BMC Medical Genomics

RESEARCH **Open Access**

Assessment of gene order computing methods for Alzheimer's disease

Benqiong Hu[1], Gang Jiang[2], Chaoyang Pang[2], Shipeng Wang[2], Qingzhong Liu[3], Zhongxue Chen[4], Charles R Vanderburg[5], Jack T Rogers[6], Youping Deng[7], Xudong Huang[6*]

From The 2011 International Conference on Bioinformatics and Computational Biology (BIOCOMP'11)
Las Vegas, NV, USA. 18-21 July 2011

Abstract

Background: Computational genomics of Alzheimer disease (AD), the most common form of senile dementia, is a nascent field in AD research. The field includes AD gene clustering by computing gene order which generates higher quality gene clustering patterns than most other clustering methods. However, there are few available gene order computing methods such as Genetic Algorithm (GA) and Ant Colony Optimization (ACO). Further, their performance in gene order computation using AD microarray data is not known. We thus set forth to evaluate the performances of current gene order computing methods with different distance formulas, and to identify additional features associated with gene order computation.

Methods: Using different distance formulas- Pearson distance and Euclidean distance, the squared Euclidean distance, and other conditions, gene orders were calculated by ACO and GA (including standard GA and improved GA) methods, respectively. The qualities of the gene orders were compared, and new features from the calculated gene orders were identified.

Results: Compared to the GA methods tested in this study, ACO fits the AD microarray data the best when calculating gene order. In addition, the following features were revealed: different distance formulas generated a different quality of gene order, and the commonly used Pearson distance was not the best distance formula when used with both GA and ACO methods for AD microarray data.

Conclusion: Compared with Pearson distance and Euclidean distance, the squared Euclidean distance generated the best quality gene order computed by GA and ACO methods.

Background

A brief introduction of Alzheimer's disease

Being the most common form of age-related dementia, Alzheimer's disease (AD) affects 5.4 million Americans, and at least $183 billion will be spent in 2011 on care of AD and other dementia patients. The problem is worsening as life expectancy continues to increase. By 2050, the projected number of AD patients could range from 11 to 16 million people in the United States alone if no cure or preventive measure for AD is found. Hence, AD has quickly become a pandemic and exacted a huge socioeconomic toll [1].

AD is named after Dr Alois Alzheimer, who has first investigated the disease [2]. Later on, the autopsies of brain examinations of most cases of senility under light microscope were discovered to be extracellular deposits of β-amyloid and intracellular deposits of neurofibrillary tangles (NFTs). Abundant amounts of these lesions in the brain were necessary for a confirmed diagnosis of AD [3]. In 1984, an possible AD-related gene on chromosome 21 was implied when Glenner and Wong reported on the amino acid sequence of the main component of β-amyloid-, an approximate 4.3 kD peptide that they coined as "amyloid-β protein"(Aβ) based on their analysis of cerebrovascular amyloid derived from patients with Down's

* Correspondence: xhuang3@partners.org
[6]Neurochemistry Laboratory, Department of Psychiatry, Massachusetts General Hospital and Harvard Medical School, Charlestown, MA 02129, USA
Full list of author information is available at the end of the article

syndrome [4]. This study has laid the foundation for AD's "amyloid hypothesis" which claims that the accumulation of Aβ, as determined by its generation versus clearance in the brain, is the primary driver of AD-related pathogenesis, including neuronal cell death.

Frangione et al reported on the sequencing of the exons 16 and 17 of amyloid precursor protein (APP) to reveal the first pathogenic mutation in APP [5]. Finally the subsequent sequencing of these same two APP exons (encoding the Aβ portion of the molecule) that were truly linked to chromosome 21 led to the discovery of the first AD-related mutation [6]. Following this finding, Pericak-Vance and colleagues reported a significant genetic linkage of the more common late-onset of AD (> 65 years) to chromosome 19 [7]. Then in 1993, they found a common polymorphism in the gene encoding Apolipoprotein E (APOE)- APOE allele 4, is associated with increased risk for AD [8]. In 1993, the first study aimed at investigating the Presenilins as putative AD genes offered evidence for a significant association between a single-nucleotide polymorphism (SNP) in intron 8 of the Presenilin 1 (PSEN1) gene and AD. Estimates were that the common variants in PSEN1 could account for nearly half of the population-attributable risk for AD than was found for the APOE4 allele [9]. Then in 2001, a report investigating a consecutive series of referral-based AD cases found coding sequence mutations in 11% of the samples, suggesting that PSEN1 mutations may indeed be more frequent in the general population than had been previously assumed [10,11]. Furthermore, reports indicated that changes in the promoter region could lead to an altered expression pattern of the protein in neurons [12].

Currently, the mainly proposed therapeutic intervention for AD is anti-amyloid approach, which ranges from interdicting amyloidogenic processing of the β-amyloid precursor protein (APP) to removing amyloid plaques in the brain [13]. In addition to therapies based on curbing the production of Aβ or enhancing its clearance, another therapeutic strategy would be aimed at attenuating Aβ toxicity and neuroinflammation in the AD brain. Perhaps, the most effective way to approach the blocking of Aβ toxicity would be to prevent the formation of neurotoxic Aβ oligomers [3,14]. As APP, the Presenilins, and APOE represent the only firmly established AD genes to date for AD, they represent the most effective means of curbing the production of Aβ or accelerating the clearance and degradation of this peptide in the brain [3]. The identification of the remaining genes involved in AD will enable investigators and clinicians to further delineate the path of biological events that lead to AD-related neurodegeneration [3].

Introduction of gene clustering and gene order

Having been applied to many biological domains, such as drug discovery, molecular diagnosis, and toxicological research, DNA microarray technology is used most importantly to generate gene data, which holds a lot of biological information. One common data structure of a microarray data set is the presentation of a matrix. In matrix X, element X_{ij} represents the expression level of the *i-th* gene in the *j-th* experiment. Then the *i-th* line vector of matrix X represents a group of expression levels of the *i-th* gene. The *i-th* line vector contains the biological information of the *i-th* gene, and it is often used as an atom object of data to be processed.

One important aspect of biology is to make similar genes cluster together. Since line vectors of a matrix contain the information of genes, clustering similar vectors together is equivalent to cluster similar genes together. A number of algorithms were proposed to cluster gene expression profiles. Eisen *et al.* [15] applied hierarchical clustering [16], a widely used tool [17-20], to solve the problem. It also has some variants [21,22]. Self-organizing maps (SOMs) [23,24] and k-means clustering [25] were also used for the same purpose. Ben-Dor *et al.* [26] developed an algorithm-cluster affinity search technique (CAST), that has a good theoretical basis. Merz and Zell [27] proposed a memetic algorithm for the problem, formulated as finding the minimum sum-of-squares clustering [28,29].

To achieve a much better quality of clustering, the computing concept of gene order has been proposed. Gene order is the permutation of all line vectors in such a way that all the line vectors are ordered one by one in a sequence, and that similar vectors are ordered together. A gene is associated with a line vector of a matrix. The optimal gene order refers to the permutation that results in a sequence that all the vectors line up via the minimal distance. Alternatively, computing optimal gene order is equivalent to identifying a route of the traveling salesman problem (TSP) in which every vector associates with a gene that has been abstracted as a virtual city [30-35].

Since TSP is an NP-hard problem, the computation of the optimal gene order is NP-hard and only the approximation of the optimal gene order can be calculated. To obtain the approximation of the optimal gene order, Tsai *et al.* applied a family competition genetic algorithm (FCGA) [33-36] and Seung-Kyu *et al.* applied a hybrid genetic algorithm (NNGA) [37].

Introduction of ant colony optimization (ACO)

First introduced in 1992, ant colony optimization (ACO) is a novel nature-inspired method based on the foraging behavior of real ants to solve TSP. (Dorigo, 1992; Dorigo *et al.*, 1996, 1999; Dorigo and Stützle, 2004) [38]. When searching for food, ants initially explore the area surrounding their nest in a random manner. As soon as an ant finds a food source, it evaluates it and carries some food back to the nest. During the return trip, the ant deposits a pheromone trail on the ground. The pheromone deposited, the amount of which may depend on the quantity and quality of the food, guides other ants to the food source. As it has been shown (Goss *et al.*, 1989), indirect communication among ants via pheromone trails enables them to find the shortest paths between their nests and food sources. ACO generates the TSP route of the highest quality in general compared with other methods. However, it is a challenge to apply ACO to calculating gene order; its running time has been too long even for input data that has less than 1000 elements when a common personal computer is used. To make ACO better suited for the computation of gene order, we have improved its running speed by factors of at least 200 [39,40].

Introduction of genetic algorithm

Genetic algorithm (GA) can be understood as an intelligent probabilistic search algorithm that works on Darwin's principle of natural selection and that can be applied to a variety of combinatorial optimization problems [41]. More to the point, GAs are based on the evolutionary process of biological organisms in nature about which theoretical foundations were originally developed by Holland [32]. During the course of evolution, natural populations evolve according to the principle of natural selection and "survival of the fittest". Individuals who are more successful in adapting to their environments will have a better chance of surviving and reproducing, whilst individuals who are less fit will be eliminated.

To understand the outline of GA as in [42], the following original statement is given:

A GA simulates these processes by taking an initial population of individuals and applying a genetic algorithm to their reproduction. In optimization terms, each individual in the population is encoded into a string or chromosome that represents a possible solution to a given problem. The fitness of an individual is evaluated with respect to a given objective function. Highly fit individuals or solutions have opportunities to reproduce by exchanging pieces of their genetic information, in a crossover procedure, with other highly fit individuals. This produces new "offspring" solutions (i.e., children), who share some characteristics taken from both parents [43].

To date, there are few types of tools to calculate gene order. In our knowledge, GA [35] and ACO [39] are mostly used methods. Our study intends to address this question- which method is a better for AD gene order computation using AD microarray data under different conditions. Herein, we reported that ACO fits the AD microarray data the best when calculating gene order in comparison to the GA methods tested in this study.

Methods

This study intends to answer the question of which algorithm, between ACO and GA, generates the optimal AD gene order. The distance formula, which measures the similarity degree of two genes, is the key parameter that affects the quality of gene order. With different distance formulas (see the following Formulae 1-3), the gene orders will be calculated using the tools of ACO and GA in this section. Then, the quality of gene order will be measured both by the fitness function and by a heat map.

Traveling salesman problem (TSP)

TSP is introduced below:

Assume that there are n cities and a distance matrix $D = [d_{ij}]$, where d_{ij} is the distance between city i and city j, and TSP is the problem of finding a permutation π of all the cities such that minimizes $\sum_{i=1}^{n-1} d_{\pi(i),\pi(i+1)} + d_{\pi(n),\pi(1)}$.

Measurement of gene similarity

As aforementioned, a gene associates with a vector and the similarity of two genes can be estimated by the distance between the two vectors.

For two genes, different metric measurements will measure out different degrees of possible similarity. That is, the estimation of gene similarity is sensitive to the distance formula.

Many distance formulas of vectors to measure the similarity of genes are presented, such as Pearson correlation, absolute correlation, Spearman rank correlation [44], Kendall rank correlation [45], and Euclidean distance. In this paper, three popular distance formulas are introduced below.

The first distance measure is the Pearson correlation:

Let k-dimensional vector $X = (x_1, x_2, ..., x_k)$ and $Y = (y_1, y_2, ..., y_k)$ be the expression levels of two genes X and Y, which are observed over a series of k conditions. The Pearson correlation of two genes X and Y is

$$s_{X,Y} = \frac{1}{k}\sum_{i=1}^{k}(\frac{x_i - \overline{X}}{\sigma_X})(\frac{y_i - \overline{Y}}{\sigma_Y})$$

where $\overline{X}$ and σ_X is the mean and the standard deviation of the expression levels, respectively. The value of σ_X is

$$\sigma_X = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(x_i - \overline{X})^2}$$

Pearson distance is defined as

$$D_P(X, Y) = 1 - s_{X,Y} \quad (1)$$

The second distance is the Euclidean distance:

$$D_E(X, Y) = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \quad (2)$$

The third distance measure is the squared Euclidean distance:

$$D_{\mathrm{SE}}(X,Y) = \sum_{i=1}^{k} (x_i - y_i)^2 \tag{3}$$

Gene order

As it is introduced before, a gene is associated with a vector that is derived from microarray data. In this way, a gene can be regarded as a virtual city whereby each coordinate is a vector. Two associated genes are more similar as the distance shortens between two virtual cities. As it is introduced at Section 1, an optimal (shortest) TSP route for a given set of virtual cities is the optimal gene order that is a permutation of all genes. In an optimal TSP route, closed cities are ordered together and the length of the route is that which is the shortest. In an optimal gene order, similar genes cluster together, and the quality of clustering is optimal globally. This is in contrast to many clustering methods that are only optimal locally.

Currently optimal gene order cannot be calculated perfectly because it is an NP-hard problem; only an approximation can be achieved. Therefore, we need a function to measure the quality of the approximation. The following function $Q(\pi)$ is called a fitness function:

$$Q(\pi) = \sum_{i=1}^{n} D(g_{\pi_i}, g_{\pi_{i+1}}) \tag{4}$$

where g_i denotes a vector associated with a gene, π denotes a gene order, n is the number of genes, $D(g_i, g_{i+1})$ is the distance between gene g_i and gene g_{i+1}, and $g_{\pi_{n+1}} = g_{\pi_1}$. The distance formula $D(g_i, g_{i+1})$ can be chosen from Pearson distance, Euclidean distance, squared Euclidean distance, Spearman distance, and other measurements.

Function $Q(\pi)$ is a measurement of the quality of the gene order. The smaller the function value $Q(\pi)$ is, the better the quality of the gene order π is.

However, the measurement of function $Q(\pi)$ is not consistent with the fact of biology, and a true review of the quality of gene order depends on the review of a biologist. A biologist often reviews the quality of gene clustering by visually observing its heat map, and he or she often gets heuristic information from that heat map.

Apply ACO to calculate optimal gene order

To generate the optimal gene order, ACO is applied as it is below:

Step 1: Use the distance formula to compute the distance between genes.

Step 2: Initialize the pheromone trails for all edges between genes (or virtual cities) and put m ants at different genes to travel. Pre-assign an iteration number $t_{\max}$ and let $t = 0$, where t denotes the t - th iteration computation.

Step 3: *while*$(t < t_{\max})$

{

Step 3.1: Each ant selects its next city according to the transition probability $p_{ij}^{k}(t)$.

The transition probability of the k - th ant from the i - th gene to j - th gene is defined as

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum\limits_{s \in allowed_k} \tau_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)}, & j \in allowed_k \\ 0 & otherwise \end{cases}$$

where $allowed_k$ denotes the set of genes that can be accessed by the k - th ant; $\tau_{ij}(t)$ is the pheromone value of the edge (i, j); $\eta_{ij}(t)$ is the local heuristic function and $\eta_{ij}(t) = 1/d_{ij}$, and where d_{ij} are the distance between the i - th gene and j - th gene; the parameters α and β determine the relative influence of the trail strength and the heuristic information, respectively.

Step 3.2: After all ants finish their travels, all pheromone values $\tau_{ij}(t)$ are updated according to the following formula.

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t)$$

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \dfrac{Q}{L_k}, e(i,j) \in L_k \\ 0, \; else \end{cases}$$

where L_k is the length of the route passed by the k - th ant; ρ is the persistence of the trail; Q denotes constant quantity of pheromone; and $e(i, j)$ represents the edge between gene i and gene j.

Step 3.3: $t = t + 1$

}

Step 4: End procedure and select the TSP route that has the minimum length as the output.

Apply GAs to calculate optimal gene order

As mentioned before, the calculation of gene order can be converted to TSP. To make GA fit to process TSP and gene order, the commonly used GA is modified a little. The modifications are listed below:

First, the roulette rule [46] is used to design selection probability.

Second, the crossover probability is set to be 1.0 in this paper. That is, the crossover will occur definitely.

Third, the mutation is designed to occur. Between the parent and mutated offspring, the one which has the better fitness value is selected as the genuine offspring, and the others are discarded.

The modified GA is described below:

Step 1: Initialization: Set the maximum iteration number to $t_{\max}$. The *t-th* iteration step is denoted by *t*. In this paper, the length of the chromosome is set to be the number of AD genes, which is denoted by *L*. The initial population is denoted by P_{old}, and its size is set to be *N*.

Step 2: The next generation is denoted by P_{new}, and it is initialized to be an empty set. In addition, a counter is used, which is denoted by *c*, and it is initialized to be 1.

Step 3: Selection

1. Calculate each chromosome's fitness value according to formula (4).
2. Calculate the proportion (ratio) of the fitness value of each chromosome.
3. A ratio is chosen by the roulette rule, and its associated chromosome is chosen too. According to this method, two chromosomes are chosen, which are denoted by C_1 and C_2.

Step 4: Crossover

1. Generate two random integer numbers between 1 and *L*, which are denoted by C_{point1} and C_{point2} ($C_{point1} < C_{point2}$), and where C_{point1} and C_{point2} are used to indicate the positions of two crossovers on chromosomes C_1 and C_2.
2. Denote the part of C_2 from C_{point1} to C_{point2} as C_{t2}, and copy it to the head of C_1. The increased chromosome C_1 is denoted by C_1'.
Denote the part of C_1 from C_{point1} to C_{point2} as C_{t1}, and copy it to the head of C_2. The increased chromosome C_2 is denoted by C_2'.
3. Find every gene that lies in chromosome C_{t2} and C_1, which is denoted by *x* (i.e., $x \in C_{t2} \cap C_1$). Delete every *x* from C_1, and add C_{t2} to the head of updated C_1 (i.e., $C_1' \leftarrow C_{t2} \cup (C_1 - \{x\})$). The updated C_1' is regarded as temporary offspring of C_1 and denoted as $T_{offspring1}$.
Using the same method, the temporary offspring of C_2 is generated, which is denoted as $T_{offspring2}$.

Step 5: Mutation

Select a point on $T_{offspring1}$ randomly as a mutation point, which is denoted by M_{point1}. Suppose the value of mutation point M_{point1} is V_{old}. Generate a random integer between 1 and *L* , which is denoted by V_{new}. Set V_{new} as the updated value of point M_{point1}.

Find the point at which value is equal to V_{new} except point M_{point1}, and update its value as V_{old}.

The chromosome $T_{offspring1}$ is updated, and it is a true offspring.

Using the above method, chromosome $T_{offspring2}$ can also be updated, and it is a true offspring.

Step 6: Add the two true offspring into the set P_{new}, which represents the new population. Update the counter: $c = c + 2$, if $c < N$, go to Step 3, or else go to Step 7.

Step 7: Joint population P_{old} and P_{new} (i.e., $P = P_{old} \cup P_{new}$). Select *N* chromosomes from set *P* to cover the old population P_{old} for which the fitness values are smaller than the other chromosomes.

Step 8: Increase the iteration step: $t = t + 1$. If $t < t_{\max}$, and go to step 2, or else go to Step 9.

Step 9: End the algorithm and choose the chromosome that has the smallest fitness value from the last population P_{old} as the output.

Kirk presented an improved GA (IGA) program [47], and it consists of three parts: mutation, group, and iteration.

Part I (operation of mutation)

Suppose there is a chromosome$\{a_1, a_2, a_3, a_4, a_5, a_6\}$, and it is a permutation of genes a_1, a_2, a_3, a_4, a_5 and a_6. Firstly, cut a sub-sequence from the chromosome randomly, and suppose it is $\{a_2, a_3, a_4, a_5\}$. Three types of mutations are listed below:

Flip operation M_f:

Flip the gene positions of the sub-sequence. For example, $\{a_2, a_3, a_4, a_5\} \xrightarrow{M_f} \{a_5, a_4, a_3, a_2\}$.

Swap operation M_s:

Swap the positions of the two terminal genes-$\{a_2, a_3, a_4, a_5\} \xrightarrow{M_s} \{a_5, a_3, a_4, a_2\}$.

Slide operation M_l:

Shift the gene to the next position by a rotation-$\{a_2, a_3, a_4, a_5\} \xrightarrow{M_l} \{a_3, a_4, a_5, a_2\}$.

Part II (group)

Suppose *N* chromosomes, denoted by s_1, s_2, s_3, ..., and s_N, are generated randomly where *N* is divisible by 4. And all chromosomes are saved in a table *T* sequentially. In table *T*, every 4 chromosomes is grouped as a team sequentially. For every team, perform the following operations:

Firstly, select the chromosome with the minimal fitness value as seed, and discard the other three chromosomes.

Secondly, let the mutation operation M_f, M_s and M_l act on the seed, respectively, and generate three mutated chromosomes.

Thirdly, all chromosomes in this team are updated as the seed and the three mutated chromosomes, which updates table T.

Part III (iteration computation)

An operation of a group is called an iteration computation. Within every iteration, an optimal chromosome will be generated for which the fitness value is minimal compared to the other N - 1 chromosomes. Suppose R_t is the optimal chromosome of the *t-th* iteration. After all, iterations are performed on the set $\{R_1, R_2, ..., R_{t_{\max}}\}$ for a given number of iteration $t_{\max}$. The solution is selected from $\{R_1, R_2, ..., R_{t_{\max}}\}$, which has a minimal fitness value.

Source data

In this paper, the AD microarray data was downloaded from GEO Datasets, NCBI [48], which includes 22283 genes. Four cases of control, incipient, moderate, and severe data are provided in the original data. Nine samples of control are organized to form a matrix with a size of 22283 lines by 9 columns. The format of this matrix is shown in Table 1. In this matrix, each line vector is a 9-dimensional vector that represents microarray data of a gene collected from nine different conditions. All line vectors form a data set.

Seven samples of incipient for each gene are selected to form a 7-dimensional vector, and the resulting 22283 vectors are used to form a data set; eight samples of moderate for each gene are selected to form an 8-dimensional vector and to form a data set; and seven samples of severe for each gene are selected to form a data set.

In addition, according to the usual practice, all data of the AD gene is log-transformed for smoothing.

Computing parameters and environment

All data tested by GAs and ACO run on a personal computer, CPU (2): 2.99 GHZ, 3.0 GHZ; Memory: 1.0 GB.

The parameters of ACO are set below:

$$\alpha = 1, \beta = 2, \rho = 0.7, Q = 100, \tau_{ij}(0) = 1, m = 50, t_{\max} = 100.$$

The parameters of GA are set below:

$$t_{\max} = 500, M = 400,$$

where $t_{\max}$ and M represents the maximal number of iterations and the size of populations, respectively.

The parameters for the improved genetic algorithm are set as below:

$$t_{\max} = 2000, M = 900.$$

In addition, in GA, parameter values of $t_{\max}$ and M are smaller than parameters in IGA, respectively. The reason that the parameter value is different is that GA is much slower than IGA, and a high value of parameter will require excessive GA program running time.

Results and discussion

The results are showed in Figure 1 to Figure 3, and Table 2 to Table 3. From these figures and tables, we discovered that:

(1) ACO was better suited than GA to calculate the gene order of the AD genes tested in this paper.

(2) Both for ACO and GAs, the use of different distance formulas generated a different quality of gene order. The squared Euclidean distance generated the best quality overall compared with the Pearson distance and Euclidean distance.

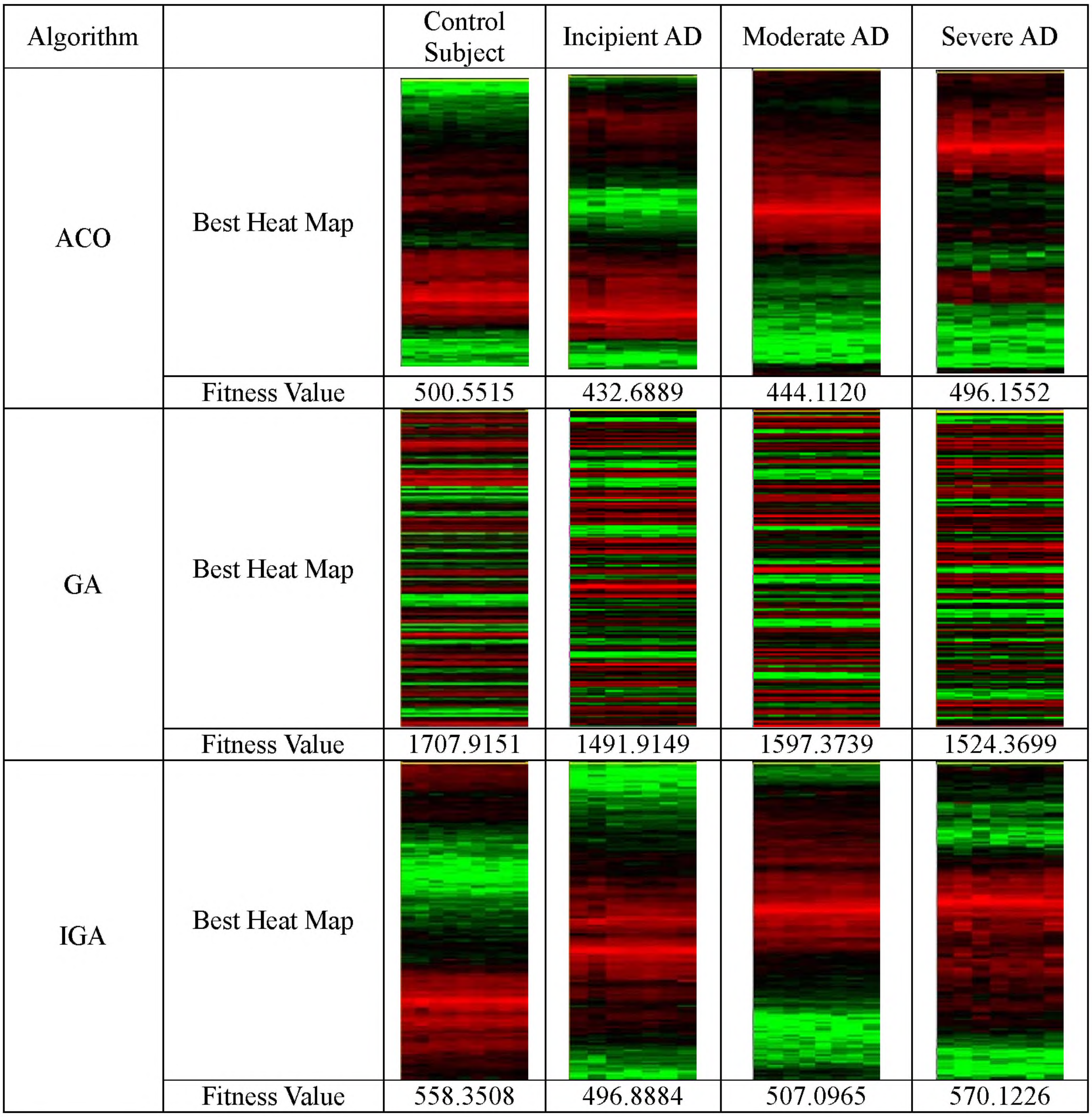
Pearson distance is a popular distance formula that is commonly used to calculate gene order. However, we found that Pearson distance is not the optimal distance formula for the calculation of gene order associated with AD genes. In this paper, the original data is not normalized, the reason for which is explained below:

Suppose two genes and their associated vectors are $X = (x_1, x_2, ..., x_k)$ and $Y = (y_1, y_2, ..., y_k)$. If all components of the vector are normalized, they become small real value that is less than 1.0. Value $S = \sum_{i=1}^{k} (x_i - y_i)^2$ is small, and it is close to zero if the two genes are very similar. Then the value of the square-root $\sqrt{S}$ has a big error because it must be expressed as base operations (+, -, ×, and ÷) to approximate. That is why Pearson distance, Euclidean distance and other distance formulas generate

Table 1 The illustration of organization of AD microarray data

AFFX -NAME	GSM 21215	GSM 2127	GSM 2128	GSM 21219	GSM 21220	GSM 21221	GSM 21226	GSM 21231	GSM 21232
BioB-5_at	8.937	9.941	8.986	9.305	9.366	8.781	9.236	9.35	9.386
BioB-M_at	9.278	10.56	9.55	10.08	10.23	9.355	9.915	10.27	10.37
BioB-3_at	7.92	9.033	8.71	8.993	9.353	8.381	8.716	9.481	9.299
BioC-5_at	10.18	11.46	10.49	10.76	10.88	10.25	10.52	10.87	10.91

*Each column of the data represents the result of one microarray test. Each line of the data represents the expression levels of the same gene under different conditions. All data was log-transformed.



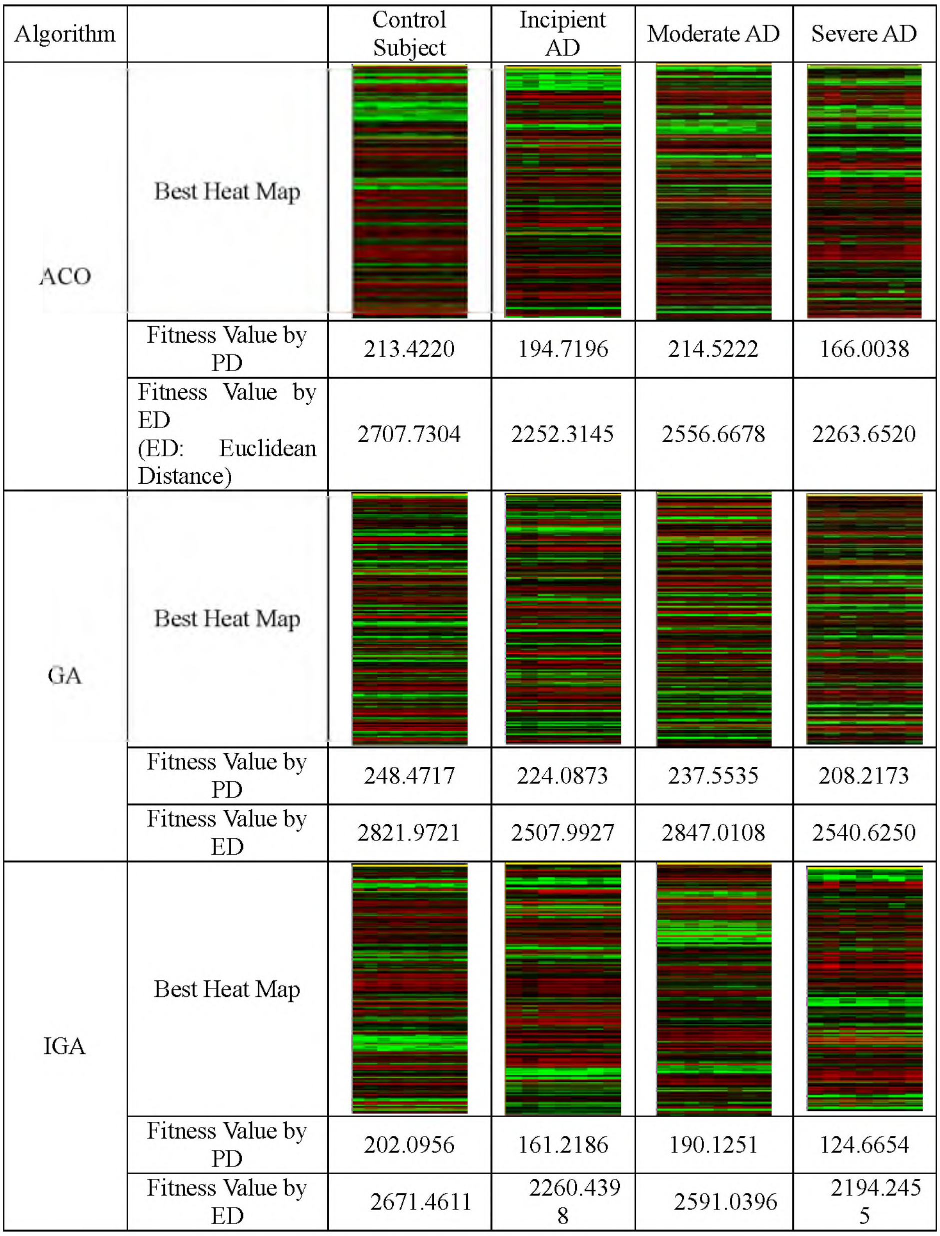
Algorithm		Control Subject	Incipient AD	Moderate AD	Severe AD
ACO	Best Heat Map				
	Fitness Value	500.5515	432.6889	444.1120	496.1552
GA	Best Heat Map				
	Fitness Value	1707.9151	1491.9149	1597.3739	1524.3699
IGA	Best Heat Map				
	Fitness Value	558.3508	496.8884	507.0965	570.1226

Figure 1 The comparison of the quality of gene order generated by ACO and GA using Euclidean distance. *Ancillary information for figures:1. All microarray data are downloaded from [48], and the data from the 1st line to 300th line are used to do experiment and for other figures and tables. 2. Every heat map is the optimal gene order, which has the smallest value of fitness function and was selected from tests performed over 40 times. In addition, the distance formula used in the fitness function (see formula 4) is the Euclidean Distance. 3. All of the figures listed in this paper are generated by TreeView, which was developed by Dr Elson, and is downloaded from the website: http://rana.lbl.gov/downloads/TreeView/TreeView_vers_1_60.exe. 4. Because most of the expression levels of the AD gene data are larger than zero, the average value of every column is subtracted when the heat map is shown. Otherwise, all heat maps are red, and the display is incorrect.

lower qualities of gene order calculation compared with squared Euclidean distance.

Conclusion

With AD being the most common form of senile dementia, the study of AD-associated genes is an imperative research subject. One important branch of an AD gene study is to cluster AD genes with the highest quality; gene order generates a better quality of clustering than other methods in general. In addition, our results of the experiment support the following conclusion: ACO is better than GA in AD gene order computation. Further, the following computational features were revealed in our study: For both ACO and GA, different distance



Algorithm		Control Subject	Incipient AD	Moderate AD	Severe AD
ACO	Best Heat Map				
	Fitness Value by PD	213.4220	194.7196	214.5222	166.0038
	Fitness Value by ED (ED: Euclidean Distance)	2707.7304	2252.3145	2556.6678	2263.6520
GA	Best Heat Map				
	Fitness Value by PD	248.4717	224.0873	237.5535	208.2173
	Fitness Value by ED	2821.9721	2507.9927	2847.0108	2540.6250
IGA	Best Heat Map				
	Fitness Value by PD	202.0956	161.2186	190.1251	124.6654
	Fitness Value by ED	2671.4611	2260.4398	2591.0396	2194.2455

Figure 3 The comparison of the quality of gene order generated by ACO and GA using Pearson distance formula.

Table 2 The statistical comparison of the quality of gene order

Algorithm	Distance	Control man	Incipient patient	Moderate patient	Severe patient
ACO	ED	507.9163	442.7255	459.7381	504.0716
GA	ED	1800.9287	1582.5394	1689.2580	1604.3304
IGA	ED	566.0912	508.6311	516.0917	579.3226
ACO	SED	484.8221	419.8804	437.9346	479.5701
GA	SED	1916.9891	1679.9281	1789.6030	1682.0008
IGA	SED	576.9810	521.2992	529.8852	593.4252
ACO	PD	2737.5938	2233.1848	2518.7568	2167.4011
GA	PD	2882.9409	2532.2205	2708.5082	2515.8520
IGA	PD	2712.5501	2319.1112	2513.9173	2218.1910

Notation: ED: Euclidean Distance; PD: Pearson Distance; SED: Squared Euclidean Distance

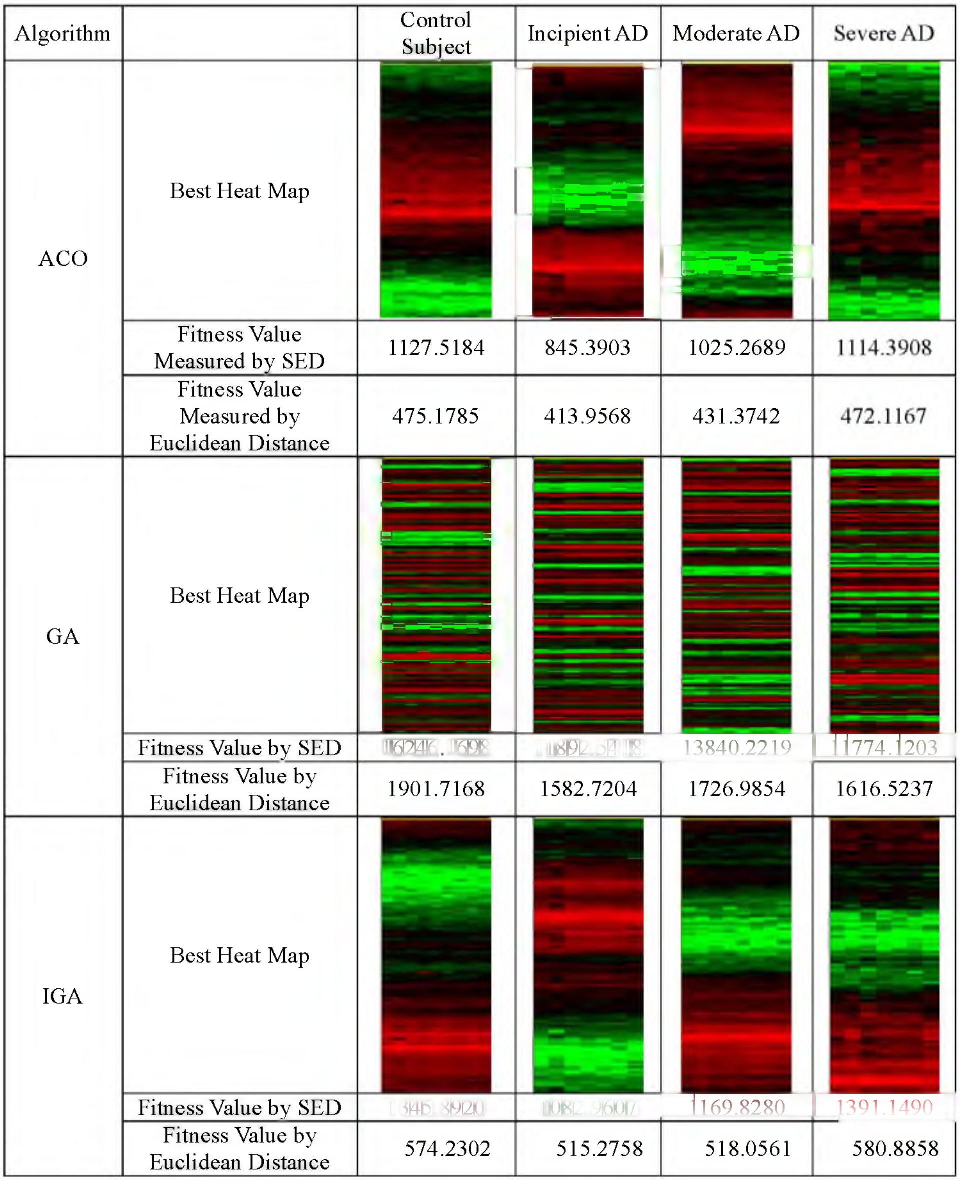
Ancillary information: all data in this table is the value of the fitness function, and it is the average of 40 times of tests. In addition, the distance formula used to calculate fitness value is ED. Every data in Table 5 corresponds to an average runtime.

Table 3 The statistical comparison of the runtime of ACO, GA and IGA

Algorithm	Distance	Control man	Incipient patient	Moderate patient	Severe patient
ACO	ED	122.0545	121.8582	121.8611	121.8653
GA	ED	580.8345	586.7355	588.9012	586.7427
IGA	ED	133.0079	131.2152	140.4218	139.1710
ACO	SED	109.8382	110.0110	109.7321	110.2532
GA	SED	186.4143	184.5551	185.1629	185.7899
IGA	SED	126.8957	126.9276	126.9757	127.0232
ACO	PD	123.0438	122.8454	122.6719	122.6450
GA	PD	186.9550	187.5644	187.0732	188.4089
IGA	PD	129.8745	127.7448	127.0051	126.4476

Notation: ED: Euclidean Distance; PD: Pearson Distance; SED: Squared Euclidean Distance

Ancillary Information: Every runtime in this table is the average of 40 times of tests. In addition, every runtime corresponds to a fitness value i listed at Figure 3.



Algorithm		Control Subject	Incipient AD	Moderate AD	Severe AD
ACO	Best Heat Map				
	Fitness Value Measured by SED	1127.5184	845.3903	1025.2689	1114.3908
	Fitness Value Measured by Euclidean Distance	475.1785	413.9568	431.3742	472.1167
GA	Best Heat Map				
	Fitness Value by SED	16246.1698	11892.5418	13840.2219	11774.1203
	Fitness Value by Euclidean Distance	1901.7168	1582.7204	1726.9854	1616.5237
IGA	Best Heat Map				
	Fitness Value by SED	1345.8920	1082.9607	1169.8280	1391.1490
	Fitness Value by Euclidean Distance	574.2302	515.2758	518.0561	580.8858

Figure 2 The comparison of the quality of gene order generated by ACO and GA using squared Euclidean distance formula.

formulas generated a different quality of gene order. Compared to Pearson distance and Euclidean distance, the squared Euclidean distance generated the best quality of AD gene order. Although Pearson distance commonly used tool, it is less optimal in AD gene order computation when employed in both ACO and GA methods.

Authors' contributions

BH formulated the computing framework in this paper with CP together. GJ performed computing and drafted this manuscript with CP together. SW wrote part of background section for this manuscript. QL, ZC, CRV, JTR, and YD assisted the study and provided some suggestions. XH initiated the project, provided the guidance for the study, and performed the final editing for the manuscript. All authors have read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The work was supported by the BWH Radiology and MGH Psychiatry research funds (to X. Huang) and the Technology Innovation fund (No. 09zz028) of Key Developing Program from Education Department of Sichuan Province, China. The authors appreciate the help from the other members of gene computation group: W. Hu, C.-B. Wang, X. Li, H. Liu, L.-J. Ye, J.-L. Zhou, P. Shuai, and S.-Q. Liu. The authors appreciate the help from Prof. J. Zhang and Prof. J. Zhou. The authors would also like to thank Ms. Kimberly Larson of BWH Radiology and Mr. Conan Huang of Brown University and MGH Psychiatry for editing the manuscript.
This article has been published as part of *BMC Medical Genomics* Volume 6 Supplement 1, 2013: Proceedings of the 2011 International Conference on Bioinformatics and Computational Biology (BIOCOMP'11). The full contents of the supplement are available online at http://www.biomedcentral.com/bmcmedgenomics/supplements/6/S1. Publication of this supplement has been supported by the International Society of Intelligent Biological Medicine.

Author details

[1]College of Management Science, Chengdu University of Technology, Chengdu 610059, China. [2]Group of Gene Computation, College of Mathematics and Software Science, Sichuan Normal University, Chengdu 610066, China. [3]Department of Computer Science, Sam Houston State University, Huntsville, TX 7734, USA. [4]Department of Epidemiology and Biostatistics, School of Public Health, Indiana University Bloomington, 1025 E. 7th Street, Bloomington, IN 47405-7109, USA. [5]Harvard NeuroDiscovery Center and Department of Neurology, Massachusetts General Hospital and Harvard Medical School, Charlestown, MA 02129, USA. [6]Neurochemistry Laboratory, Department of Psychiatry, Massachusetts General Hospital and Harvard Medical School, Charlestown, MA 02129, USA. [7]Cancer Bioinformatics, Rush University Cancer Center, and Department of Internal Medicine, Rush University Medical Center, Chicago, IL 60612, USA.

Published: 23 January 2013

References

1. Thies W, Bleiler L: **2011 Alzheimer's disease facts and figures.** *Alzheimers Dement* 2011, **7**:208-244.
2. Alzheimer A: **"Über eine eigenartige Erkrankung der Hirnrinde" - Die Alzheimersche Krankheit im Brennpunkt von Klinik und Forschung.** *Allg Zeitschr Psychiatr Psychiatr-Gerichtl Med* 1907, **109**:146-148.
3. Tanzi RE, Bertram L: **Twenty years of the Alzheimer's disease amyloid hypothesis: a genetic perspective.** *Cell* 2005, **120**:545-555.
4. Glenner GG, Wong CW: **Alzheimer's disease and Down's syndrome: sharing of a unique cerebrovascular amyloid fibril protein.** *Biochem Biophys Res Commun* 1984, **122**:1131-1135.
5. Levy E, Carman MD, Fernandez-Madrid IJ, Power MD, Lieberburg I, van Duinen SG, Bots GT, Luyendijk W, Frangione B: **Mutation of the Alzheimer's disease amyloid gene in hereditary cerebral hemorrhage, Dutch type.** *Science* 1990, **248**:1124-1126.
6. Goate A, Chartier-Harlin MC, Mullan M, Brown J, Crawford F, Fidani L, Giuffra L, Haynes A, Irving N, James L, *et al*: **Segregation of a missense mutation in the amyloid precursor protein gene with familial Alzheimer's disease.** *Nature* 1991, **349**:704-706.
7. Pericak-Vance MA, Bebout JL, Gaskell PC Jr, Yamaoka LH, Hung WY, Alberts MJ, Walker AP, Bartlett RJ, Haynes CA, Welsh KA, *et al*: **Linkage studies in familial Alzheimer's disease: evidence for chromosome 19 linkage.** *Am J Hum Genet* 1991, **48**:1034-1050.
8. Schmechel DE, Saunders AM, Strittmatter WJ, Crain BJ, Hulette CM, Joo SH, Pericak-Vance MA, Goldgaber D, Roses AD: **Increased amyloid beta-peptide deposition in cerebral cortex as a consequence of apolipoprotein E genotype in late-onset Alzheimer's disease.** *Proc Natl Acad Sci USA* 1993, **90**:9649-9653.
9. Wragg M, Hutton M, Talbot C: **Genetic association between intronic polymorphism in presenilin-1 gene and late-onset Alzheimer's disease. Alzheimer's Disease Collaborative Group.** *Lancet* 1996, **347**:509-512.
10. Rogaeva EA, Fafel KC, Song YQ, Medeiros H, Sato C, Liang Y, Richard E, Rogaev EI, Frommelt P, Sadovnick AD, *et al*: **Screening for PS1 mutations in a referral-based series of AD cases: 21 novel mutations.** *Neurology* 2001, **57**:621-625.
11. Zhang C, Wu B, Beglopoulos V, Wines-Samuelson M, Zhang D, Dragatsis I, Sudhof TC, Shen J: **Presenilins are essential for regulating neurotransmitter release.** *Nature* 2009, **460**:632-636.
12. Lambert JC, Mann DM, Harris JM, Chartier-Harlin MC, Cumming A, Coates J, Lemmon H, StClair D, Iwatsubo T, Lendon C: **The -48 C/T polymorphism in the presenilin 1 promoter is associated with an increased risk of developing Alzheimer's disease and an increased Abeta load in brain.** *J Med Genet* 2001, **38**:353-355.
13. Jakob-Roetne R, Jacobsen H: **Alzheimer's disease: from pathology to therapeutic approaches.** *Angew Chem Int Ed Engl* 2009, **48**:3030-3059.
14. Caughey B, Lansbury PT: **Protofibrils, pores, fibrils, and neurodegeneration: separating the responsible protein aggregates from the innocent bystanders.** *Annu Rev Neurosci* 2003, **26**:267-298.
15. Eisen MB, Spellman PT, Brown PO, Botstein D: **Cluster analysis and display of genome-wide expression patterns.** *Proc Natl Acad Sci USA* 1998, **95**:14863-14868.
16. Sokal RR, Michener CD: **A statistical method for evaluating systematic relationships.** *University of Kansas Science Bulletin* 1958, 1409-1438.
17. Alizadeh AA, Eisen MB, Davis RE, Ma C, Lossos IS, Rosenwald A, Boldrick JC, Sabet H, Tran T, Yu X, *et al*: **Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling.** *Nature* 2000, **403**:503-511.
18. Kawasaki S, Borchert C, Deyholos M, Wang H, Brazille S, Kawai K, Galbraith D, Bohnert HJ: **Gene expression profiles during the initial phase of salt stress in rice.** *Plant Cell* 2001, **13**:889-905.
19. Khodursky AB, Peter BJ, Cozzarelli NR, Botstein D, Brown PO, Yanofsky C: **DNA microarray analysis of gene expression in response to physiological and genetic changes that affect tryptophan metabolism in *Escherichia coli*.** *Proc Natl Acad Sci USA* 2000, **97**:12170-12175.
20. Schaffer R, Landgraf J, Accerbi M, Simon V, Larson M, Wisman E: **Microarray analysis of diurnal and circadian-regulated genes in *Arabidopsis*.** *Plant Cell* 2001, **13**:113-123.
21. Bar-Joseph Z, Gifford DK, Jaakkola TS: **Fast optimal leaf ordering for hierarchical clustering.** *Bioinformatics* 2001, **17(Suppl 1)**:S22-29.
22. Herrero J, Valencia A, Dopazo J: **A hierarchical unsupervised growing neural network for clustering gene expression patterns.** *Bioinformatics* 2001, **17**:126-136.
23. Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander ES, Golub TR: **Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation.** *Proc Natl Acad Sci USA* 1999, **96**:2907-2912.
24. Toronen P, Kolehmainen M, Wong G, Castren E: **Analysis of gene expression data using self-organizing maps.** *FEBS Lett* 1999, **451**:142-146.
25. Tavazoie S, Hughes JD, Campbell MJ, Cho RJ, Church GM: **Systematic determination of genetic network architecture.** *Nat Genet* 1999, **22**:281-285.
26. Ben-Dor A, Shamir R, Yakhini Z: **Clustering gene expression patterns.** *J Comput Biol* 1999, **6**:281-297.

27. Merz P, Zell A: **Clustering gene expression profiles with memetic algorithms.** *7th International Conference on Parallel Problem Solving from Nature* 2002, 811-820.
28. Edwards AW, Cavalli-Sforza LL: **A method for cluster analysis.** *Biometrics* 1965, **21**:362-375.
29. Ward JH Jr: **Hierarchical grouping to optimize an objective function.** *Journal of the American statistical association* 1963, **58**:236-244.
30. Bar-Joseph Z, Biedl T, Brejova B, Demaine ED, Gifford DK, Hamel AM, Jaakkola TS, Srebro N, Vinar T: **Optimal arrangement of leaves in the tree representing hierarchical clustering of gene expression data.** *Technical Report CS-2001-14* Dept. of Computer Science, University of Walterloo; 2001.
31. Goldberg DE: **Genetic algorithms in search, optimization & machine learning.** Reading, MA: Addison-Wesley; 1989.
32. Holland JH: **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.** University of Michigan Press; 1975.
33. Nagata Y, Kobayashi S: **Edge assembly crossover: a high-power genetic algorithm fort the traveling salesman problem.** *7th International Conference on Genetic Algorithms* 1977, 450-457.
34. Tsai HK, Yang JM, Kao CY: **A genetic algorithm for traveling salesman problems.** *Genetic and Evolutionary Computation Conference (GECCO 2001)* 2001, 687-693.
35. Tsai HK, Yang JM, Kao CY: **Applying genetic algorithms to finding the optimal order in displaying the microarray data.** *Genetic and Evolutionary Computation Conference (GECCO 2002)* 2002, 610-617.
36. Tsai HK, Yang JM, Kao CY: **Solving traveling salesman problems by combining global and local search mechanisms.** *Congress on Evolutionary Computation (CEC 2002)* 2002, 1290-1295.
37. Lee S-K, Kim Y-H, Moon B-R: **Finding the optimal gene order in displaying microarray data.** *Lecture Notes in Computer Science* Springer Berlin/ Heidelberg; 2003, 1611-3349.
38. Dorigo M, Maniezzo V, Colorni A: **Ant system: optimization by a colony of cooperating agents.** *IEEE Trans Syst Man Cybern B Cybern* 1996, **26**:29-41.
39. Pang C, Jiang G, Wang S, Hu B, Liu Q, Deng Y, Huang X: **Gene order computation using Alzheimer's DNA microarray gene expression data and the ant colony optimization algorithm.** *International Journal of Data Mining and Bioinformatics* .
40. Pang C, Wang C, Hu B: **Experiment study of entropy convergence of ant colony optimization.** *arXiv:09051751v4 [csNE]* 2009.
41. Reeves CR, Beasley JE: **Modern heuristic techniques for combinatorial problems.** *Blackwell Scientific* 1993.
42. Djannaty F, Doostdar S: **A hybrid genetic algorithm for the multidimensional knapsack problem.** *International Journal of Contemporary Mathematical Sciences* 2008, **3**:443-456.
43. Glover F, Kochenberger G: **Handbook of metaheuristics.** *Kluwer Academic Publisher* 2003.
44. Spearman C: **The proof and measurement of association between two things.** *American Journal of Psychology* 1904, **15**:72-101.
45. Kendall M: **A new measure of rank correlation.** *Biomerika* 1938, **30**:81-93.
46. Ueyama T, Fukuda T, Arai F: **Configuration using genetic algorithm for cellular robotic system.** *IROS* Raleigh, NC; 1992, 1542-1549.
47. Kirk J: **Traveling salesman problem-genetic algorithm.**, .
48. Blalock EM, Geddes JW, Chen KC, Porter NM, Markesbery WR, Landfield PW: **Incipient Alzheimer's disease: microarray correlation analyses reveal major transcriptional and tumor suppressor responses.** *Proc Natl Acad Sci USA* 2004, **101**:2173-2178.

doi:10.1186/1755-8794-6-S1-S8
Cite this article as: Hu *et al.*: Assessment of gene order computing methods for Alzheimer's disease. *BMC Medical Genomics* 2013 6(Suppl 1): S8.