Part One

Gene Expression Analysis and Systems Biology

Chapter 1

Hybrid of Neural Classifier and Swarm Intelligence in Multiclass Cancer Diagnosis with Gene Expression Signatures

Rui Xu, Georgios C. Anagnostopoulos, and Donald C. Wunsch II

1.1 INTRODUCTION

With the rapid advancement of deoxyribonucleic acid (DNA) microarray technologies, cancer classification through gene expression profiles has already become an important means for cancer diagnosis and treatment and attracted numerous efforts from a wide variety of research communities (McLachlan et al., 2004). Compared with the traditional classification methods that are largely dependent on the morphological appearance and clinical parameters, gene expression signature-based methods offer cancer researchers new methods for the investigation of cancer pathologies from a molecular angle, under a systematic framework, and further, to make more accurate prediction in prognosis and treatment.

Although previous research has included binary cancer classification (Alizadeh et al., 2000; Golub et al., 1999; West et al., 2001), it is more common practice to discriminate more than two types of tumors (Khan et al., 2001; Nguyen and Rocke, 2002; Ooi and Tan, 2003; Ramaswamy et al., 2001; Scherf et al., 2000). For example, Khan et al. (2001) used multilayer perceptrons to categorize small round blue-cell tumors (SRBCTs) with 4 subclasses. Scherf et al. (2000) constructed a gene expression database to investigate the relationship between genes and drugs for

Computational Intelligence in Bioinformatics. Edited by G. B. Fogel, D. Corne, and Y. Pan

60 human cancer cell lines originating from 10 different tumors, which provides an important criterion for therapy selection and drug discovery. Although these methods manifest interesting performance for some cancer data sets, their classification accuracy deteriorates dramatically with the increasing number of classes in the data, as shown in a comparative study by Li et al. (2004).

Among all the challenges encountered in gene expression data analysis, the curse of dimensionality becomes more serious as a result of the availability of overwhelming number of gene expression values relative to the limited number of samples. The existence of numerous genes in the data that are completely irrelevant to the discrimination of tumors not only increases the computational complexity but restricts the discovery of truly relevant genes. Therefore, feature selection, also known as *informative gene selection* in this context, becomes critically important (Deng et al., 2004; Golub et al., 1999; Ooi and Tan, 2003). Commonly used selection methods to discover informative genes rank the genes in terms of their expression difference in two different categories (Golub et al., 1999; Jaeger et al., 2003; Nguyen and Rocke, 2002; Ben-Dor et al., 2000). These criteria can provide some meaningful insights for binary classification. However, they may result in many highly correlated genes while ignoring the truly important ones. For multiclass cancer-type discrimination, these same approaches also cause some additional complexity due to the requirement for either one-versus-all or all-pairs comparison.

Considering this insufficiency of the existing methods in the analysis of more complicated cancer data sets, such as the NCI60 data set, in this chapter we propose a hybridized computational intelligence technology of semisupervised ellipsoid ARTMAP (ssEAM) and particle swarm optimization (PSO) (Kennedy et al., 2001) for multiclass cancer discrimination and gene selection. We apply the proposed method on publicly available cancer data sets and investigate the effect of various parameters on system performance. The experimental results demonstrate the effectiveness of ssEAM/PSO in addressing the massive, multidimensional gene expression data and are comparable to, or better than, those obtained by other classifiers. Additional analysis and experimental results on ssEAM/PSO in cancer classification are published elsewhere (Xu et al., 2007).

The rest of the chapter is organized as follows. In Section 1.2, we present the ssEAM/PSO system for multiclass cancer discrimination, together with the experiment design. In Section 1.3, we show the experimental results on two publicly accessible benchmark data sets. We conclude the chapter in Section 1.4.

1.2 METHODS AND SYSTEMS

1.2.1 EAM and Semisupervised EAM

Semisupervised ellipsoid ARTMAP is based on adaptive resonance theory (ART) (Grossberg, 1976), which was inspired by neural modeling research, and developed as a solution to the *plasticity–stability dilemma*. It is designed as an enhancement and generalization of ellipsoid ART (EA) and ellipsoid ARTMAP (EAM)

(Anagnostopoulos, 2001; Anagnostopoulos and Georgiopoulos, 2001), which, in turn, follow the same learning and functional principles of fuzzy ART (Carpenter et al., 1991) and fuzzy ARTMAP (Carpenter et al., 1992).

Ellipsoid ARTMAP accomplishes classification tasks by clustering data that are attributed with the same class label. The geometric representations of these clusters, which are called EA *categories*, are hyperellipsoids embedded in the feature space. A typical example of such a category representation, when the input space is two-dimensional, is provided in Figure 1.1, where each category C_j is described by a center location vector $\mathbf{m}_j$, orientation vector $\mathbf{d}_j$, and Mahalanobis radius M_j, which are collected as the template vector $\mathbf{w}_j = [\mathbf{m}_j, \mathbf{d}_j, M_j]$. If we define the distance between an input pattern $\mathbf{x}$ and a category C_j as

$$D(\mathbf{x}, \mathbf{w}_j) = \max\{\|\mathbf{x} - \mathbf{m}_j\|_{\mathbf{c}_j}, M_j\} - M_j, \tag{1.1}$$

$$\|\mathbf{x} - \mathbf{m}_j\|_{\mathbf{c}_j} = \sqrt{(\mathbf{x} - \mathbf{m}_j)^{\mathrm{T}} \mathbf{S}_j (\mathbf{x} - \mathbf{m}_j)}, \tag{1.2}$$

where $\mathbf{S}_j$ is the category's shape matrix, defined as

$$\mathbf{S}_j = \frac{1}{\mu^2}[\mathbf{I} - (1 - \mu^2)\mathbf{d}_j \mathbf{d}_j^{\mathrm{T}}], \tag{1.3}$$

and μ is the ratio between the length of the hyperellipsoid's minor axes (with equal length) and major axis, the *representation region* of C_j, which is the shaded area in the figure, can be defined as a set of points in the input space, satisfying the condition

$$D(\mathbf{x}, \mathbf{w}_j) = 0 \Rightarrow \|\mathbf{x} - \mathbf{m}_j\|_{\mathbf{S}_j} \leq M_j. \tag{1.4}$$

A category encodes whatever information the EAM classifier has learned about the presence of data and their associated class labels in the locality of its geometric representation. This information is encoded into the location, orientation, and size of the hyperellipsoid. The latter feature is primarily controlled via the baseline vigilance $\bar{\rho} \in [0, 1]$ and indirectly via the choice parameter $a > 0$ and a network parameter $\omega \geq 0.5$ (Anagnostopoulos and Georgiopoulos, 2001). Typically, small values of $\bar{\rho}$ produce categories of larger size, while values close to 1 produce the opposite effect. As a special case, when $\bar{\rho} = 1$, EAM will create solely point categories (one for each

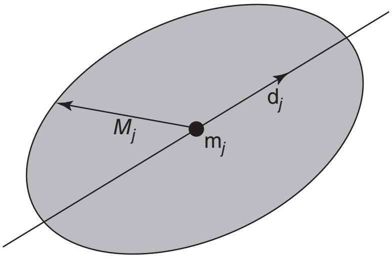


Figure 1.1 Example of geometric representation of EA category C_j in two-dimensional feature space.

training pattern) after completion of training, and it implements the ordinary, Euclidian 1-nearest neighbor classification rule. A category's particular shape (eccentricity of its hyperellipsoid) is controlled via the network parameter $\mu \in (0, 1]$; for $\mu = 1$ the geometric representations become hyperspheres, in which case the network is called *hypersphere ARTMAP* (Anagnostopoulos and Georgiopoulos, 2000).

Figure 1.2 illustrates the block diagram of an EAM network, which consists of two EA modules (ART$_a$ and ART$_b$) interconnected via an inter-ART module. The ART$_a$ module clusters patterns of the input domain while ART$_b$ clusters patterns of the output domain. The information regarding the input–output associations is stored in the weights $\mathbf{w}_j^{ab}$ of the inter-ART module, while EA category descriptions are contained in the template vectors $\mathbf{w}_j$. These vectors are the top–down weights of F_2-layer nodes in each module.

Learning in EAM occurs by creating new categories or updating already existing ones. If a training pattern $\mathbf{x}$ initiates the creation of a new category C_J, then C_J receives the class label $L(\mathbf{x})$ of $\mathbf{x}$, by setting the class label of C_J to $l(J) = L(\mathbf{x})$. The recently created category C_J is initially a *point category*, meaning that $\mathbf{m}_J = \mathbf{x}$ and $M_J = 0$. While training progresses, point categories are being updated, due to the presentation of other training patterns, and their representation regions may grow. Specifically, when it has been decided that a category C_j must be updated by a training pattern $\mathbf{x}$, its representation region expands, so that it becomes the minimum-volume hyperellipsoid that contains the entire, original representation region and the new pattern. Learning eventually ceases in EAM, when no additional categories are being created and the existing categories have expanded enough to capture all training data. Notice that, if $\mathbf{x}$ falls inside the representation region of C_j, no update occurs since C_j has already taken into account the presence of $\mathbf{x}$.

The procedure of deciding which category C_j is going to be updated, with a training pattern $\mathbf{x}$, involves competition among preexisting categories. Let us define this set of categories as N as well as the set $S \subseteq N$ of all categories that are candidates

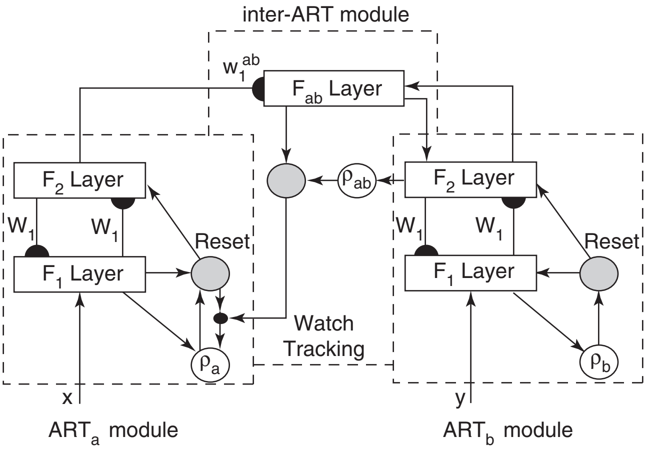


Figure 1.2 Ellipsoid ARTMAP block diagram.

in the competition; initially, $S = N$. Before the competition commences, for each category C_j, two quantities are calculated: the *category match function* (CMF) value

$$\rho(\mathbf{w}_j|\mathbf{x}) = \frac{D_{\max} - 2M_j - D(\mathbf{x}, \mathbf{w}_j)}{D_{\max}}, \tag{1.5}$$

where $D_{\max}$ is a parameter greater than 0, and the *category choice function* (CCF) value

$$T(\mathbf{w}_j|\mathbf{x}) = \frac{D_{\max} - 2M_j - D(\mathbf{x}, \mathbf{w}_j)}{D_{\max} - 2M_j + a}. \tag{1.6}$$

Next, EAM employs two category-filtering mechanisms: the *vigilance test* (VT) and the *commitment test* (CT). Both tests decide if the match between the pattern and the category's representation region is sufficient to assign that pattern to the cluster represented by the category in question. These tests can function as a novelty detection mechanism as well: If no category in S passes both tests, then $\mathbf{x}$ is not a typical pattern in comparison to the data experienced by the classifier in the past. Categories that do not pass these tests can be subsequently removed from the candidate set S. Next, the competition for $\mathbf{x}$ is won by the category C_J that features the maximum CCF value with respect to the pattern; in case of a tie, the category with minimum index is chosen. The final verdict on whether to allow C_J to be updated with $\mathbf{x}$ or not is delivered by the *prediction test* (PT): C_J is allowed to be updated with $\mathbf{x}$ only if both C_J and $\mathbf{x}$ feature the same class label, that is, if $I(J) = L(\mathbf{x})$. If C_J fails the PT, a *match tracking* process is invoked by utilizing a stricter VT in the hope that another suitable EAM category will be found that passes all three tests. If the search eventually fails, a new point category will be created as described before.

Ellipsoid ARTMAP does not allow categories to learn training patterns of dissimilar class labels. This property is ideal when the individual class distributions of the problem are relatively well separated. However, in the case of high class overlap, or when dealing with increased amount of noise in the feature domain, EAM will be forced to create many small-sized categories, a phenomenon called the *category proliferation problem*. Moreover, when EAM is trained in an offline mode to perfection, its posttraining error will be zero, which can be viewed as a form of data overfitting.

The semisupervised EAM (ssEAM) classifier extends the generalization capabilities of EAM by allowing the clustering into a single category of training patterns not necessarily belonging to the same class (Anagnostopoulos et al., 2002). This is accomplished by augmenting EAM's PT in the following manner: A winning category C_J may be updated by a training pattern $\mathbf{x}$, even if $I(J) \neq L(\mathbf{x})$, as long as the following inequality holds:

$$\frac{w_{J,I(J)}}{1 + \sum_{c=1}^{C} w_{J,c}} \geq 1 - \varepsilon, \tag{1.7}$$

where C denotes the number of distinct classes related to the classification problem at hand and the quantities $w_{J,c}$ contain the count of how many times category C_J was updated by a training pattern belonging to the cth class. In other words, Eq. (1.7) ensures that the percentage of training patterns that are allowed to update category C_J and carry a class label different than the class label $I(J)$ (the label that was initially assigned to C_J, when it was created) cannot exceed 100ε %, where $\varepsilon \in [0,1]$ is a new network parameter, the *category prediction error tolerance*, which is specific only to ssEAM. For $\varepsilon = 1$ the modified PT will allow categories to be formed by clustering training patterns, regardless of their class labels, in an unsupervised manner. In contrast, with $\varepsilon = 0$ the modified PT will allow clustering (into a single category) only of training patterns belonging to the same class, which makes the category formation process fully supervised. Under these circumstances, ssEAM becomes equivalent to EAM. For intermediate values of ε, the category formation process is performed in a semisupervised fashion.

Both EAM and ssEAM feature a common performance phase, which is almost identical to their training phases. However, during the presentation of test patterns, no categories are created or updated. The predicted label for a test pattern **x** is determined by the *dominant class label* $O(J)$ of the winning category C_J defined as

$$\hat{L}(\mathbf{x}) = O(J) = \arg\max_{c=1..C} w_{J,c} = 1. \tag{1.8}$$

When $\varepsilon < 0.5$, ssEAM's PT guarantees that throughout the training phase $O(j) = I(j)$ for any category C_j.

For $\varepsilon > 0$ ssEAM will, in general, display a nonzero posttraining error, which implies a departure from EAM's overfitting and category proliferation issues. For classification problems with noticeable class distribution overlap or noisy features, ssEAM with $\varepsilon > 0$ will control the generation of categories representing localized data distribution exceptions, thus, improving the generalization capabilities of the resulting classifier. Most importantly, the latter quality is achieved by ssEAM without sacrificing any of the other valuable properties of EAM, that is, stable and finite learning, model transparency, and detection of atypical patterns.

Semisupervised EAM has many attractive properties for classification or clustering. First, ssEAM is capable of both online (incremental) and offline (batch) learning. Using *fast learning* in offline mode, the network's training phase completes in a small number of epochs. The computational cost is relatively low, and it can cope with large amounts of multidimensional data, maintaining efficiency. Moreover, ssEAM is an *exemplar-based model*, that is, during its training the architecture summarizes data via the use of exemplars in order to accomplish its learning objective. Due to its exemplar-based nature, responses of an ssEAM architecture to specific test data are easily explainable, which makes ssEAM a *transparent learning model*. This fact contrasts other, *opaque* neural network architectures, for which it is difficult, in general, to explain why an input **x** produced a particular output **y**. Another important feature of ssEAM is the capability of detecting atypical patters during either its training or performance phase. The detection of such patterns is accomplished via the employment of a match-based criterion that decides to which degree a particular pattern matches the characteristics of an already formed category in ssEAM. Additionally, via the utilization of hyperellipsoidal categories, ssEAM can learn complex decision boundaries, which arise frequently in gene expression classification problems. Finally, ssEAM is far simpler to implement, for example, than backpropagation for feedforward neural networks and the training algorithm of support vector machines. Many of these advantages are inherited, general properties of the ART family of neural networks including fast, exemplar-based, match-based, learning (Grossberg, 1976), transparent learning (Healy and Caudell, 1997), capability to handle massive data sets (Healy et al., 1993), and implementability in software and hardware (Serrano-Gotarredona et al., 1998; Wunsch et al., 1993; Wunsch, 1991). Also, ART neural networks dynamically generate clusters without specifying the number of clusters in advance as the classical *k*-means algorithm requires (Xu and Wunsch, 2005).

1.2.2 Particle Swarm Optimization

Particle swarm optimization (PSO) is a heuristic, global, stochastic maximization meta-algorithm, motivated by the complex social behavior and originally intended to explore optimal or near-optimal solutions in sophisticated continuous spaces (Kennedy et al., 2001). Like most evolutionary computation meta-algorithms, PSO is also population based, consisting of a swarm of particles, each of which represents a candidate solution $\mathbf{x}_i$ and is associated with a random velocity $\mathbf{v}_i$. The basic idea of PSO is that each particle randomly searches through the problem space by updating itself with its own memory and the social information gathered from other particles. Specifically, at each time step, each particle is accelerated toward two best locations, based on the fitness value: p_{best} for the previous best solution for this particle and g_{best} for the best overall value in the entire swarm. Accordingly, the canonical PSO velocity and position update equations are written as

$$v_{ij}(t+1) = W_I \times v_{ij}(t) + c_1 \ \text{rand}_1[p_{\text{best}_{ij}} - x_{ij}(t)] + c_2 \ \text{rand}_2[g_{\text{best}_{ij}} - x_{ij}(t)], \quad (1.9)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \quad (1.10)$$

where W_I is an inertia weight, c_1 and c_2 are acceleration constants, and rand_1 and rand_2 are samples of random variables uniformly distributed in the range of [0, 1]. PSO has many desirable characteristics, such as a memory mechanism of keeping track of previous best solutions, the easiness to implement, fast convergence to high-quality solutions, and the flexibility in balancing global and local exploration.

Since our goal is to select important genes from a large gene pool with M genes in total, we employ a discrete binary version of PSO (Kennedy and Eberhart, 1997). The major change of the binary PSO lies in the interpretation of the meaning of the particle velocity. Given a set of N particles $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, each of which corresponds to a subset of genes, the velocity for the ith particle $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iM})$ is represented as $\mathbf{v}_i^* = (v_{i1}^*, v_{i2}^*, \ldots, v_{iM}^*)$. The possible values for each bit x_{ij} ($1 \le i \le N$, $1 \le j \le M$) are either one or zero, indicating the corresponding genes are selected or not. The velocity v_{ij}^* associated with it is defined as the probability that x_{ij} takes the value of 1 and is calculated by the logistic probability law

$$v_{ij}^*(t+1) = \frac{1}{1+\exp[-v_{ij}(t+1)]}, \tag{1.11}$$

where v_{ij} is calculated using Eq. (1.9). Accordingly, the position update equation is given as

$$x_{ij}(t+1) = \begin{cases} 1 & \text{if } \mathrm{rand}_3 + \delta < v_{ij}^*(t+1), \\ 0 & \text{otherwise,} \end{cases} \tag{1.12}$$

where rand_3 is a sample of random variable uniformly distributed in the range of [0, 1], and δ is a parameter that limits the total number of genes selected to a certain range. Compared to the original binary PSO by Kennedy and Eberhart (1997), we add the parameter δ to obtain more flexibility in controlling the number of informative genes. If the value of δ is large, the number of genes selected becomes small and vice versa.

Now, we summarize the basic procedure of binary PSO for informative gene selection as follows:

1. Initialize a population of N particles with random positions and velocities. The dimensionality M of the problem space equals the number of genes in the data.
2. Evaluate the classification performance of ssEAM and calculate the optimization fitness function for each particle. The definition of PSO's fitness function aims to minimize the classification error while favoring the subset with fewer genes; it is defined as

$$f(\mathbf{x}_i) = \mathrm{Acc}_{\mathrm{LOOCV}} + \frac{1}{M_i}, \tag{1.13}$$

   where $\mathrm{Acc}_{\mathrm{LOOCV}}$ is the *leave-one-out cross-validation* (LOOCV) (Kohavi, 1995) classification accuracy defined in Eq. (1.16) and M_i is the number of informative genes selected.
3. Compare the fitness value of each particle with its associated p_{best} value. If the current value is better than p_{best}, set both p_{best} and the particle's attained location to the current value and location.
4. Compare p_{best} of the particles with each other and update g_{best} with the greatest fitness.
5. Update the velocity and position of the particles using Eqs. (1.9), (1.11), and (1.12).
6. Return to step 2 until the stopping condition is met, which, usually, is reaching the maximum number of iterations or the discovery of high-quality solutions.

The inertial weight is similar to the momentum term in the backpropagation algorithm for multilayer perceptrons (Bergh and Engelbrecht, 2004). It specifies the trade-off between the global and local search (Shi and Eberhart, 1998). Larger values of W_I facilitate the global exploration while lower values encourage local search. W_I can take on a fixed value, or more commonly, decreases linearly during a PSO run (Shi and Eberhart, 1999),

$$W_I = (W_{I1} - W_{I2})\frac{T-t}{T} + W_{I2}, \tag{1.14}$$

where W_{I1} and W_{I2} are the initial and final values, respectively, T is the maximum number of epochs allowed, and t is the current epoch number. Alternately, the inertial weight can be set to change randomly within a range (Eberhart and Shi, 2001), for instance,

$$W_I = W_{I0} + \frac{\text{rand}}{2}, \tag{1.15}$$

where rand is a uniformly distributed random function in the range of [0, 1]. As an example, if W_{I0} is set as 0.5, Eq. (1.15) makes W_I vary between 0.5 and 1, with a mean of 0.75. In this chapter, these three methods are referred to as PSO-FIXEW, PSO-LTVW, and PSO-RADW, respectively. c_1 and c_2 are known in the PSO literature as cognition and social components, respectively, and are used to adjust the velocity of a particle toward p_{best} and g_{best}. Typically, they are both set to a value of 2, although different values may achieve better performance (Eberhart and Shi, 2001).

1.2.3 Experiment Design

Since the data sets consist of only a small number of samples for each cancer type, it is important to use an appropriate method to estimate the classification error of the classifier. In the experiment below, we perform a double cross validation [*10-fold cross validation* (CV10) with LOOCV], instead of just the commonly used LOOCV, to examine the performance of ssEAM/PSO. This is because, although the LOOCV is an unbiased point estimate of the classification error, it has high variance, which is not preferred in cancer classification. On the contrary, resampling strategies such as bootstrap (Efron, 1982), may become largely biased in some cases (Ambroise and McLachlan, 2002; Kohavi, 1995) despite having lower variance. During the double cross-validation procedure, the data set with Q samples is divided into 10 mutually exclusive sets of approximately equal size, with each subset consisting of approximately the same proportions of labels as the original data set, known as stratified cross validation (Kohavi, 1995). The classifier is trained 10 times, with a different subset left out as the test set and the other samples are used to train the classifier each time. During the training phase, gene selection is performed based on the 9 out of the 10 data subsets (without considering the test data), for which LOOCV classification accuracy is used as the fitness function in Eq. (1.13). The prediction performance of the classifier is estimated by considering the average classification accuracy of the 10 cross-validation experiments, described as

$$\mathrm{Acc}_{\mathrm{CV10}} = \left(\frac{1}{Q} \sum_{i=1}^{10} A_i \right) \times 100\%, \tag{1.16}$$

where A_i is the number of correctly classified samples. Previous studies have shown that CV10 is more appropriate when a compromise between bias and variance is preferred (Ambroise and McLachlan, 2002; Kohavi, 1995).

We also compare our approach with four other classifiers, that is, multilayer perceptrons (MLPs) (Haykin, 1999), probabilistic neural networks (PNNs) (Specht, 1990), learning vector quantization (LVQ) (Kohonen, 2001), and *k*-nearest-neighbor (kNN) (Duda et al., 2001), together with Fisher's discriminant criterion (Hastie et al., 2003), which is used for informative genes selection and defined as

$$F(i) = \frac{|\mu_+(i) - \mu_-(i)|^2}{\sigma_+^2(i) + \sigma_-^2(i)}, \tag{1.17}$$

where $\mu_+(i)$ and $\mu_-(i)$ are the mean values of gene i for the samples in class +1 and class −1, and $\sigma_+^2(i)$ and $\sigma_-^2(i)$ are the variances of gene i for the samples in class +1 and −1. The score aims to maximize the between-class difference and minimize the within-class spread. Currently, other proposed rank-based criteria with these considerations show similar performance (Jaeger et al., 2003). Since our ultimate goal is to classify multiple types of cancer, we utilize a one-versus-all strategy to seek gene predictors. In order to overcome *selection bias*, which is caused by including the test samples in the process of feature selection and which leads to an overoptimistic estimation of the performance for the classifier (Ambroise and McLachlan, 2002; Nguyen and Rocke, 2002; West et al., 2001), we utilize the strategy that separates gene selection from cross-validation assessment. Note in this case that the subsets of genes selected at each stage tend to be different.

1.3 EXPERIMENTAL RESULTS

1.3.1 NCI60 Data

The NCI60 data set includes 1416 gene expression profiles for 60 cell lines in a drug discovery screen by the National Cancer Institute (Scherf et al., 2000). These cell lines belong to 9 different classes: 8 breast (BR), 6 central nervous system (CNS), 7 colorectal (CO), 6 leukemia (LE), 9 lung (LC), 8 melanoma (ME), 6 ovarian (OV), 2 prostate (PR), and 8 renal (RE). Since the PR class only had two samples, these were excluded from further analysis. There were 2033 missing gene expression values in the data set, which were imputed by the method described by Berrar et al. (2003). This process left the final matrix in the form of $E = \{e_{ij}\}_{58\times1409}$, where $e_{i,j}$ represents the expression level of gene j in tissue sample i.

We first investigated the effect of the three different strategies for the inertia weight (i.e., PSO-FIXEW, PSO-LTVW, and PSO-RADW) on the performance of PSO. The values of c_1 and c_2 were both set to 2 in this study. We fixed the W_I at 0.8 for PSO-FIXEW and linearly decreased W_I from 0.9 to 0.4 for PSO-LTVW. For the random method PSO-RADW, we chose W_{I0} equal to 0.5, so that W_I varied in the range of [0.5, 1]. The performance of the three methods was compared in terms of the number of iterations required to reach a prespecified classification accuracy, say, 81% (47 out of 58 samples) in this case. We further set the maximum number of iterations to 100. If PSO reached 81% classification accuracy within 100 iterations, we considered PSO to have converged. The results over 20 runs are summarized in Table 1.1, which consists of the number of times that the iteration exceeded the allowed maximum and the average number of epochs if PSO converged. As indicated in Table 1.1, the most effective performance was achieved when PSO-FIXEW was used, where PSO achieves the expected accuracy within 39 iterations on average, except for 2 runs that did not converge. The result for PSO-LTVW was slightly inferior to PSO-FIXEW, as the average number of iterations was 42.9 for 16 out of 20 converging cases. PSO-RADW did not perform well in this problem and was more dependent on its initialization. In the following discussion, we used PSO-FIXEW with W_I at 0.8, and set both c_1 and c_2 to 2.

We set the parameters for ssEAM as follows: $\mu = 0.3$, $\rho = 0.4$, and $\alpha = 2.5$, learning rate = 0.8, and adjusted the value of ε, which controlled the amount of misclassification allowed in the training phase. The parameters of ssEAM were determined based on a simple selection procedure in which the data set was randomly divided into training and validation sets. We compared the different parameter combination and chose the ones that lead to relatively high performance. The parameter δ controlled the total number of genes selected in the subsets, and we evaluated the program with δ at 0.5, 0.45, 0.4, 0.3, 0.2, 0.1, and 0.0. Each time, evolution was processed for 300 generations with a swarm population of 50 particles. The algorithm was iterated 20 times using different partitions of the data set and performance was reviewed relative to mean performance. The mean and standard deviation of the classification accuracies from the 20 runs are summarized in Table 1.2, and the best results are depicted in Figure 1.3(*a*). For the purpose of comparison, we also show the results of PNN, MLP, kNN, and LVQ1, in which the Fisher criterion was

Table 1.1 Comparison of PSO-FIXEW, PSO-LTVW, and PSO-RADW on PSO Performance

	Performance	
W_I	>100 Iterations	Average Number of Iterations
PSO-FIXEW (at 0.8)	2	38.3
PSO-LTVW (0.9–0.4)	3	42.9
PSO-RADW (0.5–1)	5	45.5

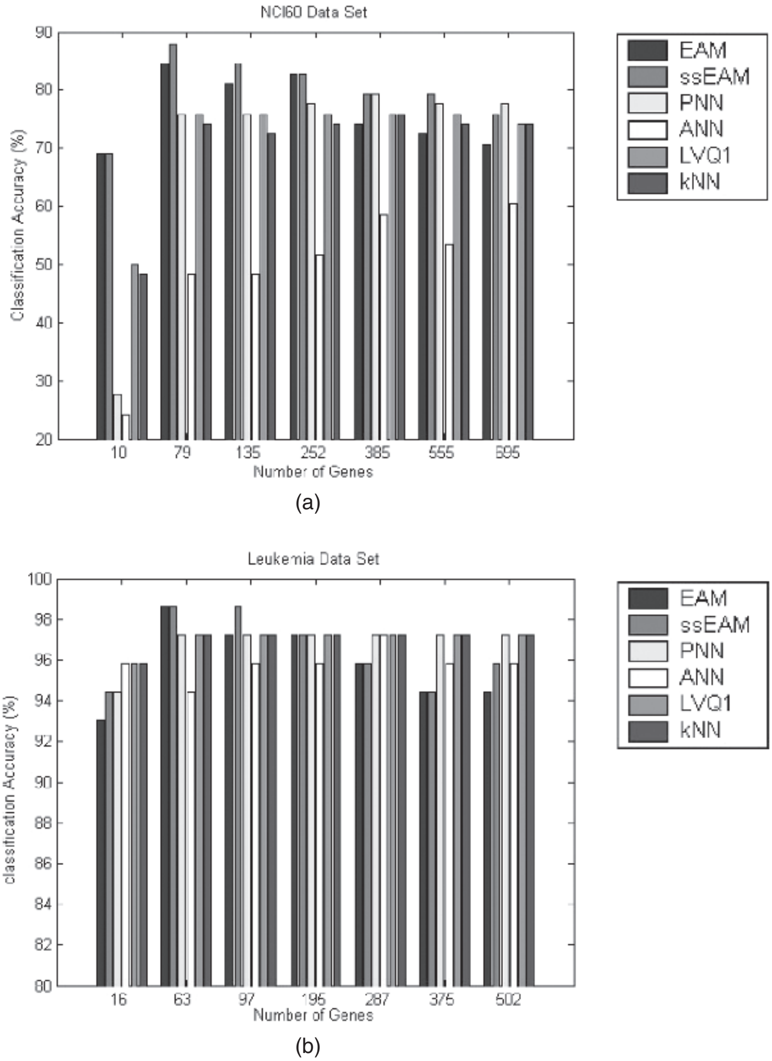
Table 1.2 Classification Accuracy for NCI60 Data Set[a]

		Number of Features (Genes)						
NCI60		10	79	135	252	385	555	695
EAM ($\varepsilon = 0$)	PSO	65.52 (1.86)	83.02 (1.51)	79.40 (1.42)	76.64 (2.47)	71.21 (1.59)	68.10 (1.90)	67.76 (1.49)
ssEAM ($\varepsilon = 0.1$)	PSO	65.78 (2.19)	84.66 (1.36)	81.12 (1.98)	78.62 (1.89)	75.26 (1.79)	73.10 (2.27)	72.50 (2.34)
PNN	Fisher criterion	24.05 (2.27)	71.12 (2.23)	72.24 (2.09)	74.65 (2.02)	76.81 (2.27)	76.03 (1.24)	76.12 (1.40)
MLP	Fisher criterion	16.81 (3.21)	39.14 (5.82)	39.40 (4.38)	44.91 (5.08)	45.43 (5.46)	45.17 (4.25)	47.59 (7.51)
kNN	Fisher criterion	41.90 (2.86)	69.22 (2.64)	69.74 (1.63)	72.59 (1.57)	73.71 (1.28)	72.59 (1.10)	71.81 (1.28)
LVQ1	Fisher criterion	42.67 (2.73)	71.81 (2.87)	72.76 (2.48)	73.10 (1.80)	73.88 (1.28)	72.33 (1.98)	72.24 (1.47)

[a] Given are the mean and standard deviation (in parentheses) of percent of correct classification of 58 tumor samples with CV10 ($\rho = 0.4$, $\mu = 0.3$, learning rate = 0.8, $\alpha = 2.5$).

used for gene selection. For PNN, the smoothing parameter of the Gaussian kernel was set to 1. The MLP consisted of a single hidden layer featuring 20 nodes and was trained with the one-step secant algorithm for fast learning. We varied the number of prototypes in LVQ1 and the value of *k* in kNN from 8 to 17 and 1 to 10, respectively. Both methods were evaluated based on average classification accuracy. From Table 1.2 it should be noted that ssEAM/PSO is superior to the other methods used in these experiments or additional ones found in the literature. Specifically, the best result attained with ssEAM/PSO is 87.9% (79 genes are selected by PSO), which is better than other results reported in the literature. We performed a formal *t* test to compare the difference between the best overall results of ssEAM and other methods. All *p* values are less than 10^{-15}, which indicates the classification accuracy for ssEAM is statistically better than those of other methods at a 5% significance level. The same conclusion can also be reached via a nonparametric Wilcoxon rank test and a Kruskal–Wallis test. Another interesting observation from the experimental results is that the introduction of the error tolerance parameter ε could provide an effective way to increase generalization capability and decrease overfitting, which is encountered frequently in cancer classification. The performance of ssEAM can usually be improved with the appropriate selection of ε (0.1 for this data set). However, the overrelaxing of the misclassification tolerance criterion during the category formation process in ssEAM training could cause the degradation of the classifier performance.

We further compared the top 100 genes selected by the Fisher criterion with those selected by PSO. This comparison shows that there is only a small fraction of overlaps between the genes chosen by these two methods. For example, for the 79



(a)

(b)

Figure 1.3 Best classification accuracy of 20 runs for the (*a*) NCI60 and (*b*) leukemia data set. Order for bars is EAM, ssEAM, PNN, ANN, LVQ1, and kNN, from left to right.

genes that lead to the best classification result, only 7 were also selected by the Fisher criterion. Although the Fisher criterion can be effective in some binary classification problems, as shown in Xu and Wunsch (2003), it does not achieve effective performance in cases of multiclass discrimination. The reason for this may lie in the fact that the Fisher criterion tends to choose many highly correlated genes, ignoring those genes that are really important for classification. In addition, the use of the Fisher discriminant criterion is justified when the data follow an approximately Gaussian distribution. This may not be true for this data set.

1.3.2 Acute Leukemia Data

The ssEAM method was evaluated on a second cancer data set. The acute leukemia data set is comprised of 72 samples that belong to 3 different leukemia types: 25 acute myeloid leukemia (AML), 38 B-cell acute lymphoblastic leukemia (ALL), and 9 T-cell ALL (Golub et al., 1999). Gene expressions for 7129 genes were measured using oligonucleotide microarrays. We ranked genes based on their variance across all the samples and chose the top 1000 for further analysis. The final matrix is in the form of $E = \{e_{ij}\}_{72\times1000}$.

As before, we compared the performance of our method with PNN, MLP, LVQ1, and kNN, based on the average results for 20 runs with different splitting. The parameters for ssEAM were $\mu = 0.9$, $\rho = 0.45$, and $a = 4$, and the learning rate was set equal to 0.8; δ is set to 0.5, 0.45, 0.4, 0.3, 0.2, 0.1, and 0.0. Additionally, the smoothing parameter of the Gaussian kernel was set to 1, as mentioned previously in this chapter. The MLP included 15 nodes in the hidden layer with the logistic function as the transfer function. The number of prototypes in LVQ1 varied from 3 to 12. For this data set, we typically achieved reasonable results after only 100 generations of evolutionary optimization. Each swarm still consisted of 50 particles. The results are shown in Table 1.3 and Figure 1.3(*b*). The best classification performance was achieved by ssEAM when 63 or 97 genes were selected with PSO. In this setting, only one sample is misclassified (i.e., a T-cell ALL67 is misclassified as a B-cell ALL). Still, classification performance deteriorates when too many or too few genes are chosen, particularly for ssEAM. The number of genes in the data does not affect much the performance of PNN, LVQ1, and kNN classifiers, although

Table 1.3 Classification Accuracy for Acute Leukemia Data Set[a]

		Number of Features (Genes)						
Acute Leukemia Data		16	63	97	195	287	375	502
EAM	PSO	89.72	94.44	95.07	94.31	93.26	93.13	92.36
($\varepsilon = 0$)		(2.08)	(2.67)	(1.65)	(1.68)	(1.58)	(1.23)	(1.39)
ssEAM	PSO	91.60	97.15	97.50	95.83	94.65	93.68	92.64
($\varepsilon = 0.1$)		(1.23)	(0.95)	(0.73)	(0.90)	(0.82)	(0.71)	(1.43)
PNN	Fisher	90.00	96.32	96.74	96.46	96.39	96.25	96.18
	criterion	(2.61)	(0.68)	(0.68)	(0.71)	(0.70)	(0.91)	(0.99)
MLP	Fisher	93.61	92.50	93.47	91.60	91.74	91.60	91.67
	criterion	(2.27)	(1.23)	(2.07)	(4.19)	(3.94)	(2.90)	(3.60)
LVQ1	Fisher	94.86	95.83	96.45	95.97	96.04	96.10	95.90
	criterion	(0.65)	(0.45)	(0.84)	(0.77)	(0.93)	(0.86)	(0.71)
kNN	Fisher	95.07	95.83	96.18	96.11	95.90	95.69	95.69
	criterion	(0.71)	(0.45)	(0.62)	(0.73)	(0.84)	(1.00)	(0.77)

[a]Given are the mean and standard deviation (in parentheses) of percent of correct classification for 72 tumor samples with CV10 ($\rho = 0.45$, $\mu = 0.9$, learning rate = 0.8, $\alpha = 4$).

there is some slight decrease when more genes are included. In contrast to the performance results obtained with the NCI60 data set, kNN and LVQ1 work well for this data set. The *p* values of the associated *t* tests comparing performances of ssEAM, PNN, ANN, LVQ1, and kNN classifiers are 0.0015, 9.9×10^{-9}, 1.6×10^{-4}, and 3.1×10^{-7}, respectively, which, again, shows the significantly better performance of ssEAM models (at a 5% significance level).

Among all examples, sample AML66 and T-cell ALL67 cause the most misclassification (e.g., when $\delta = 0.45$, 23 out of 50 particles misclassified AML66 as ALL, and 32 out of 50 particles misclassified T-cell ALL67 either as B-cell ALL or AML). This is similar to the results from other analyses (Golub et al., 1999; Nguyen and Rocke, 2002). For the acute leukemia data set, the effect of introducing the error tolerance parameter ε ($\varepsilon > 0$) is not as pronounced as in the NCI60 data set. This may be the result of reduced overlap among the Golub data in comparison to the NCI60 data set. The improvement of performance is more effective for semisupervised training when applied to higher overlapped data sets.

It is interesting to examine whether the genes selected by PSO are really meaningful in a biological sense. Among the top 50 genes selected, many of them have already been identified as the important markers for the differentiation of the AML and ALL classes. Specifically, genes such as *NME4, MPO, CD19, CTSD, LTC4S, Zyxin*, and *PRG1*, are known to be useful in AML/ALL diagnosis (Golub et al., 1999). Also, some new genes are selected that previously were not reported as being relevant to the classification problem. Additional investigation is required for these genes. Moreover, we found that the Fisher discriminant criterion can also identify genes that contribute to the diagnosis of these three leukemia types, such as with genes *Zyxin, HoxA9*, and *MB-1*. The reason for this may be due to the underlying biology represented in the Golub data: Genes express themselves quite differently under different tumor types in the AML/ALL data relative to the NCI60 data. Furthermore, we observe that different feature selection methods usually lead to different subsets of selected, informative genes with only very small overlap, although the classification accuracy does not change greatly. Genes that have no biological relevance still can be selected as an artifact of the feature selection algorithms themselves. This suggests that feature selection may provide effective insight in cancer identification; however, careful evaluation is critical due to the problems caused by insufficient data.

1.4 CONCLUSIONS

Classification is critically important for cancer diagnosis and treatment. Microarray technologies provide an effective way to identifying different kinds of cancer types, while simultaneously spawning many new challenges. Here, we utilized semisupervised ellipsoid ARTMAP and particle swarm optimization to address the multitype cancer identification problem, based on gene expression profiling. The proposed method achieves qualitatively good results on two publicly accessible benchmark data sets, particularly, with the NCI60 data set, which is not effectively dealt with by previous methods. The comparison with four other important machine learning techniques shows that the combined ssEAM/PSO scheme can outperform them on both data sets and the difference in classification accuracy is statistically significant.

With all the improvement we obtain, we also note that there are still many problems that remain to be solved in gene expression profiles-based cancer classification, particularly, the curse of dimensionality, which becomes more serious due to the rapidly and persistently increasing capability of gene chip technologies, in contrast to the limitations in sample collections. Thus, questions, such as how many genes are really needed for disease diagnosis, and whether these gene subsets selected are really meaningful in a biological sense, still remain open. Without any doubt, larger data sets would be immensely useful in effectively evaluating different kinds of classifiers and constructing cancer discrimination systems. In the meantime, more advanced feature selection approaches are required in order to find informative genes that are more efficient in prediction and prognosis.

Acknowledgments

Partial support for this research from the National Science Foundation and from the M.K. Finley Missouri endowment is gratefully acknowledged.

REFERENCES

Alizadeh, A., M. Eisen, R. Davis, C. Ma, I. Lossos, A. Rosenwald, J. Boldrick, H. Sabet, T. Tran, X. Yu, J. Powell, L. Yang, G. Marti, T. Moore, J. Hudson, Jr., L. Lu, D. Lewis, R. Tibshirani, G. Sherlock, W. Chan, T. Greiner, D. Weisenburger, J. Armitage, R. Warnke, R. Levy, W. Wilson, M. Grever, J. Byrd, D. Bostein, P. Brown, and L. Staudt (2000). "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, Vol. 403, pp. 503–511.

Ambroise, C. and G. McLachlan (2002). "Selection bias in gene extraction on the basis of microarray gene-expression data," *Proc. Natl. Acad. Sci., USA*, Vol. 99, pp. 6562–6566.

Anagnostopoulos, G. C. (2001). Novel Approaches in Adaptive Resonance Theory for Machine Learning, Doctoral Thesis, University of Central Florida, Orlando, Florida.

Anagnostopoulos, G. C. and M. Georgiopoulos (2000). "Hypersphere ART and ARTMAP for unsupervised and supervised incremental learning," Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00), pp. 59–64.

Anagnostopoulos, G. C. and M. Georgiopoulos (2001). "Ellipsoid ART and ARTMAP for incremental unsupervised and supervised learning," Proceedings of the IEEE-INNS-ENNS Intl. Joint Conf. on Neural Networks (IJCNN'01), pp. 1221–1226.

Anagnostopoulos, G. C., M. Georgiopoulos, S. Verzi, and G. Heileman (2002). "Reducing generalization error and category proliferation in ellipsoid ARTMAP via tunable misclassification error tolerance: Boosted ellipsoid ARTMAP," Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'02), pp. 2650–2655.

Ben-Dor, A., L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini (2000). "Tissue classification with gene expression profiles," Proceedings of the Fourth Annual International Conference on Computational Molecular Biology, pp. 583–598.

Bergh, F. and A. Engelbrecht (2004). "A cooperative approach to particle swarm optimization," *IEEE Trans. Evol. Computat.*, Vol. 8, pp. 225–239.

Berrar, D., C. Downes, and W. Dubitzky (2003). “Multiclass cancer classification using gene expression profiling and probabilistic neural networks,” *Pacific Symp. Biocomput.*, Vol. 8, pp. 5–16.

Carpenter, G., S. Grossberg, N. Markuzon, J. Reynolds, and D. Rosen (1992). “Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps,” *IEEE Trans. Neural Net.*, Vol. 3, pp. 698–713.

Carpenter, G, S. Grossberg, and D. Rosen (1991). “Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system,” *Neural Net.*, Vol. 4, pp. 759–771.

Deng, L., J. Pei, J. Ma, and D. Lee (2004). “A Rank Sum Test Method for Informative Gene Discovery,” Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 410–419.

Duda, R., P. Hart, and D. Stork (2001). *Pattern Classification* 2nd ed., Wiley, New York.

Eberhart, R. and Y. Shi (2001). “Particle swarm optimization: Developments, applications and resources,” Proceedings of the 2001 Congress on Evolutionary Computation, pp. 81–86.

Efron, B. (1982). *The Jackknife, the Bootstrap and Other Resampling Plans*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.

Golub, T., D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander (1999). “Molecular classification of cancer: class discovery and class prediction by gene expression monitoring,” *Science*, Vol. 286, pp. 531–537.

Grossberg, S. (1976). “Adaptive pattern recognition and universal encoding II: Feedback, expectation, olfaction, and illusions,” *Biol. Cybern.*, Vol. 23, pp. 187–202.

Hastie, T., R. Tibshirani, and J. Friedman (2003). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*, 2nd ed. Prentice Hall, Upper Saddle River, NJ.

Healy, M. and T. Caudell (1997). “Acquiring rule sets as a product of learning in the logical neural architecture LAPART,” *IEEE Trans. Neural Net.*, Vol. 8, pp. 461–474.

Healy, M., T. Caudell, and S. Smith (1993). “A neural architecture for pattern sequence verification through inferencing,” *IEEE Trans. Neural Net.*, Vol. 4, pp. 9–20.

Jaeger, J., R. Sengupta, and W. Ruzzo (2003). “Improved gene selection for classification of microarrays,” *Pacific Symp. Biocomput.*, Vol. 8, pp. 53–64.

Kennedy, J. and R. Eberhart (1997). “A discrete binary version of the particle swarm optimization,” *Proc. IEEE Intl. Conf. System, Man, Cybern.*, Vol. 5, pp. 4104–4108.

Kennedy, J., R. Eberhart, and Y. Shi (2001). *Swarm Intelligence*. Morgan Kaufmann, San Francisco.

Khan, J., J. Wei, M. Ringnér, L. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. Antonescu, C. Peterson, and P. Meltzer (2001). “Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks,” *Nature Med.*, Vol. 7, pp. 673–679.

Kohavi, R. (1995). “A study of cross-validation and bootstrap for accuracy estimation and model selection,” Proceedings of the 14th International Joint Conference Artificial Intelligence, Morgan Kaufman, San Francisco, pp. 1137–1145.

Kohonen, T. (2001). *Self-Organizing Maps*, 3rd ed., Springer, Berlin, Heidelberg.

Li, T., C. Zhang, and M. Ogihara (2004). “A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression,” *Bioinformatics*, Vol. 20, pp. 2429–2437.

McLachlan, G., K. Do, and C. Ambroise (2004). *Analyzing Microarray Gene Expression Data*. Wiley, Hoboken, NJ.

Nguyen, D. and D. Rocke (2002). “Multi-class cancer classification via partial least squares with gene expression profiles,” *Bioinformatics*, Vol. 18, pp. 1216–1226.

Ooi, C. and P. Tan (2003). “Genetic algorithms applied to multi-class prediction for the analysis of gene expression data,” *Bioinformatics*, Vol. 19, pp. 37–44.

Ramaswamy, S., P. Tamayo, R. Rifkin, S. Mukherjee, C. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. Mesirov, T. Poggio, W. Gerald, M. Loda, E. Lander, and T. Golub (2001).

"Multiclass cancer diagnosis using tumor gene expression signatures," *Proc. Natl. Acad. Sci., USA*, Vol. 98, pp. 15149–15154.

Scherf, U., D. Ross, M. Waltham, L. Smith, J. Lee, L. Tanabe, K. Kohn, W. Reinhold, T. Myers, D. Andrews, D. Scudiero, M. Eisen, E. Sausville, Y. Pommier, D. Botstein, P. Brown, and J. Weinstein (2000). "A gene expression database for the molecular pharmacology of cancer," *Nat. Genet.*, Vol. 24, pp. 236–244.

Serrano-Gotarredona, T., B. Linares-Barranco, and A. Andreou (1998). *Adaptive Resonance Theory Microchip*. Kluwer Academic, Norwell, MA.

Shi, Y. and R. Eberhart (1998). "Parameter selection in particle swarm optimization," Proceedings of the 7th Annual Conference on Evolutionary Programming, pp. 591–601.

Shi, Y. and R. Eberhart (1999). "Empirical study of particle swarm optimization," Proceedings of the 1999 Congress on Evolutionary Computation, pp. 1945–1950.

Specht, D. (1990) "Probabilistic neural networks," *Neural Net.*, Vol. 3, pp. 109–118.

West, M., C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. Olson, J. Marks, and J. Nevins (2001). "Prediction the clinical status of human breast cancer by using gene expression profile," *Proc. Natl. Acad. Sci., USA*, Vol. 98, pp. 11462–11467.

Wunsch II, D. (1991) An Optoelectronic Learning Machine: Invention, Experimentation, Analysis of First Hardware Implementation of the ART1 Neural Network. Ph.D. Dissertation, University of Washington.

Wunsch II, D., T. Caudell, D. Capps, R. Marks, and A. Falk (1993). "An optoelectronic implementation of the adaptive resonance neural network," *IEEE Trans. Neural Net.*, Vol. 4, pp. 673–684.

Xu, R., G. Anagnostopoulos, and D. Wunsch II (2006). "Multi-class cancer classification using semi-supervised ellipsoid artmap and particle swarm optimization with gene expression data," *IEEE/ACM Trans. Computat. Biol. Bioinform.*, Vol. 4, pp. 65–77.

Xu, R. and D. Wunsch II (2003). "Probabilistic neural networks for multi-class tissue discrimination with gene expression data," Proceedings of the International Joint Conference on Neural Networks (IJCNN'03), pp. 1696–1701.

Xu, R. and D. Wunsch II (2005). "Survey of clustering algorithms," *IEEE Trans. Neural Net.*, Vol. 16, pp. 645–678.