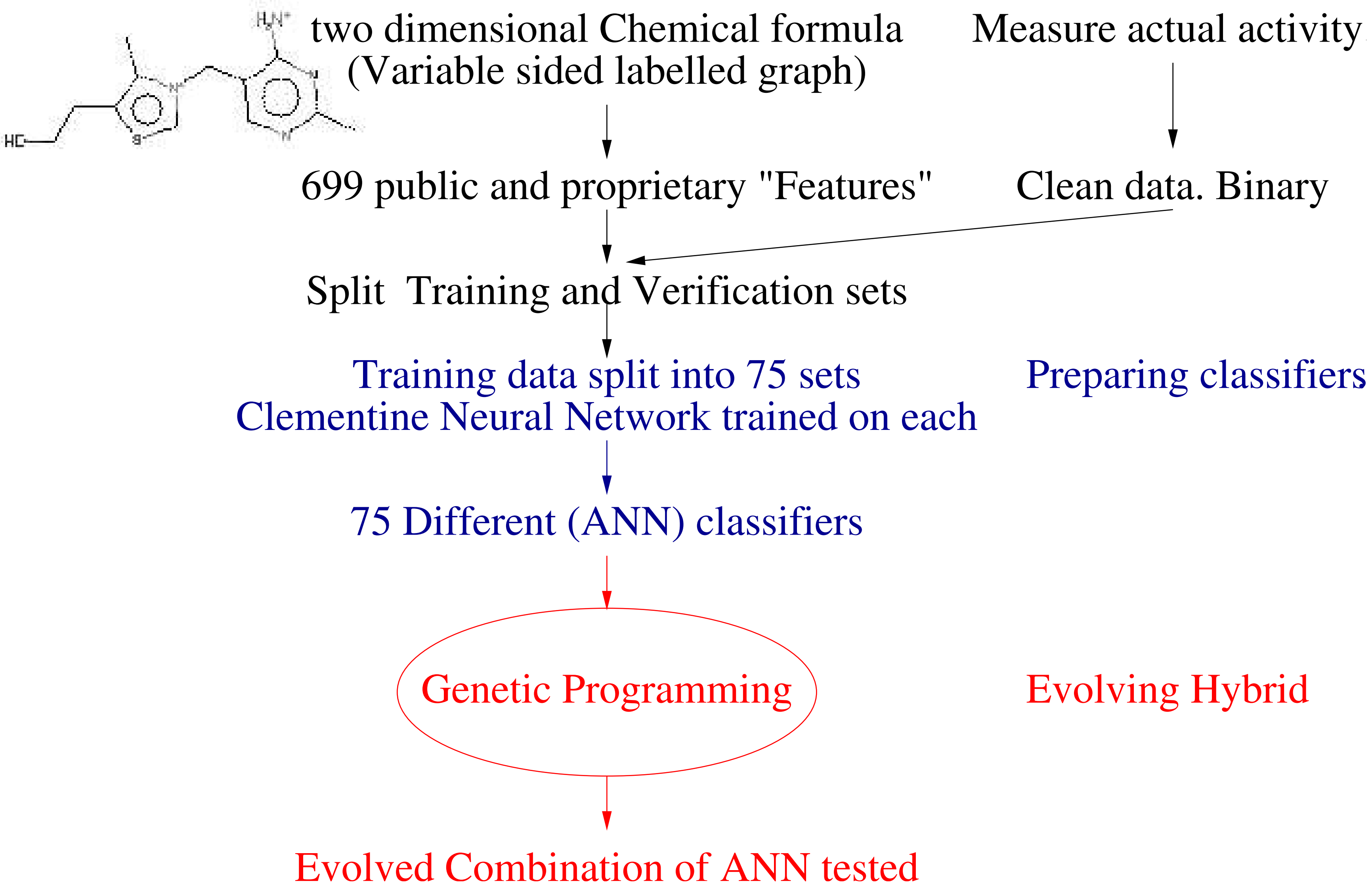
We use P450 mainly as a model of real biological targets (also cannot use real targets commercial reasons).

P450 is moderately complex enzyme with many potential sites where drugs might bind to it and disrupt its catalytic action.

Measurements of P450 inhibition will be required at some point. I.e. some commercial advantages in avoiding potential drug failing P450 check later in discovery process.

Availability of High Throughput Screening (HTS) data

Evolving a Hybrid P450 Activity Predictor



two dimensional Chemical formula
(Variable sided labelled graph)

Measure actual activity

699 public and proprietary "Features"

Clean data. Binary

Split Training and Verification sets

Training data split into 75 sets
Clementine Neural Network trained on each

Preparing classifiers

75 Different (ANN) classifiers

Genetic Programming

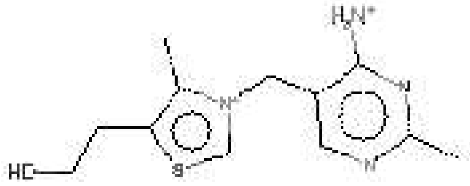
Evolving Hybrid

Evolved Combination of ANN tested

Nature of Modelling Problem

At the High throughput screening stage, the complete three dimensional structure of most compounds is not known.

Instead we use the chemical formula data



Example Compound (vitamin B1)

From this various physical properties can be estimated.
E.g. electrical charge imbalance, hydrophobicity and Hydrophilicity, presence of groups.

GSK calculated 699 “features” for each compound. (Mixture of public domain and GSK proprietary features).

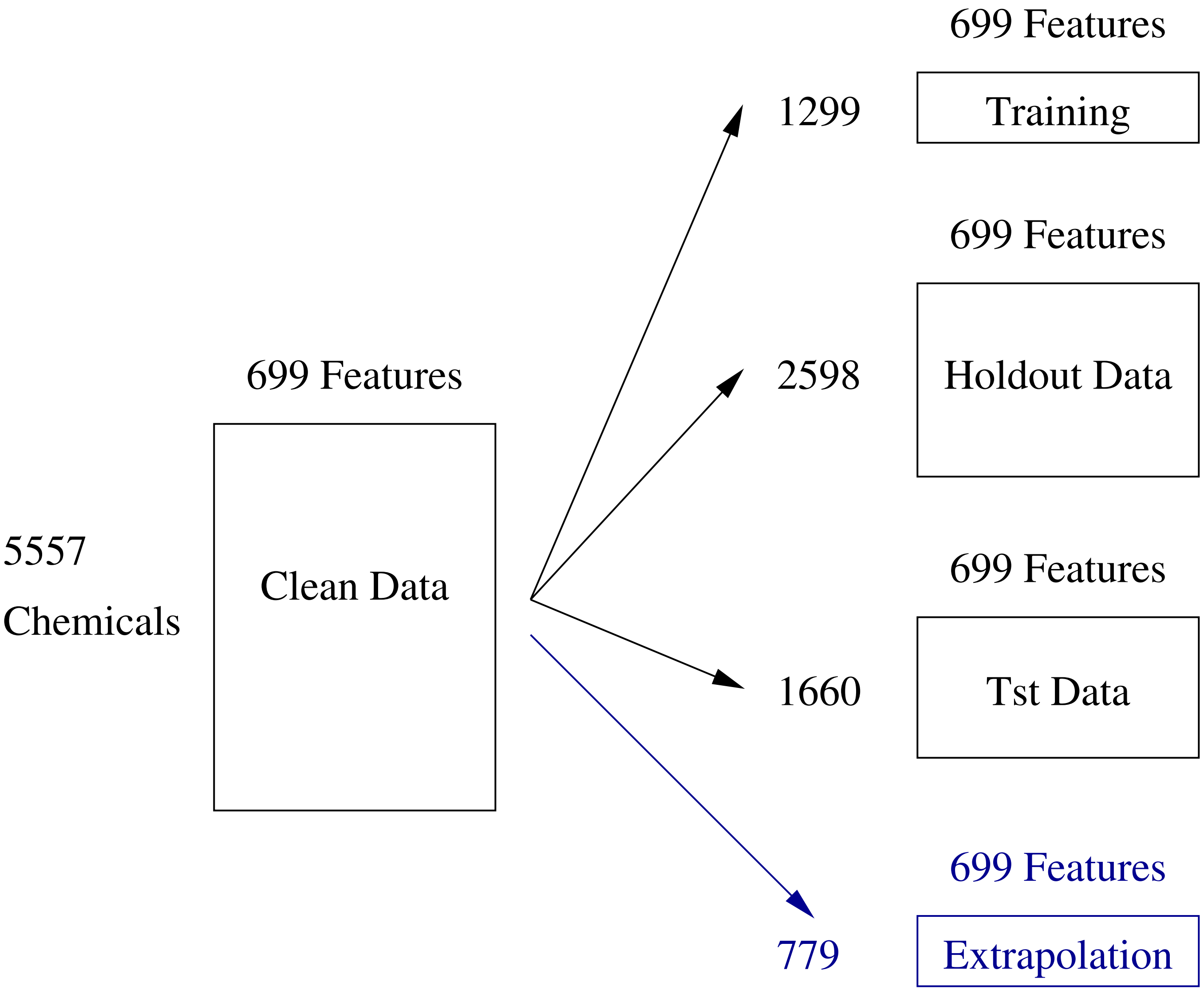
The inhibitory effect of chemicals on a P450 enzyme was measured in two triplicated High Throughput Screening (HTS) runs $(2 \times 3 = 6)$.

HTS measure effects in tiny wells, not in living tissue. HTS is essentially a crude measurement of very small quantity of materials. Both mean its results cannot be guaranteed in patients.

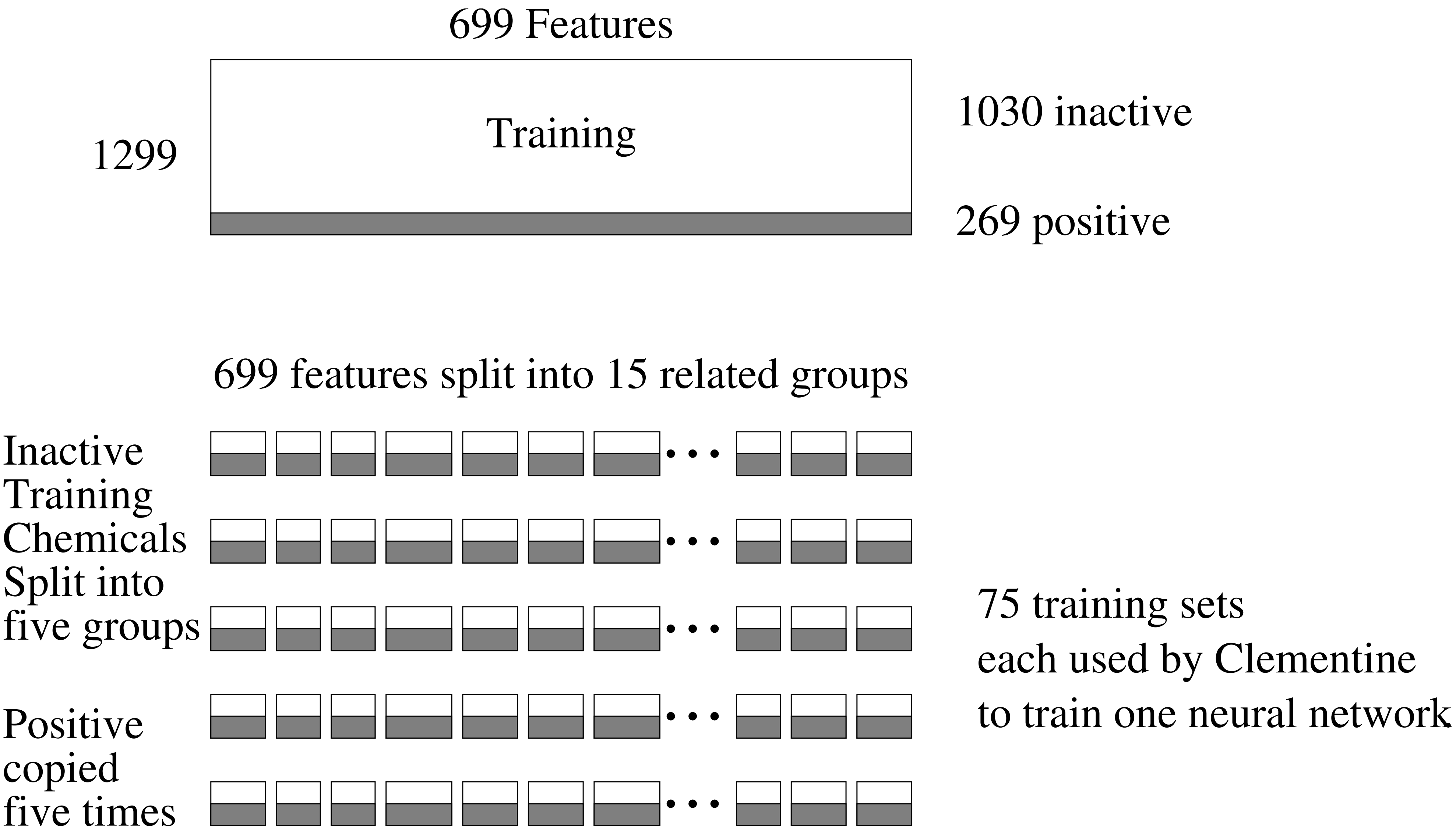
Chemicals with inconsistent readings (more than 15% variation) excluded. Average of 6 readings compared to a threshold to yield binary decisions: inhibits or not.
Boolean activity = (mean 6 readings $>$ fixed value)?

5557 data with 699 attributes and binary classification.

P450 Data Sets



P450 Clementine Training Sets



Supervised Learning. Clementine trains 75 ANN

1299 compounds randomly selected to be training set.

699 attributes split into 15 functionally related groups.

Only about 20% of compounds have an inhibitory effect.
These “positives” separated from the others.

The “negatives” randomly split into five groups.

5 “balanced” groups produced by combining the same positives with each of the five groups of negatives.

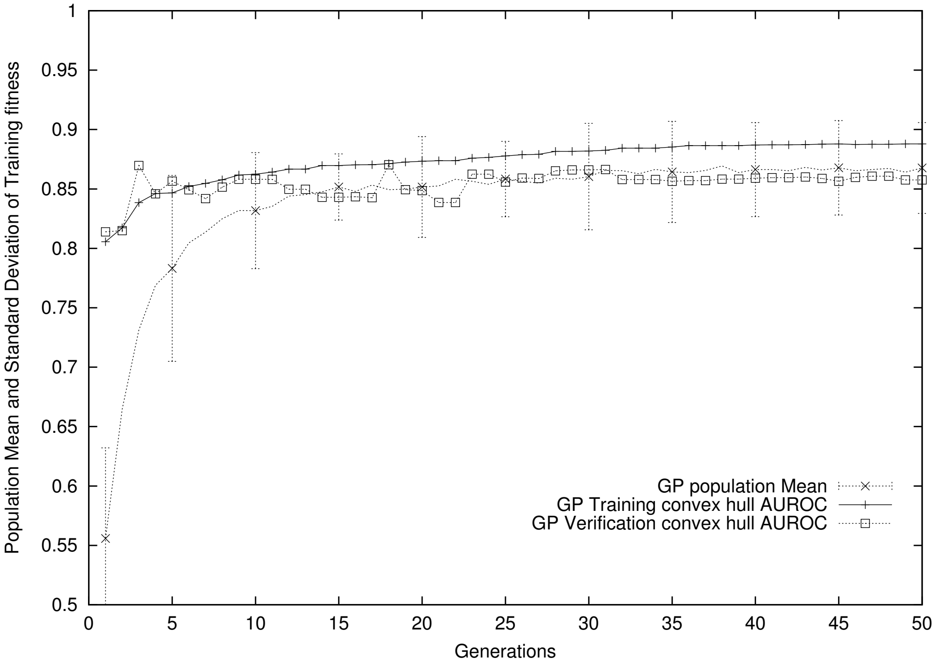
$15 \times 5 = 75$ feed forward neural networks trained by Clementine.

Composite of 75 ANN: Genetic Programming

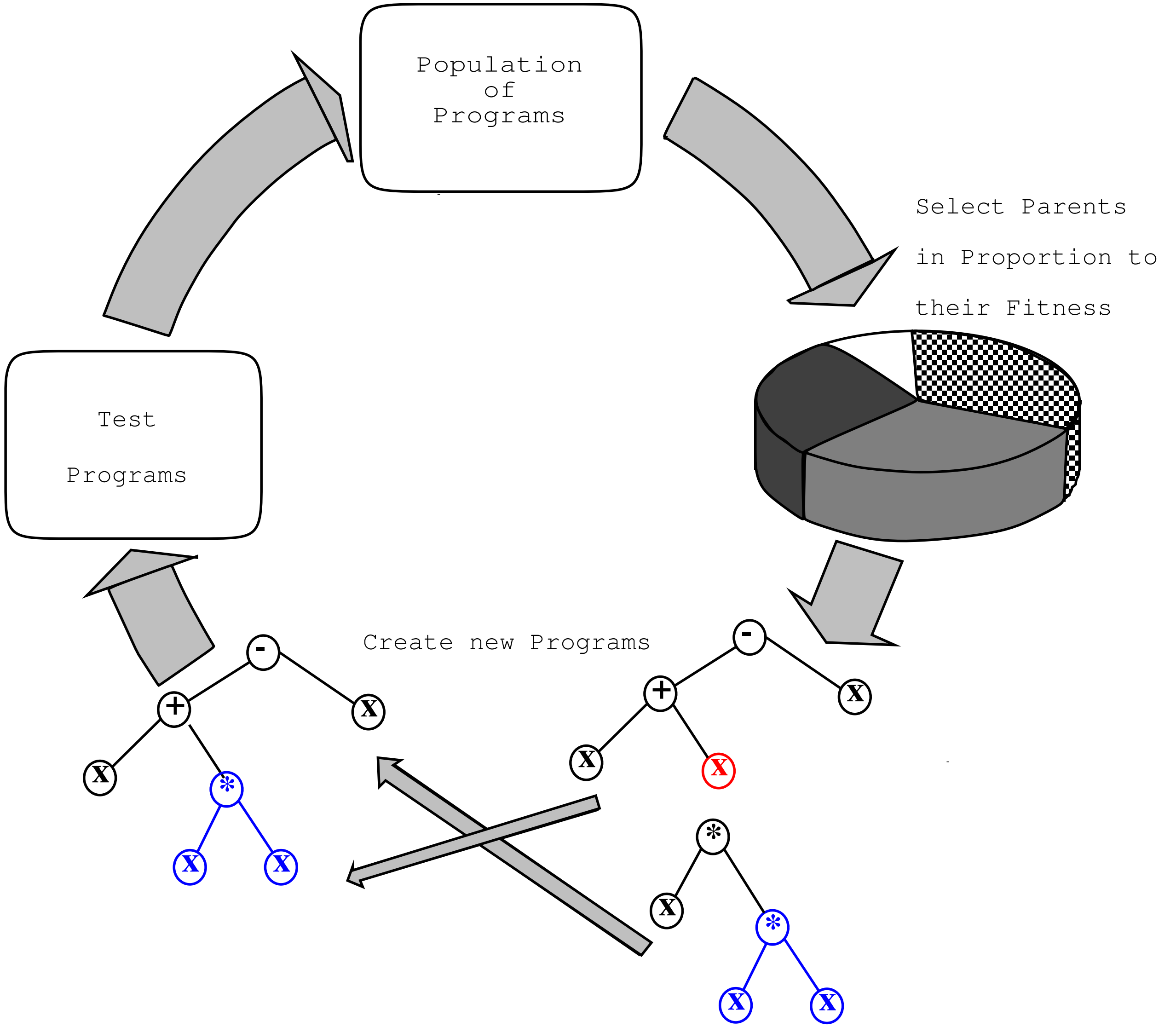
Data randomly split into training (866 compounds) and verification (433).

Clementine Neural Networks used a functions within genetic programming trees.

Performance of individual trees within GP population given by the area under the convex hull of their Receiver Operating Characteristics (Wilcox' statistic).



Evolutionary Cycle



Population of Programs

Select Parents in Proportion to their Fitness

Create new Programs

Test Programs

Evolutionary Cycle: Evolving Hybrid Classifiers

1. Classifier(s) trained in usual way

2. Classifier packaged. Positive case ⇒ positive number.
   Negative case ⇒ negative number.
   More confident bigger magnitude.

   Do not need to have direct access to problem data.

   Arithmetic operations, IF, Max, Min, constants...

3. Random initial combinations (generation 0)

4. Fitness used to select (from generation n) better combinations

5. New generation of classifier combinations (generation $n+1$)

6. Iterate **4.–5.**

7. Composite classifier demonstrated on holdout set.

GP Representation: Functions and Arguments

- Function set: $+ - \times /$ if Min Max Frac Int and **classifiers**

  Within GP, all classifiers are 1 input function (threshold). Implicitly uses current test case.

  Threshold argument allows evolved combination to bias answer given by classifier.

  Function returns its classification of test case (positive or negative) and its “confidence” (near or far from zero).

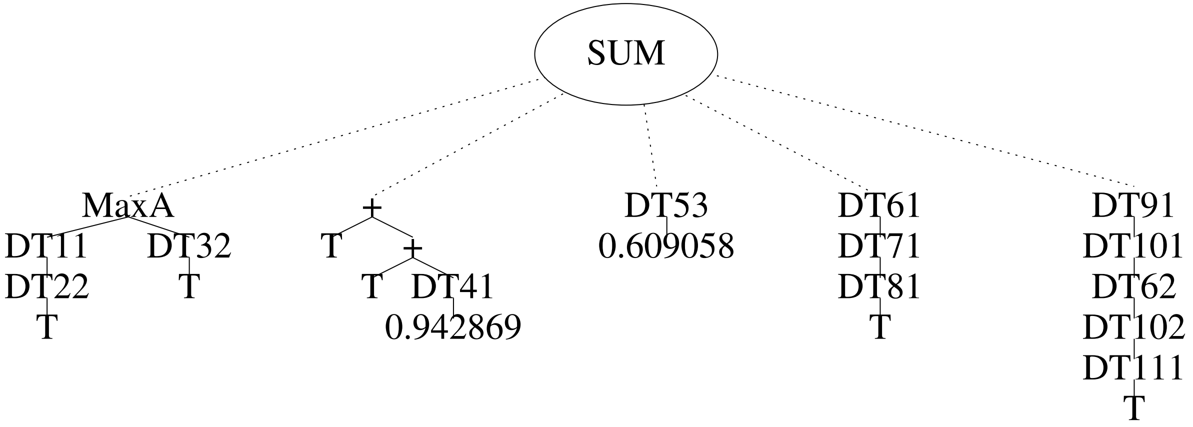
- Terminal set: constants and **threshold T**

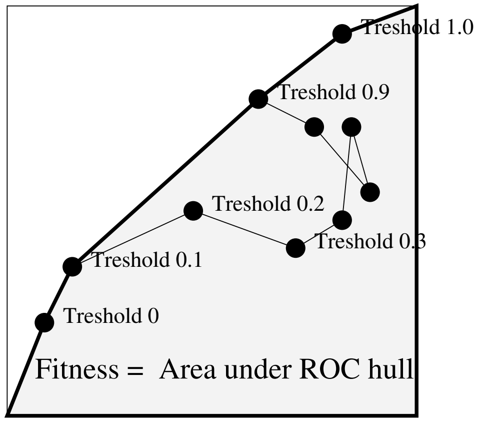
  T allows us to tune response of evolved classifier. I.e. move it up and down its ROC sensitivity curve.

GP Representation: Five Trees

5 Trees: sum value returned by each tree (i.e. weighted vote)

Sum $\geq 0 \Rightarrow$ positive class





Fitness = area under convex hull of 13 points

Population 500. 50 generations Size fair crossover & mutations

Genetic Programming Data Fusion Parameters

Objective:	Evolve a Non-Linear Combination of Neural Networks with Maximum ROC Convex Hull Area on P450
Function set:	INT FRAC Max Min MaxA MinA MUL ADD DIV SUB IFLTE
	75 ANN trained on P450 data
Terminal set:	T 0 0.5 1 plus 100 unique random constants $-1 \ldots 1$
Fitness:	Area under convex hull of 11 ROC points $(179 + 697 = 866$ chemicals)
Selection:	generational (non elitist), tournament size 7
Wrapper:	$\geq 0 \Rightarrow$ active, inactive otherwise
Pop Size:	500
No size or depth limits	
Initial pop:	ramped half-and-half (5:8) (half terminals are constants)
Parameters:	50% size fair crossover (90% must be on internal nodes) 50% mutation (point 22.5%, constants 22.5%, shrink 2.5% subtree 2.5%)
Termination:	generation 50

What is Best Classifier?

- All classifiers make a tradeoff between catching all positive examples and raising false alarms.

- The Receiver Operating Characteristics (ROC) of a classifier captures this trade off.

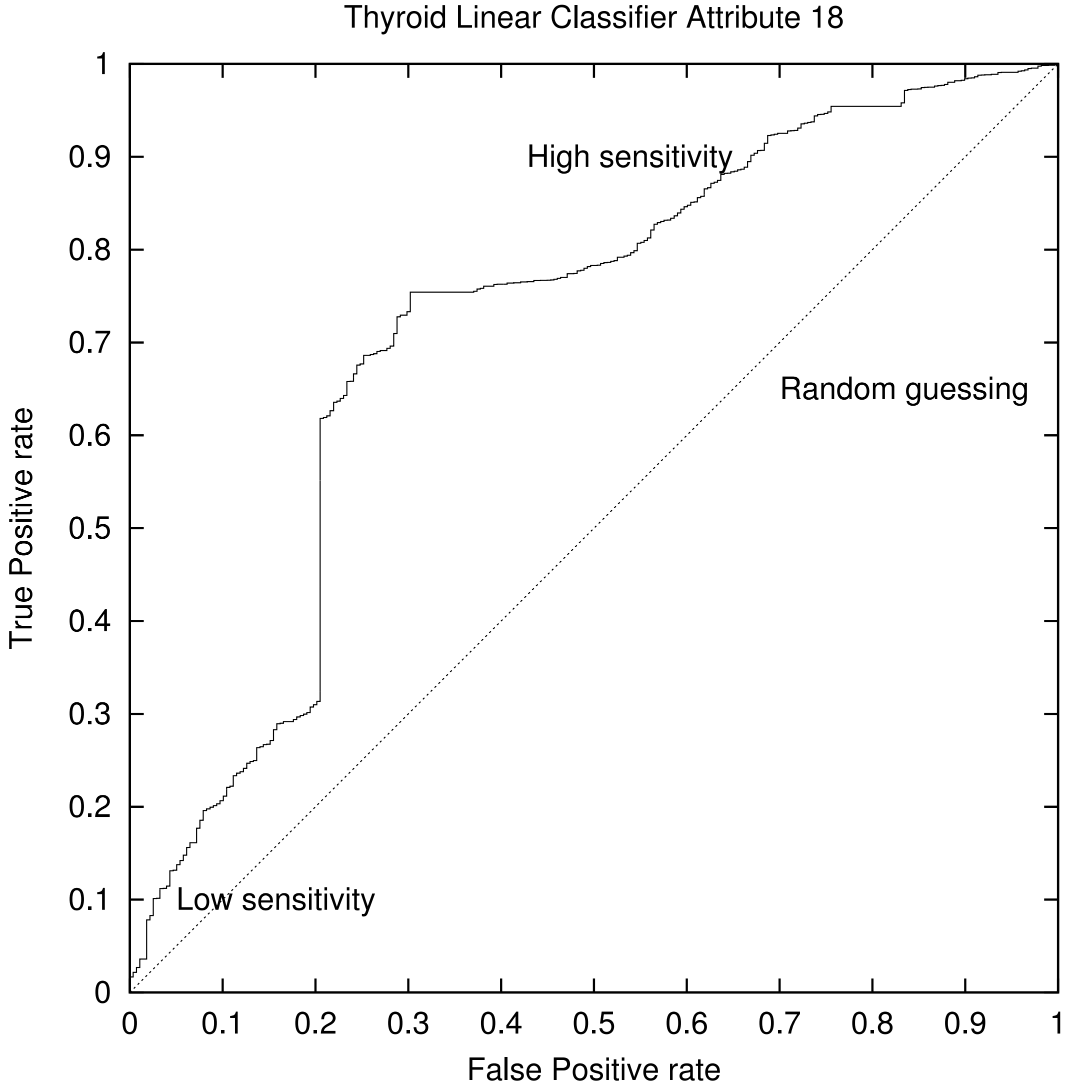
- ROC plot False Positive (false alarms) rate v. True Positives.

- ROC useful where costs FP or TP are unknown, variable or hard to determine.
  Cost of FP known (2^{nd} test?), cost missed positive unknown.

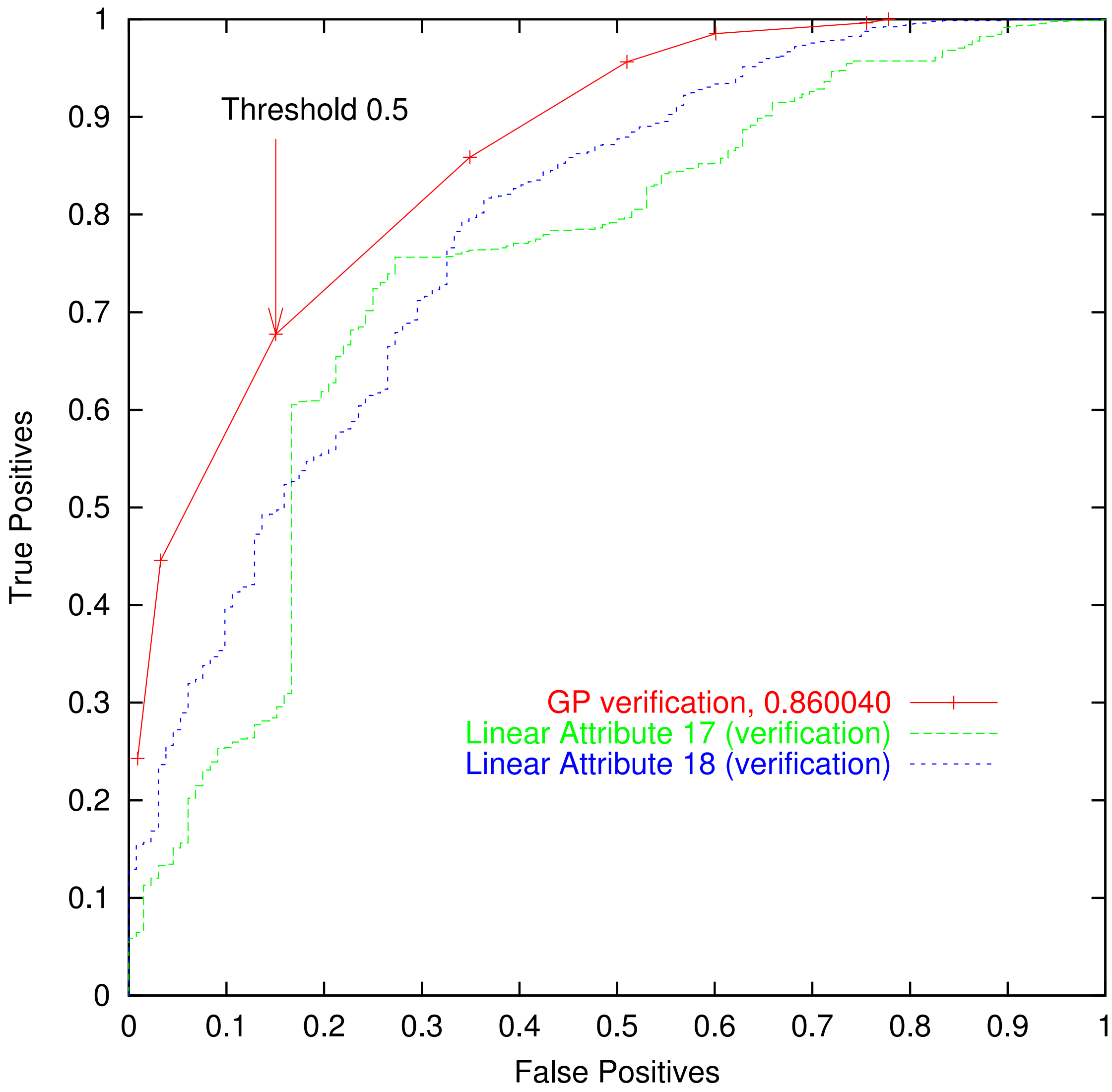
- Area under ROC curve gives overall measure of performance (Area under ROC = Wilcoxon statistic)

  Area used as objective measure for GP

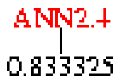
Receiver Operating Characteristics – Thyroid



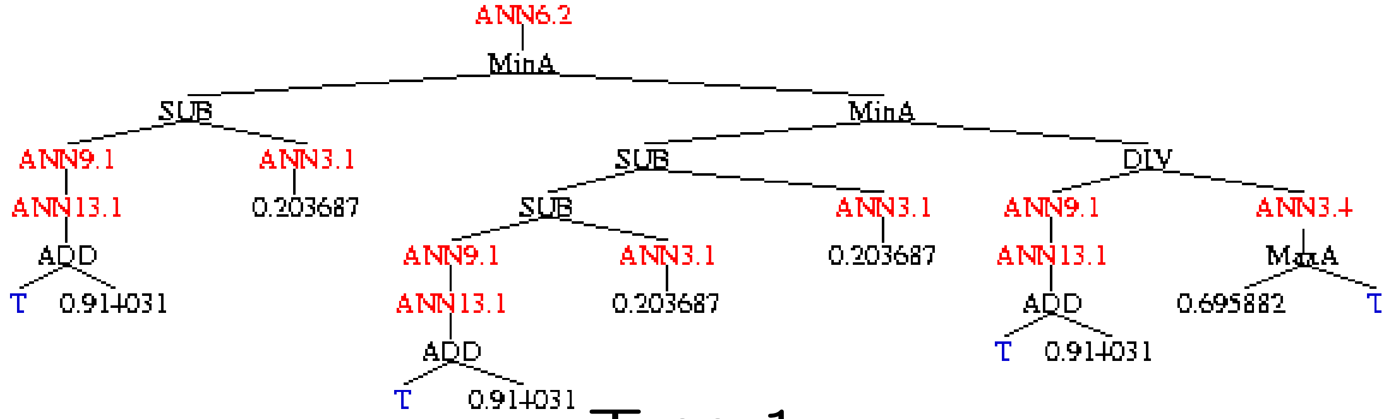
Thyroid Receiver Operating Characteristics



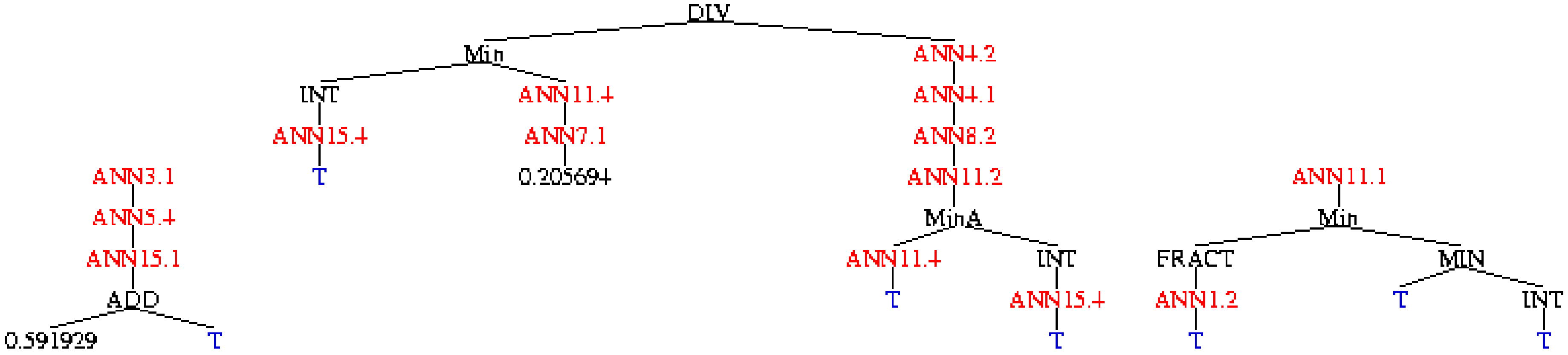
Example Genetic Programming P450 Classifier



Tree 0



Tree 1

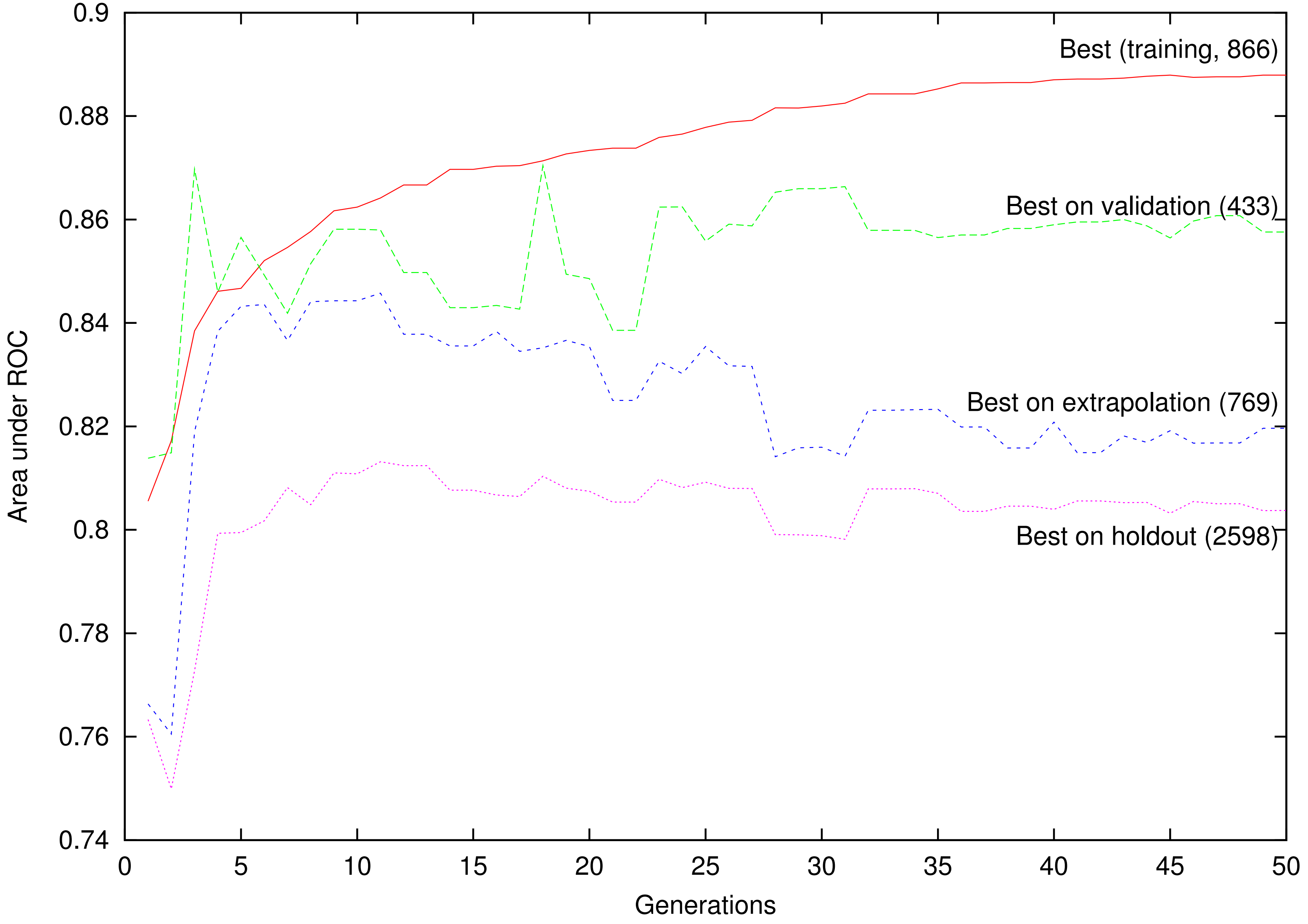


Tree 2

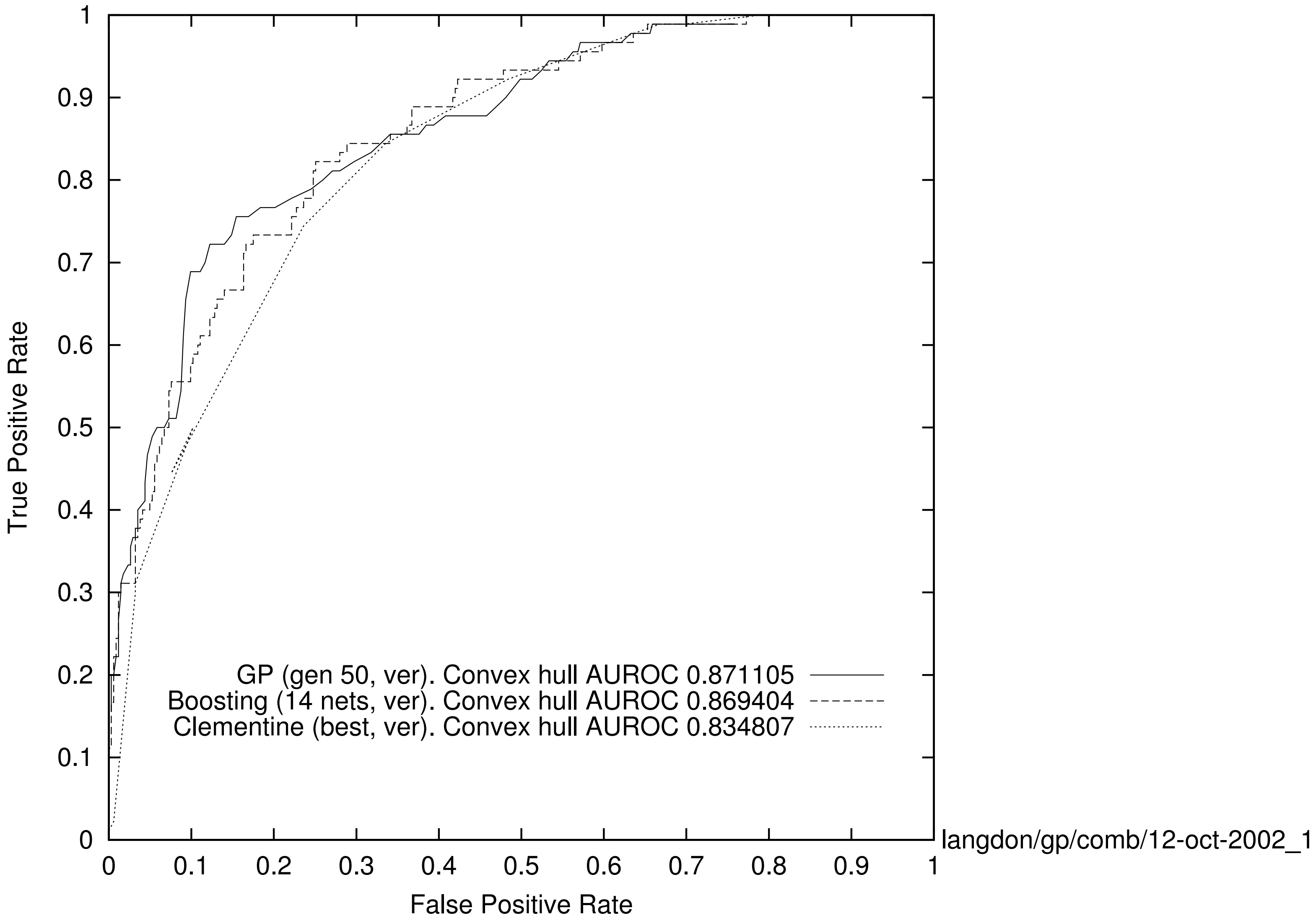
Tree 3

Tree 4

Evolution of Fitness

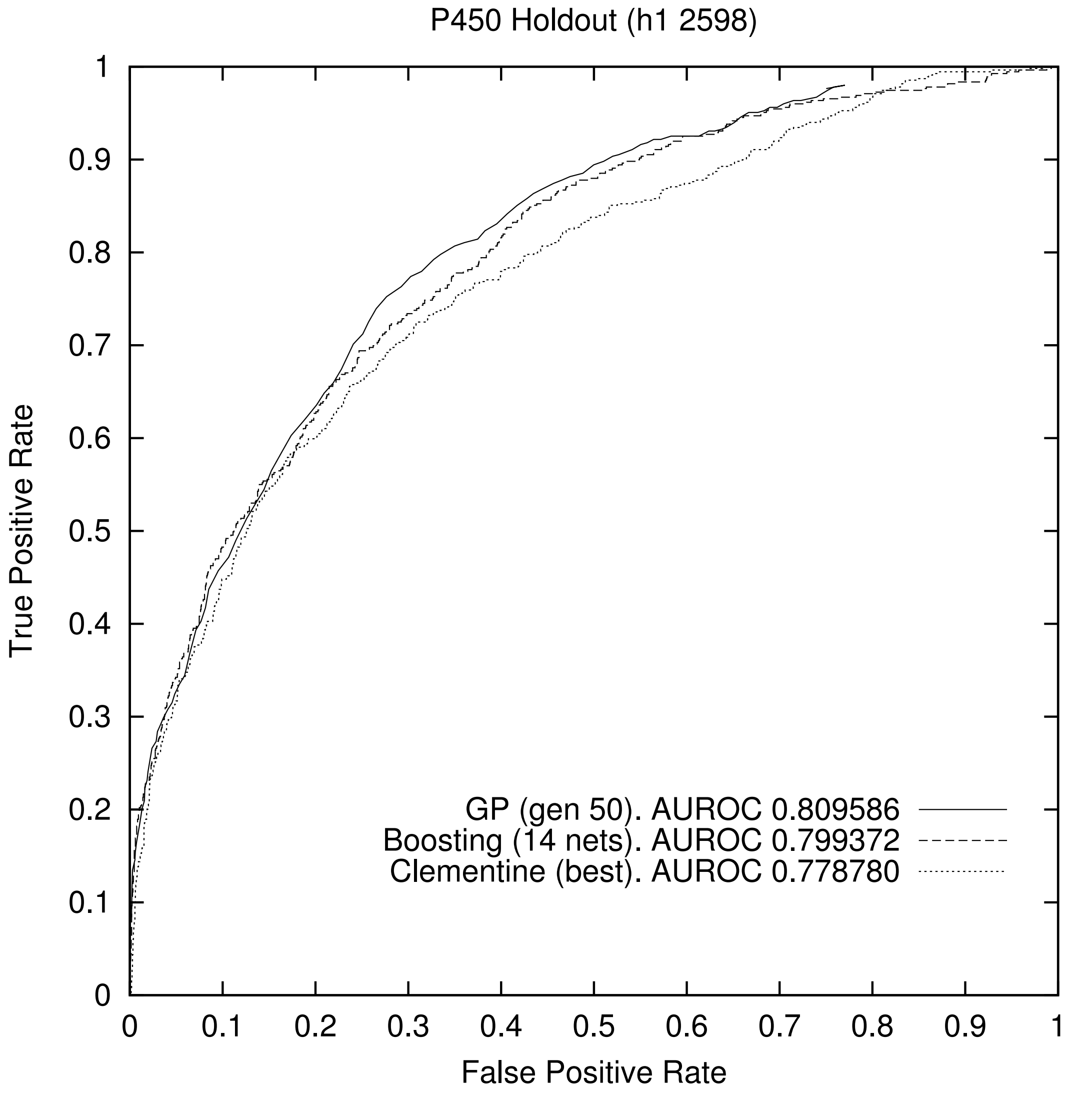


P450 Verification (433)



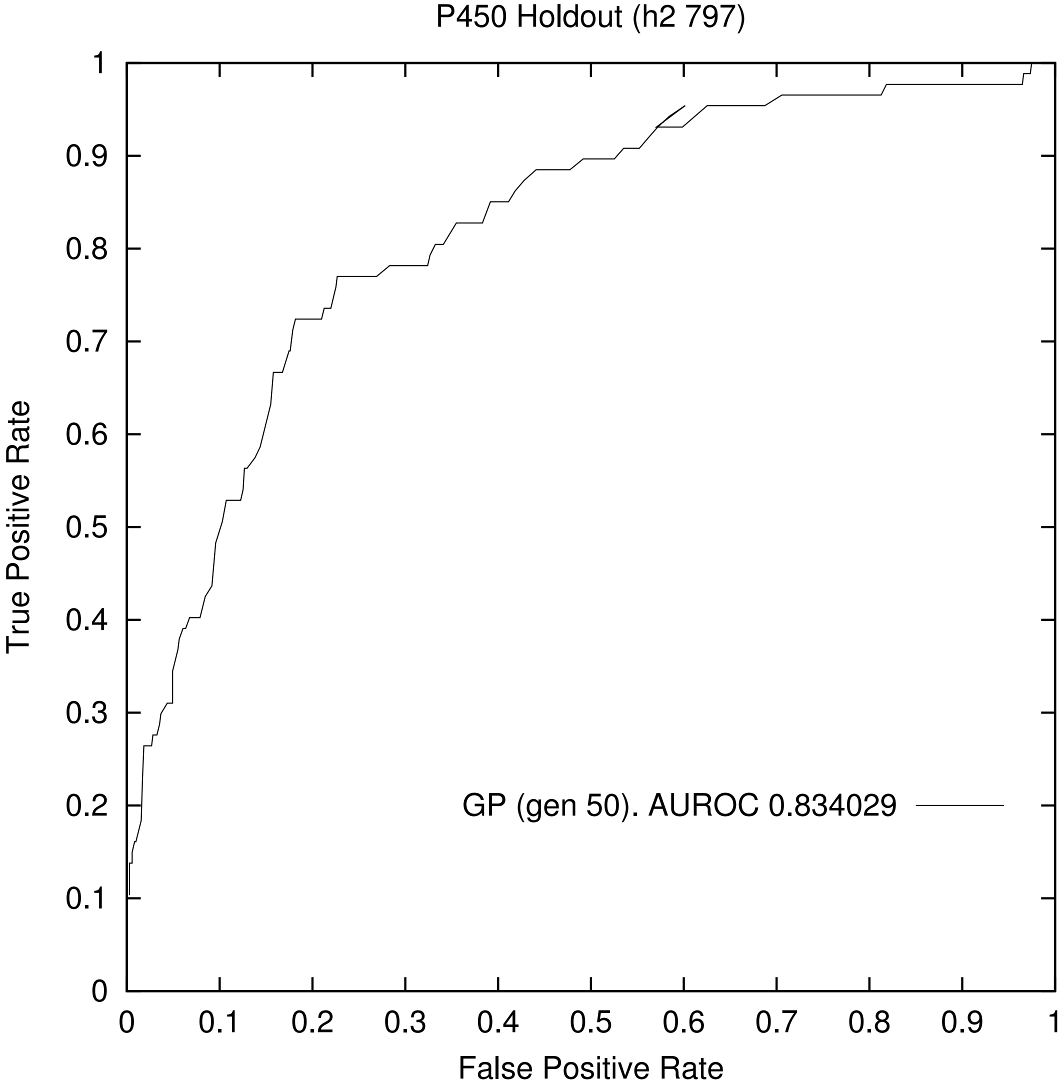
Receiver Operating Characteristics (ROC) of Combined Features

P450 Holdout (2598)



Receiver Operating Characteristics (ROC) of Combined Features

P450 Extrapolation (797)



Performance on Non Typical Chemicals

Comparison GP and AdaBoost.M2 (holdout set)

70% of inhibiting compounds (HTS) can be predicted (at the expense of misclassifying 24% of inactives) using Genetic Programming composite classifier based on readily computed features.

For a boosted combination, at 70% true positive rate, 26.6% of inactives are incorrectly predicted.

The best Clementine network, at 70% true positive rate, wrongly suggests 28.6% of inactives inhibit P450.

None of the features give adequate performance if used singularly.

Conclusions

- GP automatically gets better results from a "fusion" of classifiers. Demonstrated on:

  Same classifier
(linear, overlapping Gaussians)

  Classifiers of same type
(linear, Thyroid)

  Classifiers of different types, trained on different data
(Landsat, C4.5, naive Bayes, neural networks)

- Not specific to a domain, demonstrated both on ML benchmarks and real industrial (GSK) data.

- Generalisation (extrapolation?) performance on P450

C++, Matlab code `http://www.cs.ucl.ac.uk/staff/W.Langdon/boosting/`

Future Work

- High throughput screening data is:
  artificial (in test tube)
  noisy

  Compare with IC50:
  more accurate
  closer to end use

- Virtual chemistry

- Other forms of data mining, e.g. time sequences

Composite of 75 Networks: Boosting

GP and Matlab boosting code use identical data.

Unlike GP, combination rule is Matlab feed forward neural net with over fitting stopping rule.

Experiments (without boosting) indicated little performance difference between 2 and 20 hidden units, so smallest neural network was used (75 inputs, 2 hidden units, 2 output neurons).

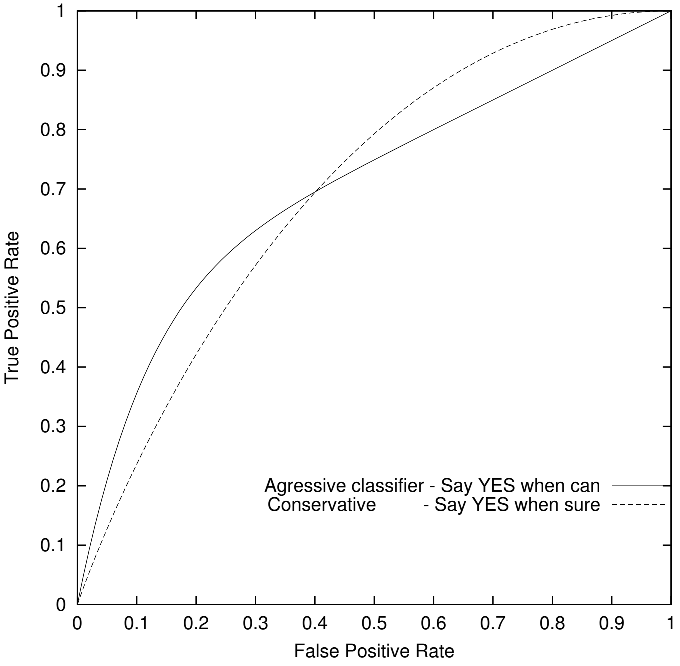
AdaBoost.M2 using training error and resampling from re-weighted training set.

When Will GP-ROC Work?

We may hope for improvement when

- Have both aggressive (say positive when can) and conservative (only when sure) classifiers
- Classifiers which are good at different parts of the feature space
- That is we have “complementary classifiers”.
- Alternatively, we have a small number of significant features interacting in a complicated way. I.e. we are seeking a non-linear combination.

Aggressive v. Conservative Classifiers


True Positive Rate
False Positive Rate
Agressive classifier - Say YES when can
Conservative - Say YES when sure