WHAT COMPUTERS CAN DO: APPLYING ARTIFICIAL INTELLIGENCE TECHNIQUES TO TWO COMPUTATIONALLY INTENSIVE SCIENTIFIC PROBLEMS

by

SERGEY VICTOR FOGELSON

(Under the Direction of Walter Potter)

ABSTRACT

Several AI techniques are applied in two scientific task domains. Genetic Programming (GP) is used to evolve a set of functions to approximate the static dielectric constant of water and several different binary classification algorithms are compared in their ability to distinguish translation start sites on two different prokaryotic genomes. GP performs very well as compared with standard statistical approaches to approximating the dielectric constant, and is a very powerful new tool that can be used for regression analysis in this and related domains. Translation start site prediction remains an open problem in bioinformatics, and several computational models for translation start site prediction have been created before. Support vector machines, decision trees, naïve bayes, artificial neural networks, and XCS are all compared in their ability to locate translation start sites. XCS has never been used for this task and performs as well as the other aforementioned techniques, making the technique a viable new candidate for generating predictive models for this and other computational biological problems.

INDEX WORDS: Static Dielectric Constant, Genetic Programming, Symbolic Regression, Translation Start Site, Neural Network, Support Vector Machine, Learning Classifier System, Decision Tree, Naïve Bayes Classifier

# WHAT COMPUTERS CAN DO: APPLYING ARTIFICIAL INTELLIGENCE TECHNIQUES TO TWO COMPUTATIONALLY INTENSIVE SCIENTIFIC PROBLEMS

by

SERGEY VICTOR FOGELSON

B.A., The University of Georgia, 2007

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2007

# WHAT COMPUTERS CAN DO: APPLYING ARTIFICIAL INTELLIGENCE TECHNIQUES TO TWO COMPUTATIONALLY INTENSIVE SCIENTIFIC PROBLEMS

by

SERGEY VICTOR FOGELSON

| | |
|---|---|
| Major Professor: | Walter Potter |
| Committee: | Khaled Rasheed |
| | Jan Mrázek |

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
August 2007

# DEDICATION

I would like to dedicate this thesis to my parents, Lidiya Grankina and Mikhail Fogelson, for their unending support in (almost) everything that I have done.

# ACKNOWLEDGEMENTS

I would like to thank Dr. Potter for being an exceptional teacher and mentor throughout my time at the AI Center. I would also like to thank Dr. Jan Mrázek and his postdoctoral student Xiangxue Guo for allowing me to use the translation start site data that they compiled as part of my thesis research. Without their cooperation one half of this thesis would never have occurred. Finally, the students of the AI Center have made the past two years a rather memorable experience; I am happy to say that they are some of the most interesting and engaging people I have ever met, and I want to thank them for letting me play a small role in their lives.

TABLE OF CONTENTS

Page

# LIST OF TABLES

Page

# LIST OF FIGURES

Page

# CHAPTER 1

## INTRODUCTION AND LITERATURE REVIEW

The current thesis serves three purposes. Firstly, it aims to apply several artificial intelligence techniques to two computationally complex scientific modeling problems: approximating the static dielectric constant of water and steam and predicting translation start sites on prokaryotic genomes. Secondly, the study aims to compare these techniques' abilities to generate accurate, robust models of the phenomena they were developed to predict or approximate. Two of the techniques described in this study (genetic programming and learning classifier systems) have never been used in either of the scientific domains explored. The models generated with these techniques should be of interest to researchers looking for new computational methods to help tackle similar scientific problems. The final purpose of this study is to create accurate, useable models of the phenomena studied, so that other researchers may incorporate them into their own research.

This thesis is the culmination of research begun as term projects for two courses offered through the Artificial Intelligence Center: Computational Intelligence and Evolutionary Computation. These term projects were initiated as a result of my interests in the application of AI techniques in novel ways to difficult scientific problems. The first project involved using genetic programming to evolve a function to approximate the relative permittivity (or static dielectric constant) of water and steam. Briefly, the relative permittivity of a substance is a measure of the ability of that substance to permit the existence of an electric field given certain thermodynamic conditions (notably the temperature and pressure of the substance). This property of a substance is

essential for understanding its electrochemical behavior in a variety of settings. In the case of water, approximating the static dielectric constant is essential for understanding its behavior in biochemical, geophysical, and industrial processes. As a result, creating a function that accurately approximates the static dielectric constant of water across a wide range of temperatures and pressures has been an open problem for researchers in physical chemistry for the past five decades. This problem has traditionally been approached by wedding statistical regression techniques to a variety of theoretical considerations that constrain the overall form that the function for approximating the dielectric constant may assume. In the first part of this thesis, comprised of the first three papers, a different approach is taken; instead of constraining the form of the function through theoretical and statistical means, a function for predicting the dielectric constant is evolved using genetic programming, a computationally intensive artificial intelligence algorithm that constrains the form of the final function based on how well the function fits the data that the algorithm is initially given. This approach to approximating the dielectric constant has never been taken, and the following results should show that genetic programming is an incredibly powerful and versatile technique that can be used to tackle similar approximation and regression problems.

In the latter part of this thesis, initiated as a second term project and comprised of the final paper, a slew of different artificial intelligence techniques are applied to the problem of developing a model for predicting translation start sites on prokaryotic genomes. Translation is one of several stages in gene expression and protein biosynthesis and involves the decoding of a given mRNA sequence into a sequence of amino acids (known as a polypeptide). Although translation termination is unambiguously coded for in a given mRNA sequence, translation start sites cannot be unambiguously identified in a simple manner. Furthermore, manual curation techniques, although able to unambiguously pinpoint the translation start site for a specific mRNA strand are time consuming and expensive. As a result, a computational model that accurately predicts translation

start sites would be a very useful tool for researchers in computational biology. For this part of the thesis, an extensive preprocessing phase occurred, where transcription start site data was compiled for two different bacterial species. Following this, datasets were ready to be used to construct predictive models. However, following the initial phase of compiling useable datasets, the dimensionality of the data relative to the number of positive examples in the datasets was prohibitively high. As a result, the dimensionality of both datasets was significantly reduced and the quality of the models being constructed saw concomitant improvement.

The techniques that are compared here (support vector machines, decision trees, naïve bayes, artificial neural networks, and XCS) have all been used to predict translation start sites before, except for learning classifier systems, which is applied to this domain here for the first time. Learning classifier systems is a robust technique that attempts to evolve a collection of rules to solve a given problem. The final collection of rules that comprise the model for the task at hand can then be used and understood fairly easily by researchers. This approach would be especially useful for bioinformatics researchers as creating human-readable rules and models out of data with very high dimensionality is proving to be quite a challenge in computational biology.

The models generated with these techniques will be shown to be robust, accurate, and capable of generalizing well to unseen data. Furthermore, the techniques that are used here for the first time will be shown to perform comparably to other techniques typically used in both domains. The research that is described in this work will allow future researchers to understand effective and appropriate ways in which problems belonging to these and similar task domains may be approached, represented, and modeled from an artificial evolutionary and heuristic-driven computational perspective. Finally, the models developed in this work may be used on their own.

# CHAPTER 2

# A GP-EVOLVED FORMULATION FOR THE RELATIVE PERMITTIVITY OF WATER AND STEAM[1]

[1] S.V. Fogelson and W.D. Potter. 2007. To appear in *Proceedings of the International Conference on Genetic and Evolutionary Methods, GEM '07.* Reprinted here with permission of the publisher, 6/18/2007.

ABSTRACT

The relative permittivity (or static dielectric constant) of water and steam has been experimentally calculated at a relatively wide range of temperatures and pressures. Two separate functions for predicting the relative permittivity of water and steam in three distinct thermodynamic regions are evolved using genetic programming. A data set comprised of all of the most accurate relative permittivity values, along with temperature, pressure, and density values from the entire experimentally calculated range of these values, found in (Fernandez et al. 1995), is used for this task. The accuracy of these two functions is evaluated by comparing the values for the relative permittivity calculated using the evolved functions and the values calculated using the latest formulation of Fernandez et al., found in (Fernandez et al. 1997) to the aforementioned data set. In all three regions, the newly evolved functions outperform the most current formulation in terms of difference between calculated and experimentally obtained values for the dielectric constant. This work heralds the first successful application of AI techniques to this important scientific application area.

## 2.1 INTRODUCTION AND BACKGROUND

The relative permittivity (or static dielectric constant) of water and steam, $\varepsilon_r$, has been experimentally calculated at a relatively wide range of temperatures and pressures. The relative permittivity is an important indicator of the solvent behavior of water in a variety of biological (cell membrane electrophysiology, intracellular biochemical processes), and industrial (geochemical high temperature, high pressure processes in deep sea vents) settings (Fernandez et al. 1997). Thus, predicting the behavior of the static dielectric constant of water is crucial for understanding a variety of phenomena, from the effects of hydrostatic pressure on protein folding and unfolding within the cell (Floriano and Nascimento 2004), to understanding the corrosive behavior of water at the high temperatures and pressures found in nuclear power plants. In electrical engineering, the relative permittivity of a substance is used in the design of capacitors. There have been many attempts at creating a single function that accurately predicts the relative permittivity of water and steam, the earliest of which was done by Quist and Marshall in 1965 (Quist and Marshall 1965), but these have suffered from a lack of experimental values across the entire temperature and pressure range. Recently, Fernandez et al. compiled all of the experimentally available data for the relative permittivity of water and steam in a single database (Fernandez et al. 1995). Furthermore, Fernandez et al. evaluated the methods used to experimentally derive the relative permittivity and chose a subset of the total data set that was the most accurate and that should be used in data correlation. Fernandez et al. proposed a new formulation in (Fernandez et al. 1997) that used this subset and approximated the relative permittivity very well across the entire temperature and pressure range.

Our proposal is that in order to more accurately model the behavior of the relative permittivity of water across all temperature and pressure values, two formulations should be created, so that each may be applied in separate thermodynamic regions. In our approach, two

functions are evolved that separately approximate the relative permittivity of water and steam across three thermodynamically distinct regions. These two functions collectively approximate the relative permittivity of water across the entire range of temperature and pressure values. The accuracy of these two functions is evaluated by comparing their values for the relative permittivity with the values obtained using the latest formulation of Fernandez et al., against the subset of dielectric constant values that Fernandez et al. chose for data correlation mentioned earlier.

The static dielectric constant (hereon relative permittivity) of a substance, $\varepsilon_r$, is roughly defined as the ability of a substance to transmit or allow the existence of an electric field. More formally, the relative permittivity of a substance, $\varepsilon_r$, is the ratio of the static permittivity of the substance, $\varepsilon_s$, to the static permittivity of a vacuum, $\varepsilon_0$ (Fernandez et al. 1995). The behavior of the relative permittivity of water is related to its physical state (as a liquid or as steam), temperature, and pressure. This allows the entire range of temperatures and pressures to be divided into 4 regions, A, B, C, and D. Region A is the normal liquid water state between the normal freezing and boiling points (~273K to ~373K). Region B refers to water along the liquid-vapor phase boundary (saturation line). In this region, which extends from 373K to approximately 647.1K (the critical point), water may exist in either the liquid or gas state (depending on the pressure value). The critical point, which occurs at approximately 647.1K with a corresponding pressure of approximately 22.1MPa, denotes the point in the phase space beyond which water ceases to exist in the liquid state. Region C is the region above 373.15K, and at lower pressures and temperatures within region C, water is in the normal gas (steam) phase. However, at higher pressures and temperatures in this region (beyond the critical point), water becomes a supercritical fluid that exhibits the properties of both a liquid and gas. Finally, region D refers to super cooled water (water below the normal freezing point of 273.15K at the standard pressure of ~.1MPa).

The behavior of the relative permittivity exhibits discontinuities along the liquid-vapor phase boundary (region B) and in the supercritical part of the region above the normal boiling point (region C). In these regions, very small changes in temperature and pressure cause very large changes in density and in the value of the relative permittivity (Harvey 2006). As a result, theoretical formulations for calculating the relative permittivity of water have mainly focused on a broad range of temperatures (~270K to ~600K) within a small range of pressures (~.1MPa to 200MPa) (Fernandez 1995). The most current formulation for approximating the relative permittivity across the entire range of experimental temperatures and pressures may be found in (Fernandez 1997). Fernandez et al.'s formulation uses an extensive adaptive regression algorithm to create an appropriate function. The final function uses 5 adjustable parameters and a total of 25 constants and domain specific non-adjustable parameters and approximates well across the entire range of experimentally available values (260K to 800K temperatures, at pressures up to 1200 MPa).

## 2.2 EXPERIMENTAL SET-UP

In our approach, a variety of different function and terminal sets were explored in an effort to evolve two functions that could model the relative permittivity of water as a function of pressure, temperature, and density. Unfortunately, no empirical temperature or pressure data for region B (along the phase boundary) is currently available (Fernandez et al. 1995), and thus a function approximating the dielectric constant in region B was not evolved. As a result, evolving 2 different functions, one specific to regions A and D, the other specific to region C, became the most logical next step in function development.

The functions for regions A, C, and D were evolved using data sets taken from (Fernandez et al. 1995) and were then compared to relative permittivity values calculated with the same input values (taken from the same data sets) using the newest formulation for dielectric constant

prediction, found in (Fernandez et al. 1997). These data sets were compiled from all previous experimentally available data, and were then corrected by Fernandez et al. to coincide with the most recent internationally accepted temperature scale, ITS-90. In most cases, values were provided for the temperature (in degrees Kelvin, or K), pressure (in megapascals, or MPa), and the corresponding dielectric constant. However, in some cases, temperature/density/dielectric constant values were given instead of temperature/pressure/dielectric constant values. In these circumstances, density values were converted into their corresponding pressures, and pressure values were converted to their corresponding densities using the IAPWS-95 formulation for the equation of state of water found in (Wagner and Pruss 2002). With this completed, the final data set uniformly represented the dielectric constant at every temperature, pressure, and density value that was experimentally available.

Both functions were evolved by generating a population of possible functions (represented as trees) as with standard genetic programming implementations. Each candidate function's fitness was taken to be the sum of the absolute values of the difference between the calculated and the experimentally measured value for the relative permittivity at every input value in the corresponding data set. The combination of input values for each function (that is, what combination of the three possible adjustable inputs was to be used) was determined by the GP module. The population of possible functions was then evolved with a variety of crossover/mutation probabilities and function sets. The data set of experimentally calculated relative permittivity values used to create a function for regions A and D consisted of 291 data points. The data set used to create the function for the one-phase supercritical region (region C) consisted of 353 data points. These data sets include all of the data points (644 total data points) that Fernandez et al. recommend for data correlations (Fernandez et al. 1995). The two evolved functions with the lowest sum of

absolute errors across the data points that were found were used as the final equations for approximating the dielectric constant across the three regions.

During any given GP run, all function and terminal sets used during function evolution always included addition, subtraction, multiplication, and division as function operators, and temperature, $T_k$, pressure, $p$, and density, $\rho$, as terminal values. All runs also used a population of 10 random floating-point constants in the range between 0 and 1, which were generated at runtime. Other function operators (sin(), cos(), ln(), $\log_{10}$, $\log_2$, and $x^y$) and terminal operators (Avogadro's number, $N_A$, permittivity of free space, $\varepsilon_0$, elementary charge, $e$, Boltzmann's constant, $k$, molar mass of water, $M_w$, mean molecular polarizability of water, $\alpha$, the dipole moment of water, $\mu$) were also used in certain GP runs. The aforementioned terminal operators are provided in table 1. A range of crossover probabilities (between .5 and 1.0, in increments of .05) and mutation probabilities (between 0 and .5, in increments of .05) were explored for all combinations of function and terminal sets. Each combination of parameter settings was implemented in 10 GP runs, each on a population of one million individuals that were evolved for 200 generations. The function length of any individual solution (a tree representing a given candidate function) never exceeded 50 functional units (where a functional unit is taken to be a single operator from the function set or a terminal value from the terminal set), as maintaining the readability of any given evolved function was a priority.

## 2.3 RESULTS

Both of the two best functions that were evolved were found during a run that used multiplication, division, subtraction, and addition as operators in the function set and temperature, pressure, and the molar mass of water as terminal operators (with the 10 additional random ephemeral constants described earlier). In addition to the above terminals, the function evolved for

region C used density, $\rho$, Avogadro's number, $N_A$, and Boltzmann's constant, $k$, as terminal operators. Both best function runs used a probability of crossover of 0.7 and a probability of mutation of 0.05. These functions (simplified with all redundancies eliminated), along with Fernandez et. al's formulation, follow:

$$\varepsilon = \varepsilon_{r,AD} \quad \text{when } T_k \leq 373.15,$$
$$\varepsilon_{r,C} \quad \text{elsewhere}$$

$$\varepsilon_{r,AD} = \frac{2.5203 * 10^4}{T_k} - \frac{M_w^3 p^2}{0.587} - 0.08133 T_k + 0.0355 p + 18.08$$

**Figure 2.1: Evolved Equation for regions A, D**

$$\varepsilon_{r,C} = \frac{0.728(\frac{N_A}{k}\rho + M_w \rho^2)}{0.971 T_k - 88.4082} - \frac{\rho p + p^2}{T_k^2} + 0.97$$

**Figure 2.2: Evolved Equation for region C**

$$\varepsilon_r = \frac{1 + 5A + 5B + \sqrt{9 + 2A + 18B + A^2 + 10AB + 9B^2}}{4 - 4B}$$

*where A and B are given by*

$$A = \frac{N_A \mu^2}{\varepsilon_0 k} \frac{\rho g}{T_k}$$

$$B = \frac{N_A \alpha}{3\varepsilon_0} \rho$$

*and where g is given by*

$$g = 1 + \sum_{k=1}^{11} N_k (\frac{\rho}{\rho_c})^{i_k} (\frac{T_c}{T})^{j_k} + N_{12} (\frac{\rho}{\rho_c})(\frac{T}{228K} - 1)^{-q}$$

*with* $\rho_c = \frac{322}{M_w}$ *and* $T_c = 647.096K$ *and values for* $N_k, i_k, j_k$, *and q given in table 2 of Appendix A.*

**Figure 2.3: Fernandez et al.'s Formulation**

The evolved functions shown above are significantly smaller than the formulation developed by Fernandez et al. and use at most three adjustable parameters (temperature, pressure, and density), three non-adjustable domain specific parameters (Avogadro's number, Boltzmann's constant, and the molar mass of water), and three of the ten possible random ephemeral constants that were available during function evolution. No domain-specific knowledge (aside from the data sets themselves) was applied to the formulation of the functions. Furthermore, the evolved functions selected different terminal values for both regions, so that the region C function uses density as an input value along with temperature and pressure, whereas the region A and D function uses temperature and pressure exclusively. This is telling because density is a much more relevant predictive parameter (varying discontinuously along with the relative permittivity while temperature and pressure monotonically increase) for the relative permittivity in the single phase and super critical region (region C) than in regions A and D. The fact that the GP approach was able to selectively choose the relevant parameters for each region is notable and significant.

Both evolved functions outperformed Fernandez et al.'s formulation across all thermodynamic regions. For regions A and D the evolved function outperformed Fernandez et al.'s formulation strictly because of one data point value (notably, a data point that occurred immediately preceding the phase boundary around 373.15K). At this temperature, Fernandez et al.'s formulation may have rounded the temperature input parameter (at 373.147K) up, causing a very sharp discontinuous drop in the calculated relative permittivity value. In region C, the evolved function consistently outperformed Fernandez et al.'s formulation, leading to an improvement in calculation accuracy across the entire range of experimentally available relative permittivity values.

## 2.4 CONCLUSIONS AND FUTURE WORK

Two functions that approximate the relative permittivity of water and steam at a variety of temperatures and pressures have been proposed. These functions were evolved using the GP

technique with a specific function and terminal set, and their accuracy has been compared to that achieved by Fernandez et al.'s most recent formulation. The evolved functions approximate the relative permittivity of water and steam for a wide range of temperature and pressure values quite well, improving on Fernandez et al.'s formulation across the entire experimentally available temperature and pressure range while being much simpler computationally. Further refinements to create more accurate approximations of the relative permittivity of water and steam will include creating an evolved function that can be used across all thermodynamically distinct temperature and pressure regions.

## 2.5 TABLES AND REFERENCES

Tables and references are available in the extended version of this paper. See Appendix A.

# CHAPTER 3

# A NEW GP-EVOLVED FORMULATION FOR THE RELATIVE PERMITTIVITY OF WATER AND STEAM[2]

[2] S.V. Fogelson and W.D. Potter. To appear in *Proceedings of the International Conference on Artificial Intelligence and Pattern Recognition, AIPR-07.* Reprinted here with permission of the publisher, 6/19/2007.

ABSTRACT

The relative permittivity (or static dielectric constant) of water and steam has been experimentally calculated at a relatively wide range of temperatures and pressures. A single function for predicting the relative permittivity of water and steam in three distinct thermodynamic regions is evolved using genetic programming. A data set comprised of all of the most accurate relative permittivity values, along with temperature, pressure, and density values from the entire experimentally calculated range of these values, found in (Fernandez et al. 1995), is used for this task. The accuracy of this function is evaluated by comparing the values for the relative permittivity calculated using the evolved function and the values calculated using the latest formulation of Fernandez et al., found in (Fernandez et al. 1997) to the aforementioned data set. In all regions, the newly evolved function outperforms the most current formulation in terms of difference between calculated and experimentally obtained values for the dielectric constant.

## 3.1 INTRODUCTION

The relative permittivity (or static dielectric constant) of water and steam, $\varepsilon_r$, has been experimentally calculated at a relatively wide range of temperatures and pressures. The relative permittivity is an important indicator of the solvent behavior of water in a variety of biological (cell membrane electrophysiology, intracellular biochemical processes) and geophysical/industrial (geochemical high temperature, high pressure processes in deep sea vents and in industrial processing plants) settings (Fernandez et al. 1997). Many prior researchers have attempted to create a single function that accurately predicts the relative permittivity of water and steam, the earliest of which being Quist and Marshall's 1965 formulation (Quist and Marshall 1965). However, these attempts have suffered from a lack of experimental values across the entire temperature and pressure range, and thus have only been able to approximate the relative permittivity of water with minimal uncertainty over a small range of temperatures and pressures. Recently, Fernandez et al. compiled all of the experimentally available data for the relative permittivity of water and steam in a single database (Fernandez et al. 1995). Furthermore, Fernandez et al. evaluated the methods used to experimentally derive the relative permittivity and chose a subset of the total data set that was the most accurate and that was recommended for use in data correlation. Fernandez et al. proposed a new formulation in (Fernandez et al. 1997) that used this subset to generate a statistical regression function that approximated the relative permittivity very well across the entire experimentally available temperature and pressure range.

In an earlier paper (Fogelson and Potter 2007), we proposed two individual functions evolved using genetic programming that divided the entire data set recommended for data correlation by Fernandez et al. into two distinct thermodynamic regions, with each equation applied to the temperature and pressure range specific to the given thermodynamic region. Although that proposed formulation outperformed Fernandez et al.'s formulation across the entire range of data

values in both thermodynamic regions, a formulation that utilizes a single equation to approximate the relative permittivity across the entire range of experimental values would seem both more natural and appropriate, and is an important goal for researchers in this area. It was hoped that an increase in the size of the evolving population of programs coupled with an increase in the maximum size any individual program could be would allow for the discovery of just such an equation. In the current approach, such an equation has been evolved and closely approximates the relative permittivity of water across the entire range of experimentally verified temperature and pressure values. The accuracy of this function is evaluated by comparing its output value for the relative permittivity of water at a given temperature and pressure with the output relative permittivity value obtained using the latest formulation of Fernandez et al., against the subset of dielectric constant values that Fernandez et al. chose for data correlation mentioned earlier.

## 3.2 THE STATIC DIELECTRIC CONSTANT

The static dielectric constant (hereon relative permittivity) of a substance, $\varepsilon_r$, is roughly defined as the ability of a substance to transmit or allow the existence of an electric field. More formally, the relative permittivity of a substance, $\varepsilon_r$, is the ratio of the static permittivity of the substance, $\varepsilon_s$, to the static permittivity of a vacuum, $\varepsilon_0$ (Fernandez et al. 1995). The relative permittivity of a substance is used for practical purposes in the design of capacitors. The behavior of the relative permittivity of water is related to its physical state or phase (as a liquid or as steam), temperature, and pressure. Experimentally verified relative permittivity values for water in its solid phase (as ice) at temperatures as low as 190K ($-83°C$) exist (Matsuoka, Fujita, and Mae 1996), however this data did not include corresponding pressure values for any of the measurements, and as a result, could not be used. Water, in its liquid or gas (steam) state can exist within a large range of temperatures and pressures, and this range has been traditionally divided into 4 regions, A, B, C, and D. Region A is the normal liquid water state between the normal freezing and boiling points

(~273K to ~373K) at pressures up to 1000MPa. Region B refers to water along the vapor-liquid phase boundary. Region C is the region with a temperature above 373.15K. At lower pressures and temperatures within region C, water is in the normal vapor (steam) state. At higher pressures and temperatures in this region, water becomes a supercritical fluid, that is, water ceases to behave as if it were in either the liquid or gas state, but rather exhibits a combination of the thermodynamic properties attributable to both liquids and gases. Finally, region D refers to super cooled water (water that exists in the liquid state below the normal freezing point of 273.15K at the standard pressure of ~.1MPa).

The behavior of the relative permittivity exhibits discontinuities along the liquid-vapor phase boundary (region B) and in the supercritical part of the region above the normal boiling point (region C), with very small changes in the temperature and pressure causing very large changes in density and in the value of the relative permittivity (Harvey 2006). As a result, theoretical formulations for calculating the relative permittivity of water have mainly focused on a narrow range of temperatures (~270K to ~315K) and pressures (~.1MPa to 100MPa) below the phase boundary (Fernandez et al. 1995). Furthermore, data points along the phase boundary (region B), although numerous, have not had their pressure values recorded, and thus have not figured in any data-driven correlations that correct for pressure differences. The most current formulation for approximating the relative permittivity across the entire range of experimental temperatures and pressures may be found in (Fernandez et al. 1997) and is also reproduced in the results section. Fernandez et al.'s formulation uses an extensive adaptive regression algorithm to create an appropriate function taking a wide variety of domain specific thermodynamic values (including first, second, and third derivatives of the temperature and pressure inputs with respect to each other) into account.

The final function uses 5 adjustable parameters and a total of 25 constants and domain specific non-adjustable parameters and approximates well across the entire range of experimentally available values.

## 3.3 EVOLUTION AND GENETIC PROGRAMMING

Genetic Programming (GP) may be seen as an abstract algorithmic implementation broadly inspired by the main principles of Darwin's theory of evolution by means of natural selection. Roughly, Darwinian evolutionary theory involves populations of interbreeding organisms (species) competing for environmental resources over time. Species share genetic material by interbreeding, and random mutations occur to members of the species that may either hinder or further their reproductive success. As the members of a given species breed with each other over time, characteristics beneficial for the species' survival propagate throughout the population, while those characteristics that are detrimental to the survival of the species do not get expressed in the population. That is, individuals with characteristics that favor their survival within the given environment tend to propagate, whereas individuals not possessing those characteristics in the environment (or those that exhibit detrimental characteristics) tend to die out.

GP applies the broad tenets of Darwinian evolutionary theory within a heuristic framework that attempts to create automatically generated programs that evolve to optimally solve user-defined problems (Koza 1992). GP is an extension of the evolutionary computational approach known as genetic algorithms (GA) first pioneered by John Holland (Holland 1992). Within the GP framework, a population of candidate solutions, each represented as an executable computer program of some finite length (an individual of a given population), evolves in response to some problem to be solved (the environmental conditions) (Koza 1992). Each GP individual/candidate program within the population is given a fitness value that is the output of a function (the fitness function) that determines the appropriateness or optimality of the program output (individual

behavior) when given the user-defined problem (the environmental conditions). This allows each individual within the GP population to be measured against every other individual, whether the individual solves the problem (optimally responds to the environment) or not. Once all of the individuals within a population have been assigned a fitness value, certain individuals are probabilistically chosen to recombine and create offspring based on their fitness values, so that individuals with higher fitness values tend to be chosen more frequently for recombination. During recombination two unique individuals are chosen to represent the parents, and may stochastically recombine to generate two offspring. Occasionally, however, (because recombination is probabilistic and does not always occur) they do not recombine and remain unchanged as offspring. After every recombination event, an offspring individual may be mutated with some small probability. The series of steps from initial population generation, parent selection, recombination, and mutation of offspring constitutes a generation of the GP run. At the start of every generation, newly created individuals in the population are evaluated by the fitness function and assigned a fitness value. The GP run continues in this manner (after the generation of the initial population, only fitness value assignment, parent selection, recombination, and mutation of offspring occur) until some stopping criteria (such as the creation of an individual with either some given minimum or maximum fitness value, or one that adequately solves the problem at hand) has been reached.

Each GP individual uses a tree-based representation scheme, where the tree completely represents a given program. Nodes for the GP program tree either come from the terminal set or the function set (both predefined by the individual implementing the GP search). The terminal set completely defines the kinds of inputs the given program can use to solve the problem. The members of the terminal set can only occur as leaf nodes within the program tree (that is, nodes that have no children). The function set defines the kinds of transformations that are permissible given any of the elements in the terminal set or any of the other elements within the function set as

arguments to each of the elements within the function set. Thus, the members of the function set may only occur as the internal nodes of a GP-generated program tree (nodes with at least one child node). These restrictions amount to the fact that the union of the function and terminal sets of a GP implementation must possess the property of closure (where closure is defined as the ability to have any composition of functions and terminals produce an executable computer program) (Ghanea-Hercock 2003). The program trees generated using GP do not have to be standard binary trees (trees where every node is either a leaf node, or has a maximum of two child nodes), as the experimenter may define a function operator within the function set that takes more than two arguments. Initially, GP individuals are randomly generated through a stochastic tree-building process where each node in the tree is chosen to be a random member of either the function or terminal sets. Traditionally, GP candidate programs are initially generated either strictly to some maximum initial tree depth limit (where all nodes up to the maximum initial tree depth are chosen stochastically exclusively from the function set and all nodes at the maximum initial depth limit are chosen exclusively from the terminal set), or until all of the branches of the tree have either gone to the maximum initial depth or have ended in terminal nodes before the maximum initial tree depth has been reached.

The genetic operators of crossover and mutation, as well as the way in which individuals are ranked according to their fitness level are modified from the GA approach (described in detail in Holland 1992) to suit the GP technique. Crossover occurs by selecting two nodes on different parent trees and then swapping all of the children of the selected nodes (as well as the selected nodes themselves) between the two individuals. Mutation, on the other hand, involves selecting a node at which mutation will occur, deleting all of the nodes that are children of the selected node, and then generating a random tree with this node as its root. The fitness evaluation and ranking method in GP are slightly different from the classic GA approach (where fitness maximization is standard) in the

fact that the highest ranking individual programs in GP have the lowest fitness values (in effect, a minimization problem). Thus, GP attempts to find a program with the globally minimal fitness value in the search space of all possible programs that may be created using the function and terminal sets used in the problem, to the tree depth or program length specified in the GP setup.

Ultimately, the GP approach involves determining a set of functions and terminals to be used in solving the problem, defining a fitness measure by which individual programs may be evaluated and assigned a fitness value, setting the specific parameters and operator probabilities that are involved in program tree generation (crossover and mutation probabilities, initial tree depth limit, maximum tree length, etc.), and developing a set of rules or stopping criteria to determine when to end a specific GP run (whether after a certain number of generations have elapsed, or after an individual program with a desired fitness threshold has been found).

## 3.4 EXPERIMENTAL SET-UP

In our approach, a variety of different function and terminal sets were explored in an effort to evolve a single function that could model the relative permittivity of water as a function of pressure, temperature, and density in thermodynamic regions A, C, and D of the temperature-pressure phase space. Unfortunately, no empirical temperature *and pressure* data for region B (along the phase boundary) is currently available (Fernandez et al. 1995), and thus a function approximating the dielectric constant in region B was not evolved.

The function for regions A, C, and D was evolved using data sets taken from (Fernandez et al. 1995) and was then compared to relative permittivity values calculated with the same input temperature/pressure/density values (taken from the same data sets) using the newest formulation for dielectric constant prediction, found in (Fernandez et al. 1997). These data sets were compiled from all previous experimentally available data, and were then corrected by Fernandez et al. to coincide with the most recent internationally accepted temperature scale, ITS-90. In most cases,

values were provided for the temperature (in degrees Kelvin, or K), pressure (in megapascals, or MPa), and the corresponding dielectric constant. However, in some cases, temperature/density/dielectric constant values were given instead of temperature/pressure/dielectric constant values. In these circumstances, density values were converted into their corresponding pressures, and pressure values were converted to their corresponding densities using the IAPWS-95 formulation for the equation of state of water found in (Wagner and Pruss 2002). With this completed, the final data set uniformly represented the dielectric constant at every temperature, pressure, and density value that was experimentally available (as of December 2006).

The function was evolved by generating a population of possible functions (represented as trees) as with standard genetic programming implementations. Each candidate function's fitness was taken to be the sum of the absolute errors between the calculated and the experimentally measured value for the relative permittivity at every input value in the corresponding data set. The combination of input values for each function (that is, what combination of the three possible adjustable inputs was to be used) was determined by the GP module. The population of possible functions was then evolved with a variety of crossover/mutation probabilities and function sets. The data set of experimentally calculated relative permittivity values used to create the function consisted of 644 data points, which represent the complete dataset that Fernandez et al. recommend for data correlations (Fernandez et al. 1995). The function with the lowest sum of absolute errors across the data points that was found after all runs had been completed was chosen as the final formulation.

During any given GP run, all function and terminal sets used during function evolution always included addition, subtraction, multiplication, and division as function operators, and temperature, $T_k$, pressure, $p$, and density, $\rho$, as terminal values. In cases where a generated function divided a value by zero, the zero-generating term was replaced by 0.00001. All runs used a

population of 100 random floating-point constants in the range between 0 and 1, which were generated at runtime. These constants would function as additional terminal values for the genetic program to use during function creation. Other function operators (sin(), cos(), ln(), $\log_{10}$, $\log_2$, and $x^y$) and terminal operators (Avogadro's number, $N_A$, permittivity of free space, $\varepsilon_0$, elementary charge, $e$, Boltzmann's constant, $k$, molar mass of water, $M_w$, mean molecular polarizability of water, $\alpha$, the dipole moment of water, $\mu$) were also used in certain GP runs. The aforementioned terminal operators are provided in table 3.1. The function length of any individual solution (a tree representing a given candidate function) never exceeded 100 functional units (where a functional unit is taken to be a single operator from the function set or a terminal value from the terminal set). The large size of the function and terminal sets causes the size of the search space (representing all of the possible unique programs of length 100 or less that can be generated from the function and terminal sets) to be enormous (more than a googol). As a result, each GP run was done on a population of one and a half million individuals that were evolved for 200 generations. This was done to ensure that the GP implementation would sample as much of the search space as possible in its effort to find a suitable function within a reasonable time. A range of crossover probabilities (between .5 and 1.0, in increments of .05) and mutation probabilities (between 0 and .5, in increments of .05) were explored for all combinations of function and terminal sets. Each combination of unique parameter settings was implemented in 10 GP runs, after which the function with the lowest total absolute error was chosen.

## 3.5 RESULTS

The optimal function that was evolved was found during a run that used multiplication, division, subtraction, and addition as operators in the function set and temperature, pressure, and density as terminal operators (with the 100 additional random ephemeral constants described

earlier). The optimal function run used a probability of crossover of 0.9 and a probability of mutation of 0.05. The final evolved function (with all expressions simplified), along with Fernandez et. al's formulation, are listed in figures 3.1 and 3.2.

The results of applying the GP-evolved function and Fernandez et al.'s formulation to the total data set are found in table 3.3. The evolved (non-simplified) function shown above is significantly smaller (31 terms versus 112 terms) than the formulation developed by Fernandez et al. and uses only three adjustable parameters (temperature, pressure, and density), zero non-adjustable domain specific parameters, and only three of the one hundred possible random ephemeral constants that were available during function evolution. No domain-specific knowledge (aside from the data sets themselves) was applied to the formation of the function. As can be seen from table 3.3, the evolved function outperformed Fernandez et al.'s formulation in all collected statistical categories except the minimum absolute difference, where both functions had at least one data point where very marginal absolute error (<0.01) existed.

$$\varepsilon_r = \frac{\dfrac{\rho^2+\rho^3}{-.02036T_k\rho+.0864\rho^2+.1194T_k p}+\dfrac{(-6.75862p^2+.313T_k p-{T_k}^2)(1+\rho)}{T_k}}{55.474T_k p+55.55p^2-.076T_k p\rho+.016p+.016p\rho+\rho+\rho^2}+$$
$$\frac{\rho^2}{.03264T_k(T_k+\rho)}-\frac{\dfrac{{T_k}^2}{\rho^2}+2.617p-1.617\rho+1.617T_k}{T_k+.0486-p+\rho}$$

**Figure 3.1: GP-Evolved Equation, regions A, C, and D**

$$\varepsilon_r = \frac{1+5A+5B+\sqrt{9+2A+18B+A^2+10AB+9B^2}}{4-4B}$$

*where A and B are given by*

$$A = \frac{N_A \mu^2}{\varepsilon_0 k}\frac{\rho g}{T_k}$$

$$B = \frac{N_A \alpha}{3\varepsilon_0}\rho$$

*and where g is given by*

$$g = 1 + \sum_{k=1}^{11} N_k \left(\frac{\rho}{\rho_c}\right)^{i_k} \left(\frac{T_c}{T}\right)^{j_k} + N_{12}\left(\frac{\rho}{\rho_c}\right)\left(\frac{T}{228K} - 1\right)^{-q}$$

*with* $\rho_c = \frac{322}{M_w}$ *and* $T_c = 647.096K$ *and values for* $N_k$, $i_k$, $j_k$, *and q given in table 2.*

**Figure 3.2: Fernandez et al.'s Formulation**

## 3.6 CONCLUSIONS AND FUTURE WORK

A new function that approximates the relative permittivity of water and steam at a variety of temperatures and pressures has been developed. This function was evolved using the GP technique with a specific function and terminal set, and its accuracy has been compared to that achieved by Fernandez et al.'s most recent formulation. The evolved function approximates the relative permittivity of water and steam for a wide range of temperature and pressure values extremely well, improving on Fernandez et al.'s formulation across the entire experimentally available temperature and pressure range while being simpler computationally. Further refinements to create more accurate approximations of the relative permittivity of water and steam will include creating an evolved function that can be used across all thermodynamically distinct temperature and pressure regions, including regions where water is in the solid phase, or where a phase boundary exists. This

can be done when experimental values for the temperature, pressure, and relative permittivity in these regions are obtained. A refined fitness function that takes more than the absolute distance between expected and calculated values may also prove useful in creating a new, more accurate formulation. Introducing a penalty for very large and difficult to read programs may also help in finding a function that is both compact and generalizes well across the entire thermodynamic space. However, significant improvements to the evolution of an appropriate function will most surely come from an increase in experimentally verifiable values for the relative permittivity, and thus any new accurate data that may be found should be used to refine the current formulation.

## 3.7 TABLES

**Table 3.1: Constants used in the relative permittivity formulation**

| Parameter | Value |
|---|---|
| Permittivity of free space, $\varepsilon_0$ | $[4*10^{-7}\pi(299792458)^2]^{-1}C^2J^{-1}m^{-1}$ |
| Elementary charge, $e$ | $1.60217733*10^{-19}C$ |
| Boltzmann's constant, $k$ | $1.380658*10^{-23}JK^{-1}$ |
| Avogadro's number, $N_A$ | $6.0221367*10^{23}mol^{-1}$ |
| Molar mass of water, $M_w$ | $0.018015268kg*mol^{-1}$ |
| Mean molecular polarizability of water, $\alpha$ | $1.636*10^{-40}C^2J^{-1}m^{-2}$ |
| Dipole moment of water, $\mu$ | $6.138*10^{-30}Cm$ |

**Table 3.2: Coefficients $N_k$, and exponents $i_k$, $j_k$, and q of the equation for g**

| $k$ | $N_k$ | $i_k$ | $j_k$ |
|---|---|---|---|
| 1 | $0.978224486826$ | 1 | 0.25 |
| 2 | $-0.957771379375$ | 1 | 1 |
| 3 | $0.237511794148$ | 1 | 2.5 |
| 4 | $0.714692244396$ | 2 | 1.5 |
| 5 | $-0.298217036956$ | 3 | 1.5 |
| 6 | $-0.108863472196$ | 3 | 2.5 |
| 7 | $0.949327488264*10^{-1}$ | 4 | 2 |
| 8 | $-0.980469816509*10^{-2}$ | 5 | 2 |
| 9 | $0.165167634970*10^{-4}$ | 6 | 5 |
| 10 | $0.937359795772*10^{-4}$ | 7 | 0.5 |
| 11 | $-0.123179218720*10^{-9}$ | 10 | 10 |
| 12 | $0.196096504426*10^{-2}$ | | *q=1.2* |

**Table 3.3: Results and numeric comparison**

| | Evolved GP result | Fernandez |
|---|---|---|
| Sum Absolute Difference | 103.12 | 149.73 |
| Mean Absolute Difference | 0.16 | 0.23 |
| Standard Deviation Absolute Difference | 0.25 | 2.15 |
| Sum Squared Difference | 55.8 | 3026.43 |
| Mean Squared Difference | 0.09 | 4.68 |
| Standard Deviation Squared Difference | 0.54 | 117.24 |
| Minimum Absolute Difference | 0 | 0 |
| Maximum Absolute Difference | 3.55 | 54.61 |
| # Data Points Absolute Difference formulation < Absolute Difference Fernandez | 331 | |
| Percentage of Total Data Points better than Fernandez | 51.08% | |
| Total Data Points | 644 | |

## 3.8 REFERENCES

Fernandez, D.P., Y. Mulev, A.R.H. Goodwin, and J.M.H. Levelt-Sengers. 1995. A Database for the Static Dielectric Constant of Water and Steam. *Journal of Physical and Chemical Reference Data* 24(1): 33-69.

Fernandez, D.P., A.R.H. Goodwin, E.W. Lemmon, J.M.H. Levelt-Sengers, and R.C. Williams. A Formulation for the Static Permittivity of Water and Steam at Temperatures from 238K to 873K at Pressures up to 1200MPa, Including Derivatives and Debye-Huckel Coefficients. *Journal of Physical and Chemical Reference Data* 26(4): 1125-1166.

Fogelson, S., and W. Potter. 2007. A GP-Evolved Formulation for the Relative Permittivity of Water and Steam. Submitted to *IEA-AIE07.*

Ghanea-Hercock, R. 2003. *Applied Evolutionary Algorithms in Java.* New York, NY: Springer-Verlag.

Harvey, A. 2006. NIST. Personal communication.

Holland, J. 1992. *Adaptation in Natural and Artificial Systems: 2nd Edition.* Cambridge, MA: MIT Press.

Koza, J.R. 1992. *Genetic Programming.* Cambridge, MA: MIT Press.

Matsuoka, T., S. Fujita, and S. Mae. 1996. Effect of temperature on dielectric properties of ice in the range 5-39 GHz. *Journal of Applied Physics* 80(10): 5884-5890.

Quist, A.S., and W.L. Marshall. 1965. Estimation of the Dielectric Constant of Water to $800°$. *Journal of Physical Chemistry* 9: 3165.

Wagner, W. and A. Pruss. 2002. The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use. *Journal of Physical and Chemical Reference Data* 31(2): 387-535.

# CHAPTER 4

# A FORMULATION FOR THE RELATIVE PERMITTIVITY OF WATER AND STEAM TO HIGH TEMPERATURES AND PRESSURES EVOLVED USING GENETIC PROGRAMMING[3]

[3] S.V. Fogelson and W.D. Potter. Submitted to *Journal of Physical Chemistry*, 7/16/2007.

ABSTRACT

The relative permittivity (or static dielectric constant) of water and steam has been experimentally obtained from a relatively wide range of temperatures and pressures. A suite of functions for predicting the relative permittivity of water and steam in four distinct thermodynamic regions is evolved using genetic programming. A data set comprised of all of the most accurate relative permittivity values, along with temperature, pressure, and density values from the entire experimentally obtained range of these values, is used for this task. The accuracy of these functions is evaluated by comparing the values for the relative permittivity calculated using the evolved function and the values calculated using the latest formulation of Fernandez et al. to the aforementioned data set. In all regions, the newly evolved function performs comparably to or better than the most current formulation in terms of difference between calculated and experimental values for the dielectric constant. This research heralds the first successful application of artificial evolutionary techniques to relative permittivity prediction in physical chemistry.

## 4.1 INTRODUCTION

The relative permittivity (or static dielectric constant) of water and steam, $\varepsilon_r$, has been experimentally determined for a relatively wide range of temperatures and pressures. The relative permittivity is an important indicator of the solvent behavior of water in a variety of biological (cell membrane electrophysiology, intracellular biochemical processes) and geophysical/industrial (geochemical high temperature, high pressure processes in deep sea vents and in industrial processing plants) settings (Fernandez et al. 1997). Over the years, many researchers have worked to derive a single function that accurately predicts the relative permittivity of water and steam, the earliest of which being Quist and Marshall's 1965 formulation (Quist and Marshall 1965). As research in this area progressed, work was done to explore more of the temperature and pressure spectrum, refine experimental results, and propose alternate formulations in order to enhance relative permittivity prediction. Recently, Fernandez et al. compiled all of the experimentally available data for the relative permittivity of water and steam in a single database (Fernandez et al. 1995). Furthermore, they evaluated the methods used to experimentally derive the relative permittivity and chose a subset of the total data available that represented the most accurate values and that was recommended for use in data correlation. In 1997, Fernandez et al. proposed a new formulation in (Fernandez et al. 1997) based on a portion of this subset. Their new formulation, based on a statistical regression function, approximated the relative permittivity very well across the entire experimentally available temperature and pressure range.

In an earlier paper (Fogelson and Potter 2007), we proposed a formulation evolved using the genetic programming technique, to approximate the static dielectric constant across three (A,C,D) of the four thermodynamic regions (A,B,C,D) characterized by Fernandez et al. in

(Fernandez et al. 1995). This function was evolved by applying the genetic programming technique to a dataset that was recommended for data correlation by Fernandez et al. Although the proposed formulation performed comparably to Fernandez et al.'s formulation across the entire range of data values in the three thermodynamic regions (A,C,D), a formulation that can approximate the relative permittivity across the entire range of experimental values seemed both more natural and appropriate, and has been an important goal for researchers in this area. Unfortunately, the dataset for region B was incomplete (did not contain pressure values) and thus not used in our earlier formulation. In the current approach, we have incorporated the data from region B with our earlier formulation by evolving separate additional functions to approximate the static dielectric constant in this region. The accuracy of this suite of functions is evaluated by comparing each function's output value for relative permittivity at a given temperature, pressure, and density with the output value obtained using the latest formulation of Fernandez et al., against the subset of dielectric constant values that Fernandez et al. recommended for data correlation mentioned earlier, as well as the smaller subset of values that Fernandez et al. actually used to create their formulation.

## 4.2 BACKGROUND: THE STATIC DIELECTRIC CONSTANT

The static dielectric constant (hereon relative permittivity) of a substance, $\varepsilon_r$, is roughly defined as the ability of a substance to transmit or allow the existence of an electric field. More formally, the relative permittivity of a substance, $\varepsilon_r$, is the ratio of the static permittivity of the substance, $\varepsilon_s$, to the static permittivity of a vacuum, $\varepsilon_0$ (Fernandez et al. 1995). The relative permittivity of a substance is used for practical purposes in the design of capacitors. The behavior of the relative permittivity of water is related to its physical state or phase (as a liquid or as vapor), temperature, and pressure. Experimentally verified relative permittivity values for

water in its solid phase (as ice) at temperatures as low as 190K ($-83°C$) exist (Matsuoka, Fujita, and Mae 1996), however these data do not include corresponding pressure values for any of the measurements. Water, in its liquid or vapor (steam) state, exists within a large range of temperatures and pressures, and this range has been traditionally divided into 4 regions, A, B, C, and D. Region A is the normal liquid water state between the normal freezing and boiling points (~273K to ~373K) at pressures up to 1000MPa. Region B refers to water along the liquid-vapor phase boundary. For water located in this region of the thermodynamic space, in contrast to all other regions, every pair of temperature/pressure values takes on two density values (corresponding to the phase, either liquid or vapor/steam, in which the water occurs). Region C is the region with a temperature above 373.15K. At lower pressures and temperatures within region C, water is in the normal vapor (steam) state. At higher pressures and temperatures in this region, water becomes a supercritical fluid, that is, water ceases to behave as if it were in either the liquid or vapor state, but rather exhibits a combination of the thermodynamic properties attributable to both liquids and gases. Finally, region D refers to super cooled water (water that exists in the liquid state below the normal freezing point of 273.15K at the standard pressure of ~.1MPa).

The behavior of the relative permittivity exhibits discontinuities along the liquid-vapor phase boundary (region B) and in the supercritical part of the region above the normal boiling point (region C), with very small changes in the temperature and pressure causing very large changes in density and in the value of the relative permittivity (Harvey 2006). As a result, theoretical formulations for calculating the relative permittivity of water have mainly focused on a broad range of temperatures (~270K to ~1000K) and a narrow range of pressures (~.1MPa to 100MPa) (Fernandez et al. 1995). Furthermore, data points along the phase boundary (region B),

although numerous, have not had their pressure or density values recorded, and thus have not figured in any data-driven correlations that correct for pressure and density differences. The most current formulation for approximating the relative permittivity across the entire range of experimental temperatures and pressures may be found in (Fernandez et al. 1997) and is also reproduced in the results section. Fernandez et al.'s formulation uses an extensive adaptive regression algorithm to create an appropriate function taking a wide variety of domain specific thermodynamic values into account. Furthermore, they analyze the first, second, and third derivatives of the change in the dielectric constant with respect to both temperature and pressure inputs in order that their function accords with theoretical considerations of how the static dielectric constant is to behave across all thermodynamic regions. The final function uses 5 adjustable parameters and a total of 25 constants and domain specific non-adjustable parameters and approximates well across the entire range of experimentally available values.

## 4.3 BACKGROUND: ARTIFICIAL EVOLUTION AND GENETIC PROGRAMMING

Genetic Programming (GP) may be seen as an abstract algorithmic implementation broadly inspired by the main principles of Darwin's theory of evolution by means of natural selection. Roughly, Darwinian evolutionary theory involves populations of interbreeding organisms (species) competing for environmental resources over time. Species share genetic material by interbreeding, and random mutations occur to members of the species that may either hinder or further their reproductive success. As the members of a given species breed and reproduce over time, characteristics beneficial for the species' survival propagate throughout the population, while those characteristics that are detrimental to the survival of the species do not get expressed in the population. That is, individuals with characteristics that favor their survival within the given environment tend to propagate, whereas individuals not possessing those

characteristics in the environment (or those that exhibit detrimental characteristics) tend to die out.

GP applies the broad tenets of Darwinian evolutionary theory within a heuristic framework that attempts to create automatically generated computer programs that evolve to optimally solve user-defined problems (Koza 1992). GP is an extension of the evolutionary computational approach known as genetic algorithms (GA) first pioneered by John Holland (Holland 1992) (see also Forrest 1993). Within the GP framework, a population of candidate solutions, with each solution representing an executable computer program of some finite length (an individual of a given population), evolves in response to some problem to be solved (the environmental conditions) (Koza 1992). Each GP individual/candidate computer program within the population is given a fitness value that is the output of a function (the fitness function) that determines the appropriateness or optimality of the program output (individual behavior) when given the user-defined problem (the environmental conditions). This allows each individual within the GP population to be measured against every other individual, whether the individual solves the problem (favorably responds to the environment) or not. Once all of the individuals within a population have been assigned a fitness value, certain individuals are stochastically chosen to recombine and create offspring based on their fitness values, so that individuals with higher fitness values tend to be chosen more frequently for recombination. During recombination two unique individuals are chosen to represent the parents, and may stochastically recombine to generate two offspring. Occasionally, however, (because recombination is probabilistic and does not always occur) they do not recombine and remain unchanged as offspring. After every recombination event, an offspring individual may be mutated with some small probability. The series of steps following initial population generation include parent selection, recombination,

and mutation of offspring, and constitutes a generation of the GP run. At the start of every generation, newly created individuals in the population are evaluated by the fitness function and assigned a fitness value. The GP run continues in this manner (after the generation of the initial population, only fitness value assignment, parent selection, recombination, and mutation of offspring occur) until some stopping criteria (such as the creation of an individual with either some given minimum or maximum fitness value, or one that adequately solves the problem at hand) has been reached.

Each GP individual uses a tree-based data structure representation scheme, where the tree structure completely represents a given program. A tree structure resembles a company organization chart with a root node (president), subtrees (subordinate divisions under the president managed by division chiefs), and continuing down the branches until reaching leaf nodes (nodes without subordinates). Nodes for the GP program tree either come from the terminal set or the function set (both predefined by the individual implementing the GP search). The terminal set completely defines the kinds of inputs (independent variables) the evolving computer programs (individuals) can use to solve the problem. The elements of the terminal set can only occur as leaf nodes within the program tree (that is, nodes that have no children). The function set defines the kinds of transformations that are permissible given any of the elements in the terminal set or any of the other elements within the function set as arguments to each of the elements within the function set. Thus, the elements of the function set may only occur as the internal nodes of a GP-generated program tree (nodes with at least one child node). These restrictions amount to the fact that the union of the function and terminal sets of a GP implementation must possess the property of closure (where closure is defined as the ability to have any composition of functions and terminals produce a syntactically correct, executable

computer program) (Ghanea-Hercock 2003). The program trees generated using GP do not have to be standard binary trees (trees where every node is either a leaf node, or has a maximum of two child nodes), as the experimenter may define a function operator within the function set that takes more than two arguments. Initially, GP individuals are randomly generated through a stochastic tree-building process where each node in the tree is chosen to be a random member of either the function or terminal sets. Traditionally, GP candidate programs are initially generated either strictly to some maximum initial tree depth limit (where all nodes up to the maximum initial tree depth are chosen stochastically exclusively from the function set and all nodes at the maximum initial depth limit are chosen exclusively from the terminal set), or until all of the branches of the tree have either gone to the maximum initial depth or have ended in terminal nodes before the maximum initial tree depth has been reached.

The genetic programming operators of crossover and mutation, as well as the way in which individuals are ranked according to their fitness level are modified from the genetic algorithm (GA) approach (described in detail in Holland 1992) to suit the GP technique. Crossover occurs by selecting two nodes on different parent trees and then swapping the corresponding subtrees, that is, all of the descendants of the selected nodes (as well as the selected nodes themselves) between the two individuals.

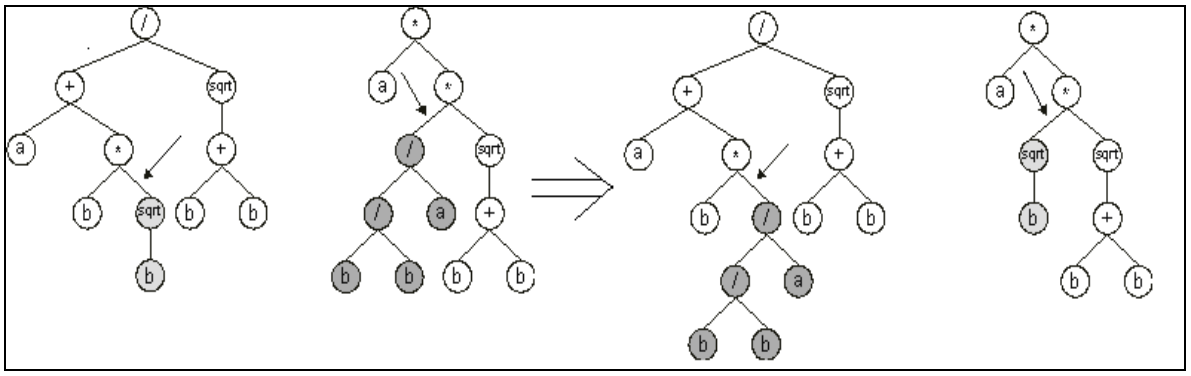

**Figure 4.1: GP Crossover**

Mutation, on the other hand, involves selecting a node at which mutation will occur, deleting all of the nodes that are descendants of the selected node, and then generating a random subtree with this node as its root.

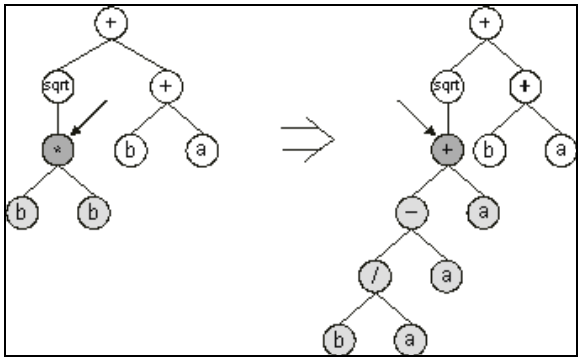


**Figure 4.2: GP Mutation**

The fitness evaluation and ranking methods in GP are slightly different from the classic GA approach (where fitness maximization is standard) in the sense that the highest ranking individual programs in GP have the lowest fitness values (in effect, a minimization problem). Thus, GP attempts to find a computer program with the globally minimal fitness value in the search space of all possible computer programs that may be created using the function and terminal sets used in the problem, to the tree depth or program length specified in the GP setup.

Ultimately, the GP approach involves determining a set of functions and terminals to be used in solving the problem, defining a fitness measure by which individual programs may be evaluated and assigned a fitness value, setting the specific parameters and operator probabilities that are involved in program tree generation (crossover and mutation probabilities, initial tree depth limit, maximum tree length, etc.), and developing a set of rules or stopping criteria to determine when to end a specific GP run (whether after a certain number of generations have elapsed, or after an individual program with a desired fitness threshold has been found).

## 4.4 EXPERIMENTAL SET-UP

In our experiments, a variety of different function and terminal sets were explored in an effort to evolve a single function that could model the relative permittivity of water as a function of pressure, temperature, and density in thermodynamic regions A, C, and D of the temperature-pressure phase space. Unfortunately, no temperature *and pressure* data for region B (along the phase boundary) is currently available (Fernandez et al. 1995), and thus two functions approximating the dielectric constant in region B (one for vapor saturation and one for liquid saturation) taking only temperature into account were evolved separately.

The suite of functions for regions A, B, C, and D was evolved using experimentally verified input data sets taken from (Fernandez et al. 1995) and output results were then compared to relative permittivity values experimentally observed from the same input temperature/pressure/density values, or only input temperature values, as in the case of region B, (taken from the same data sets) using the newest formulation for dielectric constant prediction, found in (Fernandez et al. 1997). These data sets were compiled from all previous experimentally available data, and were then corrected by Fernandez et al. to coincide with the most recent internationally accepted temperature scale, ITS-90. In most cases, values were provided for the temperature (in degrees Kelvin, or K), pressure (in megapascals, or MPa), and the corresponding dielectric constant. However, in some cases, temperature/density/dielectric constant values were given instead of temperature/pressure/dielectric constant values. In these circumstances, density values were converted into their corresponding pressures, and pressure values were converted to their corresponding densities using the IAPWS-95 formulation for the equation of state of water found in (Wagner and Pruss 2002). With this completed, the final data set uniformly represented

the dielectric constant at every temperature, pressure, and density value that was both accurate and experimentally available (as of December 2006).

Our functions were evolved by generating a population of possible functions (represented as trees) as with standard genetic programming implementations. Each candidate function's fitness was taken to be the sum of the absolute errors between the computed and the experimentally measured value for the relative permittivity at every input value in the corresponding data set. The combination of input values for each function (that is, what combination of the three possible adjustable inputs was to be used) was determined by the GP module. The population of possible functions was then evolved with a variety of crossover/mutation probabilities and function sets. The data set of experimentally obtained relative permittivity values used to create the function consisted of 771 data points, which represent the complete dataset that Fernandez et al. recommend for data correlations for regions A,C, and D (644 data points) and region B (127 data points) (Fernandez et al. 1995). The function with the lowest sum of absolute errors across the data points that was found after all runs had been completed was chosen as the final function for each part of the formulation.

During any given GP run, all function and terminal sets used during function evolution always included addition, subtraction, multiplication, and division as function operators, and temperature, $T_k$, pressure, $p$, and density, $\rho$, as terminal values. In cases where a generated function divided a value by zero, the zero-generating term was replaced by 0.00001. All runs used a population of 100 random floating-point constants in the range between 0 and 1, which were generated at runtime. These constants would play the role of additional terminal values for the genetic program to use during function creation. Other function operators (sin(), cos(), ln(), $\log_{10}$, $\log_2$, and $x^y$) and terminal operators (Avogadro's number, $N_A$, permittivity of free

space, $\varepsilon_0$, elementary charge, $e$, Boltzmann's constant, $k$, molar mass of water, $M_w$, mean molecular polarizability of water, $\alpha$, the dipole moment of water, $\mu$) were also used in various GP runs. The aforementioned terminal operators are reproduced from (Fernandez et al. 1997) in table 4.1. The function length of any individual solution (a tree representing a given candidate function) never exceeded 100 functional units (where a functional unit is taken to be a single operator from the function set or a terminal value from the terminal set). The large size of the function and terminal sets causes the size of the search space (representing all of the possible unique programs of length 100 or less that can be generated from the function and terminal sets) to be enormous (more than a googol possible syntactically correct possible functions). As a result, each GP run was done on a population of two million individuals that were evolved for 200 generations. This was done to ensure that the GP implementation would sample as much of the search space as possible in its effort to find a suitable function within a reasonable time. A range of crossover probabilities (between .5 and 1.0, in increments of .05) and mutation probabilities (between 0 and .5, in increments of .05) were explored for all combinations of function and terminal sets. Each combination of unique parameter settings was implemented in 10 GP runs, after which the function with the lowest total absolute error was chosen.

## 4.5 RESULTS

The optimal functions that were evolved were found during runs that used multiplication, division, subtraction, and addition as operators in the function set and temperature, pressure, and density as terminal operators (with the 100 additional random ephemeral constants described earlier). The best function runs used a crossover probability of 0.9 and a mutation probability of 0.05. The final evolved functions along with Fernandez, et. al.'s formulation, follow:

$$\varepsilon_{A,C,D} = \frac{\dfrac{\rho^2+\rho^3}{-.02036T_K\rho+.0864\rho^2+.1194T_K p} + \dfrac{(-6.75862p^2+.313T_K p-{T_K}^2)(1+\rho)}{T_K}}{55.474T_K p+55.55p^2-.076T_K p\rho+.016p+.016p\rho+\rho+\rho^2} + \frac{\rho^2}{.03264T_K(T_K+\rho)} - \frac{\dfrac{{T_K}^2}{\rho^2}+2.617p-1.617\rho+1.617T_K}{T_K+.0486-p+\rho}$$

**Figure 4.3: GP-Evolved Equation, regions A, C, and D**

$$\varepsilon_{B,liquid} = \frac{0.0209}{0.2038-\dfrac{88000+\dfrac{{T_K}^3-0.2864{T_K}^2}{{T_K}^2-0.8234}}{{T_K}^2+0.0896T_K}} + 0.1553 - \frac{87968.427}{{T_K}^2+0.97T_K+0.196} * \frac{-0.31093*(T_K+0.567)*(T_K+1.6)}{T_K}$$

**Figure 4.4: GP-Evolved Equation, region B, liquid saturation**

$$\varepsilon_{B,vapor} = \frac{T_K-0.33439}{-0.0015136{T_K}^2+2T_K-1.08451-\dfrac{0.108579T_K}{-0.003027{T_K}^2+1.8963794T_K+1.4214339}} + \frac{0.1085797T_K}{-0.003027{T_K}^2+2T_K+1.05097} + 0.067589$$

**Figure 4.5: GP-Evolved Equation, region B, vapor saturation**

$$\varepsilon_r = \frac{1 + 5A + 5B + \sqrt{9 + 2A + 18B + A^2 + 10AB + 9B^2}}{4 - 4B}$$

*where A and B are given by*

$$A = \frac{N_A \mu^2}{\varepsilon_0 k} \frac{\rho g}{T_k}$$

$$B = \frac{N_A \alpha}{3\varepsilon_0} \rho$$

*and where g is given by*

$$g = 1 + \sum_{k=1}^{11} N_k \left(\frac{\rho}{\rho_c}\right)^{i_k} \left(\frac{T_c}{T}\right)^{j_k} + N_{12} \left(\frac{\rho}{\rho_c}\right) \left(\frac{T}{228K} - 1\right)^{-q}$$

*with* $\rho_c = \frac{322}{M_w}$ *and* $T_c = 647.096K$ *and values for* $N_k$, $i_k$, $j_k$, *and q given in table 4.2.*

**Figure 4.6: Fernandez' formulation, reproduced from (Fernandez et al. 1997)**

The results of applying the GP-evolved functions and Fernandez et al.'s formulation to the total data set are found in tables 4.3 through 4.6. The evolved functions shown above are significantly smaller than the formulation developed by Fernandez et al. (31 terms for the regions A,C, D function, 24 terms for the liquid saturation function, and 17 terms for the vapor saturation function versus 112 terms for Fernandez et al.'s formulation) and uses only three adjustable parameters (temperature, pressure, and density), zero non-adjustable domain specific parameters, and only fifteen of the one hundred possible random ephemeral constants that were available during function evolution. No domain-specific knowledge (aside from the data sets themselves) was applied to the formulation of the suite of functions.

As can be seen from tables 4.3 through 4.5, the evolved function performed comparably to Fernandez et al.'s formulation in all collected statistical categories except the minimum

absolute difference, where each function had at least one data point where very marginal absolute error (<0.01) existed. However, it must be mentioned that our proposed formulation and Fernandez et al.'s formulation differ in the way each was generated. Fernandez et al. did not use the entire set of data points that they recommended for data correlation, but rather a significantly reduced subset of this total dataset (127 data points out of the total 771 data points for all four regions). Table 4.6 compares the accuracy of our formulation to that of Fernandez et al. on this reduced dataset. As should be expected, our formulation does not perform as well as Fernandez et al.'s on these selected data points, but does have a smaller maximum absolute difference across the data set. This dataset does not cover the entire thermodynamic space, and thus, Fernandez et al. used a variety of theoretical considerations to buttress the sparseness of their dataset. In the case of region B, Fernandez et al. also provide pressure and density values for the data points that they used in constructing their formulation, even though no experimental pressures and densities for this region exist. Furthermore, Fernandez et al. weighted each data point in their dataset differently, based on certain theoretical and experimental factors. Although it was our goal to create a formulation following steps as similar to Fernandez et al.'s as possible, we could not follow this aspect of their experimental methodology. This approach entailed creating copies of each data point in proportion to its weight and then reinserting those copies back into the data set or multiplying the error on each data point in proportion to its weight. This seems tractable until one realizes that the weights of some data points are very small (<.01, where the sum of the weights of all 127 data points is 100). As a result, certain data points with much larger weight values (weights between 1 and 3) would skew the evolutionary search to find functions that approximate those points well, but approximate the low-weight data points very poorly. Furthermore, this would come at the expense of not covering the entire thermodynamic

space (even when other experimental points that do cover the thermodynamic space exist). Thus, we decided to leave the weights of all data points equal, and to use all of the experimental data that was available and recommended for correlation.

## 4.6 CONCLUSIONS AND FUTURE WORK

A suite of functions that approximate the relative permittivity of water and steam across the entire experimentally verified range of temperatures and pressures have been developed. These functions were evolved using the GP technique with a specific function and terminal set, and their accuracy has been compared to that achieved by Fernandez et al.'s most recent formulation. This approach uses no theoretical domain-specific knowledge to obtain a useable function. The evolved functions approximate the relative permittivity of water and steam extremely well, comparing favorably with Fernandez et al.'s formulation across the entire experimentally available temperature and pressure range, while being simpler computationally. Further refinements to create more accurate approximations of the relative permittivity of water and steam will include creating a single evolved function that can be used across all thermodynamically distinct temperature and pressure regions, including regions where water is in the solid phase, or where a phase boundary exists. This can be done when experimental values for the temperature, pressure, and relative permittivity in these regions (especially region B) are obtained. A refined fitness function that takes more than the absolute difference between expected and calculated values may also prove useful in creating a new, more accurate formulation. Introducing a penalty for very large and difficult to read functions may also help in finding a function that is both compact and generalizes well across the entire thermodynamic space. However, significant improvements to the evolution of an appropriate function will most surely come from an increase in experimentally verifiable values for the relative permittivity, and thus any new accurate data that may be found should be used to refine the current formulation.

## 4.7 TABLES

**Table 4.1: Constants used in the relative permittivity formulation, reproduced from (Fernandez et al. 1997)**

| Parameter | Value |
|---|---|
| Permittivity of free space, $\varepsilon_0$ | $[4*10^{-7}\pi(299792458)^2]^{-1}C^2J^{-1}m^{-1}$ |
| Elementary charge, $e$ | $1.60217733*10^{-19}C$ |
| Boltzmann's constant, $k$ | $1.380658*10^{-23}JK^{-1}$ |
| Avogadro's number, $N_A$ | $6.0221367*10^{23}mol^{-1}$ |
| Molar mass of water, $M_w$ | $0.018015268kg*mol^{-1}$ |
| Mean molecular polarizability of water, $\alpha$ | $1.636*10^{-40}C^2J^{-1}m^{-2}$ |
| Dipole moment of water, $\mu$ | $6.138*10^{-30}Cm$ |

**Table 4.2: Coefficients $N_k$, and exponents $i_k$, $j_k$, and q of the equation for g, reproduced from (Fernandez et al. 1997)**

| $k$ | $N_k$ | $i_k$ | $j_k$ |
|---|---|---|---|
| 1 | $0.978224486826$ | 1 | 0.25 |
| 2 | $-0.957771379375$ | 1 | 1 |
| 3 | $0.237511794148$ | 1 | 2.5 |
| 4 | $0.714692244396$ | 2 | 1.5 |
| 5 | $-0.298217036956$ | 3 | 1.5 |
| 6 | $-0.108863472196$ | 3 | 2.5 |
| 7 | $0.949327488264*10^{-1}$ | 4 | 2 |
| 8 | $-0.980469816509*10^{-2}$ | 5 | 2 |
| 9 | $0.165167634970*10^{-4}$ | 6 | 5 |
| 10 | $0.937359795772*10^{-4}$ | 7 | 0.5 |
| 11 | $-0.123179218720*10^{-9}$ | 10 | 10 |
| 12 | $0.196096504426*10^{-2}$ | | *q=1.2* |

**Table 4.3: Results and numeric comparison, regions A,C,D**

| REGIONS A,C,D | Evolved GP result | Fernandez |
|---|---|---|
| Sum Absolute Difference | 103.12 | 95.20 |
| Mean Absolute Difference | 0.16 | 0.15 |
| Standard Deviation Absolute Difference | 0.25 | 0.22 |
| Sum Squared Difference | 55.80 | 44.30 |
| Mean Squared Difference | 0.09 | 0.07 |
| Standard Deviation Squared Difference | 0.54 | 0.54 |
| Minimum Absolute Difference | 0.00 | 0.00 |
| Maximum Absolute Difference | 3.55 | 3.60 |
| # Data Points Absolute Difference formulation < Absolute Difference Fernandez | 330 | |
| % Total Data Points better than Fernandez | 50.93% | |
| Total Data Points | 644 | |

**Table 4.4: Results and numeric comparison, region B, liquid saturation**

| REGION B, liquid | Evolved GP result | Fernandez |
|---|---|---|
| Sum Absolute Difference | 27.11 | 37.49 |
| Mean Absolute Difference | 0.22 | 0.30 |
| Standard Deviation Absolute Difference | 0.18 | 0.33 |
| Sum Squared Difference | 9.74 | 24.64 |
| Mean Squared Difference | 0.08 | 0.20 |
| Standard Deviation Squared Difference | 0.13 | 0.65 |
| Minimum Absolute Difference | 0.00 | 0.00 |
| Maximum Absolute Difference | 0.90 | 2.56 |
| # Data Points Absolute Difference formulation < Absolute Difference Fernandez | 71 | |
| % Total Data Points better than Fernandez | 56.35% | |
| Total Data Points | 126 | |

**Table 4.5: Results and numeric comparison, region B, vapor saturation**

| REGION B, vapor | Evolved GP result | Fernandez |
|---|---|---|
| Sum Absolute Difference | 0.273 | 0.335 |
| Mean Absolute Difference | 0.007 | 0.009 |
| Standard Deviation Absolute Difference | 0.016 | 0.017 |
| Sum Squared Difference | 0.011 | 0.013 |
| Mean Squared Difference | 0.0003 | 0.0004 |
| Standard Deviation Squared Difference | 0.001 | 0.001 |
| Minimum Absolute Difference | 0.00 | 0.00 |
| Maximum Absolute Difference | 0.075 | 0.074 |
| # Data Points Absolute Difference formulation < Absolute Difference Fernandez | 22 | |
| % Total Data Points better than Fernandez | 59.46% | |
| Total Data Points | 37 | |

**Table 4.6: Results and numeric comparison, reduced dataset used by Fern for correlation, ALL REGIONS**

| REGIONS A,B,C,D | Evolved GP result | Fernandez |
|---|---|---|
| Sum Absolute Difference | 26.25 | 16.18 |
| Mean Absolute Difference | 0.21 | 0.13 |
| Standard Deviation Absolute Difference | 0.37 | 0.34 |
| Sum Squared Difference | 22.70 | 16.38 |
| Mean Squared Difference | 0.18 | 0.13 |
| Standard Deviation Squared Difference | 1.13 | 1.15 |
| Minimum Absolute Difference | 0.00 | 0.00 |
| Maximum Absolute Difference | 3.55 | 3.60 |
| # Data Points Absolute Difference formulation < Absolute Difference Fernandez | 40 | |
| % Total Data Points better than Fernandez | 31.50% | |
| Total Data Points | 127 | |

## 4.8 REFERENCES

Fernandez, D.P., Y. Mulev, A.R.H. Goodwin, and J.M.H. Levelt-Sengers. 1995. A Database for the Static Dielectric Constant of Water and Steam. *Journal of Physical and Chemical Reference Data* 24(1): 33-69.

Fernandez, D.P., A.R.H. Goodwin, E.W. Lemmon, J.M.H. Levelt-Sengers, and R.C. Williams. A Formulation for the Static Permittivity of Water and Steam at Temperatures from 238K to 873K at Pressures up to 1200MPa, Including Derivatives and Debye-Huckel Coefficients. *Journal of Physical and Chemical Reference Data* 26(4): 1125-1166.

Fogelson, S., and W. Potter. 2007. A GP-Evolved Formulation for the Relative Permittivity of Water and Steam. Submitted to *IEA-AIE07.*

Forrest, S. 1993. Genetic Algorithms: Principles of Natural Selection Applied to Computation. *Science* 261:872-878.

Ghanea-Hercock, R. 2003. *Applied Evolutionary Algorithms in Java.* New York, NY: Springer-Verlag.

Harvey, A. 2006. NIST. Personal communication.

Holland, J. 1992. *Adaptation in Natural and Artificial Systems: 2nd Edition.* Cambridge, MA: MIT Press.

Koza, J.R. 1992. *Genetic Programming.* Cambridge, MA: MIT Press.

Matsuoka, T., S. Fujita, and S. Mae. 1996. Effect of temperature on dielectric properties of ice in the range 5-39 GHz. *Journal of Applied Physics* 80(10): 5884-5890.

Quist, A.S., and W.L. Marshall. 1965. Estimation of the Dielectric Constant of Water to $800°$. *Journal of Physical Chemistry* 9: 3165.

Wagner, W. and A. Pruss. 2002. The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use. *Journal of Physical and Chemical Reference Data* 31(2): 387-535.

# CHAPTER 5

# COMPARING MACHINE LEARNING TECHNIQUES IN PREDICTING TRANSLATION START SITES IN PROKARYOTIC GENOMES[4]

[4] S.V. Fogelson, K. Rasheed, X. Guo, and J. Mrázek. To appear in *Proceedings of the International Conference on Machine Learning: Models, Technologies, and Applications, MLMTA '07.* Reprinted here with permission of the publisher, 6/18/2007.

ABSTRACT

Accurate discovery of translation start sites in prokaryotic genomes remains an open problem. We compare the performance of several different machine learning techniques on a database of experimentally verified translation start sites from two different prokaryotic genomes (E. coli and Synechocystis PCC6803). The SVM, ANN, and XCS learning algorithms performed well on the database. Furthermore, XCS has never been used in any kind of computational biological approach, and we provide evidence that it is an effective new tool to be utilized in this field. The reasons for why each of these techniques performed well are explored, and possible directions for future work in this area are offered.

## 5.1 INTRODUCTION

Gene discovery in genomics is a rapidly developing field. With the recent mapping of the human genome and the genomes of a significant number of other species within the past decade, genetic discovery is only just beginning. However, after the mapping of a given genome has been completed, pinpointing exactly where genes begin on the genome becomes a costly, time-consuming process. Whereas translation stop sites are unambiguously determined by the first in-frame stop codon (with very few exceptions), translation start sites are generally ambiguous, and the most accurate predictions are achieved by combinations of automated predictions and manual curation. Needless to say, this process is time consuming and automating the discovery of the actual translation start sites would greatly reduce the amount of time and effort spent on this task. In this paper, we compare the performance of several machine learning techniques (XCS, SVM, C4.5, Naïve Bayes, and Artificial Neural Network) on a database of prokaryotic gene start sequences taken from two distinct prokaryotic genomes (*E. coli* and *Synechocystis* PCC6803). XCS (an extension of the standard learning classifier system approach), SVM (support vector machines), and ANN (artificial neural networks) perform very well on this task, with SVM achieving the highest overall percentage of correctly classified instances and correctly classified true negative instances (TN), and ANN achieving the highest percentage of correctly classified true positive instances (TP). The reasons for this discontinuity and directions for future work are offered.

## 5.2 RELATED WORK

Several prior attempts for detecting gene start sequences in the genomes of both prokaryotic and eukaryotic organisms have been made, and the effective automation of the task still remains an open problem. In eukaryotes, most efforts center on prediction of transcription

start sites. Bajic and Seah (Bajic and Seah 2003) describe an attempt to predict transcription start sites (or promoter sequences) on human chromosomes 4, 21, and 22. This approach used an Artificial Neural Network (ANN) to predict promoter sequences based on the existence of CpG islands, or segments of DNA with relatively high (>50%) concentrations of CG dinucleotides. CG dinucelotides are generally methylated in human DNA and CpG islands comprise non-methylated CG dinucleotides associated with promoters of actively transcribed genes. Information regarding the presence of these sites along the chromosomes was fed into an ANN, which then predicted the most likely areas where the RNA polymerase binds to DNA. The system was able to predict close to half of the actual transcription start sites, and limited its rate of false positive predictions to less than half that of other algorithms.

In prokaryotes, accurate prediction of both transcription and translation start sites are unresolved problems. Probably the most efficient algorithm to predict translation start sites was developed by Besemer, Lomsadze, and Borodovsky (Besemer, Lomsadze, and Borodovsky 2001). Theirs was an unsupervised learning approach that combines gene predictions by GeneMark.hmm (Lukashin and Borodovsky 1998) with statistical models for ribosome binding sites. The method accurately predicted translation start sites for 83.2% and 94.4% on testing datasets from *B. subtilis* ad *E. coli*, respectively, but it may be lower in genomes where fewer genes have recognizable ribosome binding sites.

## 5.3 MACHINE LEARNING TECHNIQUES USED

Support vector machines, XCS (a modified version of traditional learning classifier systems), C4.5 (a decision tree algorithm), artificial neural networks, and naïve bayes are all compared in this paper on their ability to predict gene start sequences. Support vector machines are a kernel-based approach for developing a linear classifier that searches for the maximum-

margin hyperplane (if one exists) between two disjoint classes. SVMs were originally described in (Boser, Guyon, and Vapnik 1992), and a voluminous literature exists on the topic. XCS is an extension of the traditional learning classifier system (hereon, LCS) approach first proposed by John Holland and described in (Holland 1992). LCS evolves a population of rules based on each rule's ability to predict adequate behavior within some environment. XCS is described in (Wilson 1995) and extends LCS by making each rule's fitness value dependent on the accuracy of the rule with respect to the kinds of actions it recommended during earlier training pattern presentations. C4.5 is a decision tree algorithm designed by Quinlan and described in (Quinlan 1993) that improves upon the ID3 decision tree algorithm in order to address overfitting of a data set. C4.5 attempts to generate an optimal decision tree for classifying the data set it is trained on, and does so with aggressive tree pruning and it may be extended to handle attributes with continuous values. Artificial neural networks, described in (Rumelhardt and McClelland 1986), are a machine learning technique developed as a universal function approximating method that attempts to minimize the squared error between expected output and the actual output of the network by modifying the weights between the connections of the artificial neurons within the model. Finally, naïve bayes, described in (Rish 2001) is a statistical learning technique that calculates the posterior probability of any given possible output class (maximum likelihood) by utilizing the prior probability of the class and the features (variables) used to classify each given data pattern within a data set.

## 5.4 EXPERIMENTAL SET-UP

In order for the current investigation to be carried out, an extensive data preprocessing phase occurred before the data set of gene starts was fed, as appropriately formatted numerical data, into the machine learning algorithms. Originally, a set of genes with experimentally

verified gene-starts was compiled from the genes' respective protein (amino acid) sequences. This was done by querying the NCBI database of complete prokaryotic genomes of DNA sequences for homologous genes with a given protein sequence using the TBLASTX version of the BLAST algorithm (Altschul et al. 1990). Once the homologous genes for a given query were obtained, DNA sequences spanning 200 base pairs (hereon, bp) upstream and 600bp downstream (or to the end of the gene, if less than 600bp long) from the original translation start site were extracted from the original query sequence and all homologous sequences. Homologous sequences were then aligned with the query sequence using the CLUSTALW algorithm (Thompson, Higgins, and Gibson 1994). The alignment was further processed by removing all columns corresponding to gaps in the query sequence. Once aligned, the information content at each position *i* in the alignment was calculated using Shannon's information gain function:

$$R_i = 2 + \sum_{j=A,C,G,T} p_j \log_2 p_j$$

Where $p_j$ is the frequency of nucleotide *j* at position *i*. The values of $R_i$ range from 0 (when all four nucleotides are equally likely at position *i*) to 2 (when a single nucleotide appears in position *i* for all aligned sequences). A normalized 3bp autocorrelation function (since 3bp code a single amino acid) for $R_i$ was then derived using the following equation:

$$C_i = R_i R_{i+3} - \frac{1}{2}(R_i R_{i+2} + R_i R_{i+4})$$

The functions $R_i$ and $C_i$ were then converted into the corresponding cumulative functions (hereon Cumulated and Correlated, respectively):

$$R_m = \sum_{i=0}^{m} R_i \qquad C_m = \sum_{i=0}^{m} C_i$$

At this point, the reading frame was selected to be of length 60 and the reporting frequency was set to 1. Thus, for every alignment, two distinct patterns were created where each pattern was of length 120, corresponding to 60bp upstream and 60bp downstream from all possible translation start sites (initially identified as any ATG or GTG codon in the query sequence). 3 additional values, obtained with the same $R_i$ equation, were appended to each respective $R_m$ and $C_m$ pattern and corresponded to the level of conservation of the start codon itself for every alignment. Positive examples correspond to the experimentally verified translation start sites whereas all other ATG and GTG codons in the vicinity represent negative examples. In this manner, two total data sets of 1310 patterns of 124 attributes each were generated (123bp plus the target attribute, corresponding to the presence or lack of a verified translation start site) using the correlated and cumulated schemes previously described, with each data set containing 260 positive and 1050 negative examples.

Following this, because of the high dimensionality of the data set and the sparseness of the positive examples (123 attributes excluding the target binary attribute and 260 positive examples), the dimensionality of the data set was significantly reduced using the WEKA machine learning package, described in (Witten and Frank 2005) and freely available online through Waikato University, to prevent overfitting of the data and to automate the relevant feature extraction process. WEKA is an incredibly versatile machine learning package that implements a large number of different machine learning algorithms as well as having feature extraction, data clustering, and meta-learning capabilities. Dimensionality reduction was done by running a genetic algorithm (Holland 1992) built into the package that evolved a population of support vector machines, each of which used a random subset of the total attribute set. Genetic algorithms (GA) are global optimization techniques inspired by Charles Darwin's theory of

natural selection and attempt to evolve an optimal solution to a given problem from a population of potential solutions for that problem. The genetic algorithm used one point crossover, roulette wheel parent selection, bit flip mutation, a population size of 200, and was evolved for 200 generations. The SVMs underwent 10-fold cross validation (90% of each data set was used for training, 10% was used for testing) during the evolutionary run to make sure that overfitting of the data did not occur. The evolutionary procedure was repeated 10 times and the set of attributes that consistently ranked in the top 30% of the attribute subsets (those attributes that were chosen at least 70% of the time in the trained SVMs with the highest accuracy on the cross-validation set) were chosen from each data set to be used in the next stage of model development. Preliminary experiments showed that using this percentage was an effective way to reduce the dimensionality of the data sets.

With the decrease in dimensionality, the next task was discretizing the attributes (which were all initially real-valued) so that they may be appropriately incorporated into the XCS framework. Recently, XCS has been extended for handling real-valued classification in (Wilson 2000). However, no implementation of the real-valued approach is currently available, and as a result, discretization of the attributes was a necessity. All other machine learning techniques used could handle real-valued attributes, and in their case, no further processing was required.

Discretization of the attributes was done so that the mean and standard deviation of each attribute across all patterns was calculated, and the range of possible values was divided according to where a given value fell in relation to the mean and to a certain number of standard deviations from the mean. Thus, values that were further than two standard deviations to the left of the mean were assigned a value of 0, those that fell between two and one standard deviation to the left of the mean were assigned a value of 1, and those that fell between one standard

deviation to the left of the mean and the mean itself were assigned a value of 2. The same procedure was used to discretize values that were to the right of the mean, and these were placed into their appropriate ranges and assigned discrete values between 3 and 5. This, in turn, caused different attributes to have different discrete ranges of possible values as certain attributes had no values that, for example, fell two standard deviations to the left or the right of the mean value. However, this only affected a small number of the attributes.

Once the discretization process was completed, each data set was incorporated into a modified version of Martin Butz' Java implementation of XCS (Butz 2000) that could handle wider discrete-valued range of condition values than the standard '1', '0', and '#' setup. The data sets were treated as the total environment in which the XCS was evolved, and a state of the environment corresponded to a given pattern (DNA segment) being presented to the classifier system. With all of the necessary preprocessing stages complete, the XCS implementation was ready to be run.

Initially, the XCS implementation was run given an extensive range of different possible parameter settings to determine the optimal size of the population of classifiers, the learning rate used to update each individual rule's accuracy based fitness measure, the probability of assigning a '#', or "don't care" value, during the initial creation of a random classifier, the probabilities of crossover and mutation during the GA-mediated rule discovery mechanism, and the number of pattern presentations (environmental states) needed until no additional improvement in the accuracy of the classifier system occurs. These preliminary explorations were done for both data sets and yielded the same parameter settings, with 10-fold cross validation being used to ensure that parameter tuning was not being affected by overlearning of the data set. With parameter tuning complete, the XCS was run on each data set twenty times and average total percentage

correct, true positive, false positive, true negative, and false negative rates were all recorded. Finally, these results were compared to three other machine learning algorithms implemented in WEKA- support vector machines, naïve bayes, and the WEKA implementation of the C4.5 decision tree algorithm, and to a separately implemented artificial neural network package provided by Brian Smith of the UGA AI Center. This ANN package was used because of its modularity and ease of incorporation into the current approach. In the case of the WEKA-implemented algorithms, the default settings for the algorithms were used because tuning the parameter settings for these techniques to locate optimal values would have both taken a prohibitively large amount of time (several days for each algorithm tried), and because some initial experimentation with parameter settings in these cases showed that the improvement in performance was marginal. In the case of the neural network, the only parameter tuned was the number of nodes in the hidden layer, with the learning rate set at a default 0.3, and the momentum term set to a default 0.2. For the non-XCS machine learning algorithms a single run took a much shorter time to execute than the XCS. As a result, a single run where results could be accurately compared was taken to be the run that yielded the best of the aforementioned values out of a set of back-to-back runs of the given method (with different initial random seed values, where appropriate). Furthermore, the time to execute the set of back-to-back runs was equal to the time necessary to execute a single XCS run. Again, this was done on each data set, with 20 such ‘runs’ being done and the average values across the 20 runs ultimately reported.

## 5.5 RESULTS

The process of dimensionality reduction winnowed the number of attributes in the Cumulated and Correlated data sets considerably to 21 and 36 attributes (excluding the target binary value), respectively. The standard three layer artificial neural network finally used in the

experiment had 10 hidden nodes. The optimal parameter settings that were found during the parameter tuning stage for XCS model development are recorded in table 5.1.

**Table 5.1: Optimal Parameter Settings for the XCS Model**

| | |
|---|---|
| Learning Rate | 0.2 |
| Probability of '#' (don't care) | 0.6 |
| Crossover Probability | 0.7 |
| Mutation Probability | 0.02 |
| Number of Pattern Presentations | 50000 |
| Maximum Classifier Population Size | 2000 |

The results of the experiments on both reduced data sets across all machine learning approaches used may be found in tables 5.2 and 5.3.

**Table 5.2: 10-Fold Cross Validation Results for the 21 Attribute Cumulated Data Set**

| Algorithm | Avg. % Correct | TP | TN | FP | FN |
|---|---|---|---|---|---|
| XCS | 91.08% | 0.7126 | 0.9445 | 0.0555 | 0.2874 |
| SVM | 92.44% | 0.751 | 0.968 | 0.0302 | 0.249 |
| C4.5 | 88.35% | 0.6552 | 0.9106 | 0.0894 | 0.3448 |
| Naïve Bayes | 80.58% | 0.5517 | 0.8709 | 0.1291 | 0.4483 |
| ANN | 90.08% | 0.7805 | 0.9434 | 0.0566 | 0.2195 |

**Table 5.3: 10-Fold Cross Validation Results for the 36 Attribute Correlated Data Set**

| Algorithm | Avg. % Correct | TP | TN | FP | FN |
|---|---|---|---|---|---|
| XCS | 92.22% | 0.7667 | 0.955 | 0.045 | 0.2333 |
| SVM | 91.05% | 0.7005 | 0.9437 | 0.0563 | 0.2995 |
| C4.5 | 88.17% | 0.6333 | 0.9268 | 0.0732 | 0.3667 |
| Naïve Bayes | 83.98% | 0.6026 | 0.900 | 0.1 | 0.3974 |
| ANN | 91.22% | 0.7222 | 0.9502 | 0.0498 | 0.2778 |

The support vector machine algorithm outperformed all other models on the Cumulated data set with respect to the overall percent correct (Overall>92%) and achieved a very high negative identification rate (TN>96%). The SVM trained on the Cumulated data set achieved the highest overall percent correct across both data sets as well. However, the artificial neural network model trained and tested on the Cumulated data set achieved the highest positive identification rate (TP>78%) across both data sets. XCS was able to outperform all other algorithms on the Correlated data set with respect to the average percent correct. XCS has never been used to predict gene starts before, and its ability to outperform other standard machine learning techniques in predicting gene starts is notable. From the results, it seems that the discretization of attributes required for XCS does not hamper the predictive ability of the method; in fact, it may be that the discretization is what allowed the technique to perform as well as it did. Naïve bayes achieved the poorest performance on both data sets. The C4.5 algorithm consistently outperformed naïve bayes, but was unable to compete with the XCS, ANN, or SVM approaches. All of the machine learning approaches used would have significantly outperformed a random classifier, since a random classifier would have had about a 20% true positive success rate (based on the number of positive examples relative to the number of overall examples), whereas the classifier with the lowest true positive rate (naïve bayes) was able to correctly predict over 50% of the positive examples. It is also interesting to note that although the overall predictive capacity of XCS and ANN improved with a significant increase in the dimensionality of the attribute space (as evinced by the dimensionality difference between the two data sets), the performance of the SVM model worsened as a result of the increase in dimensionality. This may be attributed to either a fundamental difference in the way the three approaches selectively magnify and minimize aspects of the attribute space (so that an increase in the attribute

dimensionality allowed the XCS to better distinguish the appropriate gene start codon from a set of all possible start codons, whereas the increase confused the SVM and worsened its general performance) or to a fundamental difference in the structure of the numerical data between the two data sets (since the numerical attribute values in the two data sets, although culled from the same DNA sequence data, represent different mathematical approaches towards modeling this genetic data).

## 5.6 FINAL REMARKS

Several distinct machine learning techniques for predicting translation start sites across multiple bacterial genomes have been tested. XCS, an accuracy-based classifier system that evolves a population of rules to solve some computationally intensive task, has not been used to predict start sequences before, and the approach works as well as other, more common algorithms typically used in this area of bioinformatics. In the case of XCS, although a population of rules has been evolved, the population size of the rules is very large and thus does not provide an adequate way of choosing the most salient, representative rules to be used by researchers in genetic research. However, certain approaches for extracting general rules from the rule set have been offered. Wilson proposed a method for reducing the size of the rule set in XCS in (Wilson 2002) and was able to generate a small rule set using a specific data set (Wisconsin Breast Cancer dataset), where the final rule set was understandable and represented knowledge interpretable to researchers within the field. Kharbat, Odeh, and Bull (Kharbat, Odeh, and Bull 2006) also recently described an approach where they were able to cluster the total rule set into aggregate regions of rules. Their approach is able to extract an average rule from each rule cluster, representing the most common features of the rules within the cluster. Perhaps applying these methods to the rule set obtained for the current model will provide useful

knowledge to researchers attempting to understand the logic behind gene starts. In any case, future work should involve both improving on the accuracy of the current approach through further, more sensitive discretizing of the attributes, and the reduction of the total rule set to a smaller set of comprehensible and general rules.

The SVM and ANN techniques compared in this approach have been used for predicting translation and transcription start sites before, but the kind of automatic dimensionality reduction used in this approach has not been used. The feature extraction approach described here may prove useful to researchers that are attempting to automate the extraction of useful features in this and other areas of computational biology.

## 5.7 REFERENCES

Altschul, S.F., W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. 1990. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3): 403-410.

Bajic, V.B., and S.H. Seah. 2003. Dragon Gene Start Finder: An Advanced System for Finding Approximate Locations of the Start of Gene Transcriptional Units. *Genome Research*, 13: 1923-1929.

Besemer, J., A. Lomsadze, and M. Borodovsky. 2001. GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. *Nucleic Acids Research*, 29(12): 2607-2618.

Boser, B.E., I.M. Guyon, and V.N. Vapnik. 1992. A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory*, 144-152.

Butz, M. 2000. XCSJava 1.0: An implementation of the XCS classifier system in Java (IlliGAL report 2000027). Technical Report, University of Illinois at Urbana-Champaign: Illinois Genetic Algorithms Laboratory.

Holland, J. 1992. *Adaptation in Natural and Artificial Systems: 2nd Edition*. Cambridge, MA: MIT Press.

Kharbat, F., M. Odeh, and L. Bull. 2006. New Approach for Extracting Knowledge from XCS Learning Classifier System. Learning Classifier Systems Group Technical Report– UWELCSG06-002, 1-24.

Lukashin, A.V., and M. Borodovsky. 1998. GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Research,* 26(4): 1107-1115.

Quinlan, R. 1993. *C4.5: Programs for Machine Learning.* San Francisco, CA: Morgan Kaufmann.

Rish, I. 2001. An empirical study of the naïve bayes classifier. *Proceedings of the IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence,* 42-46.

Rumelhardt, D.E., and J.L. McClelland. 1986. *Parallel Distributed Processing: explorations in the microstructure of cognition. 2 vols.* Cambridge: MIT Press.

Thompson, J.D., D.G. Higgins, and T.J. Gibson. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties, and weight matrix choice. *Nucleic Acids Research*, 22(22): 4673-4680.

Wilson, S.W. 1995. Classifier Fitness Based on Accuracy. *Evolutionary Computation*, 3(2): 149-175.

Wilson, S.W. 2000. Get Real! XCS With Continuous-Valued Inputs. *Learning Classifier Systems: From Foundations to Applications*, 209-219. Berlin: Springer.

Wilson, S.W. 2002. Compact Rulesets from XCSI. *Advances in Learning Classifier Systems: 4th International Workshop, IWLCS 2001*, 197-208.

Witten, I.A., and E. Frank. 2005. *Data Mining: Practical machine learning tools and techniques, 2nd Edition*. San Francisco, CA: Morgan Kaufmann.

# CHAPTER 6

# CONCLUSIONS

The models generated with the techniques described in the previous four chapters have been shown to be robust, accurate, and to generalize well to unseen data. Two of the techniques described have never been used in their respective domains of application before and have been shown to perform comparably to other techniques typically used in these domains. The research described in these studies will now allow future researchers to understand effective and appropriate ways in which problems belonging to these and similar task domains may be approached, represented, and modeled from an artificial evolutionary and heuristic-driven computational perspective. Some of the models that have been developed in this work, especially the final formulation for the static dielectric constant, may be used on their own by a variety of researchers needing accurate approximations and predictions in both of these task domains.

The research directions presented in this thesis have not been exhausted. For the static dielectric constant formulation, cross-validation may be used on the dataset to evaluate the interpolative abilities of the GP approach. The traditional handling of constants in the GP framework may be modified as well, so that constants are dealt with more dynamically and fluidly (generated based on the kinds of functional forms being created by the GP system, for example) than in the standard approach. For translation start site prediction, other methods may be compared to those described here so that further insight into what kinds of techniques work for tackling this problem may occur. The classifiers may also be combined into a single meta-classifier that may exploit the strengths of each technique. Ultimately, research into these areas

may continue as long as the techniques and methods used to solve these problems are refined and improvements in the qualities of the models do not become insignificant.

# APPENDIX

# A GP-EVOLVED FORMULATION FOR THE RELATIVE PERMITTIVITY OF WATER AND STEAM (EXTENDED VERSION)

## I. INTRODUCTION

The relative permittivity (or static dielectric constant) of water and steam, $\varepsilon_r$, has been experimentally calculated at a relatively wide range of temperatures and pressures. The relative permittivity is an important indicator of the solvent behavior of water in a variety of biological (cell membrane electrophysiology, intracellular biochemical processes), and industrial (geochemical high temperature, high pressure processes in deep sea vents) settings (Fernandez et al. 1997). Thus, predicting the behavior of the static dielectric constant of water is crucial for understanding a variety of phenomena, from the effects of hydrostatic pressure on protein folding and unfolding within the cell (Floriano and Nascimento 2004), to understanding the corrosive behavior of water at the high temperatures and pressures found in nuclear power plants. In electrical engineering, the relative permittivity of a substance is used in the design of capacitors. There have been many attempts at creating a single function that accurately predicts the relative permittivity of water and steam, the earliest of which was done by Quist and Marshall in 1965 (Quist and Marshall 1965), but these have suffered from a lack of experimental values across the entire temperature and pressure range. Recently, Fernandez et al. compiled all of the experimentally available data for the relative permittivity of water and steam in a single database (Fernandez et al. 1995). Furthermore, Fernandez et al. evaluated the methods used to

experimentally derive the relative permittivity and chose a subset of the total data set that was the most accurate and that should be used in data correlation. Fernandez et al. proposed a new formulation in (Fernandez et al. 1997) that used this subset and approximated the relative permittivity very well across the entire temperature and pressure range. AI techniques have never been used to approximate the relative permittivity of water across any range of temperatures and pressures.

Our proposal is that in order to more accurately model the behavior of the relative permittivity of water across all temperature and pressure values, two formulations should be created, so that each may be applied in separate thermodynamic regions. In our approach, two functions are evolved that separately approximate the relative permittivity of water and steam across three thermodynamically distinct regions. These two functions collectively approximate the relative permittivity of water across the entire range of temperature and pressure values. The accuracy of these two functions is evaluated by comparing their values for the relative permittivity with the values obtained using the latest formulation of Fernandez et al., against the subset of dielectric constant values that Fernandez et al. chose for data correlation mentioned earlier. Any differences between the functional forms of the two evolved functions are explained and ideas for future work regarding a more accurate formulation are offered.

## II. BACKGROUND: THE STATIC DIELECTRIC CONSTANT

The static dielectric constant (hereon relative permittivity) of a substance, $\varepsilon_r$, is roughly defined as the ability of a substance to transmit or allow the existence of an electric field. More formally, the relative permittivity of a substance, $\varepsilon_r$, is the ratio of the static permittivity of the substance, $\varepsilon_s$, to the static permittivity of a vacuum, $\varepsilon_0$ (Fernandez et al. 1995). The behavior of the relative permittivity of water is related to its physical state (as a liquid or as steam),

temperature, and pressure. This allows the entire range of temperatures and pressures to be divided into 4 regions, A, B, C, and D. Region A is the normal liquid water state between the normal freezing and boiling points (~273K to ~373K). Region B refers to water along the liquid-vapor phase boundary (saturation line). In this region, which extends from 373K to approximately 647.1K (the critical point), water may exist in either the liquid or gas state (depending on the pressure value). The critical point, which occurs at approximately 647.1K with a corresponding pressure of approximately 22.1MPa, denotes the point in the phase space beyond which water ceases to exist in the liquid state. Region C is the region above 373.15K, and at lower pressures and temperatures within region C, water is in the normal gas (steam) phase. However, at higher pressures and temperatures in this region (beyond the critical point), water becomes a supercritical fluid, that is, water recondenses back into a semi-liquid state, but exhibits the properties of both a liquid and gas. Finally, region D refers to super cooled water (water below the normal freezing point of 273.15K at the standard pressure of ~.1MPa).

The behavior of the relative permittivity exhibits discontinuities along the liquid-vapor phase boundary (region B) and in the supercritical part of the region above the normal boiling point (region C). In these regions, very small changes in temperature and pressure cause very large changes in density and in the value of the relative permittivity (Harvey 2006). As a result, theoretical formulations for calculating the relative permittivity of water have mainly focused on a broad range of temperatures (~270K to ~600K) within a small range of pressures (~.1MPa to 200MPa) (Fernandez et al. 1995). The most current formulation for approximating the relative permittivity across the entire range of experimental temperatures and pressures may be found in (Fernandez et al. 1997). Fernandez et al.’s formulation uses an extensive adaptive regression algorithm to create an appropriate function taking a wide variety of domain specific

thermodynamic values (including first, second, and third derivatives of the temperature and pressure inputs with respect to each other) into account. The final function uses 5 adjustable parameters and a total of 25 constants and domain specific non-adjustable parameters and approximates well across the entire range of experimentally available values (260K to 800K temperatures, at pressures up to 1200 MPa).

## III. BACKGROUND: GENETIC PROGRAMMING

What follows is a very brief summary of the genetic programming technique, for further explanation and clarification see (Koza 1992). Genetic Programming (GP) is the evolutionary computing technique that attempts to evolve computer programs using a tree based representation scheme and GP-specific modified versions of the traditional evolutionary operators of crossover and mutation (Ghanea-Hercock 2003). This technique attempts to evolve an executable computer program that solves a specific user-defined problem from a set of functions, which are individual processes that manipulate and convert data elements, and terminals, which are the data elements themselves. The GP approach involves determining a set of functions and terminals to be used in solving the problem, defining a fitness measure by which individual programs may be evaluated regarding the extent to which they may solve the specified problem, setting the specific parameters and operator probabilities that are involved in program tree generation (crossover and mutation probabilities, initial tree depth limit, maximum tree length, etc.), and developing a set of rules to determine when to end a specific GP run (whether after a certain number of generations have elapsed, or after an individual program with a desired fitness threshold has been found).

The genetic operators of crossover and mutation, as well as the way in which individuals are ranked according to their fitness level are modified from the GA approach (described in

detail in Holland 1992) to suit the GP technique. Crossover occurs by selecting two nodes on different parent trees and then swapping all of the children of the selected nodes (as well as the selected nodes themselves) between the two individuals. Mutation, on the other hand, involves selecting a node at which mutation will occur, deleting all of the nodes that are children of the selected node, and then generating a random tree with this node as its root. The fitness evaluation and ranking method in GP is slightly different from the classic GA approach (where fitness maximization is standard) in the fact that the highest ranking individual programs in GP have the lowest fitness values (in effect, a minimization problem). Thus, GP attempts to find a program with the globally minimal fitness value in the search space of all possible programs that may be created using the function and terminal sets used in the problem, to the tree depth or program length specified in the GP setup.

## IV. EXPERIMENTAL SET-UP

In our approach, a variety of different function and terminal sets were explored in an effort to evolve two functions that could model the relative permittivity of water as a function of pressure, temperature, and density. Initially, a continuous function was evolved to approximate the relative permittivity of water across regions A, C, and D, but the results of this function were highly unsatisfactory because the function could not accurately model the behavior of the relative permittivity of water in regions where discontinuities in the relative permittivity were observed (region C, as described earlier). Unfortunately, no empirical temperature/pressure data for region B (along the phase boundary) is currently available (Fernandez et al. 1995), and thus a function approximating the dielectric constant in region B was not evolved. As a result, evolving 2 different functions, one specific to regions A and D (which are contiguous with respect to each other and where the relative permittivity does not exhibit discontinuous behavior), the other

specific to region C (where the relative permittivity behaves nonlinearly with respect to linear changes in the temperature and pressure), became the most logical next step in function development.

The functions for regions A, C, and D were evolved using data sets taken from (Fernandez et al. 1995) and were then compared to relative permittivity values calculated with the same input values (taken from the same data sets) using the newest formulation for dielectric constant prediction, found in (Fernandez et al. 1997). These data sets were compiled from all previous experimentally available data, and were then corrected by Fernandez et al. to coincide with the most recent internationally accepted temperature scale, ITS-90. In most cases, values were provided for the temperature (in degrees Kelvin, or K), pressure (in megapascals, or MPa), and the corresponding dielectric constant. However, in some cases, temperature/density/dielectric constant values were given instead of temperature/pressure/dielectric constant values. In these circumstances, density values were converted into their corresponding pressures, and pressure values were converted to their corresponding densities using the IAPWS-95 formulation for the equation of state of water found in (Wagner and Pruss 2002). With this completed, the final data set uniformly represented the dielectric constant at every temperature, pressure, and density value that was experimentally available.

Both functions were evolved by generating a population of possible functions (represented as trees) as with standard genetic programming implementations. Each candidate function's fitness was taken to be the sum of the absolute values of the difference between the calculated and the experimentally measured value for the relative permittivity at every input value in the corresponding data set. The combination of input values for each function (that is,

what combination of the three possible adjustable inputs was to be used) was determined by the GP module. The population of possible functions was then evolved with a variety of crossover/mutation probabilities and function sets. The data set of experimentally calculated relative permittivity values used to create a function for regions A and D consisted of 291 data points. The data set used to create the function for the one-phase supercritical region (region C) consisted of 353 data points. These data sets include all of the data points (644 total data points) that Fernandez et al. recommend for data correlations (Fernandez et al. 1995). The two evolved functions with the lowest sum of absolute errors across the data points that were found were used as the final equations for approximating the dielectric constant across the three regions.

During any given GP run, all function and terminal sets used during function evolution always included addition, subtraction, multiplication, and division as function operators, and temperature, $T_k$, pressure, $p$, and density, $\rho$, as terminal values. All runs also used a population of 10 random floating-point constants in the range between 0 and 1, which were generated at runtime. Other function operators (sin(), cos(), ln(), $\log_{10}$, $\log_2$, and $x^y$) and terminal operators (Avogadro's number, $N_A$, permittivity of free space, $\varepsilon_0$, elementary charge, $e$, Boltzmann's constant, $k$, molar mass of water, $M_w$, mean molecular polarizability of water, $\alpha$, the dipole moment of water, $\mu$) were also used in certain GP runs. The aforementioned terminal operators are provided in table A.1. A range of crossover probabilities (between .5 and 1.0, in increments of .05) and mutation probabilities (between 0 and .5, in increments of .05) were explored for all combinations of function and terminal sets. Each combination of parameter settings was implemented in 10 GP runs, each on a population of one million individuals that were evolved for 200 generations. The function length of any individual solution (a tree representing a given candidate function) never exceeded 50 functional units (where a functional unit is taken to be a

single operator from the function set or a terminal value from the terminal set), as maintaining the readability of any given evolved function was a priority.

## V. RESULTS

Both of the two best functions that were evolved were found during a run that used multiplication, division, subtraction, and addition as operators in the function set and temperature, pressure, and the molar mass of water as terminal operators (with the 10 additional random ephemeral constants described earlier). In addition to the above terminals, the function evolved for region C used density, $\rho$, Avogadro's number, $N_A$, and Boltzmann's constant, $k$, as terminal operators. Both best function runs used a probability of crossover of 0.7 and a probability of mutation of 0.05. These functions (simplified with all redundancies eliminated), along with Fernandez et. al's formulation, follow:

$$\varepsilon = \varepsilon_{r,AD} \quad when\ T_k \le 373.15,$$
$$\varepsilon_{r,C} \quad elsewhere$$

$$\varepsilon_{r,AD} = \frac{2.5203*10^4}{T_k} - \frac{M_w{}^3 p^2}{0.587} - 0.08133T_k + 0.0355p + 18.08$$

**Figure A.1: Evolved equation, regions A, D**

$$\varepsilon_{r,C} = \frac{0.728(\frac{N_A}{k}\rho + M_w\rho^2)}{0.971T_k - 88.4082} - \frac{\rho p + p^2}{T_k{}^2} + 0.97$$

**Figure A.2: Evolved equation, region C**

$$\varepsilon_r = \frac{1 + 5A + 5B + \sqrt{9 + 2A + 18B + A^2 + 10AB + 9B^2}}{4 - 4B}$$

*where A and B are given by*

$$A = \frac{N_A \mu^2}{\varepsilon_0 k} \frac{\rho g}{T_k}$$

$$B = \frac{N_A \alpha}{3\varepsilon_0} \rho$$

*and where g is given by*

$$g = 1 + \sum_{k=1}^{11} N_k \left(\frac{\rho}{\rho_c}\right)^{i_k} \left(\frac{T_c}{T}\right)^{j_k} + N_{12} \left(\frac{\rho}{\rho_c}\right) \left(\frac{T}{228K} - 1\right)^{-q}$$

*with* $\rho_c = \frac{322}{M_w}$ *and* $T_c = 647.096K$ *and values for* $N_k$, $i_k$, $j_k$, *and q given in table A.2.*

**Figure A.3: Fernandez' formulation**

The results of applying these functions to their respective partitions of the total data set are found in table A.3. The evolved functions shown above are significantly smaller than the formulation developed by Fernandez et al. and use at most three adjustable parameters (temperature, pressure, and density), three non-adjustable domain specific parameters (Avogadro's number, Boltzmann's constant, and the molar mass of water), and three of the ten possible random ephemeral constants that were available during function evolution. No domain-specific knowledge (aside from the data sets themselves) was applied to the formulation of the functions. Furthermore, the evolved functions selected different terminal values for both regions, so that the region C function uses density as an input value along with temperature and pressure, whereas the region A and D function uses temperature and pressure exclusively. This is telling because density is a much more relevant predictive parameter (varying discontinuously along with the relative permittivity while temperature and pressure monotonically increase) for the

relative permittivity in the single phase and super critical region (region C) than in regions A and D. The fact that the GP approach was able to selectively choose the relevant parameters for each region is notable and significant.

As can be seen from table A.3, both evolved functions outperformed Fernandez et al.'s formulation across all thermodynamic regions. For regions A and D the evolved function outperformed Fernandez et al.'s formulation strictly because of one data point value (notably, a data point that occurred immediately preceding the phase boundary around 373.15K). At this temperature, Fernandez et al.'s formulation may have rounded the temperature input parameter (at 373.147K) up, causing a very sharp discontinuous drop in the calculated relative permittivity value. In region C, the evolved function consistently outperformed Fernandez et al.'s formulation, leading to an improvement in calculation accuracy across the entire range of experimentally available relative permittivity values.

## VI. CONCLUSIONS AND FUTURE WORK:

Two functions that approximate the relative permittivity of water and steam at a variety of temperatures and pressures have been proposed. These functions were evolved using the GP technique with a specific function and terminal set, and their accuracy has been compared to that achieved by Fernandez et al.'s most recent formulation. The evolved functions approximate the relative permittivity of water and steam for a wide range of temperature and pressure values quite well, improving on Fernandez et al.'s formulation across the entire experimentally available temperature and pressure range while being much simpler computationally. Further refinements to create more accurate approximations of the relative permittivity of water and steam will include creating an evolved function that can be used across all thermodynamically distinct temperature and pressure regions. This can be done when experimental values for the

temperature, pressure, and relative permittivity along the phase boundary and more values in the supercritical region are obtained. A refined fitness function that takes more than the absolute difference between expected and calculated values may also prove useful in creating a new formulation. However, significant improvements to the evolution of an appropriate function will most surely come from an increase in experimentally verifiable values for the relative permittivity, and thus any new accurate data that may be found should be used to refine the current formulation.

## VII. TABLES

**Table A.1: Constants used in the relative permittivity formulation**

| Parameter | Value |
|---|---|
| Permittivity of free space, $\varepsilon_0$ | $[4*10^{-7}\pi(299792458)^2]^{-1}C^2J^{-1}m^{-1}$ |
| Elementary charge, $e$ | $1.60217733*10^{-19}C$ |
| Boltzmann's constant, $k$ | $1.380658*10^{-23}JK^{-1}$ |
| Avogadro's number, $N_A$ | $6.0221367*10^{23}mol^{-1}$ |
| Molar mass of water, $M_w$ | $0.018015268kg*mol^{-1}$ |
| Mean molecular polarizability of water, $\alpha$ | $1.636*10^{-40}C^2J^{-1}m^{-2}$ |
| Dipole moment of water, $\mu$ | $6.138*10^{-30}Cm$ |

**Table A.2: Coefficients $N_k$, and exponents $i_k$, $j_k$, and q of the equation for g**

| $k$ | $N_k$ | $i_k$ | $j_k$ |
|---|---|---|---|
| 1 | $0.978224486826$ | 1 | 0.25 |
| 2 | $-0.957771379375$ | 1 | 1 |
| 3 | $0.237511794148$ | 1 | 2.5 |
| 4 | $0.714692244396$ | 2 | 1.5 |
| 5 | $-0.298217036956$ | 3 | 1.5 |
| 6 | $-0.108863472196$ | 3 | 2.5 |
| 7 | $0.949327488264*10^{-1}$ | 4 | 2 |
| 8 | $-0.980469816509*10^{-2}$ | 5 | 2 |
| 9 | $0.165167634970*10^{-4}$ | 6 | 5 |
| 10 | $0.937359795772*10^{-4}$ | 7 | 0.5 |
| 11 | $-0.123179218720*10^{-9}$ | 10 | 10 |
| 12 | $0.196096504426*10^{-2}$ | | *q=1.2* |

**Table A.3: Results and numeric comparison**

| | REGION C | | | REGION A,D | |
|---|---|---|---|---|---|
| | Evolved GP result | Fernandez | | Evolved GP result | Fernandez |
| Sum Absolute Difference | 57.21 | 69.15 | | 44.85 | 80.58 |
| Mean Absolute Difference | 0.16 | 0.20 | | 0.15 | 0.27 |
| Standard Deviation Absolute Difference | 0.25 | 0.25 | | 0.26 | 3.18 |
| Sum Squared Difference | 30.82 | 35 | | 26.9 | 2991.42 |
| Mean Squared Difference | 0.09 | 0.1 | | 0.09 | 10.18 |
| Standard Deviation Squared Difference | 0.75 | 0.70 | | 0.41 | 173.92 |
| Minimum Absolute Difference | 0 | 0.00 | | 0.001 | 0 |
| Maximum Absolute Difference | 3.76 | 3.6 | | 1.97 | 54.61 |
| # Data Points Absolute Difference formulation < Absolute Difference Fernandez | 181 | | | 142 | |
| Percentage of Total Data Points better than Fernandez | 51.27% | | | 48.80% | |
| Total Data Points | 353 | 353 | | 291 | 291 |

## VIII. REFERENCES

Fernandez, D.P., Y. Mulev, A.R.H. Goodwin, and J.M.H. Levelt-Sengers. 1995. A Database for the Static Dielectric Constant of Water and Steam. *Journal of Physical and Chemical Reference Data* 24(1): 33-69.

Fernandez, D.P., A.R.H. Goodwin, E.W. Lemmon, J.M.H. Levelt-Sengers, and R.C. Williams. 1997. A Formulation for the Static Permittivity of Water and Steam at Temperatures from 238K to 873K at Pressures up to 1200MPa, Including Derivatives and Debye-Huckel Coefficients. *Journal of Physical and Chemical Reference Data* 26(4): 1125-1166.

Floriano, W., and M.A.C Nascimento. 2004. Dielectric Constant and Density of Water as a Function of Pressure at Constant Temperature. *Brazilian Journal of Physics* 34(1): 38-41.

Ghanea-Hercock, R. 2003. *Applied Evolutionary Algorithms in Java.* New York, NY: Springer-Verlag.

Harvey, Allan. 2006. NIST. Personal communication.

Holland, J. 1992. *Adaptation in Natural and Artificial Systems: 2nd Edition.* Cambridge, MA: MIT Press.

Koza, J.R. 1992. *Genetic Programming.* Cambridge, MA: MIT Press.

Quist, A.S., and W.L. Marshall. 1965. Estimation of the Dielectric Constant of Water to 800°. *Journal of Physical Chemistry* 9: 3165.

Wagner, W and Pruss, A. 2002. The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use. *Journal of Physical and Chemical Reference Data* 31(2): 387-535.