An analysis of the use of probabilistic modeling for synaptic connectivity prediction from genomic data

Roberto Santana
Intelligent Systems Group.
School of Computer Science
University of the Basque Country
UPV/EHU
Email: roberto.santana@ehu.es

Alexander Mendiburu
Intelligent Systems Group
School of Computer Science
University of the Basque Country
UPV/EHU
Email: alexander.mendiburu@ehu.es

Jose A. Lozano
Intelligent Systems Group
School of Computer Science
University of the Basque Country
UPV/EHU
Email: ja.lozano@ehu.es

***Abstract*—The identification of the specific genes that influence particular phenotypes is a common problem in genetic studies. In this paper we address the problem of determining the influence of gene joint expression in synapse predictability. The question is posed as an optimization problem in which the conditional entropy of gene subsets with respect to the synaptic connectivity phenotype is minimized. We investigate the use of single- and multi-objective estimation of distribution algorithms and focus on real data from *C. elegans* synaptic connectivity. We show that the introduced algorithms are able to compute gene sets that allow an accurate synapse predictability. However, the multi-objective approach can simultaneously search for gene sets with different number of genes. Our results also indicate that optimization problems defined on constrained binary spaces remain challenging for the conception of competitive estimation of distribution algorithm.**

I. Introduction

Neurons communicate by means of chemical and electrical contacts called synapses. Neural circuits provide the basics for information processing in animals and investigating the principles that regulate neuronal connectivity is an important problem in neuroscience [9]. One of the aspects that influences the ways in which synapses are formed is genetics. Genetically encoded molecular markers can play an important role in synaptic formation. Therefore, the identification of these markers and the elucidation of the way they guide the formation of synapses is a relevant issue.

There are important voids in the knowledge about the functioning of neural circuits in mammals. However, for other less evolved organisms different characterizations of the factors that determine neural organization have been established. One of these organisms is the nematode *C. elegans*. The synaptic connectivity network of this worm has been identified and several works have investigated its neuronal circuitry [7], [27], [26], [28]. Gene expression patterns of most of this worm's neurons are also available. Therefore, information compiled for *C. elegans* represents a unique opportunity for analyzing the influence of genetic factors in the formation of synapses. In this paper, we address the question of identifying modules of genes that synergestically contribute to the formation of synapses. We build on work presented in [25], where this question is posed as a combinatorial optimization problem.

A number of works have addressed the question of predicting the formation of synapses between any pair of neurons in *C. elegans* based on the expression pattern of the genes. In [12], the prediction of neurons connectivity from gene expression patterns was initially addressed using standard weighted K-nearest neighbor (KNN) prediction algorithm with multiclass targets. The authors also analyzed to what extent the neighborhood relations between neurons in one space (e.g., expression) were similar to their neighborhood relations in the other space (e.g., synaptic connectivity). Varadan et al. [25] introduced an entropy minimization and Boolean parsimony approach to identify sets of synergistically interacting genes whose joint expression predicts neural connectivity. Baruch et al. [4] also investigate the predictability of the neuronal synapses in *C. elegans*. They construct a probabilistic model to explain the neuronal connectivity diagram of *C. elegans* as a function of the expression patterns of its neurons. As additional data, they use information about the physical proximity of the neurons, limiting the predictions to pairs of neurons that are certain to be in physical proximity to each other in the worms body.

In [4], it is shown that synapse prediction patterns can be achieved by means of only a small number of specific genes that interact in a combinatorial fashion. However, the identification of subsets of genes that synergistically determine synaptic patterns is a difficult combinatorial problem. To deal with the exponential increase in the number of gene sets that are potentially associated with synaptic connectivity, Varadan et al. [25] propose the use of a local optimization procedure and simulated annealing [13].

In this paper, we propose the use of single- and multi-objective optimization based on probabilistic modeling of the search space to solve this problem. We show that estimation of distribution algorithms (EDAs) [14], [17], [18], a class of evolutionary algorithms that explicitly model the search space regularities in terms of probabilistic dependencies, are able to find robust solutions to this problem. The results achieved by multi-objective EDAs are particularly encouraging since they reach high-quality solutions while simultaneously finding gene sets with different number of genes. We also analyze the limitations that binary contraints arising in this problem

impose to the efficiency of EDAs.

The paper is organized as follows: In the next section, the main elements that influence synapse connectivity and neuron gene expressions in *C. elegans* are introduced. The conditional entropy minimization approach for synapse connectivity prediction are explained in Section III. Section IV defines the optimization problem and describes the used representation. Section V presents the single- and multi-objective estimation of distribution algorithms that are the essential component of our proposal. Section VI describes the experimental framework to evaluate our proposal and presents the numerical results. The main contributions of the paper are summarized in Section VII where some lines for future research are also discussed.

II. Neural wiring and gene expression patterns in *C. elegans*

C. elegans is a phylum Nematoda that lives in the soil. It has a relatively simple organism, consisting of 302 neurons and approximately 5600 synapses [28]. Since this organism has a simple neuronal connectivity pattern and its entire genome has been sequenced, it is very suitable for analyzing the relationship between gene expression and synaptic connectivity.

Synapses are morphologically distinct subcellular junctional structures, composed of a presynaptic terminal, a postsynaptic target, and the synaptic cleft aligning pre- and postsynaptic specializations [8], [11]. There are two different types of synapses in *C. elegans*: chemical synapses and gap junctions. The former occur between neighboring parallel nerve processes, or between nerve processes and muscle arms [28]. The second are organelles that mediate electrical and metabolic coupling between cells [5]. Figure 1 shows the local synaptic connectivity pattern[1] between neuron ASH and all its neighbors. In Figure 1, chemical synapses are represented by arrows and gap junctions with bars.

To introduce the synaptic connectivity problem addressed in this paper we use the definitions originally introduced by Varadan et al. [25]:

1) Neurons are represented as a list $c_1, c_2 \ldots, c_K$, where K is the total number of neurons.
2) The topology of the chemical synapses in the wiring diagram is specified by a $K \times K$ adjacency matrix A, defined so that A_{ij} is 1 if presynaptic c_i connects to postsynaptic c_j with at least one chemical synapse, and 0 otherwise.
3) List of genes for which expression data is available: $l_1, \ldots, l_M$, where M is the total number of such genes.
4) Genes that are expressed in any neuron are specified by a gene expression matrix E, defined so that E_{ij} is 1 if l_i is expressed in c_j and 0 otherwise.

Our analysis is based on the connectivity data previously used in [7], [25]. In this paper we focus our analysis on the class of chemical synapses. The number of neurons is $K = 280$.

[1]This figure has been generated using the interactive tool for visualization *C. elegans* neural network available at http://wormweb.org/details.html

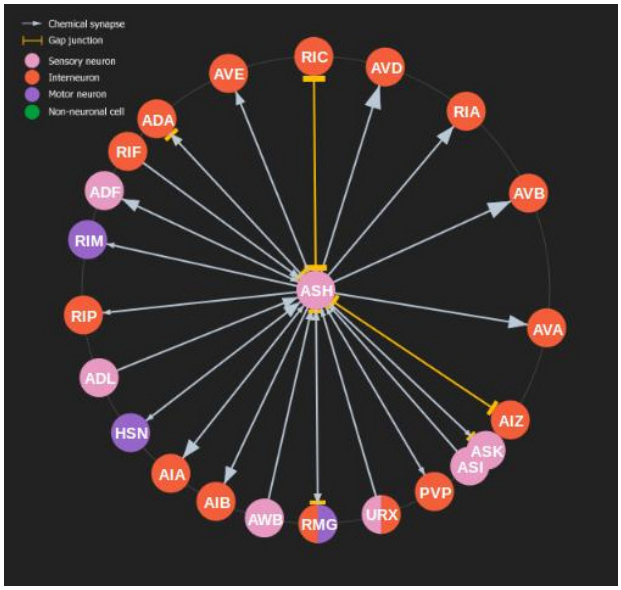

Fig. 1. Chemical and gap junction synapses of neuron ASH

III. Synaptic connectivity prediction problem definition

In [25], the following problem is defined: *Given a number m, identify the set of m genes (subset of the set of all M genes corresponding to the rows of matrix E), each of which is associated to either the presynaptic or the postsynaptic neuron, whose joint expression pattern predicts the existence of a synapse with minimum uncertainty.*

Basically, we intend to find a set of m genes, where m is given, such that the genetic expression of these genes could predict with a high probability whether there is chemical synapse between any pair of neurons c_i (pre-synaptic) and c_j (post-synaptic). A gene can serve as a good predictor of synapse in the pre-synaptic neuron, in the post-synaptic neuron, or in both neurons simultaneously. In the next section we show how the search of the genes that predict the synapse with minimum uncertainty can be solved by finding the set of genes for which the conditional entropy, i.e. the entropy of the genes states conditioned on the existence of a synapse, is minimized.

A. Genes as predictors of synaptic connectivity

To give an intuitive idea of how the genes are used as predictors of synapses we present a simple illustrative example. Table I represents the gene expression states $(l_1^{pre}, l_2^{pre}, \ldots, l_M^{pre}, l_1^{post}, l_2^{post}, \ldots, l_M^{post})$ in every possible pair of neurons c_i, c_j where c_i and c_j are assumed to be the putative synaptic neurons. When there is actually a synapse between the two neurons the random binary variable C is one, otherwise it is zero. Notice that any gene l_i is considered twice, according to whether it is in the pre- or post-synaptic neuron.

Now imagine that we want to measure how good genes l_r^{pre}, l_s^{post} are for predicting a synapse. From Table I, we could

	l_1^{pre}	l_2^{pre}	...	l_M^{pre}	l_1^{post}	l_2^{post}	...	l_M^{post}	C
c_1, c_1	1	0	...	0	0	1	...	1	1
c_1, c_2	1	0	...	0	0	0	...	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
c_1, c_n	0	1	...	1	0	1	...	1	1
c_i, c_1	0	1	...	1	1	1	...	1	1
c_i, c_2	1	1	...	1	0	1	...	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
c_i, c_n	1	0	...	0	0	0	...	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
c_n, c_n	1	1	...	0	1	0	...	0	1

TABLE I
EXAMPLE OF THE USE OF GENES AS PREDICTOR VARIABLES.

compute the probabilities $p(C = 1 \mid l_r^{pre}, l_s^{post})$ for all possible states of l_r^{pre}, l_s^{post} (for these two genes there are only four possible states). If the two genes can serve to predict the synapse with absolute certainty, then we expect that for all possible states of l_r^{pre}, l_s^{post}. $p(C = 1|l_r^{pre}, l_s^{post})$ will be either zero or one indicating that when the genes are in this joint state it is sure that there will be synaptic connexion or there will be no synapse. This rationale leads to the use of the conditional entropy to evaluate different gene sets.

B. Conditional entropy

The conditional entropy [23] quantifies the remaining entropy (i.e. uncertainty) of a random variable $\mathbf{Y}$ given that the value of another random variable $\mathbf{X}$ is known. It is referred to as the entropy of $\mathbf{Y}$ conditional on $\mathbf{X}$, and it is denoted as $H(\mathbf{Y} \mid \mathbf{X})$:

$$H(\mathbf{Y} \mid \mathbf{X}) = \sum_{\mathbf{x} \in \mathcal{X}} p(\mathbf{x}) H(\mathbf{Y} \mid \mathbf{X} = \mathbf{x}) \tag{1}$$

where, taking into account that the variable follows a Bernoulli distributio, $H(q) = -qlog_2(q) - (1-q)log_2(1-q)$.

Let $\mathbf{G} = (G_1, G_2, \ldots, G_m)$ be the set of m binary variables representing a set of genes. $\mathbf{G}$ can be partitioned into two sets. The sets $\mathbf{G}^{pre}$ and $\mathbf{G}^{post}$ that respectively represent genes in the pre- and post-synaptic neurons ($\mathbf{G} = \mathbf{G}^{pre} \cup \mathbf{G}^{post}$). $\mathbf{g} = (g_1, g_2, \ldots, g_m)$ will be an assignment specifying the joint expression state of $\mathbf{G}$, where g_i defines the individual expression state for the i-th gene. We use $g_i = 1$ to represent that gene G_i is expressed and $g_i = 0$ otherwise. Each set of m genes has 2^m possible gene expression states.

Let $P(\mathbf{g})$ and $Q(\mathbf{g})$ respectively be the probability of state $\mathbf{g}$ in a randomly ordered pair of neurons, and the probability of a synapse given state $\mathbf{g}$. Then, the conditional entropy of C given $g_1, g_2, \ldots, g_m$ is:

$$H(C \mid g_1, g_2, \ldots, g_m) = \sum_{\mathbf{g}, P(\mathbf{g})>0} P(\mathbf{g}) H(Q(\mathbf{g})) \tag{2}$$

$P(\mathbf{g})$ and $Q(\mathbf{g})$ were calculated [25] as:

$$P(\mathbf{g}) = \frac{N_0(\mathbf{g}) + N_1(\mathbf{g})}{K^2} \tag{3}$$

$$Q(\mathbf{g}) = \frac{N_1(\mathbf{g})}{N_0(\mathbf{g}) + N_1(\mathbf{g})}, \text{ if } P(\mathbf{g}) > 0 \tag{4}$$

where for each expression state $\mathbf{g}$, the number $N_1(\mathbf{g})$ represents the number of times, from the $K \times K$ possible neuron pairs, that the given state is associated with a synapse. Similarly, $N_0(\mathbf{g})$ is the number of times that the state $\mathbf{g}$ was in a pair that it is not associated with a synapse.

Equation (2) can be also expressed as [25]:

$$H(C \mid g_1, g_2, \ldots, g_m) = H(C) - I(g_1, g_2, \ldots, g_m; C) \tag{5}$$

Finally, to ensure that the range of possible values extends from 0 to 1, we normalize the conditional entropy by dividing by $H(C)$, the entropy corresponding to the "null probability" Q_{null}. of a synapse in a randomly chosen pair of neurons. Therefore, for a given set of m genes, our fitness function $f(\mathbf{g})$ is computed as:

$$f(\mathbf{g}) = \frac{H(C \mid g_1, g_2, \ldots, g_m)}{H(C)} \tag{6}$$

pre	post	post				
vab-7	unc-8	str-3	N_0	N_1	Q	Q/Q_{null}
0	0	0	8765	145	0.02	0.58
0	0	1	64171	0	0	0
0	1	0	0	1979	1	35.71
0	1	1	0	540	1	35.71
1	0	0	4	326	0.01	0.58
1	0	1	0	4	1	35.71
1	1	0	2384	0	0	0
1	1	1	20	66	0.77	27.40

TABLE II
EXAMPLE OF THE PROBABILISTIC TABLE COMPUTATION FOR SYNAPTIC CONNECTIVITY.

Table II shows an example of the way the probabilistic table is computed. In this example, there are 3 genes; 1 presynaptic gene (vab-7) and two post-synaptic genes (unc-8 and str-3). All the 8 joint expression states for these genes are shown in the table. For each expression state, N_0 and N_1 have been computed. Notice, for instance, that whenever gene vab-7 is not expressed in the pre-synaptic neuron and gene unc-8 is expressed in the post-synaptic neuron there is a synapse between the two neurons. Using N_0 and N_1, the probability of a synapse (Q) is computed for all states. In this example $Q_{null} = 0.028$. From the ratio between Q and Q_{null} (last column in Table II) it is possible to identify which genes states are more likely to produce a synapse.

IV. OPTIMIZATION PROBLEM

A. Solution representation

Let $\mathbf{X} = X_1, X_2, \ldots, X_n$ be a set of $n = 2M$ binary variables. For $1 \leq i \leq M$, $x_i = 1$ will represent that G_i belongs to the set of pre-synaptic neurons, ($x_i = 0$) otherwise.

Similarly, for $M+1 \leq i \leq 2M$, $x_i = 1$ will represent that G_{i-M} belongs to the set of post-synaptic neurons. Each feasible solution $\mathbf{x}$ should satisfy the following constraints:

$$\sum_{i=1}^{M} x_i \geq 1 \tag{7}$$

$$\sum_{i=M+1}^{n} x_i \geq 1 \tag{8}$$

$$\sum_{i=1}^{n} x_i = m \tag{9}$$

The first two constraints enforce that there is at least one pre-synaptic and one post-synaptic neuron. The third constraint guarantees that there are exactly m genes.

B. Previous optimization approaches

In [25], two different search methods were used to determine the gene set with minimum conditional entropy. The first method is a greedy algorithm that starts with a randomly chosen gene set of size m, and iteratively modifies it by replacing one of its genes, chosen at random, with a new gene, also chosen at random from the entire set of $2M$ genes, such that the entropy is minimized. The process is terminated when the entropy has converged to zero or after a given time. Local minima are avoided by repeating the iterative algorithm with random initial conditions of the same size and selecting the gene set that yields the overall lowest entropy. For gene sets of size $m+1$, at least one of the initial solutions was forced to contain the solution found for the gene set of size m. The second algorithm that was used is an adapted version of simulating annealing [13].

A limitation of these techniques is that they are based on a single-point search and do not use information about the solutions that have been already generated. Finally, a more fundamental limitation is that, in general, we do not know in advance which the number m of genes that can be involved in the synapses is. It would be more convenient a search technique that simultaneously searches in an interval of possible values for m.

V. Probabilistic-based optimization in a constrained space

The main idea of our approach is to use probabilistic modeling of the search space to increase the search efficiency. We also intend to determine if by simultaneously searching for different m values the accuracy and efficiency of the search can be improved. Finally, we investigate if a hybrid scheme that combines global search with local improvement of the best found solutions can outperform the local and global optimization algorithms for these problems. From the point of view of the analysis of the evolutionary algorithms it should be noted that the binary search space is constrained by the specific characteristics of the problem solutions. Therefore the algorithm should be adjusted to deal with this situation. This is a complex issue because constraints can determine the arousal of dependencies that are not due to the problem characteristics [20]. To deal with this problem we investigate a number of variants of estimation of distribution algorithms (EDAs) [14], [17], [18].

EDAs are a class of evolutionary algorithms that apply probabilistic modeling of the selected solutions instead of crossover operators. The rationale behind probabilistic modeling is to explicitly model the relationships between the variables of the problem in terms of probabilistic dependencies which are captured by probabilistic graphical models (PGMs). PGMs are employed to generate new solutions that will likely resemble the selected points. EDAs are specially relevant for their application in problems from Bioinformatics since they can be used to extract relevant a priori unknown information about the problem domain [2], [21], [10].

Distinctive features of EDAs are the type of probabilistic model, and the particular class of learning and sampling methods. The models may differ in the order and number of probabilistic dependencies that they represent.

A. Univariate and bivariate marginal approximations

The EDAs that we use differ in the number of function objectives (single-objective versus two-objective optimization) and the probabilistic method used (univariate versus bivariate approximation).

In the univariate marginal product model all variables are independent, i.e. no dependencies are represented in the model. The joint probability distribution of such univariate marginal product model [17] can be factorized as follows:

$$p_u(\mathbf{x}) = \prod_{i=1}^{n} p(x_i). \tag{10}$$

In the univariate marginal distribution algorithm (UMDA) [17], the marginal probabilities are computed from the selected population (see Step 5 of Algorithm 1). New solutions are generated by independently sampling from each $p(x_i)$.

The second model learns a probabilistic model based on a tree. In this model, each variable may depend on no more than one variable that is called the parent. The probability distribution $p_{Tree}(\mathbf{x})$ used by Tree-EDA [3], [22] is defined as

$$p_{Tree}(\mathbf{x}) = \prod_{i=1}^{n} p(x_i \mid pa(x_i)), \tag{11}$$

where $pa(X_i)$ is the parent of variable X_i in the tree, and $p(x_i \mid pa(x_i)) = p(x_i)$ when $pa(X_i) = \emptyset$, i.e. when X_i is the root of the tree.

In UMDA, new solutions are generated by sampling each variable independently. To generate a new solution in Tree-EDA, each variable is sampled given the assignment of each parent in the tree. The way sampling is conducted in UMDA and Tree-EDA does not guarantee that the new generated solutions will satisfy constraints represented in Equations (7), (8), and (9). The first two constraints are relatively easy

to satisfy and we deal with then during the evaluation of solutions. In the unlikely case when a solution does not satisfy one of these two constraints it is discarded by assigning it a very low fitness evaluation. The fulfillment of Equation (9) is more difficult and discarding unfeasible solutions is not a valid alternative in this case.

Algorithm 1: **EDA**

```
1  D_0 ← Sample I individuals using a uniform distribution
2  t ← 1
3  do {
4  Evaluate D_{t-1}
5      D^{Se}_{t-1} ← Select N individuals from D_{t-1} using truncation selection (single-objective case) or Pareto-ranking selection (multi-objective case)
6      Learn a probabilistic model from D^{Se}_{t-1}
7      D_t ← Sample I individuals from the probabilistic model according to sample method
8  } until Stop criterion is met
```

1) Repairing unfeasible solutions: One straightforward way to guarantee the feasibility of solutions is by repairing them. There are two possible situations. If $\sum_i^n x_i > m$, then $\sum_i^n x_i - m$ randomly selected ones are changed to zeros. If $\sum_i^n x_i < m$, the most likely case given that $m << n$, the repairing procedure changes $m - \sum_i^n x_i$ zeros to ones. The repairing procedure introduces a mutation-like effect into the search. The reparing procedures may also have an important impact in the generated solutions, particularly at the beginning of the search when many solutions are unfeasible. However, as evolution advances and the values of some variables are fixed, the influence of the repairing procedure decreases.

B. Evolutionary multi-objective optimization based on EDAs

As previously commented, in searching for the set of genes that allows to predict synaptic connectivity there is no reason in assuming a fixed value of m. We could be interested in knowing the minimum conditional entropies for different values of m and also in investigating whether there are genes that belong to the optimal solutions for different m. In this paper we propose to simultaneously search for a set of optimal solutions for different m using a multi-objective approach.

The main idea of our approach is to search for optimal sets of m genes where $m \in [m_1, m_2]$. Since the conditional entropy tends to decrease as m is increased, minimizing the conditional entropy and minimizing m are conflicting objectives. In this situation a Pareto-set optimization method that provides a diverse set of solutions can simultaneously provide optimal or close-to-optimal solutions for different values of m.

1) Pareto dominance: We consider a maximization problem with $k = 2$ objective functions $f_i(\mathbf{x}) \rightarrow \mathbb{R}, i \in \{1, \ldots, k\}$, where the vector function $\mathbf{f}$ maps each solution $\mathbf{x} \in \mathcal{X} \subset \{0,1\}^n$ to an objective vector $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})) \in \mathbb{R}^k$.

It is also assumed that the underlying dominance structure is given by the Pareto dominance relation "$\mathbf{y}$ dominates $\mathbf{x}$" that is defined as $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X},\ \mathbf{x} \preceq_{\mathcal{F}} \mathbf{y} \iff f_i(\mathbf{x}) \leq f_i(\mathbf{y})\, \forall i$ and $\exists j :\ f_j(\mathbf{x}) < f_j(\mathbf{y})$, where $\mathcal{F} = \{f_1, \ldots, f_k\}$. The Pareto (optimal) set is given as $\{\mathbf{x} \in \{0,1\}^n \mid \nexists \mathbf{y} \in \{0,1\}^n \setminus \{\mathbf{x}\} : \mathbf{x} \preceq_{\mathcal{F}} \mathbf{y}\}$. It contains solutions that are non-dominated. The associated Pareto front contains the vector of function evaluations for each of the Pareto set members. The extreme points of the Pareto set include the solutions that maximize each of the objectives.

For implementational reasons, we have transformed our original minimization problems into the maximization of the corresponding negative problems. Therefore $f_1 = -\sum_{i=1}^n x_i$ and f_2 will be the negative of the conditional entropy (Equation 6).

2) EDA implementation: There may be important differences between EDA implementations for single and multi-objective problems. Enforcing the population diversity needed to guarantee a good covering of the Pareto set is particularly important for multi-objective problems and specialized learning and sampling methods may be conceived to fulfil this goal. Different EDAs have been used to address multi-objective optimization problems [6], [15].

We have adapted the EDAs to deal with multi-objective problems by modifying the selection step. The selection method employed uses Pareto ranking selection where individuals are ordered according to the front they belong to. Individuals in the first front (non-dominated solutions) come first. Then individuals that are only dominated by those in the first front and so on. Within each front, they are ordered according to the average rank of their fitness functions. After all the population has been ordered, truncation selection of the T percentage of the population is done. A pseudocode of the EDAs is shown in Algorithm 1.

Summarizing, the four EDA variants used in our experiments are: UMDA, Multi-UMDA, Tree-EDA, and Multi-Tree-EDA. All these algorithms have been programmed in MATLAB [24] using MATEDA2.0 software [19].

VI. Experiments

The main goal of our experiments is to investigate whether optimization based on probabilistic models is a feasible alternative for synaptic connectivity prediction from genomic data. As mentioned before, problems with constraints are challenging for EDAs and they have been scarcely investigated. The second goal of our experiments is to determine whether the use of probabilistic dependencies between the variables can play a relevant role to improve the EDA results. To this end, we compare the results of UMDA and Tree-EDA. Another question to be addressed in our experiments is the evaluation of the performance of the multi-objective approach for simultaneous determination of the relevant genes for different m.

We search for optimal solutions for $m \in [3, 12]$. The population size was $N = 200$ and 40 generations were allowed. Since best elitism was used, the total number of function

evaluations was $200+100\cdot 39 = 4100$. The computation of the fitness function is relatively costly, particularly for large m. We compare the EDA results with a local optimization procedure as implemented by Varadan et al. [25]. We allow the local optimization method a maximum of 4100 evaluations. The code of the simulated annealing used in the same paper was not available from the authors. We conducted 30 experiments for all the optimization algorithms. For the multi-objective experiments, we compute a unique Pareto-set approximation from all the solutions evaluated in a single run of the algorithm. Therefore, we have 30 Pareto-sets for each EDA. Similarly, we construct an absolute Pareto-set approximation by joining the Pareto-sets from the 30 runs.

In our first experiment, we computed the best and mean fitness values, out of the 30 experiments for the EDAs and the local optimization algorithm. These results are shown in Table III. The local optimization algorithm outperforms the EDAs both in terms of the best solutions found and the mean of the solutions. Exceptions are problems with $8 \leq m \leq 10$ for which Tree-EDA was better in terms of the best solution or the average of the best solutions found. Regarding the use of dependencies, results shown in Table III indicate that using probabilistic dependencies notably improves the results of the EDAs even when constraints are present.

m	UMDA		Tree-EDA		Local	
	best	*mean*	*best*	*mean*	*best*	*mean*
3	0.9221	0.9259	0.9221	0.9264	0.9221	0.9252
4	0.8983	0.9031	0.8973	0.9011	0.8973	0.8996
5	0.8796	0.8885	0.8796	0.8852	0.8765	0.8832
6	0.8689	0.8786	0.8620	0.8680	0.8538	0.8660
7	0.8576	0.8668	0.8345	0.8516	0.8331	0.8458
8	0.8424	0.8500	0.8126	0.8298	0.8142	0.8293
9	0.8186	0.8367	0.7956	0.8091	0.7965	0.8097
10	0.7956	0.8178	0.7739	0.7924	0.7747	0.7839
11	0.7890	0.8020	0.7571	0.7698	0.7516	0.7644
12	0.7672	0.7828	0.7305	0.7433	0.7275	0.7389

TABLE III
RESULTS OF THE LOCAL OPTIMIZATION METHOD AND THE EDAS

m	Multi-UMDA			Multi-Tree-EDA		
	napp	*best*	*mean*	*napp*	*best*	*mean*
3	30	0.9239	0.9331	30	0.9221	0.9251
4	29	0.8973	0.9145	30	0.8973	0.8997
5	27	0.8875	0.8955	30	0.8765	0.8823
6	26	0.8768	0.8860	30	0.8538	0.8659
7	26	0.8493	0.8731	30	0.8331	0.8484
8	25	0.8419	0.8594	30	0.8140	0.8314
9	25	0.8263	0.8503	30	0.7995	0.8184
10	22	0.8233	0.8394	28	0.7867	0.8079
11	27	0.8170	0.8294	20	0.7756	0.7985
12	21	0.7974	0.8187	5	0.7716	0.7873

TABLE IV
RESULTS OF MULTI-UMDA AND MULTI-TREE-EDA

We also investigate how the results achieved by the multi-objective EDAs compare with the ones achieved by their single-objective counterparts and whether the use of dependencies can improve the results also in the multi-objective case.

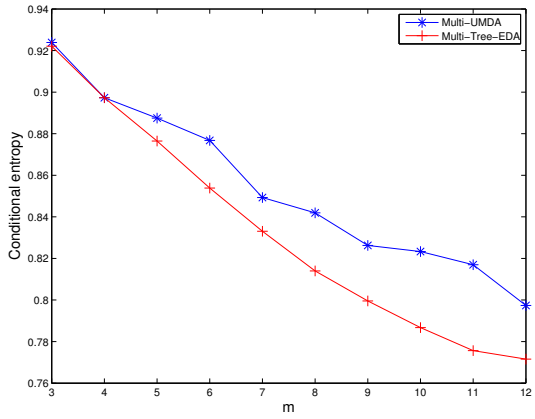

Fig. 2. Pareto set approximation found by the Multi-objective EDAs

Table IV shows the results of Multi-UMDA and Multi-Tree-EDA for the multi-objective optimization. In the table, *napp* is the number of times that there was a (non-dominated) solution with value m in the Pareto-set approximation. Solutions with a particular m value might be dominated and will not appear in the Pareto set. In Table IV, *best* and *mean* are respectively the best and mean computed from the solutions from the Pareto set that have the same number of genes m. In the table, the solutions where the multi-objective EDAs have equal or better values than their respective single-objective variants appear underlined. It is remarkable that the Multi-Tree-EDA achieves equal or better results than Tree-EDA for all $m \leq 7$. Both, in terms of the best fitness and of the mean fitness, Multi-Tree-EDA can also outperform the local optimization algorithm in some of the problems. These results show that multi-objective optimization can be a more efficient alternative to deal with this problem although it may need a higher population size than the single-objective EDAs.

It can also be seen in Table IV that Multi-Tree-EDA always finds solutions with lower conditional entropy than Multi-UMDA. The Pareto sets found by Multi-Tree-EDA are more diverse, at least for $m \leq 10$. For higher values of m, the Pareto sets found by Multi-Tree-EDA are less diverse. However, the conditional entropies are still better than for Multi-UMDA. In Figure 2, we also compare the absolute Pareto-set approximations obtained by Multi-UMDA and Multi-Tree-EDA.

We hypothesized that the single-objective EDAs were able to reach solutions close to the optimal but, as is common to other evolutionary algorithms, they were unable to conduct a fine-grain search leading to the optimum. To test this hypothesis, we apply the local optimization algorithm to the best solution found in each run of the algorithm. A maximum of 2000 evaluations was allowed. For fair comparison, we applied the local optimization algorithm also to the 30 best solutions found in the previous application of this method and with the same number of function evaluations. Results are shown in Table V.

An analysis of Table V reveals that although the results of the enhanced EDAs, in particular of the Tree-EDA, equals

m	UMDA		Tree-EDA		Local	
	best	*mean*	*best*	*mean*	*best*	*mean*
3	0.9221	0.9238	0.9221	0.9239	0.9221	0.9245
4	0.8973	0.8995	0.8973	0.8987	0.8973	0.8987
5	0.8796	0.8835	0.8765	0.8827	0.8765	0.8823
6	0.8620	0.8672	0.8538	0.8643	0.8538	0.8643
7	0.8345	0.8481	0.8331	0.8479	0.8331	0.8415
8	0.8146	0.8272	0.8126	0.8248	0.8127	0.8243
9	0.7958	0.8054	0.7956	0.8050	0.7956	0.8054
10	0.7782	0.7881	0.7728	0.7843	0.7739	0.7804
11	0.7521	0.7639	0.7530	0.7624	0.7516	0.7603
12	0.7294	0.7412	0.7262	0.7348	0.7236	0.7343

TABLE V
RESULTS OF THE LOCAL OPTIMIZATION METHOD AND THE HYBRID EDAS

the results of the local optimizer in terms of the best solution found, the average results are not competitive with the simpler local optimization algorithm. The reasons for this behavior may be that the search regularities captured by the probabilistic models are somewhat disrupted by the application of the repairing procedure. Probabilistic dependencies between the variables arise and can be used to improve the search, as the results achieved by Tree-EDA show. However, more sophisticated methods seem to be needed to deal with the binary constraints.

Another important question is whether there are genes that contribute to descreasing the conditional entropy for different values of m. These would be the critical genes that influence synaptic connectivity. We computed the most frequent genes that are in the optimal solutions obtained using UMDA and Multi-UMDA for different m values. These genes are displayed in Figures 3 and 4. It can be seen in the figures, that for both algorithms there is a gene set that is comprised in many of the optimal solutions while the majority of genes are never selected. There are also similarities between the gene sets captured in the best solutions found by UMDA and Multi-UMDA. These results seem to indicate that the heuristic of starting from previously best found solutions may indeed lead to savings in the search for optimal gene sets for higher m values.

VII. CONCLUSIONS

In this paper we have proposed the use of EDAs for synaptic connectivity prediction from genomic data. We have also proposed to simultaneously address the determination of gene sets for different values of m and presented initial results that show this approach outperforming, at least for certain values of m, the single-objective EDAs and the local optimization algorithms. More research is needed to determine if a better selection of the parameters (i.e. higher population size) or more sophisticated diversity-preserving evolutionary operators can further improve the results of the multi-objective EDAs.

We have also shown that the Pareto-sets found by the EDAs can be used to investigate the relationships between gene sets with different values of m and whether there are genes that are frequent in solutions for different m. In future work we intend to mine the EDA probabilistic models looking for interactions which could reveal interesting insights for the biological domain. Our results point to the importance of solving current EDA limitations to deal with constrained problems. Finally, we emphasize that the identification of interacting genes is not only of interest for predicting synapse formation [1]. Phenotypical characteristics are usually determined by sets of genes involved in pathways. Efficient EAs able to solve this problem could be extended to discover gene modules involved in pathways associated to other phenotypical characteristics. One possible direction for improving the results presented in this paper is the combination of the local optimization tecnique with the EDA approaches, similarly to the way memetic algorithms [16] are usually applied.

VIII. ACKNOWLEDGMENTS

This work has been partially supported by the Saiotek and Research Groups 2007-2012 (IT-242-07) programs (Basque Government), TIN2010-14931 and Consolider Ingenio 2010 - CSD 2007 - 00018 projects (Spanish Ministry of Science and Innovation) and COMBIOMED network in computational biomedicine (Carlos III Health Institute).

REFERENCES

[1] D. Anastassiou. Computational analysis of the synergy among multiple interacting genes. *Molecular systems biology*, 3(1), 2007.

[2] R. Armañanzas, I. Inza, R. Santana, Y. Saeys, J. L. Flores, J. A. Lozano, Y. Van de Peer, R. Blanco, V. Robles, C. Bielza, and P. Larrañaga. A review of estimation of distribution algorithms in bioinformatics. *BioData Mining*, 1(6):doi:10.1186/1756–0381–1–6, 2008.

[3] S. Baluja and S. Davies. Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. In D. H. Fisher, editor, *Proceedings of the 14th International Conference on Machine Learning*, pages 30–38, San Francisco, CA., 1997. Morgan Kaufmann.

[4] L. Baruch, S. Itzkovitz, M. Golan-Mashiach, E. Shapiro, and E. Segal. Using expression profiles of caenorhabditis elegans neurons to identify genes that mediate synaptic connectivity. *PLoS Computational Biology*, 4(7):e1000120, 2008.

[5] M. Bennett, L. Barrio, T. Bargiello, D. Spray, E. Hertzberg, J. Saez, et al. Gap junctions: new tools, new answers, new questions. *Neuron*, 6(3):305, 1991.

[6] P. A. Bosman and D. Thierens. Multi-objective optimization with diversity preserving mixture-based iterated density estimation evolutionary algorithms. *International Journal of Approximate Reasoning*, 31(3):259–289, 2002.

[7] B. L. Chen, D. H. Hall, and D. B. Chklovskii. Wiring optimization can relate neuronal structure and function. *Proceedings of the National Academy of Sciences (PNAS)*, 103(12):4723–4728, 2006.

[8] W. Cowan, T. Südhof, and C. Stevens. *Synapses*. Johns Hopkins Univ Pr, 2003.

[9] J. DeFelipe. From the connectome to the synaptome: an epic love story. *Science*, 330(6008):1198, 2010.

[10] H. Franken, A. Seitz, R. Lehmann, H. Häring, N. Stefan, and A. Zell. Inferring disease-related metabolite dependencies with a bayesian optimization algorithm. In *Proceedings of the Conference Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, pages 62–73, Malaga, Spain, 2012. Springer.

[11] Y. Jin. Synaptogenesis. 2005.

[12] A. Kaufman, G. Dror, I. Meilijson, and E. Ruppin. Gene expression of caenorhabditis elegans neurons carries information on their synaptic connectivity. *PLoS computational biology*, 2(12):e167, 2006.

[13] S. Kirkpatrick, C. D. J. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, May 1983.

[14] P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, Boston/Dordrecht/London, 2002.

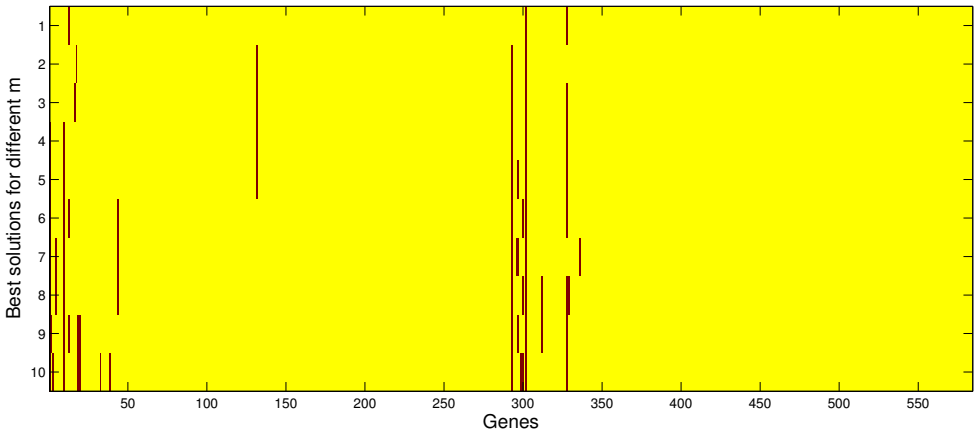

Fig. 3. Most frequent genes found in the best solutions obtained using UMDA for different m.

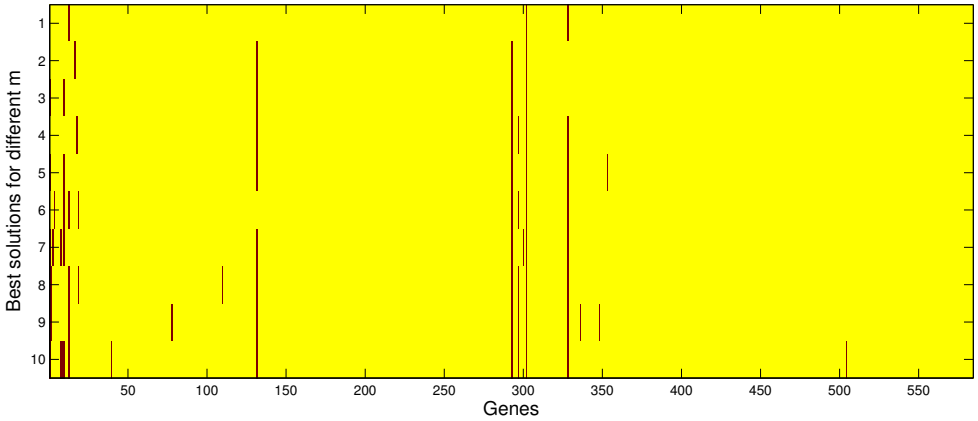

Fig. 4. Most frequent genes found in the absolute Pareto-set obtained using Multi-UMDA for different m.

[15] L. Marti, J. Garcia, A. Berlanga, C. A. Coello, and J. M. Molina. On current model-building methods for multi-objective estimation of distribution algorithms: Shortcommings and directions for improvement. Technical Report GIAA2010E001, Department of Informatics of the Universidad Carlos III de Madrid, Madrid, Spain, 2010.

[16] P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. C3p report caltech concurrent computation program, California Institute of Technology, 1989.

[17] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN IV*, volume 1141 of *Lectures Notes in Computer Science*, pages 178–187, Berlin, 1996. Springer.

[18] M. Pelikan, D. E. Goldberg, and F. Lobo. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1):5–20, 2002.

[19] R. Santana, C. Bielza, P. Larrañaga, J. A. Lozano, C. Echegoyen, A. Mendiburu, R. Armañanzas, and S. Shakya. Mateda-2.0: A MATLAB package for the implementation and analysis of estimation of distribution algorithms. *Journal of Statistical Software*, 35(7):1–30, 2010.

[20] R. Santana, E. P. de León, and A. Ochoa. The incident edge model. In A. Ochoa, M. R. Soto, and R. Santana, editors, *Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99)*, pages 352–359, Havana, Cuba, March 1999.

[21] R. Santana, P. Larrañaga, and J. A. Lozano. Adding probabilistic dependencies to the search of protein side chain configurations using EDAs. In G. Rudolph, T. Jansen, S. Lucas, C. Poloni, and N. Beume, editors, *Parallel Problem Solving from Nature - PPSN X*, volume 5199 of *Lecture Notes in Computer Science*, pages 1120–1129, Dortmund, Germany, 2008. Springer.

[22] R. Santana, A. Ochoa, and M. R. Soto. The mixture of trees factorized distribution algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2001*, pages 543–550, San Francisco, CA, 2001. Morgan Kaufmann Publishers.

[23] C. Shannon and W. Weaver. *The mathematical theory of communication*, volume 19. University of Illinois Press Urbana, 1962.

[24] The MathWorks, Inc. *MATLAB – The Language of Technical Computing, Version 7.5*. The MathWorks, Inc., Natick, Massachusetts, 2007.

[25] V. Varadan, D. Miller III, and D. Anastassiou. Computational inference of the molecular logic for synaptic connectivity in C. elegans. *Bioinformatics*, 22(14):e497–e506, 2006.

[26] L. Varshney, B. Chen, E. Paniagua, D. Hall, and D. Chklovskii. Structural properties of the caenorhabditis elegans neuronal network. *PLoS Computational Biology*, 7(2):e1001066, 2011.

[27] S. Wang, W. Pei, and Z. He. Random walks on the neural network of C. elegans. In *Proceedings of the 2008 International Conference on Neural Networks and Signal Processing*, pages 142–145, 2008.

[28] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner. The structure of the nervous system of the nematode Caenorhabditis elegans. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 314:I–340, 1986.