Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
Alignment using genetic programming with causal trees for identification of protein functions

Chun-Min Hung\textsuperscript{a}, Yueh-Min Huang\textsuperscript{a,},*, Ming-Shi Chang\textsuperscript{b}

\textsuperscript{a} Department of Engineering Science, National Cheng Kung University, No.1, Ta-Hsueh Road, Tainan 701, Taiwan, ROC
\textsuperscript{b} Department of Biochemistry, National Cheng Kung University, No.1, Ta-Hsueh Road, Tainan 701, Taiwan, ROC

Abstract

A hybrid evolutionary model is used to propose a hierarchical homology of protein sequences to identify protein functions systematically. The proposed model offers considerable potentials, considering the inconsistency of existing methods for predicting novel proteins. Because some novel proteins might align without meaningful conserved domains, maximizing the score of sequence alignment is not the best criterion for predicting protein functions. This work presents a decision model that can minimize the cost of making a decision for predicting protein functions using the hierarchical homologies. Particularly, the model has three characteristics: (i) it is a hybrid evolutionary model with multiple fitness functions that uses genetic programming to predict protein functions on a distantly related protein family, (ii) it incorporates modified robust point matching to accurately compare all feature points using the moment invariant and thin-plate spline theorems, and (iii) the hierarchical homologies holding up a novel protein sequence in the form of a causal tree can effectively demonstrate the relationship between proteins. This work describes the comparisons of nucleocapsid proteins from the putative polyprotein SARS virus and other coronaviruses in other hosts using the model.

© 2005 Elsevier Ltd. All rights reserved.

Keywords: Bioinformatics protein databases; Evolutionary computing and genetic algorithms; Splines; Invariants; Moments
1. Introduction

Identification of protein function is the main theme of the post-genome era. Recently, biomedical scientists have been striving to study proteomics and explore the therapeutic potential of genes for curing disease. Thus, a powerful and integrated methodology is required for predicting novel protein function. During the past decade, many algorithms applied to computational biology have been used to solve several of the above problems, with the most frequently used methods including dynamic programming [1], the hidden Markov model [2, 3], the Bayesian theorem, and probabilistic modeling [4–6]. The Journal of Molecular Biology contains an introduction to hidden Markov models for biological sequences written by Krogh [7]. As a statistical model, the hidden Markov model (HMM) is appropriate for many tasks in molecular biology. A profile-like HMM architecture can be used for searching against sequence databases for new family members and also can make multiple alignments of protein families. Moreover, the latest review by Tsakonas and Dounias [8] indicates that artificial intelligence computational methodologies have been widely applied in bioinformatics, for example genetic algorithms [9], genetic programming [10], neural networks [11,12], fuzzy logic [13,14], and some classification and clustering techniques used in data mining [15]. Design of automatic models for comparing protein sequences that are homologous to a known family or super-family has become increasingly important. One standard approach to solving this problem uses ‘profiles’ of position-specific scoring measure estimated using multiple sequence alignment (MSA) [16]. In fact, a potential member of a distantly related protein family cannot be identified with the other members in the same family using only the primary protein structures. Consequently, a new protein, such as the case of the SARS virus, with numerous unknown functions cannot be accurately predicted by modeling a learning model. This study applies a hybrid methodology based on genetic programming with a causal tree [4,28,31] model to predicting protein function. The causal tree [28,31] with a cause–effect form is proposed for locally comparing protein sequences across each pairwise alignment of MSA. The clues to this proposal are that comparisons of distantly related sequences may carry more sensitive and accurate messages regarding protein function if given a segmented alignment using the more comprehensive depiction of the protein family. Additionally, the probabilistic scoring measures, similar to profile-to-profile comparisons [17], can be used to achieve one objective of genetic programming, namely aligning the hierarchical homologies based on profile comparisons. The resulting output is capable of function-related representation for further linkages of protein function. Furthermore, the proposed model can generate a tree structure output with a special aligned format rather than a traditional local or global alignment format such as CLUSTALW [23]. Traditionally, a local alignment based on the Smith–Waterman algorithm [7] has been used to compare sequences with gaps using various scoring systems. Using scoring systems, many methods are used to evaluate the similarity of two profile columns, for example the FFAS method [18], and prof_sim [19]. The FFAS method can calculate the correlation coefficients and thus score the ‘dot-product’ scores of the amino acid frequencies in the two columns. Meanwhile, the prof_sim method utilizes a single similarity score combined with the divergence score and the significance score of the probability distributions in two columns. Recently, the method COMPASS [17,20] generated the occurrence probability of target residue in one profile column based on the other column of given the target residue frequencies to construct the local profile-to-profile alignment. However, the proposed model was designed to detect the potential functions of a novel protein with a hierarchical homology structure. The hybrid model, namely Alignment using Genetic programming with Causal Tree (AGCT), is a heuristic
evolutionary method that contains three basic components: (i) genetic programming with inner-exchanged individual strategy, (ii) causal trees \([4,28,31]\) with probabilistic reasoning, and (iii) construction of hierarchical homologies with local block-to-block alignment using the methods of moment invariant and robust points matching (RPM) \([24]\). Locally, a standard construction of an alignment requires at least a two-step design: the first step is a scoring evaluation for similarity between two given positions, while the second step is an alignment extension that considers the gap penalties and an extension algorithm. The AGCT model applies two modified steps to an alignment with a standard construction. The first step is to change the compared objects from positions to blocks that initialize extents based on a particular signal wave, while the second step is to iteratively adjust the boundaries using trimming local fragments rather than the extension algorithm. Actually, in BLAST \([21]\) and its successors \([22]\), the effective local extension is displayed for the cases of sequence-to-sequence and sequence-to-profile comparison. Similarly, in AGCT, the approach to a local block-to-block comparison modified from profile-to-profile alignment described in \([17]\) shifts the leftmost and rightmost boundaries in relation to the parent segments when high local alignment scores are achieved. Notably, a detailed and refined procedure of alignment such as the Smith–Waterman algorithm \([7]\) is also used in a part of the proposed model for confirming a meaningful fragment detected, because a fragment with a short length reduces the computation time and only some individuals in the population have to be compared making the combination of heuristic and exhaustive search models feasible.

2. System modeling

The proposed AGCT model is an evolutionary method based on a probabilistic inference model. The proposed model also incorporates a modified RPM \([24]\) method and a local sequence alignment into the probabilistic model. Meanwhile, the model uses a moment invariant theorem \([32]\) for pattern recognitions to extract the feature-nodes from some fragmented signal curves. The RPM compares feature points to precisely construct a soft match for two sets of points. Moreover, the modification of RPM changes the original transformation between matrices to the current transformation between a matrix and a tree. For achieving this matrix-to-tree transformation, the proposed constraints, namely rooted one-way winner-take-all (RO-WTA), resembling the two-way winner-take-all (WTA) constraints \([27]\), are also used. The proposed RO-WTA method is a WTA method that initially takes a unique root node by applying WTA to the slacked matrix column that is involved for the null point correspondences. Then WTA is applied to all rows to ensure each row only produces a corresponding feature-node in a causal tree, and finally at most three nodes are taken and used as the branches of the parent node using the first three high values in the same column. Moreover, the tree-to-matrix transformation simply visits a tree in preorder to derive a set of pairwise points. Finally, an equation, Eq. (28), is applied to yield a correspondence matrix for optimizing the RPM.

2.1. Transformation of problem

A traditional MSA method for protein classification compares more than two sequences to provide the aligned sequences for the observation of conserved domains. However, in a distantly related protein family, the data provides neither meaningful knowledge of biochemical function nor a consistent outcome when using different MSA algorithms. Clearly, the MSA comparisons would be worst if it is applied with the distantly related protein family members. In this model, a small protein fragment transformed into a feature-node would be a compared target point with
spatially localized features. Individual feature-nodes simplified by six moment invariants [32] are matched with each other for the signal registration of 2-D curves [25,30]. The 2-D curve is depicted using a block-to-block method that can transform a fragmenting sequence of protein residues into a signal curve by iteratively accumulating a block of amino acid properties. The signal curves are plotted by accumulating hydrophobicity against protein sequence here. Because the problem has been transformed into the problem of comparing feature-nodes (points), the comparison of spatially localized features for protein function can be considered an optimization problem of point match for the 2-D shape. In fact, the optimization problem is a difficult problem. Especially, the mapping problem must also consider the rigid and non-rigid deformations. A hybrid technique, combining genetic programming and the causal tree, is ideal for solving such difficult problems. Genetic programming has emerged from the evolutionary based systems [26, 40–42], and the causal tree is derived from probabilistic reasoning in intelligent systems [4]. Since the genetic programming can avoid local minima of an energy function, the model is appropriate for recognizing a potential pattern which is never found via direct comparisons.

2.2. Feature matching

This model uses a popular technique for synthesizing a special signal indicating specific protein functions by accumulating the properties of a group of protein residues. Comparing these special signals with each other can provide a wider window for viewing the world of the protein. Essentially, the signals must be decomposed for comparison using a method that can prevent a suitable waveform from destroying the protein signals. In the implementation, a technique of wavelet reconstruction for signals [29] is used to depict a noiseless signal curve skeleton. Next, the reconstructed curve is differentiated with respect to the positions of each protein residue, and let its results equal zero. The technique thus can easily identify some positions with local minimum accumulating properties. Subsequently, to preventing the generation of less than 40 residues, the method would discard some of the neighbor positions around the minima. Next, all of the fragmented signals are used to provide an initial dataset for the input of evolutionary computation. Although the local minimum values can not determine the true boundaries of the signals at the beginning step, the local alignment afterwards for fragmented proteins using the Smith–Waterman algorithm [7] would iteratively regularize the boundaries until the suboptimal boundaries are found. Next, the model must further extract signal features from the fragmented input signals after decomposing the original signals. Technically, the vector features can be associated with the signal features. However, the vector value representing a signal must be invariant when the physical linear-transforms deform the signal curve (shape), such as move, rescale, stretching, and rotation. Once the problem is transformed into the image and signal processing domains, the RPM method is easily applied to protein function identification. Conceptually, this proposed model includes two stages for feature matching. The first stage is designed to obtain the initial signal features for the signal registration [25,30]. During the second stage, an inner-exchanged genetic programming is used to repeatedly refine the RPM parameters.

3. Theorem

This work utilizes the theorem of two-dimensional moment invariants [32] for planar geometric figures from the theorem of moment spaces [33] to extract features of the protein fragments. Accumulating hydrophobicity of surrounding residues for each position of protein
enables the protein fragments to generate signal curves. Next, the signal curve further reduces to a fixed length vector, denoting a feature node, using moment invariants for matching the nodes. Next, one of the energy functions in the model introduces the energy function of a correspondence algorithm of non-rigid mapping [24] derived from the TPS theorem [34]. Finally, genetic programming with the inner-exchanged subpopulation strategy is used to refine the comparison results.

3.1. Moment invariant

The theorem of moment invariants, which is extensively used in computer science, was first proposed in [32] for recognizing visual patterns. In the visual field, an essential step must extract patterns for recognizing visual objects from the original objects. The extracted patterns must also be independent of position, size and orientation. Likewise, the model uses the moment invariants of the signal curves to substantially reduce the computational complexity. Because the flexibility of the moment invariants used for recognizing protein function was such that the model could find more domains of conservation, the domains should be able to attain more biological meaning by this moment invariant based comparison. The following subsections detail how the model obtains a feature vector denoting a feature-node in a causal tree by using the moment invariant theorem for signal recognition.

A. Uniqueness theorem regarding moments

Riemann integrals [32] define two-dimensional \((k + l)\)th order moments for a density distribution \(f(x, y)\) as follows:

\[
M_{k,l} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^k y^l f(x, y) \, dx \, dy \quad \text{where } k, l = 0, 1, 2 \ldots
\]

where \(f(x, y)\) denotes a piecewise continuous bounded function, and has nonzero values if and only if it is on the finite part of the \(xy\) plane. Vice versa, the \(f(x, y)\) may derive the unique double moment sequence \(\{\mu_{k,l}\}\) in the moment of any order.

B. Central moments

By directly integrating the double moment sequence, Eq. (2) can express a central moment \(\mu_{k,l}\) derived from the ordinary moments.

\[
\mu_{k,l} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^k (y - \bar{y})^l f(x, y) \, dx \, dy
\]

where \(\bar{x} = \bar{m}_{1,0}/\bar{m}_{0,0}\) and \(\bar{y} = \bar{m}_{0,1}/\bar{m}_{0,0}\) is an expected value of \(x\) and \(y\), respectively. The point \((\bar{x}, \bar{y})\) is termed a center of gravity or center of centroid for \(x\) and \(y\). For example, the first four orders have

\[
\begin{align*}
\mu_{0,0} &= \bar{m}_{0,0}, \quad \mu_{1,0} = \bar{m}_{1,0} - \bar{x}, \quad \mu_{0,1} = \bar{m}_{0,1}, \quad \mu_{2,0} = \bar{m}_{2,0} - \bar{x}^2, \\
\mu_{1,1} &= \bar{m}_{1,1} - \bar{m}_{1,0} \bar{x}, \quad \mu_{2,0} = \bar{m}_{2,0} - \bar{x}^2, \quad \mu_{3,0} = \bar{m}_{3,0} - 3 \bar{m}_{2,0} \bar{x} + 2 \bar{x}^3, \\
\mu_{2,1} &= \bar{m}_{2,1} - \bar{m}_{2,0} \bar{y} - 2 \bar{m}_{1,1} \bar{x} + 2 \bar{x}^2 \bar{y}, \\
\mu_{1,2} &= \bar{m}_{1,2} - \bar{m}_{1,0} \bar{y} - 2 \bar{m}_{1,1} \bar{y} + 2 \bar{x} \bar{y}^2, \quad \mu_{0,3} = \bar{m}_{0,3} - 3 \bar{m}_{0,2} \bar{y}^2 + 2 \bar{y}^3.
\end{align*}
\]
In a similar mathematical sense, a measure of area for a two-value image \( b(m, n) \) with \((k+1)\)th order moment can approximately obtain Eq. (4) from Eq. (1).

\[
M_{k,l} = \sum_{m=0}^{\hat{M}-1} \sum_{n=0}^{\hat{N}-1} m^k n^l b(m, n).
\] (4)

Moreover, the corresponding central moment is expressed as Eq. (5):

\[
\mu_{k,l} = \sum_{m=0}^{\hat{M}-1} \sum_{n=0}^{\hat{N}-1} (m - \bar{x})^k (n - \bar{y})^l b(m, n)
\] (5)

where \( \bar{x} = \frac{M_{1,0}}{M_{0,0}}, \bar{y} = \frac{M_{0,1}}{M_{0,0}}, \hat{M} \) and \( \hat{N} \) are the values for the area length and width, respectively.

C. Pattern identification

A pattern of the two-value image which must be independent of position, size, and rotation is derived from similitude moment invariants. Likewise, the function \( b(m, n) \) with respect to a pair of fixed axes can display the signal curve of protein sequences. Moreover, the two-dimensional central moments \( \mu_{k,l} \) expresses image patterns. Additionally, numerous properties of the second central moment are equivalent to a covariance matrix in probability fields. Therefore, the covariance matrix for the second central moment is expressed as follows:

\[
U = \begin{bmatrix}
\mu_{2,0} & \mu_{1,1} \\
\mu_{1,1} & \mu_{0,2}
\end{bmatrix}
\] (6)

By the diagonalization, Eq. (7) is obtained as follows:

\[
E^T U E = E U E^T = \Lambda
\] (7)

where \( E = \begin{bmatrix} e_{1,1} & e_{1,2} \\ e_{2,1} & e_{2,2} \end{bmatrix} \), a column vector of matrix \( E \) is an eigenvector of matrix \( U \), and the eigenvalues of \( U \) determine \( \Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \). The values of \( \lambda_{\text{max}} \) and \( \lambda_{\text{min}} \) corresponding to \((\lambda_1, \lambda_2)\) and \((\lambda_2, \lambda_1)\) are shown in Eqs. (8) and (9), respectively.

\[
\lambda_{\text{max}} = \frac{1}{2} (\mu_{2,0} + \mu_{0,2}) + \frac{1}{2} \sqrt{\mu_{2,0}^2 + \mu_{0,2}^2 - \mu_{0,2}\mu_{2,0} + 4\mu_{1,1}^2}
\] (8)

\[
\lambda_{\text{min}} = \frac{1}{2} (\mu_{2,0} + \mu_{0,2}) - \frac{1}{2} \sqrt{\mu_{2,0}^2 + \mu_{0,2}^2 - \mu_{0,2}\mu_{2,0} + 4\mu_{1,1}^2}
\] (9)

Moreover, Eq. (10) denotes a direction angle of the area of the image. From the above theoretical deviations, Eqs. (1)–(10) can easily formularize some basic properties of the pattern.

\[
\theta = \tan^{-1} \left\{ \frac{\lambda_{\text{max}} - \mu_{2,0}}{\mu_{1,1}} \right\}
\] (10)

Notably, an added restriction such as \( \mu_{2,0} > \mu_{0,2} \) can determine the unique angle of \( \theta \). Clearly, a discrimination property of the patterns increases when the higher order moments are used as the pattern. The higher order moments with respect to the principal axes can also be easily determined using the method of the principal axes described in [32]. Because \( \mu_{0,0} = M_{0,0} \) can be used to measure the area, a factor \( \sqrt{\alpha} \) of proportionality, a divisor, should be used to further normalize the central moment in Eq. (5) by dividing the variables of the function of Eq. (2) by the factor. Eq. (2) then can adopt the form \( f \left( \frac{x}{\sqrt{\alpha}}, \frac{y}{\sqrt{\alpha}} \right) \). As a result the normalized central
moment \( v_{k,l} \) equals the original central moment of \( f(x, y) \) divided by \((\sqrt{\alpha})^{k+l+2} \). Because of \( \mu_{0,0} \) being able to measure area, \( \alpha = \mu_{0,0} \). Eq. (11) then can be used to define the normalized central moment, as follows:

\[
v_{k,l} = \frac{\mu_{k,l}}{(\sqrt{\mu_{0,0}})^{k+l+2}}. \tag{11}
\]

An identification of the pattern independent of position, size and orientation then can be suitably employed to formulize the moment invariants for any signal curve using Eq. (11).

**D. Absolute moment invariants**

To improve pattern recognition capability, an absolute moment invariant should further be achieved by using multiple moment invariants. The normalized central \( v_{k,l} \) in the second and third orders can generate the following six absolute and orthogonal moment invariants listed in Eq. (12).

\[
\begin{align*}
\omega_1 &= v_{2,0} + v_{0,2}, \quad \omega_2 = (v_{2,0} - v_{0,2})^2 + 4v_{1,1}^2, \\
\omega_3 &= (v_{3,0} - 3v_{1,2})^2 + (3v_{2,1} - v_{0,3})^2, \quad \omega_4 = (v_{3,0} + v_{1,2})^2 + (v_{2,1} + v_{0,3})^2 \\
\omega_5 &= (v_{3,0} - 3v_{1,2}) (v_{3,0} + v_{1,2}) \left[ (v_{3,0} + v_{1,2})^2 - 3(v_{2,1} + v_{0,3})^2 \right] \\
&\quad + (3v_{2,1} - v_{0,3}) (v_{2,1} + v_{0,3}) \left[ 3(v_{3,0} + v_{1,2})^2 - (v_{2,1} + v_{0,3})^2 \right] \\
\omega_6 &= (v_{2,0} - v_{0,2}) \left[ (v_{3,0} + v_{1,2})^2 - (v_{2,1} + v_{0,3})^2 \right] \\
&\quad + 4v_{1,1} (v_{3,0} + v_{1,2}) (v_{2,1} + v_{0,3}). \tag{12}
\end{align*}
\]

While the skewed invariant listed in the original study [32] is useful for distinguishing mirror images, the model discards this invariant since one skewed orthogonal invariant is not necessary for signal recognition here. The six moment invariants in Eq. (12) are not only independent of position, size and orientation, but also can be used to present a feature-node for a fragmented protein sequence. Finally, a six-valued vector \( \vec{\omega} = [\omega_1 \ \omega_2 \ \omega_3 \ \omega_4 \ \omega_5 \ \omega_6]^T \) can denote a feature-node to enable further convenient processing.

**3.2. Thin-plate splines**

Suppose that a feature-node \( \vec{\omega}_a = (\omega_{a1}, \omega_{a2}, \omega_{a3}, \omega_{a4}, \omega_{a5}, \omega_{a6}) \) can represent a signal curve from the fragmented protein corresponding to one point in the 2-D space. Assume a set of vectors \( \vec{\tilde{\Psi}} = \{\vec{\tilde{\omega}}_a \mid a = 1, 2, \ldots, K\} \) and the other set of \( \vec{\tilde{\Psi}} = \{\vec{\tilde{\Psi}}_{a'} \mid a' = 1, 2, \ldots, N\} \). Since vector \( \vec{\omega}_a \) with six variables would increase the problem complexity, a simple linear-transform is used to obtain a scalar for solving the point-matching problem using the TPS of two variables. One problem of the fragmented sequence-mapping then can simply be transformed into the other new problem. The new problem involving the two feature-node sets of \( \vec{\tilde{\Psi}} \) and \( \vec{\tilde{\Psi}} \) fits a mapping function \( f(\vec{\omega}_a) \) using TPS, and simultaneously minimizes the TPS energy function defined as follows:

\[
E_{\text{TPS}}(f) = \sum_{a=1}^{K} \|x_a - f(\vec{\omega}_a)\|^2 + \lambda \times \iint \left[ \left( \frac{\partial^2 f}{\partial^2 \tilde{x}} \right)^2 + 2 \left( \frac{\partial^2 f}{\partial \tilde{x} \partial \tilde{y}} \right)^2 + \left( \frac{\partial^2 f}{\partial^2 \tilde{y}} \right)^2 \right] \, d\tilde{x} d\tilde{y}. \tag{13}
\]
Typically, two types of deformations of a compared object exist, rigid (affine) and non-rigid (non-affine). The rigid deformation includes parallel translation and rotation. Meanwhile, the non-rigid deformation includes shrink, dilation, and distortion. By minimizing the first error measurement term of (13), the feature-node set $\tilde{X}$ maps as closely as possible to the other feature-node set $\tilde{\Psi}$. Generally, an infinite number of mappings $f$ can minimize the first term since the mapping is non-rigid. The energy function of Eq. (13) can uniquely determine a minimizing function $f$ specified by two parameter matrices $\tilde{A}$ and $\tilde{W}$ in Eq. (14) if and only if it has a fixed value $\lambda$.

$$f(\tilde{\omega}) = \tilde{\omega} \times \tilde{A} + \phi(\tilde{\omega}) \times \tilde{W}$$

where $\tilde{A}$ represents a $(D + 1) \times (D + 1)$ affine transformation matrix, $D = 2$ in $\mathfrak{R}^2$, and $\tilde{W}$ is a $K \times (D + 1)$ non-affine warping coefficient matrix. The vector $\phi(\tilde{\omega})$, which defines a $1 \times K$ vector of each entry $\phi_a(\tilde{\omega}) = c\|\tilde{\omega} - \omega_a\|^2 \log \|\tilde{\omega} - \omega_a\|$ with $c$ being a constant for any $\omega$ [34], is related to the TPS kernel. The kernel comprises the information regarding the internal structural relationships of the feature-node set and a non-rigid warp comes into play when the warping coefficient matrix $\tilde{W}$ is considered in Eq. (14). The second term of Eq. (13), which is responsible for regularizing the mapping function between $\tilde{X}$ and $\tilde{\Psi}$, is essentially a smoothness constraint. The Lagrange parameter $\lambda$ in Eq. (13) regularizes the warping of the matches of feature-nodes. The precise matches appear if $\lambda$ approaches zero. When a solution for $f$ in Eq. (14) is substituted into the TPS energy function in Eq. (13), and using Mercer–Hilbert–Schmidt theorems of Riesz and Sz-Nagy [35] to prove Eq. (15), Eq. (16) is obtained:

$$\int \int \left[ \left( \frac{\partial^2 f}{\partial^2 \tilde{x}^2} \right)^2 + 2 \left( \frac{\partial^2 f}{\partial \tilde{x} \partial \tilde{y}} \right)^2 + \left( \frac{\partial^2 f}{\partial^2 \tilde{y}^2} \right)^2 \right] d\tilde{x} d\tilde{y} = \sum_{a=1}^{K} \hat{\lambda}_a^2 = \text{trace}(\tilde{W}^t \phi \tilde{W})$$

where $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \cdots \hat{\lambda}_K \geq 0$ denote eigenvalues of the continuous eigenfunction $\phi_1, \phi_2, \ldots, \phi_K$, and $\tilde{W}^t$ is the transposed matrix of $\tilde{W}$.

$$E_{\text{TPS}}(\tilde{A}, \tilde{W}) = \|\tilde{X} - \tilde{\Psi} \tilde{A}_{D+1,D+1} - \phi_{K,K} \tilde{W}_{K,D+1}\|^2$$

$$+ \lambda \text{trace}(\tilde{W}_{K,D+1}^t \phi_{K,K} \tilde{W}_{K,D+1})$$

where $\tilde{X}$ and $\tilde{\Psi}$ are merely concatenated versions of the vector values of the nodes $\tilde{x}_a \in \tilde{X}$ and $\tilde{\omega}_a \in \tilde{\Psi}$. Additionally, $\phi$ is a $(K \times K)$ matrix generated from the $\phi(\tilde{\omega}_a)$ of the $(1 \times K)$ vectors. Each row of the matrix derives from the original vectors. In the model, an energy function with the affine $\tilde{A}$ and warping $\tilde{W}$ parameters, which is derived from Eqs. (13)–(16), is used as a fitness function with genetic programming. Furthermore, the local border between two regions of protein fragments should be considered an important feature for understanding protein functions. Therefore, a distinctive characteristic of the TPS can always decompose a geometrical transformation into a global affine transformation and a local non-affine warping component. Using this characteristic, a functional signal generally can be detected based on the same essential protein family even if it is undergoing various environment parameters. The parameters include the setting of primary forming signal of the sequences, for example the fragment length in relation to signal response. Alternatively, the rotation, translation, and global shear of the signals should slightly influence the detection of the functional signals based on the feature geometrical
characteristics (pattern). Consequently, the smoothness term $\tilde{W}$ in Eq. (16) is specifically applied to be the warping components, and this term should be penalized when signal deformation is ineffective to their essential properties. Visually, two compared objects with similar patterns should increase the influence of TPS to interpolate a fitting spline. Correspondingly, the similar protein fragments compared with one another can be identified by TPS matching.

3.3. Optimization of mapping, correspondence, and heaping

The proposed model uses TPS methods to compare with two feature-node sets for multiple optimization problems. The TPS method raises three issues of optimizations. Traditionally, the main issues include optimizations of mapping and correspondence. Generally, the work of correspondence must permute a one-to-one relationship between two feature-node sets, and the work of mapping must fit the two feature-node sets to an optimal function $f$. However, this model defines a new issue that the problem solved must simultaneously optimize the hierarchical heaping structure by constructing a hierarchical relationship among feature-nodes. Thus, the work of correspondence in the model must achieve the hierarchical relationship of homologies rather than the one-to-one relationship. Alternatively, the model must also optimize heaping. Heaping optimization should meet the following minimum requirements: (i) only generating the node of a unique root, (ii) a parent node can compare with multiple children nodes, (iii) using the limitation of the number of branches, the model must only choose the first certain children nodes depending on homology likelihood to attach to a parent node, and (iv) the model must place the predicted feature-nodes and the predicting feature-nodes to the external and internal nods of a causal tree, respectively. These requirements describe the one-to-many query strategy in the homology search model. More precisely, heaping also accomplishes a descending order of homology probability for children nodes. Nevertheless, the computations have increased difficulty in simultaneously optimizing the hierarchical correspondence, non-rigid mapping, and heaping.

3.3.1. One-to-many query strategy

The subsection formally defines a correspondence matrix for a causal tree via the strategy of a one-to-many query. Assume that a sequence $\{Q_1\}$ with unknown functions is used as a query sequence to compare with the fragmented member sequences $\{Q_x\}_{x=2}^Z$ for constructing an optimal hierarchical structure. The optimal structure should comprise the maximum of homology relationships among these protein fragments of both ‘query’ and ‘member’. Moreover, $Z$ denotes the number of the input sequence classified to the same distantly related protein family. Furthermore, let $\tilde{M}$ denote a correspondence matrix representing the correspondence between two node-sets in a tree. Assume the matrix contains $(N+1) \times (K+1)$ elements:

$$\tilde{M} = \left\{ [m_{ia}]_{N+1,K+1} \mid m_{ia} \in [0, 1], \sum_{i=S_1+1}^{N} m_{i(N+1)} = 1 \land \forall i \leq S_1 \implies m_{i(N+1)}, \sum_{a=S_1+1}^{N+1} m_{aa} = 0, \sum_{a=S_1+1, i \neq a}^{N+1} m_{ia} = 1, \forall i = 1, 2, \ldots, N \right\}$$

(17)

where each $m_{ia}$ is an element of the matrix, with a value that is a real number between zero and one. One query sequence $Q_1$ is divided into $S_1$ fragments. The summation of $m_{ia}$ in Eq. (17) for the constraints of each row in $\tilde{M}$ must equal one. The node of the same label thus appears in the
a maximum of once. The WTA method and Sinkhorn balancing theorem \cite{36} can be used to normalize $m_{ia}$ to satisfy the row and column constraints in Refs. \cite{24,27,36}. However, this model only uses the row constraint to modify WTA as RO-WTA. Alternatively, the correspondence matrix for a causal tree must satisfy the constraint $\sum_{a=S_1+1}^{N+1} m_{ia} = 1$ in Eq. (17) to implement a one-to-many relationship. Simultaneously, a constraint $\sum_{a=S_1+1}^{N+1} m_{aa} = 0$ is used to prevent a self-correspondence. The self-correspondence that appears when one feature-node points to itself is invalid. Furthermore, a constraint $\sum_{i=S_1+1}^{N} m_{i(N+1)} = 1$ and $\forall i \leq S_1 \Rightarrow m_{i(N+1)}$ can decide the node of the unique root. From the definition of the correspondence matrix in Eq. (17), let:

$$K = N - S_1$$

(18)

where $K$ represents the number of parent nodes with the starting number one and ending at number $K$. However, the alternative representation is beginning at number $S_1 + 1$ and ending at number $N$. In fact, the two numbering representations are identical for the convenience of representation and implementation. From Eqs. (18), (19) and (20) define the set of parent and children nodes in a tree, respectively.

$$\tilde{X} = \left\{ x_j \mid x_j = x(l_x, l_y), \forall j \in \bar{P}, \bar{P} = \{ S_1 + 1, 2, \ldots, N \},
\begin{align*}
    j &= \sum_{p=0}^{l_x-1} S_p + l_y, l_x \in \{ 1, 2, \ldots, Z \}, l_y \in \{ 1, 2, \ldots, S_y \}, S_0 = 0
\end{align*}
\right\}$$

(19)

$$\tilde{\Psi} = \left\{ \omega_j \mid \omega_j = \omega(l_x, l_y), \forall j \in \bar{C}, \bar{C} = \{ 1, 2, \ldots, N \},
\begin{align*}
    j &= \sum_{p=0}^{l_x-1} S_p + l_y, l_x \in \{ 2, \ldots, Z \}, l_y \in \{ 1, 2, \ldots, S_y \}, S_0 = 0
\end{align*}
\right\}$$

(20)

where $j$ and $(l_x, l_y)$ denote different coordinates of the one and two dimensions, respectively. Moreover, the variable $S_p$ denotes the number of fragments in each protein sequence.

Next, Eqs. (21)–(25) further define a causal tree denoted $CT$ using the correspondence matrix $\tilde{M}$. A set of causal trees $\mathcal{Z}$, similarly described in \cite{35}, can be expressed in terms of $\mathcal{Z}_i$ as follows:

$$\mathcal{Z} = \{ \mathcal{Z}_i \mid \mathcal{Z}_i = R_i \cup \bar{I}_i \cup \bar{E}_i, i = 1, 2, \ldots, \infty \}$$

(21)

where $\mathcal{Z}$ denotes the set of causal trees $\mathcal{Z}_i$. A causal tree includes the components $R_i$, $\bar{I}_i$ and $\bar{E}_i$, which denote a root node, internal node-set, and external node-set, respectively. The causal tree is a tree network that is constructed from the binary random variables labeled, and is also illustrated in Fig. 1. Fig. 2 shows an example of internal and external node assignment. Of nine fragments, five corresponding fragments of internal nodes 5, 6, 7, 8, represent a set of fragmenting members from each member sequence $\{ Q_x \}_{x=2}^{\bar{C}}$. The external nodes numbered 1, 2, 3, and 4 in Fig. 1 are for a query sequence $\{ Q_1 \}$ in accordance with the query fragments of Fig. 2 in the numbering representation. In fact, the internal nodes can be used for inferring the probabilistic reasoning \cite{4} for predicting protein function. The reasoning process infers along a cost path in the causal tree by assessing orderly conditional probabilities combined together from a root-node to an external feature-node. Each internal feature-node $\bar{\omega} \in \bar{I}$ indicates a given protein fragment of different properties. Likewise, each external feature-node $\bar{\omega} \in \bar{E}$ indicates an unknown protein fragment function.
For heaping the causal tree, the constraints $\bar{H}_0$, $\bar{H}_1$ and $\bar{H}_2$ are used to allocate a correct location of feature-nodes in the causal tree. The $\bar{H}_0$ in Eq. (22) defines a root-node $R$ by applying the WTA method to column $(N + 1)$ of correspondence matrix $\tilde{M}$. Alternatively, a feature-node is a root-node if and only if no parent node points to it.

$$\bar{H}_0 = \{ \text{root} \} = \{ m_{\text{root}(N+1)} \mid \sum_{i \in S_1+1} m_{i(N+1)} \neq 1 \land \forall i \in S_1 \exists m_{i(N+1)} \}.$$ (22)

Additionally, Eqs. (23) and (24) define the correct locations of internal and external nodes, respectively. The node-pairs $(a, \bar{i})$ and $(a, a')$ represent a direction from a parent node to a child node. For clarity, Fig. 3 illustrates an example of the correspondence matrix for a causal tree. The proposed RO-WTA selects the maximum value from the real numbers within the parenthesis only for each row and the $N + 1$ column. Besides RO-WTA, the diagonal elements must all be zero to eliminate self-correspondence.

$$\bar{H}_1 = \{ (a, \bar{i}) \mid m_{\bar{i}} = m_{a(N+1)} \land \forall a \in \bar{P} \subseteq \bar{I}, \forall i \in (\bar{C} - \bar{P}) \subseteq \bar{E} \}.$$ (23)

$$\bar{H}_2 = \{ (a, a') \mid m_{a'(N+1)} = m_{a(N+1)} \land a \land a' \in \bar{P} \subseteq \bar{I} \}.$$ (24)

Eventually, Eq. (25) defines a causal tree with the entities of a correspondence matrix $\tilde{CT}$ as follows:
Because the causal tree $CT$ may heap some feature-nodes to a cause–effect form, a set of given causes can also be inferred to obtain an unknown effect. The model has transformed the TPS method of two point-sets correspondence into an optimization of the causal tree. Next, the model must control the tree growth by adding an equation in Eq. (26) into the resulting energy function. The value $N$ in Eq. (26) denotes the total number of all input fragments as the tree size. By adding Eq. (26) to the energy function, the model can attain an approximated tree in accordance with the correspondence matrix.

$$
\sum_{i=1}^{N} n_i = N, n_i = \begin{cases} 
0 & \text{if } i \notin \mathcal{I}_j \\
1 & \text{if } i \in \mathcal{I}_j \end{cases}, \quad \forall i \in \mathcal{C}
$$

where $i$ denotes the labeled identification of a feature-node. In the presence of this feature-node added to a causal tree $\mathcal{I}_j$, the value $n_i$ returns 1, and otherwise it returns 0. Finally, Eq. (27) combines Eqs. (16), (17) and (26) to create the final energy functions, as follows:

$$
E(CT, \tilde{A}, \tilde{W}) = \sum_{i=1}^{N} \sum_{a=\delta_1+1,a\neq i}^{N} m_{ia} \| \tilde{x}_i - \tilde{\omega}_a \tilde{A} - \phi_i(\omega_a) \tilde{W} \|^2 \\
+ \zeta \left( (N - 1) - \sum_{i=1}^{N} \sum_{a=\delta_1+1,a\neq i}^{N} m_{ia} \right) \\
+ T \sum_{i=1}^{N} \sum_{a=\delta_1+1,a\neq i}^{N} m_{ia} \log m_{ia} + \lambda_1 \text{trace} \left[ (\tilde{A} - I)^t (\tilde{A} - I) \right] \\
+ \lambda_2 \text{trace} \left[ \tilde{W}^t \phi \tilde{W} \right] + \lambda_3 \left| N - \sum_{i=1}^{N} n_i \right|.
$$

By minimizing Eq. (27), the model can use genetic programming to obtain a globally suboptimal solution with respect to mapping, correspondence, and heaping. The first term in Eq. (27) is an error measure term for matching the similarity between two feature-nodes. Meanwhile, the second term in Eq. (27) can guard against all matches, and a controlling
parameter $\zeta$ determines a degree of influence for the term. Next, the third term of Eq. (27), which refers to a barrier function, is an entropy term with an annealing temperature $T$ in statistical physics. By controlling the entropy barrier function, the distribution of random variables $m_{ia}$ in $CT$ gradually stabilizes. Alternatively, the model can generate the minimum number of states required to push the objective minimum away from the discrete points such that some rules emerge from $m_{ia}$. A gradually reduced temperature $T$ can suitably control thermal fluctuation to yield an optimal solution. Regarding related applications, the application of this entropy term to statistical physics is also briefly described in [37–39]. Subsequently, the fourth term in Eq. (27) is used to penalize the differences of an affine mapping parameter $\tilde{A}$ and an identity matrix $I$. The differences stimulate some unphysical reflections that result from flipping through the whole plane. The fifth term in Eq. (27), which is a standard TPS regularization term, penalizes the local warping coefficients $\tilde{W}$ [24]. Using an approximated least-squares approach and QR-decomposition technique, the model can solve the parameters $(\tilde{A}, \tilde{W})$ given fixed $CT$. Additionally, two parameters $\lambda_1$ and $\lambda_2$ are used to control the influence of the fourth and fifth terms, respectively. Finally, keeping a matrix consistent with a tree during the evolutionary process of genetic programming is very difficult. Thus, the final term in Eq. (27) is used to control the reasonable size of a causal tree. Optimizing Eq. (27), then differentiating both sides of Eq. (27) yields Eq. (28), as follows:

$$
m_{ia} = \exp \left( -\|\tilde{x}_i - \tilde{\omega}_a \tilde{A} - \phi(\tilde{\omega}_a) \tilde{W}\|^2 - \zeta \right) / T \tag{28}
$$

where $m_{ia}$ must conform to RO-WTA. Currently, a parent node can only connect a maximum of three children nodes for this model. Consequently, the summation of all elements in each column must equal three. When the model attains an optimal causal tree, a probability theorem is further used to model a training model for identifying protein function.

3.3.2. Identification of protein function

Consideration of a given protein family can be requested and answers obtained to a query with an unknown protein sequence of what the interrelationship among the particular protein fragments is. Unfortunately, the calculation of a probability distribution in Bayes networks is computationally intractable (NP-complete). However, a causal tree of fixed structure may efficiently compute the result of the probabilistic distribution [31]. Using the basic probability theorem, then assuming that $P(x_i, x_a)$ denotes the joint probability distribution of two random variables $x_i$ and $x_a$, and that $P(x_i | x_a)$ denotes the conditional probability of $x_i$ given $x_a$; this conditional probability is defined as follows:

$$
P(x_i | x_a) = \frac{P(x_i, x_a)}{P(x_a)}. \tag{29}
$$

Using the Bayes inference described in [4], Eqs. (30) and (31) are easily derived by Eq. (29).

$$
P(x_i | x_a) = \frac{P(x_a | x_i)P(x_i)}{P(x_a)} \tag{30}
$$

$$
P(x_i | x_a', x_a) = P(x_a | x_i) \tag{31}
$$

where $x_i$ and $x_a'$ are conditionally independent given $x_a$. Subsequently, a well defined Bayes model can be obtained by further generalizing beyond simple Markov models and considering
the probability distribution in the form of general trees. By the expansions of Eqs. (30) and (31), a probability distribution can be applied to learning and inferring for biological models when the tree network has a specific predetermined structure. During a learning phase, the variables within internal nodes and external nodes represent a set of member states and query states, respectively. These variables are both observable during the learning phase. Instead, these variables are hidden during an inferring phase when the structure and probability distributions of a causal tree are temporally fixed. This process fully corresponds to the operation of hidden states in a hidden Markov model. Iteratively, the model eventually produces a refined prediction result based on fitness function estimation. Recalling the example of a causal tree, Fig. 1 defines nine random binary variables on the tree. The case produces a joint probability distribution of the nine variables using a product of the following terms:

\[
P(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9) = P(x_7) P(x_6 \mid x_7) P(x_9 \mid x_7) P(x_5 \mid x_7) P(x_1 \mid x_6) P(x_4 \mid x_6) P(x_3 \mid x_9) P(x_8 \mid x_5) P(x_2 \mid x_8)
\]

where each internal node \(x_i\) separates variables above and below it. For example, using Eq. (31) can show that the node \(x_7\) gains the conditional probability \(P(x_6 \mid x_7, x_6, x_1, x_4) = P(x_6 \mid x_7)\). Initially, the root variable \(x_7\) in Eq. (32) first generates a value using the result from Eq. (28). Given \(x_7\), the probabilistic model can determine the conditional probabilities \(P(x_6 \mid x_7)\), \(P(x_9 \mid x_7)\), and \(P(x_5 \mid x_7)\). This model then can generate the values of \(x_6, x_9,\) and \(x_5\), respectively. This model thus can calculate the remaining variables recursively. Finally, this model obtains an estimated value as one of the multiple fitness functions described in the following section. By rearranging and observing the terms in Eq. (32), an ordered product of terms is also very easily implemented by a preorder tree search. Besides the fitness function of the probabilistic and RPM models mentioned above, the model needs another fitness function to determine the boundaries of protein fragments. Eq. (33) shows the normalized measures of a local alignment score \(\sigma\) divided by the sequence length and obtained by the Smith–Waterman algorithm [7].

\[
\sigma = \sum_{i=1}^{N} \sum_{a=1}^{K} m_{ia} \delta_{ia} / \rho_{ia}
\]

where \(\delta_{ia}\) denotes local alignment score and \(\rho_{ia}\) represents sequence length.

Furthermore, Eqs. (34) and (35) are used to adjust a value of parameters \(\zeta\) and \(\lambda_2\) for adjusting the reasonable parameters in Eq. (27). Since temperature \(T\) in Eq. (27) is gradually decreased during the evolution, these \(m_{ia}\) values are rapidly and accordingly decreased. From Eq. (28), it is best for \(m_{ia}\) to be in the interval of \([0.0 \ 1.0]\) to protect it from an infinite drop in value that cannot be computed due to the reasonable computation capacity limitation. Thus, Eq. (34) can be easily obtained from the inequality equation and normal distribution.

\[
\zeta = -T e^1 + \|\overline{\mu} + 2\sigma\|^2
\]

where \(e^1\) is the natural logarithm that provides a rough scope for estimating the resolution of 256 states that sufficiently fit this computing issue. Moreover, \(\overline{\mu}, \sigma\) represent the mean and standard deviation of all feature data, respectively. Furthermore, Hall and Titterington [47] proposed a
solution for estimating the smoothing parameter $\lambda_2$, by Eq. (35).

\[
\frac{\| (I - F(\lambda_2)) \tilde{X} \|^2}{\text{trace}(I - F(\lambda_2))} = \sigma^2
\]  

(35)

where $F(\lambda_2)$ denotes an influence matrix, $I$ represents an identity matrix, and $\sigma$ is a standard deviation of all feature data. A method of QR-decomposition can be used to determine the value of $(I - F(\lambda_2)) = N\lambda_2 Q_2 (Q_2^T (\Phi + N\lambda_2 I) Q_2)^{-1} Q_2^T$. The method decomposes the data of feature-nodes $\tilde{\Psi} = (Q_1 : Q_2)(\tilde{\Phi})$ into the product of orthonormal matrix and upper triangular matrix. Finally, a process of iteratively estimating Eq. (35) is used to find the optimal value $\lambda_2$ with respect to minimizing the error in Eq. (35).

4. Implementation

This section focuses on an implementation design of genetic programming that evolves three related subpopulations with inner-exchanged individuals.

4.1. Genetic programming

In 1992, Koza et al. were early pioneers of the concept of genetic programming [26,40–42] which was extended from the genetic algorithm. Genetic programming can automatically create programs for solving general problems. Genetic programming is based on a grammar-based methodology of an evolutionary theorem using the Darwinian survival principle. Therefore, genetic programming can describe highly complicated models for any domain problem without sufficient domain knowledge. First, problem solving requires randomly yielding an initial population of individuals. Subsequently, a fitness function can be used to evaluate individuals in the population and select excellent individuals for further performing specific genetic operations. The genetic operations typically include crossover, replication, and mutation operations. The operations enable the parent population at a given generation to form a new offspring population at the next generation. Iteratively, genetic programming continuously performs the above evolutionary steps until obtaining an optimal individual or satisfying certain conditions. A representation of genetic programming for a domain problem is composed of several functional programs. The programs form a rooted tree structure as a solution for solving the problem. Because genetic programming uses a tree form, the model can easily provide a flexible hierarchical presentation to estimate the relationship of homologies for predicting protein function. Characteristically, genetic programming provides a mechanism for searching for the fittest individual. The individual is the problem solution provided by using the computer program in the search space of all possible computer programs. The search space comprises terminal-nodes (external nodes) and function-nodes (internal nodes) that are appropriate to represent the specific problem [26]. When a causal tree following genetic operations generates an infeasible individual, the model must regularize this infeasible individual by using a tournament pick (TP) method. The TP method using the depth-first search (DFS) strategy is proposed to accelerate the performance of regularizing this infeasible individual to a feasible one. The TP+DFS method recursively cures the infeasible nodes residing in this individual. The method replaces an infeasible node with a feasible node, in a preorder visiting for trees, randomly selected from the tournament that contains numerous feasible nodes. This advantage of the TP+DFS method results from the mutation appearing at a suitable time, and estimates the fitness function simultaneously following the crossover. When all individuals conform to all constraints of a
causal tree, a fitness function can be used for individual estimation. To improve prediction speed and accuracy, this model uses a multiple fitness function strategy. This strategy implements an exchanged individual mechanism involving three subpopulations. The model next considers the following parameters for developing a co-evolution with the strategy of inner-exchanged individuals in a population. The model exchanges the individuals in the three subpopulations 0, 1, and 2. There are \( n_N \) individuals in subpopulation 0, \( n_M \) individuals in subpopulation 1, and \( n_K \) individuals in subpopulation 2, respectively. The size of the subpopulations should have \( n_N \gg n_M, n_K \) for considering a trade-off between speed and accuracy. The selection method used in the model always uses the tournament selection strategy rather than the fitness-proportionate selection strategy. Additionally, the tournament selection (TS) differs from TP in the target level. The former selects the individuals in the population as its targets on the size 7 tournament. Meanwhile, the latter selects the programs (nodes) of an individual (tree) to do the same. Regarding the probabilities of genetic operations, the genetic operation of crossover and reproduction gives values of 0.9 and 0.1, respectively. To control tree shape, the tree depth must be 20 or less after performing the crossover operation. Furthermore, each internal node has between one and three children connected to it. Related applications employing genetic programming can be found in [43–45].

4.2. Cooperation of multiple fitness functions

Fig. 4 illustrates the inner-exchange strategy of individuals in three subpopulations of a population for achieving the cooperation of the multiple fitness function. Initially, a set of initial feature nodes is fed into the subpopulation 0. Subsequently, two feature-node sets produced by random orders of all feature-nodes match each other with TPS methods to find the best correspondence between them. The subpopulation 0 is responsible for comparing the functional signals via heuristic matching. For effective heuristic match, the model assumes that the identical conserved protein domains demonstrate genetically similar functions given analogous functional signal shapes. Consequently, a physical deformation of the signal shapes should be preserved in somewhat ancestral contours. Subsequently, in Fig. 4, the \( n \) best individuals of subpopulation 0 must emigrate into subpopulation 1 for modifying the correct boundaries of the protein fragments. In subpopulation 1, an exhausted local alignment of two fragments (nodes) is used to adjust the boundaries of two protein fragments. Meanwhile, a moment invariant
model calculates a new set of feature data for updating the signal data. Moreover, the model must realign the protein fragments once the boundaries of the related fragments have been changed. Next, the $m$ best individuals of subpopulation 1 must emigrate into subpopulation 2 to confirm the significant protein fragments by training of real cases. Moreover, the $n$ best individuals of subpopulation 0 also must emigrate into subpopulation 2 to do the same work. Finally, the $k$ best individuals of subpopulation 2 must emigrate into subpopulation 0 to resume the individuals having specific biological evidence. These resumed individuals can improve the quality of the individuals of subpopulation 0 using the guidelines from real cases. Simultaneously, subpopulation 2 eliminates the worst individuals from subpopulation 2 to prevent them from backing to subpopulation 0 or 2. When all of the individuals in the three subpopulations have been coevolved by genetic programming, a final solution should select the optimum individual from subpopulation 0. Specifically, the solution takes the form of a causal tree. Each path from the root node to any external node represents the functional heredity process. This path can provide an auxiliary guide to biologists for comprehensively understanding the unknown function of query sequences. The strategies used by exchanged individuals are feasible and essential for improving the final solution quality. More specifically, the model cooperatively evolves the causal trees via the cooperation of multiple fitness functions. The multiple fitness functions use Eq. (27) as the fitness function of subpopulation 0, Eq. (33) as the fitness function of subpopulation 1, and the product of Eq. (30) for all nodes as the fitness function of subpopulation 2. \textbf{Fig. 5} describes the statements of the whole algorithm constructing this AGCT model.

5. Experiments and discussions

The protein nucleocapsid (nsp1) of a coronavirus family from various species was tested to clarify the protein function of nsp1 of SARS coronavirus.
Table 1
Target sequences

| S. No. | Name   | Accession | Definition                                      | Source                                  |
|--------|--------|-----------|------------------------------------------------|-----------------------------------------|
| 1      | STW1   | AAP37015  | Putative polyprotein region: 819..3240 (putative nsp1) | SARS coronavirus TW1                    |
| 2      | HcoV   | NP_835345 | Coronavirus p195/p210 protein (nsp1)             | Human coronavirus 229E                  |
| 3      | TGEV   | NP_840002 | Putative coronavirus nsp1                       | Transmissible gastroenteritis virus     |
| 4      | BcoV   | NP_742169 | Coronavirus nsp1 (PL1-RO, PL2-RO, HD)           | Bovine coronavirus                      |
| 5      | MHV    | NP_740609 | Coronavirus nsp1 (PL1-RO, PL2-RO, HD, HD-1)     | Murine hepatitis virus                  |
| 6      | PEDV   | NP_839958 | Putative coronavirus nsp1                       | Porcine epidemic diarrhea virus         |
| 7      | AIBV   | NP_740622 | Coronavirus nsp1 (HD1) hydrophobic domain 1     | Avian infectious bronchitis virus       |

5.1. Data set

For seven input sequences, one sequence is a query sequence and six are member sequences. The model simultaneously inputs the sequences to analyze its performance. For convenience, Table 1 lists the information on sequence number, abbreviated names, accessions, definitions, and sources obtained from the NCBI web site for the proteins, as follows:

These coronaviruses are members of a family of enveloped viruses that replicate in the cytoplasm of animal host cells [46]. SARS coronavirus TW1 is a severe acute respiratory syndrome associated coronavirus known as the TW1 isolate. For simplicity, the model only inputs the sequence of putative nsp1 protein of TW1 for analysis with the known protein nsp1 in other hosts. However, this analysis is useful for understanding the novel protein based on inference using other proteins with known functions. Although the traditional MSA can provide accurate knowledge regarding the conserved domains for these proteins, it cannot clearly express homology relationships over these conserved domains. In fact, the proposed model can provide a hierarchical contour of protein fragments. The contour can also help biologists to increase understanding of the functional heredity relationships only depending on the primary structure of proteins. Due to the space limitation, this matching causal tree for the problem using this model is not shown.

5.2. Analysis of the AGCT model

This section analyzes the performance of convergence of energy at different evolutionary generations using the AGCT model. Table 2 lists some of test parameters used for the AGCT model to obtain a suboptimal solution to the problems. The parameters which are appropriate for the present case are obtained using Eqs. (33)–(35). Moreover, a reasonable value of $\lambda_2 = 1.8$ is used for Eq. (27) through analysis using Eq. (35). The analysis attains the reasonable value using a general cross validation. To analyze the AGCT model, cases (I) to (V) confirm these typically varied situations. First, case (I) designs for illustrating the ideal solution can be found prematurely. Next, case (II) designs a situation involving a sufficiently large subpopulation size but related small migration size for the subpopulations. Next, case (III) designs a situation involving a small population size and small migration size. Case (IV) then designs a situation involving sufficient population size and large migration size. Finally, case (V) attempts to understand the real circumstances of evolution at subpopulations 0 and 1 when the energy of subpopulation 2 demonstrates convergence. Experimentally, the key parameters are the rates of
temperature decrease and emigration size for each subpopulation, which are used to analyze model performance.

For clarity, Table 3 lists the results of the five cases. The sensitivity column and specificity column of Table 3 display the prediction and discrimination capacities, respectively. Since the capacity estimation is based on averaging all generations, the ratios of comparison between the heuristic signal match and exhaustive sequence alignment are quite low. In fact, the high prediction capacity can improve the final solution quality. The high discrimination capacity can increase the execution speed. Although the model obtains low sensitivity results, the model attains relatively high specificity results. Actually, the results of the later generations have higher sensitivity than previous generations (data not shown). Thus, the sensitivity results, which decrease the execution performance during the co-evolution in this model, do not influence the prediction accuracy. In fact, the rises and falls depend on a threshold (cut-off) value, because of a trade-off between sensitivity and specificity. Thus, Eq. (33) specifies a threshold value of 1.5 to determine whether or not two protein fragments are homologous. Moreover, Eq. (27) specifies a threshold of value of 0.3 to decide whether or not two signals are analogous.

From the observation of the data in Table 3, case II has better sensitivity than the other cases. Furthermore, case II is also competitive with other reported results. Although fewer individuals are involved in case III than case II, the resulting reports in case III are the best except for this sensitivity result. The following discussion can further explain the above confusion. Although cases II and IV involve the same numbers of individuals, case IV involves more immigrated or emigrated individuals than case II. Consequently, case IV obtains worse results than case II. The above demonstrates that, appropriate individual migration is essential for improving solution quality. However, the amount of migration should not be too large to perform the normal evolution by genetic programming. Meanwhile, population size is not particularly important in the present cases since the RPM method and Smith–Waterman algorithm can undoubtedly

| Case no. and name | Subpopulation | Convergence generation |
|-------------------|---------------|------------------------|
|                   | Energy | Sensitivity (%) | Specificity (%) | Energy | Energy |
| Case 1, I         | 0.515 | 8.5         | 93.8         | 0.316 | 0.182 | 40 |
| Case 2, II        | 0.419 | 16.2        | 94.2         | 0.210 | 0     | 30–50 |
| Case 3, III       | 0.403 | 11.3        | 94.6         | 0.210 | 0     | 30–100 |
| Case 4, IV        | 0.426 | 11.2        | 94.0         | 0.301 | 0.150 | 100–120 |
| Case 5, V         | 0.413 | 10.8        | 94.4         | 0.212 | 0     | 40–80 |
obtain the local optimal result for each generation of genetic programming. Therefore, even though case III involves fewer individuals than case II, case III can still achieve the performance results approaching case II. Finally, the mechanism of premature convergence in cases I and IV attained the worst results. Clearly, case V improves the results of case V through a long-term evolution.

Accordingly, Fig. 6 displays the performance of convergence in each subpopulation on fixed temperature. The energy of subpopulations 0 converges at 0.488 and has a different rate of convergence to the other two subpopulations. Clearly, subpopulation 2 evolves faster than the other two subpopulations. Fig. 7 illustrates the improvement of the energy convergence of subpopulation 0 by using the five temperature rates (Tr) for decreasing annealing temperature, including 1, 0.95, 0.90, 0.85, and 0.80.

Eventually, the suboptimal solution to this present experiment is yielded using the genetic programming via a cooperation of the multiple fitness functions. This cooperation can
continuously change the positions appropriate for splitting the protein sequence. Due to the space limitation in this paper, the starting and ending positions of each protein fragment are not shown.

Subsequently, because of case II containing more individuals than the others, Fig. 8 compares the energy convergences based on case II with each other for obtaining an appropriate Tr value. More precisely, the rate of value 0.85 is the best choice for situations involving rapid and steady convergence. Additionally, Fig. 9 compares four different Tr values 0.97, 0.95, 0.9, and 0.85 for analyzing the energy of subpopulation 0. Clearly, case IV really is the worst case. Next, Fig. 10 simultaneously illustrates the energy convergence of three subpopulations based on case V by varying the rates of temperature decrease. Clearly, the model proportionally and consistently reduces the energies of each subpopulation. From the observation of generations in Fig. 10, subpopulations 1 and 2 already begin to converge at around the fortieth generation. Finally, the
model gradually performs the RPM work in subpopulation 0 between the fortieth and eightieth generations.

6. Conclusions and future work

The AGCT model utilizes a novel representation of tree structure to illustrate the concept of applying functional homologies to predict protein function. This model differs in numerous respects from the other traditional model of MSA. Nevertheless, this model still includes various traditionally popular methods used for the high level application, predicting protein function. These methods include genetic programming, wavelet transform, profile comparison, local alignment with the Smith–Waterman algorithm, moment invariant theorem, RPM with TPS, Bayes inference, and the causal tree model. Besides the above methods, this model introduces various strategies, including the inner-exchanged individuals in subpopulations, cooperation of multiple fitness functions, guided evolution involving real cases, and TP+DFS regularization. The prediction accuracy thus can be improved. This hybrid model is developed to exploit global search capabilities in genetic programming for predicting protein functions of a distantly related protein family that have difficulties in the conserved domain identification. Moreover, the RPM involved can identify more sequence homologies through softening comparisons. Essentially, this work contributed a complex and integrated methodology that enables new advances in predicting protein function in the post-genome era. We believe that the model robustness can be increased in the future if biologists generate more experimental data in laboratories. Future works will apply this model to investigate issues related to protein interaction.

References

[1] A. Delcoigne, P. Hansen, Sequence comparison by dynamic programming, Biometrika 62 (1975) 661–664.
[2] A. Krogh, S. Mian, D. Haussler, A hidden Markov model that finds genes in e. coli DNA, Nucleic Acids Res. 22 (1994) 4768–4778.
[3] A. Krogh, M. Brown, I.S. Mian, K. Sjolander, D. Haussler, Hidden Markov models in computational biology: Applications to protein modeling, J. Mol. Biol. 235 (1994) 1501–1531.
[4] P. Judea, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, Los Angeles, 1997.
[5] D. Cai, A. Delcher, B. Kao, S. Kasif, Modeling splice sites with Bayes networks, Bioinformatics 16 (2000) 152–159.
[6] P.Y. Chou, G.D. Fasman, Prediction of the secondary structure of proteins from their amino acid sequence, Adv. Enzymol. Relat. Areas Mol. Biol. 47 (1978) 45–148.

[7] T.F. Smith, M.S. Waterman, Identification of common molecular subsequences, J. Mol. Biol. 147 (1981) 195–197.

[8] A. Tsakonas, G. Dounias, Hybrid computational intelligence schemes in complex domains: An extended review, in Proc. Methods and Applications of Artificial Intelligence: Second Hellenic on AI (SETN), 2002, pp. 494–512.

[9] P. Willett, Genetic algorithms in molecular recognition and design, Trends Biotechnol. 13 (1995) 497–537.

[10] Z.R. Yang, R. Thomson, T.C. Hodgman, J. Dry, A.K. Doyle, A. Narayanan, X. Wu, Searching for discrimination rules in protease proteolytic cleavage activity using genetic programming with a min-max scoring function, Comput. Intell. Bioinformatics 72 (2003) 159–176.

[11] S. Dreiseitl, L. Ohno-Machado, Logistic regression and artificial neural network classification models: a methodology review, J. Biomed. Informatics 35 (2002) 352–360.

[12] A.C.C. Coolen, V.D. Prete, Statistical mechanics beyond the Hopfield model: solvable problems in neural network theory, Rev. Neurosci. 14 (2003) 181–193.

[13] H.T.B. Jason, P.Y. Michael, Applying fuzzy logic to medical decision making in the intensive care unit, Am. J. Respir. Crit. Care Med. 167 (2003) 948–952.

[14] H. Ressom, R. Reynolds, R.S. Varghese, Increasing the efficiency of fuzzy logic-based gene expression data analysis, Physiol. Genomics 13 (2003) 107–123.

[15] K. Reichard, M. Kaufmann, EPPS: mining the COG database by an extended phylogenetic patterns search, Bioinformatics 19 (2003) 784–788.

[16] R. Durbin, S. Eddy, A. Krogh, G. Mitchison, Biological Sequence Analysis Probabilistic Models of Proteins and Nucleic Acids, Cambridge University Press, 1998.

[17] D. Mittelman, R. Sadreyev, N. Grishin, Probabilistic scoring measures for profile–profile comparison yield more accurate short seed alignments, Bioinformatics (2003) 1531–1539.

[18] L. Rychlewski, L. Jaroszewski, W. Li, A. Godzik, Comparison of sequence profiles. Strategies for structural predictions using sequence information, Protein Sci. 9 (2000) 232–241.

[19] G. Yona, M. Levitt, Within the twilight zone: a sensitive profile–profile comparison tool based on information theory, J. Mol. Biol. 315 (2002) 1257–1275.

[20] R. Sadreyev, N. Grishin, COMPASS: a tool for comparison of multiple protein alignments with assessment of statistical significance, J. Mol. Biol. 326 (2003) 317–336.

[21] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, D.J. Lipman, Basic local alignment search tool, J. Mol. Biol. 215 (1990) 403–410.

[22] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, D.J. Lipman, Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res. 25 (1997) 3389–3402.

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

[24] H. Chui, A. Rangarajan, A new algorithm for non-rigid point matching, in: Computer Vision and Pattern Recognition, in: Proc. IEEE Conference, vol. 2, 2000, pp. 44–51.

[25] S. Meshoul, M. Batouche, Ant colony system with extremal dynamics for point matching and pose estimation, in: Proc. 16th International Conference, August 11–15, 2002, Pattern Recognit. 3 (2002) 823–826.

[26] J.R. Koza, Genetic Programming On the Programming of Computers by Means of Natural Selection, Massachusetts Institute of Technology, USA, 1992.

[27] S. Gold, A. Rangarajan, C.P. Lu, S. Pappu, E. Mjolsnes, New algorithms for 2-D and 3-D point matching: pose estimation and correspondence, Pattern Recognit. 31 (1998) 1019–1031.

[28] A.L. Delcher, S. Kasif, H.R. Goldberg, W.H. Hsu, Probabilistic prediction of protein secondary structure using causal networks, in: Proc. 11th AAAI National Conference on Artificial Intelligence, 1993, pp. 316–321.

[29] C.C. Holmes, D. Denison, Perfect sampling for wavelet reconstruction of signals, IEEE Trans. Signal Process. 50 (2002) 237–244.

[30] Q. Zheng, R. Chellappa, A computational vision approach to image registration, IEEE Trans. Image Process. 2 (1993) 313–326.

[31] L.S. Steven, B.S. David, K. Simon, Computational Methods in Molecular Biology, vol. 32, Elsevier, New York, 1998.

[32] M.K. Hu, Visual pattern recognition by moment invariants, IRE Trans. Inform. Theory 8 (1962) 179–187.

[33] M. Dresher, Moment spaces and inequalities, Duke Math. J. 20 (1953) 261–271.

[34] G. Wahba, Spline Models for Observational Data, SIAM, Philadelphia, PA, 1990.
[35] F. Riesz, B. Sz-Nagy, Functional Analysis, Frederick Ungar, New York, 1955.
[36] R. Sinkhorn, A relationship between arbitrary positive matrices and doubly stochastic matrices, Ann. Math. Statist. 35 (1964) 876–879.
[37] C. Peterson, B. Soderberg, A new method for mapping optimization problems onto neural networks, Internat. J. Neural Systems 1 (1989) 3–22.
[38] A.L. Yuille, J.J. Kosowsky, Statistical physics algorithms that converge, Neural Comput. 6 (1994) 341–356.
[39] A. Rangarajan, S. Gold, E. Mjolsness, A novel optimizing network architecture with applications, Neural Comput. 8 (1996) 1041–1060.
[40] D.E. Goldberg, Genetic and evolutionary algorithms come of age, Commun. ACM 37 (1994) 113–119.
[41] J.R. Koza, F.H. Bennett, D. Andre, M.A. Keane, Genetic programming III: Darwinian invention and problem solving: Book Review, IEEE Trans., Evolutionary Comput. 3 (1999) 251–253.
[42] I. Kushch, Genetic programming and evolutionary generalization, IEEE Trans., Evolutionary Comput. 6 (2002) 431–442.
[43] M.L. Wong, W. Lam, K.S. Leung, P.S. Ngan, J.C.Y. Cheng, Discovering knowledge from medical databases using evolutionary algorithms, IEEE Eng. Med. Biol. Mag. 19 (2000) 45–55.
[44] C.C. Bojarczuk, H.S Lopes, A.A. Freitas, Genetic programming for knowledge discovery in chest-pain diagnosis, IEEE Eng. Med. Biol. Mag. 19 (2000) 38–44.
[45] J.R. Koza, F.H. Bennett, D. Andre, Classifying proteins as extracellular using programmatic motifs and genetic programming, in: Evolutionary Computation Proc., 1998. IEEE World Congress on Computational Intelligence, 1998, pp. 212–217.
[46] B.N. Fields, D.M. Knipe, P.M. Howley, D.E. Griffin, Fields Virology, 4th edition, Lippincott Williams & Wilkins, Philadelphia, 2001.
[47] P. Hall, D.M. Titterington, Common structure of techniques for choosing smoothing parameters in regression problems, J. Roy. Statist. Soc. Ser. B 49 (1987) 184–198.