Mathematical and Computational Applications

MDPI

Article

A Peptides Prediction Methodology for Tertiary Structure Based on Simulated Annealing

Juan P. Sánchez-Hernández [1,†], Juan Frausto-Solís [2,*,†], Juan J. González-Barbosa [2], Diego A. Soto-Monterrubio [2], Fanny G. Maldonado-Nava [2] and Guadalupe Castilla-Valdez [2]

[1] Dirección de Informática, Electrónica y Telecomunicaciones, Universidad Politécnica del Estado de Morelos, Boulevard Cuauhnáhuac 566, Jiutepec 62574, Mexico; juan.paulosh@upemor.edu.mx

[2] Graduate Program Division, Tecnológico Nacional de México/Instituto Tecnológico de Ciudad Madero, Cd. Madero 89440, Mexico; jjgonzalezbarbosa@hotmail.com (J.J.G.-B.); diego_060787@hotmail.com (D.A.S.-M.); fanny_mn@hotmail.com (F.G.M.-N.); gpe_cas@yahoo.com.mx (G.C.-V.)

* Correspondence: juan.frausto@gmail.com

† These authors contributed equally to the development of this paper.

Abstract: The Protein Folding Problem (PFP) is a big challenge that has remained unsolved for more than fifty years. This problem consists of obtaining the tertiary structure or Native Structure (NS) of a protein knowing its amino acid sequence. The computational methodologies applied to this problem are classified into two groups, known as Template-Based Modeling (TBM) and ab initio models. In the latter methodology, only information from the primary structure of the target protein is used. In the literature, Hybrid Simulated Annealing (HSA) algorithms are among the best ab initio algorithms for PFP; Golden Ratio Simulated Annealing (GRSA) is a PFP family of these algorithms designed for peptides. Moreover, for the algorithms designed with TBM, they use information from a target protein's primary structure and information from similar or analog proteins. This paper presents GRSA-SSP methodology that implements a secondary structure prediction to build an initial model and refine it with HSA algorithms. Additionally, we compare the performance of the GRSAX-SSP algorithms versus its corresponding GRSAX. Finally, our best algorithm GRSAX-SSP is compared with PEP-FOLD3, I-TASSER, QUARK, and Rosetta, showing that it competes in small peptides except when predicting the largest peptides.

Keywords: protein structure prediction; Hybrid Simulated Annealing; Template-Based Modeling; structural biology; Metropolis

check for updates

Citation: Sánchez-Hernández, J.P.; Frausto-Solís, J.; González-Barbosa, J.J.; Soto-Monterrubio, D.A.; Maldonado-Nava, F.G.; Castilla-Valdez, G. A Peptides Prediction Methodology for Tertiary Structure Based on Simulated Annealing. *Math. Comput. Appl.* **2021**, *26*, 39. https://doi.org/10.3390/mca26020039

Academic Editor: Leonardo Trujillo

Received: 23 February 2021
Accepted: 27 April 2021
Published: 29 April 2021

1. Introduction

Proteins or polypeptides are macromolecules built from amino acids (aa) and are mainly responsible for living beings' functionality. Proteins are essentials elements because every protein has a specific function related to its unique three-dimensional structure named Native Structure (NS). All the proteins consist of a polymer chain of aa; the junctions with a small number of them are named peptides. The peptides have significant importance in the science community because of their multiple applications, for instance, in pharmaceutical research [1–4], drug design [5–7], diagnosis [8–10], and therapy [11,12]. To obtain the NS of proteins from an amino acid sequence could bring benefits to human beings.

The PFP has been identified as an important problem since Kendrew and Perutz's research teams obtained the myoglobin and hemoglobin molecules' tertiary structure, respectively [13,14]. These studies established the relation between function and structure. PFP consists of obtaining the three-dimensional structure of a protein with the lowest Gibbs free energy, thermodynamically stable three-dimensional conformation [15].

The PFP is considered an NP-hard problem [16]. Thus, presumably, none of the known exact algorithms can solve it in polynomial time. In other words, the execution time grows exponentially when using them. In contrast, any protein passes from the aa sequence to its NS three-dimensional structure very rapidly in nature. The latter issue is known as the Levinthal Paradox [17].

Several algorithms have been applied to solve the PFP successfully, and one of the most effective algorithms has been the Simulated Annealing algorithm (SA). The SA is commonly hybridized with other methods; the combination algorithms are called Hybrid Simulated Annealing algorithms (HSA). These algorithms successfully applied to peptides are the following:

(a) The classical Monte Carlo Method, or SA, was applied to the PFP [18,19]. Additionally, an analytical tuning method to SA was proposed [20].
(b) Golden Ratio Simulated Annealing (GRSA) family: Original GRSA proposing a cooling strategy [21], Evolutionary Golden Ratio SA (EGRSA) using genetic operators [22], and GRSA2, which is hybridization with the GRSA and Chemical Reaction Optimization algorithm (CRO) [23].
(c) Metropolis and multiobjective optimization methods were applied in the previous CASP competitions. The approaches that traditionally have obtained the best results were Rossetta [24], QUARK [25], and I-TASSER [26]. However, deep learning applied by the Alphafold algorithm [27] achieved the best score in the CASP13 and CASP14.
(d) PEP-FOLD3 algorithm, which uses secondary structure information and a Monte Carlo method, and is very successful for small peptides (5 to 50 aa) [28].

The HSA algorithms previously mentioned obtained excellent results for small proteins or peptides. However, when the number of aa increases, the variables (torsional angle of aa) are also increased, the computational time for exploring the solution space is considerable. As a result, the PFP area needs new approaches to obtaining better solutions for large peptides or proteins.

This paper proposes the methodology GRSA-SSP that combines GRSA algorithms with the Secondary Structure Prediction (SSP). For a given chain of aa representing a peptide or a protein, the GRSA-SSP performs two processes:

(a) To obtain the first protein prediction from the secondary structure of the amino-acids sequence.
(b) To refine the previous protein prediction by using GRSA family algorithms.

These two processes are performed in several steps described in this paper. The algorithms used in the second phase of GRSA-SSP can be one of the GRSA family algorithms. This paper named these hybrid algorithms GRSAX-SSP, where X is used to distinguish the GRSA algorithm. We evaluate our methodology using RMSD and TM-score metrics [29]. Additionally, experimentation is performed with a set of forty-five instances of peptides and a set of six mini proteins, which are compared with the most popular algorithms in the literature, such as PEP-FOLD3 [28], I-TASSER [30,31], Rosetta [24,32], and QUARK [25,33].

The paper's organization is as follows: first, we present the introduction to PFP and HSA algorithms. Then, in the Background section, we review the Protein Folding Problem definition and some relevant research in the literature, and we explain the GRSA family of algorithms. In the next section, we describe the GRSA-SSP methodology. In the Results section, we present experimentation comparing the GRSA algorithms with those of the literature; also, we analyze the presented methodology's performance. Finally, the conclusions of this research are presented.

2. Background

The PFP is a significant multidisciplinary problem that has been investigated for over half a century [34]. Different scientific areas have been studied, for example, computer science, bioinformatics, and molecular biology, concerning this problem, and three questions in particular need to be answered [34].

- Which is the physical code in which an amino-acids sequence dictates an NS?
- Why in nature do proteins fold very quickly while in silicon they fold relatively slower?

- Is there an algorithm that predicts the protein structure from the amino-acids sequence?

This paper is related to the last question. We propose different strategies to obtain the NS tertiary structure using GRSA family algorithms and secondary structure prediction. As we mentioned before, finding new algorithms for PFP is significant not only because of its potential applications but also because it is an NP-hard problem [16], and the number of combinations that determine which algorithms must be explored in a very large solution space.

2.1. Definition of Ab-Initio and Force Fields

The ab initio modeling can be defined as an optimization problem where the Gibbs free energy is the objective function f(n), and this has to be minimized. Thus, this problem is defined as follows: let there be a sequence of amino acids: n = a1, a2, . . . , an; every amino acid has associated with it a set of angles σ1, σ2, . . . , σm where m represents a particular dihedral angle; then, minimizing the energy function f(σ∣1, σ2, . . . , σm) provides the best tertiary structure or NS. The energy functions (force fields) are used for determining the energy of a protein structure [35], and some examples of these are AMBER [36], CHARMM [37], ECEPP/2, and ECEPP/3 [38]. The potential energy of ECEPP/2 is given by Equation (1), which is calculated in vacuo for only intramolecular energies, and this is the energy function to be minimized [39].

$$E_{total} = \sum_{j>i}\left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{{r_{ij}}^{6}}\right) + 332\sum_{j>i}\frac{q_i q_j}{\varepsilon r_{ij}} + \sum_{j>i}\left(\frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{{r_{ij}}^{10}}\right) + \sum_{n} U_n(1 \pm cos(k_n \varphi_n)) \quad (1)$$

where: r_{ij} is the distance in Å (angstroms) between the atoms i and j; A_{ij}, B_{ij}, C_{ij}, and D_{ij} are the parameters of the empirical potentials; q_i and q_j are the partial charges in the atoms i and j, respectively; ε is the dielectric constant; U_n is the energetic torsion barrier of rotation about the bond n; k_n is the multiplicity of the torsion angle φ_n.

In this paper, we use the potential energy of ECEPP/2 as an objective function because we explore the conformational space, and when the energy of the protein structure is minimized, then the protein structure is accepted.

2.2. Computational Approaches for PFP

The CASP organization has classified PFP models into two main groups:

Group 1: Template-based modeling (TBM). In this group, we find algorithms that use biological information obtained from the secondary structure of the target protein, homology, and fragments of other proteins. These algorithms have achieved good results for predicting protein structures in the CASP [32,40,41]. TBM involves several strategies; some of the most common are homology [42,43], threading [44], and fragment assembly [30,45].

Group 2: Ab initio. This prediction approach classically refers to the determination of the NS using only the aa sequence information. Unfortunately, ab initio algorithms have achieved good PFP results but only for small proteins with less than 120 residues [46]. The Ab initio modeling is the most challenging approach because it uses the amino acids' sequence as unique information. Finding an optimal solution with ab initio is very difficult for big proteins because the solution space is enormous.

These two groups can be applied to small proteins or peptides (between 5 to 50 aa) [28,47]. There are successful studies applied to protein prediction using SA [48–50] or Monte Carlo algorithms with Metropolis-Hasting [26,27]. The Monte Carlo algorithms are also applied to the inverse protein folding problem, which objective function is to find a sequence given a structure [51,52]. This paper focuses on the classical PFP that consists of finding the functional structure given a sequence aa.

The Rosetta is a protein structure prediction or de novo approach that performs models for the tertiary structure using the primary and secondary structure predictions. The algorithm generates a local sequence to produce local structures (fragments) that form a target protein template. Additionally, the fragments are then assembled by randomly using a Monte Carlo simulated annealing algorithm. Finally, the fitness of individual conformation interactions is evaluated based on a scoring function derived from known protein structures. However, only peptides longer than 27 aa can be provided as input [32].

Another PFP approach is I-TASSER (Iterative Threading ASSEmbly Refinement). It has four principal parts: generating a template using a multi-threading method, fragments' assembly method, refinement process, final model selection, and annotation tools. The I-TASSER applies an alignment of the target sequence and divides it into aligned using LOMETS [53,54] and nonaligned regions using the Monte Carlo algorithm. In the last step, annotation of functions is performed based on the structural models obtained using the BioLIP [55] database of ligand-protein interactions. Finally, the I-TASSER predicts protein structures from 10 to 1500 amino acids [31].

PEP-FOLD3 has a framework to predict the tertiary structure of peptides using de novo structure modeling. The process of predicting structure consists of three stages. Firstly, for a peptide amino acid sequence, a support vector machine is applied to predict the structural alphabet of fragments. Secondly, several models are generated using series of states and refined by a Monte Carlo algorithm. Finally, the five best conformations are selected [28].

Another approach is QUARK [33], in which an ab initio strategy is used to predict protein structures in ranges of 20 to 200 aa. Additionally, an assembly process of fragments with small structures is carefully selected and applied in the target sequence using a Monte Carlo algorithm.

SAINT2 is a fragment-based de novo structure prediction approach that has been successfully compared with the CASP12 approaches [56], which consists of a sequence-to-structure pipeline divided into four principal sections: (a) the secondary structure prediction where PSI-PRED [57] is applied, (b) the torsion angles prediction using SPINE-X [58], (c) a fragment library with the Flib package, and (d) the residue-residue contact prediction applying metaPSICOV [59]. Finally, the highest-scoring model is selected. In our methodology, sections (a) and (b) are applied, and they are shown in Figure 1.

The GRSA Family Algorithms

The SA algorithm is inspired by the physical annealing process of metals [60,61]. The algorithm has been applied with success in many NP-hard problems [20], including the PFP. SA employs the Metropolis algorithm to efficiently explore the solution space and obtain a good solution to optimization problems. We show the pseudocode of SA in Algorithm 1. T_i and T_f parameters define the initial and final temperatures, respectively; the α parameter represents the cooling factor. In the Metropolis cycle, new solutions are generated by a perturbation function. Finally, to accept or reject a new solution, an acceptance criterion based on Boltzmann distribution is applied (lines 11–14). The SA algorithm is executed until the final temperature, $\mathrm{T_f}$, is reached. The SA algorithm source code is available at https://github.com/DrJuanFraustoSolis/SimulatedAnnealing.git (accessed on 28 April 2021).

Algorithm 1. SA algorithm Procedure.

```
Data: $T_i$, $T_f$, $\alpha$
$T_k = T_i$; $\alpha = 0.95$
$S_i$ = generateSolution()
    while $T_k \ge T_f$ do  //Temperature cycle
        while Metropolis length do  //Metropolis cycle
            $S_j$=perturbation($S_i$)
            $\Delta E$=Energy($S_j$) – Energy($S_i$)
            if $\Delta E \le 0$ then
                $S_i$=$S_j$
                $E$ = Energy($S_i$)
            else if $e^{-\Delta E/Ti}$ < random [0-1] then
                $S_i$=$S_j$
                $E$ = Energy($S_i$)
            end if
        end while  //End Metropolis cycle
        $T_k = T_k*\alpha$
    end while  //End Temperature cycle
end Procedure
```

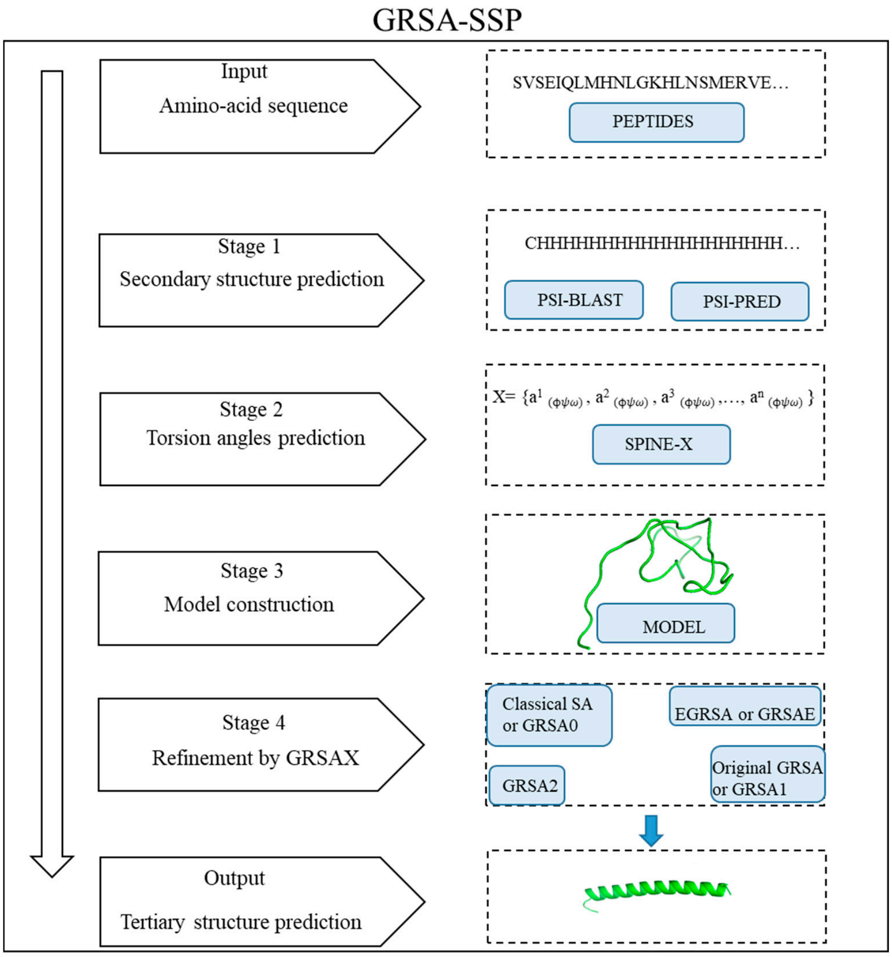


Figure 1. Methodology GRSA-SSP for peptide prediction.

However, when the solution space is very large, the algorithm's exploration takes a long time to obtain optimal solutions. Thus, new algorithms are necessary. The GRSA algorithm was proposed, which has been successfully applied in different NP problems [62,63], including the PFP [18]. The main characteristics of GRSA are the cooling scheme that decreases according to T_{fp} temperature cuts calculated by the golden number (ϕ) and then a stop criterion that reduces the cost of exploration (Algorithm 2). GRSA has a similar structure to the SA algorithm (lines 4 to 16). The difference with SA is that the GRSA calculates T_{fp} temperature cuts (five cuts are recommended), and in each cut, an α parameter in the range [0.7, 1] is associated (the common higher value is 0.95); the intermediate α values in this range are determined with an increment δ which represent the α increment since the lowest until the highest α value (in this case, $\delta = 0.05$). These alpha values are associated with each temperature cut (line 17). The algorithm reduces the temperature cooling speed; thus, the execution time, corresponding to lines 18 to 23, decreases. Finally, to reduce wasting time in low temperatures, where the quality of the result is not improved, a stop criterion was implemented using the least-squares method (lines 24 to 29). This stop criterion detects the stochastic equilibrium for some i Metropolis cycles. We measure the slope (m is a global variable) of the linear regression of the energy of these cycles. In this regression, we used the coordinates $(E_i,\ i)$; where i is in the range [2, κ_{max}]. In our case, we used κ_{max} = 5. The equilibrium is found when m is close to zero, calculated by (2).

$$m = \frac{\kappa \sum_{i=2}^{\kappa} iE_i - (\sum_{i=2}^{\kappa} i)(\sum_{i=2}^{\kappa} E_i)}{\kappa \sum_{i=2}^{\kappa} i^2 - (\sum_{i=2}^{\kappa} i)^2} \tag{2}$$

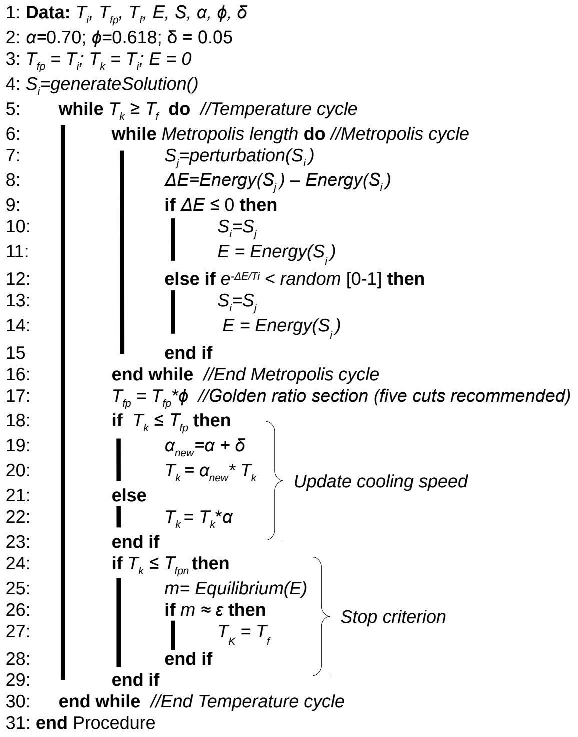
The Equation (2) can be written as follows (3):

$$m = \frac{12 \sum_{i=2}^{\kappa} iE_i - 6(\kappa - 1)(\sum_{i=1}^{\kappa} E_i)}{\kappa^3 - \kappa} \tag{3}$$

where: κ is the number of metropolis cycles for measuring the slope, i is the iteration of every metropolis cycle, and E_i the energy in each iteration.

The evaluation of m in Equation (2) does not imply a significative execution time; the summations on Equation (3) are only cumulative operations in Algorithm 3. This algorithm determines the equilibrium with this Equation (3). The GRSA algorithm source code is available at https://github.com/DrJuanFraustoSolis/GRSA.git (accessed on 28 April 2021).

Algorithm 2. GRSA algorithm Procedure.



```
1: Data: $T_i$, $T_{fp}$, $T_f$, $E$, $S$, $\alpha$, $\phi$, $\delta$
2: $\alpha$=0.70; $\phi$=0.618; $\delta$ = 0.05
3: $T_{fp} = T_i$; $T_k = T_i$; $E = 0$
4: $S_i$=generateSolution()
5:      while $T_k \geq T_f$ do //Temperature cycle
6:          while Metropolis length do //Metropolis cycle
7:              $S_j$=perturbation($S_i$)
8:              $\Delta E$=Energy($S_j$) – Energy($S_i$)
9:              if $\Delta E \leq 0$ then
10:                 $S_i$=$S_j$
11:                 $E$ = Energy($S_i$)
12:             else if $e^{-\Delta E/Ti}$ < random [0-1] then
13:                 $S_i$=$S_j$
14:                 $E$ = Energy($S_i$)
15              end if
16:         end while //End Metropolis cycle
17:         $T_{fp} = T_{fp}*\phi$ //Golden ratio section (five cuts recommended)
18:         if $T_k \leq T_{fp}$ then
19:             $\alpha_{new}$=$\alpha + \delta$
20:             $T_k = \alpha_{new} * T_k$                 Update cooling speed
21:         else
22:             $T_k = T_k*\alpha$
23:         end if
24:         if $T_k \leq T_{fpn}$ then
25:             m= Equilibrium(E)
26:             if $m \approx \varepsilon$ then                     Stop criterion
27:                 $T_K = T_f$
28:             end if
29:         end if
30:     end while //End Temperature cycle
31: end Procedure
```

Algorithm 3. Equilibrium Function.

```
Equilibrium(E)
i = 1; CE=i*E; Kmax=5; SumE=E; m=0
    if i < Kmax then
        CE=CE+i*E
        SumE=SumE+E
        i = i+1
    end if
    if i==Kmax then
        m = $(((12 * CE)-(6*(i-1)*SumE))/(i^3 - i))$
    end if
    return m
end Function
```

The EGRSA (Algorithm 4) is an algorithm integrated by the hybridization of GRSA with evolutionary techniques. This algorithm has an evolutionary perturbation (EGRSApert) in the GRSA phase (line 7), where a genetic algorithm is used. The EGRSA algorithm starts with a set of individuals generated for determining the initial solution designed as S_i. Then in the Metropolis Cycle, the S_i is perturbated by EGRSApert to generate new solutions. Next, the best individual generated S_j solution is selected of the population (lines 9 and 10). EGRSA is similar to GRSA, and both applied a stop criterion (see Algorithm 2.1) by the least-squares method [64,65] (lines 24–29). Algorithm 5 presents EGRSApert function, where one individual is a set of dihedral angles [ϕ_1, Ψ_1, X_1, ω_1, ϕ_2, Ψ_2, X_2, ω_2, …, ϕ_n, Ψ_n, X_n, ω_n] and a population is a set of individuals. Then crossover and mutation operators are applied to generate new solutions by the perturbation function. Finally, when the number of generations is reached, the best individual of the population is selected. The EGRSA algorithm source code is available at https://github.com/DrJuanFraustoSolis/EGRSA.git (accessed on 28 April 2021).

Algorithm 4. EGRSA algorithm Procedure.

```
1:  Data: $T_i$, $T_{fp}$, $T_f$, E, S, $\alpha$, $\phi$
2:  $\alpha$=0.70; $\phi$=0.618; $\delta$ = 0.05
3:  $T_{fp} = T_i$; $T_k = T_i$; E = 0
4:  $S_i$=generateSolution()
5:      while $T_k \geq T_f$ do  //Temperature cycle
6:          while Metropolis length do  //Metropolis cycle
7:              $S_j$ = EGRSApert($S_i$)
8:              $\Delta E$ = Energy($S_j$) – Energy($S_i$)
9:              if $\Delta E \leq 0$ then
10:                 $S_i = S_j$
11:                 E =Energy($S_i$ )
12:             else if $e^{-\Delta E/Ti}$ < random [0-1] then
13:                 $S_i = S_j$
14:                 E =Energy($S_i$ )
15:             end if
16:         end while  //End Metropolis cycle
17:         $T_{fp} = T_{fp}*\phi$  //Golden ratio section (five cuts recommended)
18:         if  $T_k \leq T_{fp}$ then
19:             $\alpha_{new}=\alpha + \delta$
20:             $T_k = \alpha_{new} * T_k$          } Update cooling speed
21:         else
22:             $T_k = T_k*\alpha$
23:         end if
24:         if $T_k \leq T_{fpn}$ then
25:             m= Equilibrium(E)
26:             if m ≈ ε then                       } Stop criterion
27:                 $T_K = T_f$
28:             end if
29:         end if
30:     end while  //End Temperature cycle
31: end Procedure
```

Algorithm 5. EGRSApert Function.

```
EGRSApert($S_i$)
$n$= numGen, bestSol[ ], bestEnergy
pob=initialPob()
    while $gen \leq n$ do
        population = tournament()
        population = crossPopulation()
        population =  mutatePopulation()
    end while
    pop* = bestIndividual()
    return(bestSol[ ], bestEnergy)
end Function
```

The GRSA2 algorithm [23] is a hybridization of GRSA with the CRO algorithm [66]. GRSA2 (Algorithm 6) is an enhancement of GRSA. It has the same structure as the previous algorithms revised in this paper. Specifically, GRSA2 has two principal differences in the perturbation phase, applying decomposition and soft collision (line 8) and the acceptance criterion (lines 10 to 14). In Algorithm 7, we show the perturbation process implemented in the GRSA2pert function. In GRSA2, two soft collisions are used (unimolecular, Intermolecular). This algorithm has been applied only in the PFP with a set of 19 peptides and compared with I-TASSER and PEP-FOLD3 approaches obtaining outstanding results in the case of peptides [23]. The GRSA2 algorithm source code is available at https://github.com/DrJuanFraustoSolis/GRSA2.git (accessed on 28 April 2021).

Algorithm 6. GRSA2 algorithm Procedure.

```
1: Data: $T_i$, $T_{fp}$, $T_f$, KE, E, S, $\alpha$, $\phi$
2: $\alpha$=0.70; $\phi$=0.618; $\delta$ = 0.05
3: KE=0;$T_{fp} = T_i$; $T_k = T_i$; E = 0
4: $S_i$=generateSolution()
5:      while $T_k \geq T_f$ do  //Temperature cycle
6:          while Metropolis length do  //Metropolis cycle
7:              Eold=Energy($S_i$)
8:              $S_j$= GRSA2pert($S_i$)
9:              EP = Energy($S_j$)
10:             if (EP ≤ Eold + KE) then
11:                 $S_i = S_j$
12:                 E=Energy($S_i$ )
13:                 KE = ((Eold+KE)-EP)*random[0,1]
14:             end if
15:         end while  //End Metropolis cycle
16:         $T_{fp} = T_{fp}*\phi$  //Golden ratio section (five cuts recommended)
17:         if  $T_k \leq T_{fp}$ then              ⎫
18:             $\alpha_{new}=\alpha + \delta$              ⎪
19:             $T_k = \alpha_{new} * T_k$          ⎬ Update cooling speed
20:         else                            ⎪
21:             $T_k = T_k*\alpha$                   ⎪
22:         end if                          ⎭
23:         if $T_k \leq T_{fpn}$ then               ⎫
24:             m= Equilibrium(E)           ⎪
25:             if $m \approx \varepsilon$ then               ⎬ Stop criterion
26:                 $T_K = T_f$                 ⎪
27:             end if                      ⎪
28:         end if                          ⎭
29:     end while  //End Temperature cycle
30: end Procedure
```

Algorithm 7. GRSA2pert Function.

```
GRSA2pert($S_i$)
moleColl, b
    if b > moleColl then
        Randomly select one particle Mω
        if Decompositioncriterionmet
            $S_j$ = Decomposition($S_j$)
        else if
            $S_j$ = SoftCollition($S_i$)
        end if
    end if
    return $S_j$
end Function
```

3. GRSA-SSP Methodology

In this section, we present the GRSA-SSP methodology (Figure 1). This methodology has two main processes:

(a) The prediction of the torsion angles (initial solution) from the secondary structure; that corresponds to stages 1 to 4 in Figure 1.

(b) The refinement of the solution obtained from the secondary structure. This is performed with GRSA algorithms showed in stage four (Figure 1).

The GRSA-SSP methodology has an input (amino acid sequence), an output (tertiary structure prediction), and four stages: (1) secondary structure prediction, (2) torsion angles prediction, (3) template construction, and (4) refinement by GRSAX algorithms. Next, we explain each of these stages:

Input (Amino acid sequence). The amino acid sequences are taken as input.

(1) Secondary structure prediction. This secondary structure, which corresponds to the amino acid sequence and is predicted using PSI-PRED [57]. This algorithm generates a sequence profile with PSI-BLAST [67] and performs the prediction of the stage, such as the helix (H), strand (E), and coil (C). PSI-PRED calculates the probability of each possible state and defines the most likely structure.

(2) Torsion angles prediction. The secondary structure's prediction is essential for this stage, where SPINE-X is used to obtain the torsion angles (ϕ, Ψ, and ω) of each amino acid. This process is realized through the Position-Specific Score Matrix and Physical Parameters [58]. SPINE-X applies artificial neural networks to obtain the best predictions of the target's proteins.

(3) Model construction. In this stage, the torsion angles or variables are used to construct a template as initial solution $S_i = [\phi_1, \Psi_1, X_1, \omega_1, \phi_2, \Psi_2, X_2, \omega_2, \ldots, \phi_n, \Psi_n, X_n, \omega_n]$ that is represented by amino acids subscript 1 to n and the same form by the following amino acids up to n; n is dependent on the size of an amino acid sequence of the target protein. The torsion angles represent the base column of the peptide on which the refinement will be performed.

(4) Refinement by GRSAX. When the previous stages construct the peptide template, we can apply a GRSAX algorithm such as GRSA (renamed GRSA1), EGRSA (renamed GRSAE), and GRSA2, as well as the classical SA (GRSA0). The GRSAX algorithms are tested individually for comparison, which obtains a better tertiary structure of the target peptide. Moreover, once the energy and three-dimensional structure is obtained, the structure is evaluated with the RMSD and TM-score [29] metrics.

Output. The GRSAX-SSP algorithm obtains the tertiary structure prediction.

4. Results

We performed the next GRSAX-SSP algorithms with the proposed methodology: (a) GRSA0-SSP using classical SA [19], (b) GRSA1-SSP using original GRSA [21], (c) GRSAE-SSP using EGRSA [22], and (d) GRSA2-SSP using GRSA2 [23]. For all of them, we used the methodology presented in Figure 1. The peptides in this experimentation have 9 to 49 amino acids. The number of variables (torsion angles) for each peptide in this data set is in the range [49, 304]. We chose this set because these instances (peptides) were used before in the literature. This set was also useful for comparing the GRSA2-SSP algorithm with the top-performing approaches of the CASP, which can be used for small peptides. We compared the last algorithm with I-TASSER, PEP-FOLD3, QUARK, and Rosetta, which are among the best algorithms in the CASP competition. We noted a difference between the GRSAX-SSP algorithms and the one that only applies ab initio by naming it GRSAX. Table 1 presents the set of 45 instances sorted by the number of variables taken from [23,28,68,69] and a PDB code represents each peptide.

Table 1. Set of instances (peptides).

N°	PDB-Code	aa	Number of Variables (Torsion Angles)	N°	PDB-Code	aa	Number of Variables (Torsion Angles)
1	1uao	10	47	24	1wz4	23	123
2	1egs	9	49	25	1rpv	17	124
3	1eg4	13	61	26	1pef	18	124
4	1l3q	12	62	27	1du1	20	134
5	2evq	12	66	28	1pei	22	143
6	1le1	12	69	29	1yyb	27	160
7	1in3	12	74	30	1t0c	31	163
8	3bu3	14	74	31	1by0	27	193
9	1gjf	14	79	32	2bn6	33	200
10	1rnu	13	81	33	1wr4	36	206
11	1lcx	13	81	34	1yiu	37	206
12	1k43	14	84	35	2ysh	40	213
13	1a13	14	85	36	1bhi	38	216
14	1nkf	16	86	37	1i6c	39	218
15	1le3	16	91	38	1wr7	41	222
16	1pgbF	16	93	39	2dmv	43	229
17	1dep	15	94	40	1bwx	39	242
18	1niz	16	97	41	1f4i	45	276
19	2bta	15	100	42	1dv0	47	279
20	1l2y	20	100	43	1ify	49	290
21	1e0q	17	109	44	2p81	44	295
22	1b03	18	109	45	1pgy	47	304
23	1wbr	17	120	-	-	-	-

In the experimentation, the GRSAX-SSP algorithms were executed 30 times to validate the results. The energy function ECEPP/2 is determined with SMMP framework [38]; it is the objective function of our optimization algorithms. An analytical tuning [20] was performed to obtain the initial and final temperature for each instance. In GRSA0-SSP the α value is 0.95, and the temperature range has zero golden sections. For GRSA1-SSP, GRSAE-SSP, and GRSA2-SSP algorithms, the same cooling scheme was used, using the α parameter with values from 0.75 to 0.95 with five golden ratio sections, which was determined by experimentation [21–23]. The GRSAX-SSP algorithms were executed in one of the terminals of the Ehecatl cluster in TecNM/IT Ciudad Madero, and it has the following characteristics: Intel® Xeon® processor at 2.30 GHz, Memory: 64 GB (4 × 16 GB) ddr4-2133, Linux CentOS operating system, and Fortran language.

We used the minimum energy quality values, the RMSD, and TM-score to evaluate the results, which are two metrics of the structural quality used for PFP algorithms. The RMSD is a structural measure between the native structure and the one predicted by the GRSAX-SSP and classical SA named here as GRSA0:

(a) If the RMSD has a value close to zero, the quality of the structure is considered excellent. On the contrary, the quality is worse.

(b) The TM-score is also used to measure the similarity between two structures. When the TM-score is greater than 0.5, it indicates that there is a good similarity between the two structures, and the tested one has the same fold. Otherwise, as the TM-score is lower than 0.5, the target peptide has a different fold [29].

The TM-score metrics can be calculated using the TM-align [70] (an algorithm to obtain the best structural alignment between two proteins) or in a classical formulation [29]. In this paper, we use the classical formulation of TM-score.

GRSAX-SSP algorithms use a model determined by the secondary structure, and then it is refined for obtaining a better prediction. The results are compared with the GRSAX based on ab initio that only uses the amino acid sequence as information. Figures 2–5 show average results related to energy (kcal/mol), RMSD, and TM-score for each peptide. The numbers in the x-axis, represent the instances or peptides of Table 1, and each instance is a set of torsional angles $X = [\phi_1, \Psi_1, X_1, \omega_1, \phi_2, \Psi_2, X_2, \omega_2, \ldots, \phi_n, \Psi_n, X_n, \omega_n]$ associated to each amino acid. We averaged the results of 30 executions for comparison.

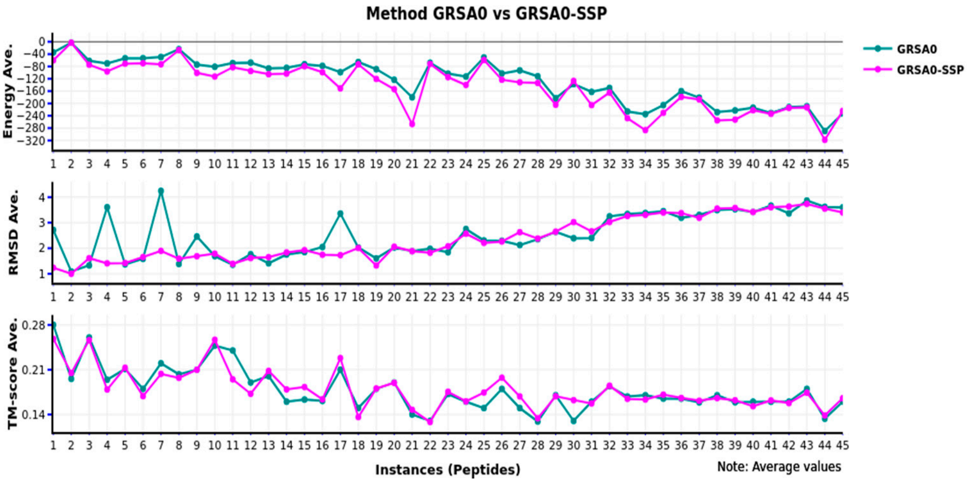


Figure 2. Comparison of GRSA0 versus GRSA0-SSP.

Figure 2 shows that GRSA0-SSP has better behavior than GRSA0 or classical SA. Note that in all the peptides, GRSA0-SSP obtained the lowest energy. In other cases, the RMSD is more stable with small instances (1–16), and in the next instances, the behavior is equal. Additionally, when we compared with TM-score, the behavior, in general, is similar. In conclusion, by implementing this methodology in GRSA0-SSP with these instances, we obtained slightly improved results.

Figure 3 presents the comparison of the GRSA1-SSP versus GRSA1 with the same metrics; we observed the behavior with the 45 instances evaluated. In terms of energy, RMSD, and TM-score, the performance of GRSA1-SSP is equivalent to GRSA1.

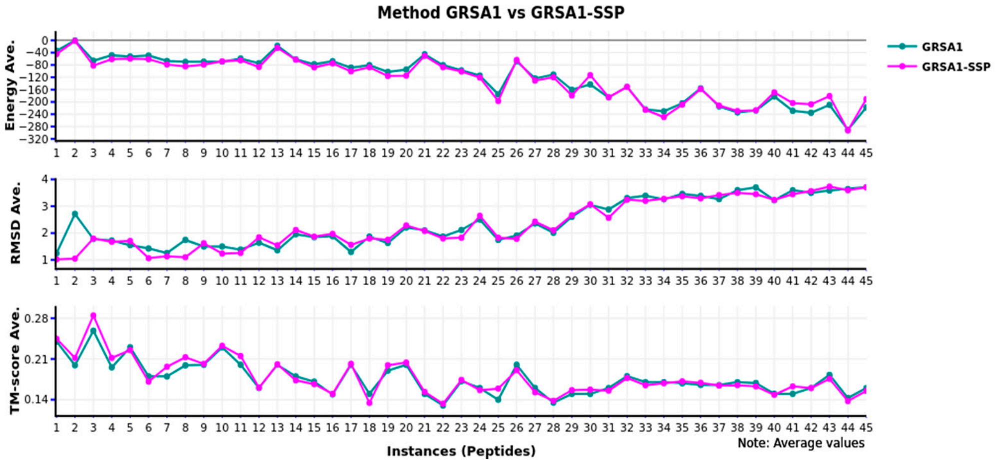


Figure 3. Comparison of GRSA1 versus GRSA1-SSP.

Figure 4 shows the behavior of GRSAE-SSP, and we compared it with the original GRSAE algorithm. In this figure, we can appreciate that the results are equivalent in all cases when energy, RMSD, and TM-score are used for comparison.

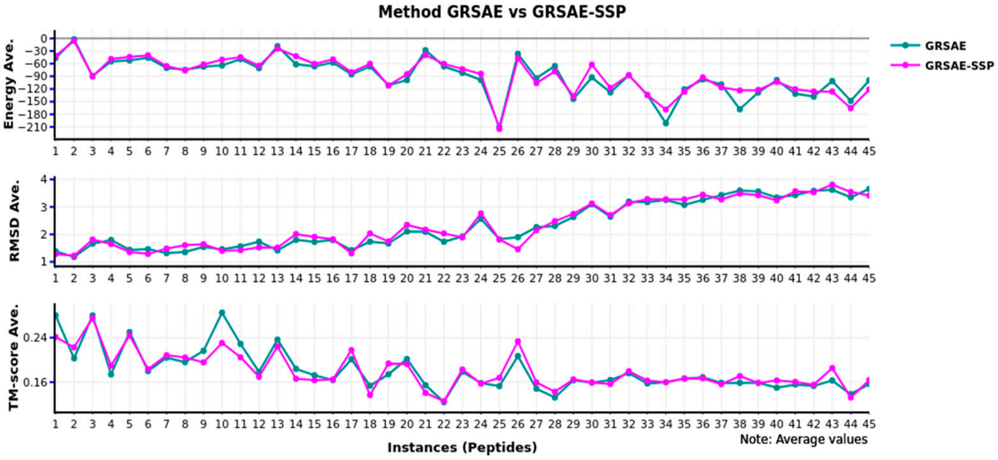


Figure 4. Comparison of GRSAE versus GRSAE-SSP.

In Figure 5, we present the comparison of GRSA2 versus GRSA2-SSP. Note that the results obtained in every instance are very remarkable, and the superiority of GRSA2-SSP uses the metrics of energy, RMSD, and TM-Score. In this case, we applied the methodology GRSA-SSP to improve the behavior of the classical GRSA2 algorithm.

Finally, in Figure 6, we present the comparison of the GRSAX-SSP family algorithms. We observe that GRSA2-SSP has the best values in several instances against the other algorithms, being higher than the others. Therefore, the best behavior of the algorithms with secondary structure prediction is GRSA2-SSP.

Furthermore, Figure 7 presents the computational time of the GRSAX-SSP family algorithms. The GRSA2-SSP has the best behavior in time with low values in most of the instances compared to the other algorithms.

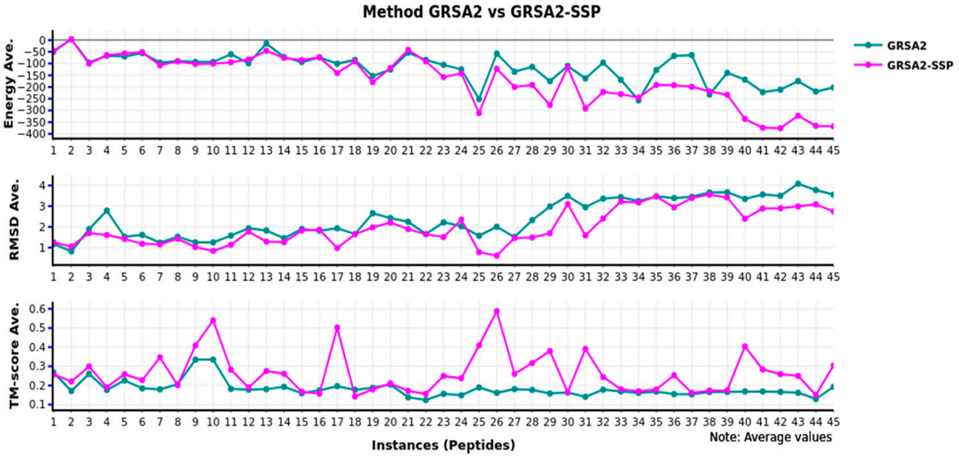


Figure 5. Comparison of GRSA2 versus GRSA2-SSP.

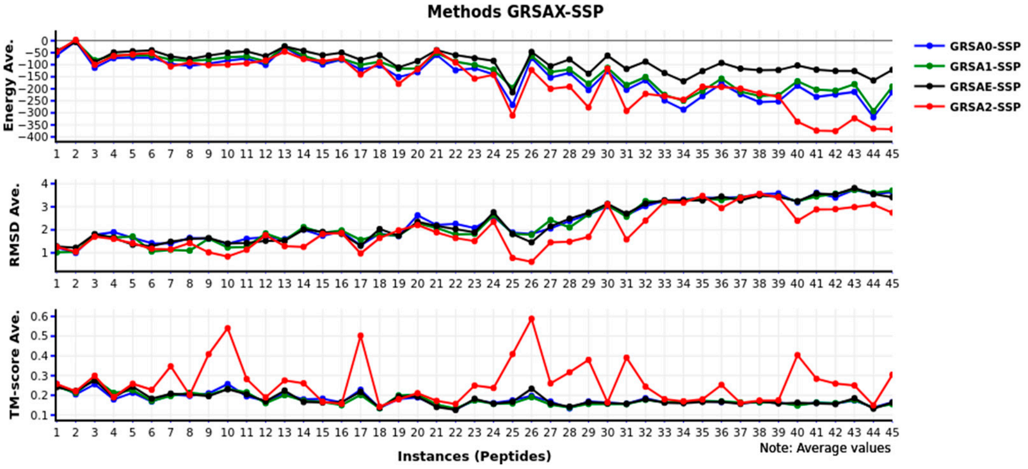


Figure 6. Comparison of GRSAX-SSP algorithms.

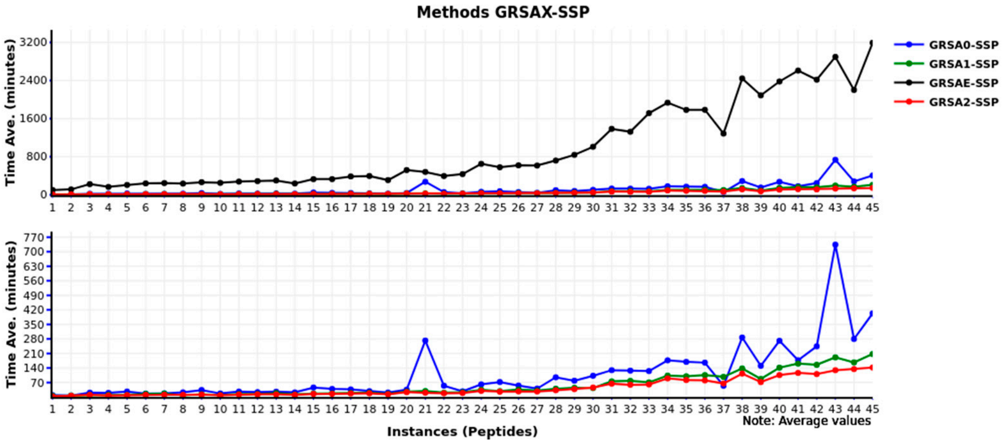


Figure 7. Comparison of the average time of the 30 execution of GRSAX-SSP algorithms.

Table 2 presents the results obtained by GRSA2-SSP. For each instance, we show the best TM-score and their RMSD. Additionally, we calculated the average of the RMSD and TM-score for the five best predictions. Complementing the results, we determined the standard deviation (std) of the RMSD and TM-score for the five best predictions and included the best type of secondary structure: A (mainly alpha), B (mainly beta), and N (mainly none). This classification as A, B, and N is based on the secondary structure predominating in each peptide [27,68,69,71,72]. We sort Table 2 by the number of amino acids for comparing the best results obtained by GRSA2-SSP with the best algorithms of the literature. This comparison is presented in Figures 9–11.

Table 2. Results obtained by GRSA2-SSP.

N°	PDB Code	aa	SS	RMSD	RMSD Ave	RMSD std	TM^1 Best	TM^1 Ave	TM^1 std	N°	PDB Code	aa	SS	RMSD	RMSD Ave	RMSD std	TM^1 Best	TM^1 Ave	TM^1 std
1	1egs	9	N	1.47	0.728	0.737	0.411	0.3630	0.043	24	1pef	18	A	1.5	0.706	0.468	0.686	0.661	0.014
2	1uao	10	B	0.71	1.214	0.828	0.401	0.375	0.022	25	1l2y	20	A	0.77	2.268	0.914	0.258	0.243	0.008
3	1l3q	12	N	1.55	1.486	0.727	0.271	0.252	0.025	26	1du1	20	A	1.13	1.62	0.463	0.266	0.266	0.001
4	2evq	12	B	2.43	1.274	1.020	0.382	0.318	0.031	27	1pei	22	A	2.02	1.43	0.366	0.379	0.364	0.010
5	1le1	12	B	0.38	1.356	1.208	0.316	0.301	0.011	28	1wz4	23	A	2.66	2.66	0.424	0.272	0.265	0.015
6	1in3	12	A	1.07	1.054	0.341	0.395	0.387	0.007	29	1yyb	27	A	1.47	1.75	0.306	0.397	0.395	0.002
7	1eg4	13	N	1.59	1.632	0.397	0.339	0.330	0.006	30	1by0	27	A	1.16	1.44	0.217	0.413	0.408	0.003
8	1rnu	13	A	0.26	0.288	0.033	0.628	0.616	0.010	31	1t0c	31	N	2.73	3.04	0.344	0.216	0.2	0.009
9	1lcx	13	N	1.08	1.412	0.422	0.334	0.323	0.009	32	2bn6	33	A	2.17	2.33	0.22	0.329	0.319	0.010
10	3bu3	14	N	1.02	1.122	0.47	0.294	0.263	0.019	33	1wr4	36	B	3.18	3.09	0.55	0.243	0.21	0.018
11	1gjf	14	A	1.37	0.874	0.461	0.561	0.547	0.040	34	1yiu	37	B	3.01	3.17	0.455	0.221	0.202	0.011
12	1k43	14	B	2.92	1.488	0.916	0.303	0.261	0.027	35	1bhi	38	N	2.76	2.736	0.794	0.306	0.296	0.007
13	1a13	14	N	1.38	1.29	0.126	0.313	0.302	0.007	36	1i6c	39	B	4.29	3.51	0.505	0.205	0.191	0.010
14	1dep	15	A	0.98	0.762	0.352	0.641	0.603	0.023	37	1bwx	39	A	2.98	2.58	0.282	0.451	0.443	0.005
15	2bta	15	N	2.47	1.716	0.455	0.227	0.196	0.018	38	2ysh	40	B	3.21	3.46	0.493	0.243	0.222	0.016
16	1nkf	16	A	3.03	1.838	0.842	0.287	0.278	0.009	39	1wr7	41	B	3.71	3.55	0.146	0.223	0.208	0.011
17	1le3	16	B	1.02	1.25	0.77	0.224	0.215	0.007	40	2dmv	43	B	3.27	3.402	0.6	0.217	0.201	0.013
18	1pgbF	16	B	1.54	2.03	0.409	0.229	0.209	0.018	41	2p81	44	A	3.52	3.21	0.476	0.185	0.178	0.007
19	1niz	16	B	2.4	1.77	0.572	0.235	0.214	0.016	42	1f4i	45	A	3.13	3.46	0.221	0.31	0.302	0.006
20	1e0q	17	B	0.79	1.494	0.536	0.226	0.221	0.008	43	1dv0	47	A	2.65	2.94	0.437	0.303	0.283	0.011
21	1wbr	17	N	1.68	1.31	0.363	0.295	0.2716	0.016	44	1pgy	47	A	3.22	2.62	0.46	0.345	0.336	0.006
22	1rpv	17	A	0.81	0.71	0.096	0.469	0.463	0.005	45	1ify	49	A	2.56	2.77	0.4	0.311	0.297	0.008
23	1b03	18	B	3.04	2.356	0.629	0.2143	0.208	0.004	-	-	-	-	-	-	-	-	-	-

Note: PDB code (Instance), number of amino acids (aa), SS is the predominant secondary structure type: beta strand (B), alpha-helix (A) and none (N), TM^1 = TM-score.

Figure 8 shows the GRSA2-SSP algorithm performance with instances classified by secondary structure. We show that the GRSA2-SSP algorithm has the best behavior in alpha structure instances evaluated with TM-score in Figure 8a and RMSD metrics in Figure 8b. The values in Figure 8 are the best obtained using TM-score and their RMSD. In Figure 8c,d, we present the TM-score average for the five best predictions and their RMSD average.

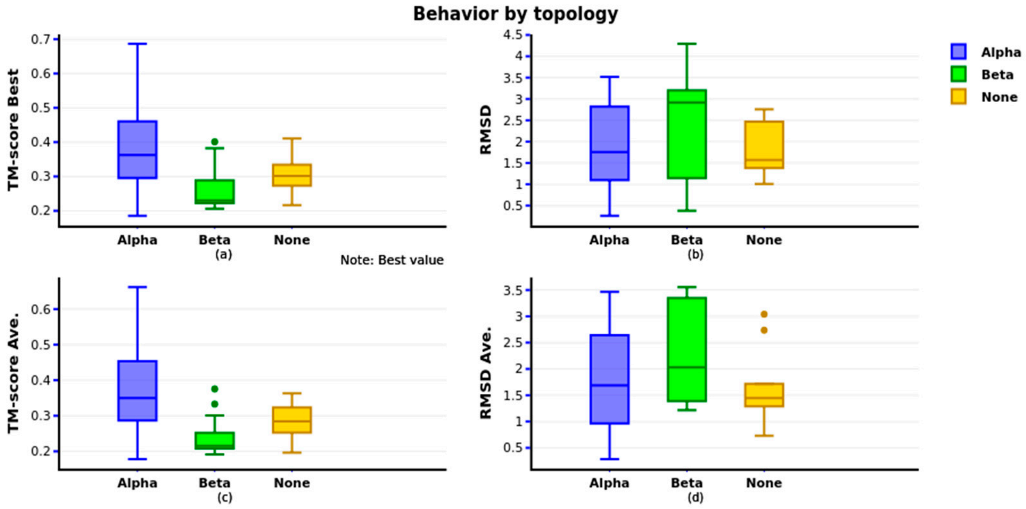


Figure 8. GRSA2-SSP according to the type of secondary structure.

In Figures 9–11, we present the behavior of the GRSA2-SSP algorithm, and we compare it with the results obtained from the approaches PEP-FOLD3, I-TASSER, QUARK, and Rosetta. We divided the dataset of Table 1 into three groups of 15 instances; groups 1, 2, and 3 have instances 1–15, 16–30, and 31–45. We compared these groups using the metrics RMSD, TM-score, GDT-TS [73], and TM-score (classical), and we present the best TM-score, the average of the five best predictions of the TM-score, and their RMSD. Additionally, we present the GDT-TS average and TM-score average.

In Figure 9, we introduced the comparison of the first group, and we observed that GRSA2-SSP behaves similarly to I-TASSER and PEP-FOLD3, but in this group of small peptides, PEP-FOLD3 is slightly better than our algorithm and I-TASSER when GDT-TS is compared (Figure 9e). Furthermore, we observed that our algorithm is competitive in this group. In this comparison, Rossetta and QUARK were not added because the minimal number of amino acids predicted are 27 and 20, respectively.

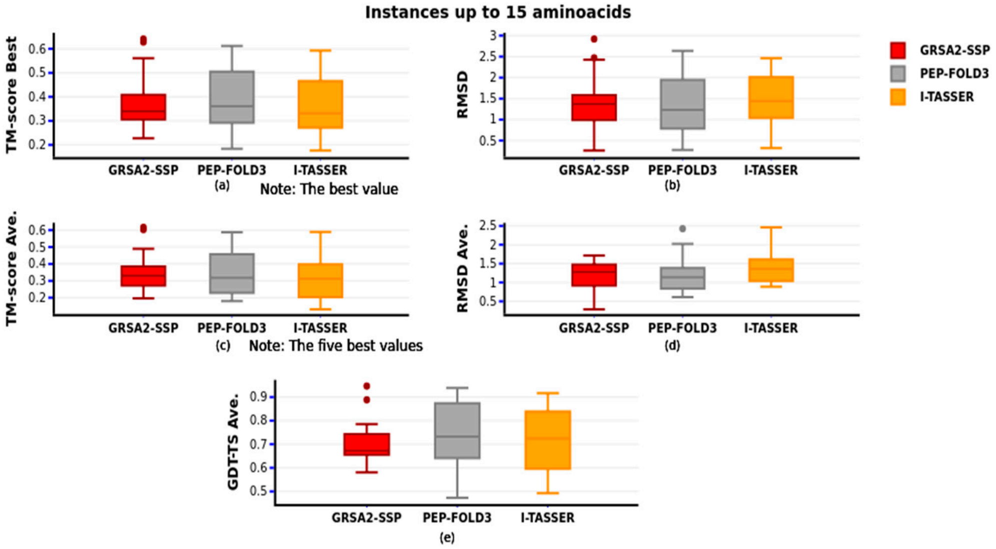


Figure 9. Comparison of GRSA2-SSP, PEP-FOLD3, and I-TASSER by RMSD (up to 15 amino acids). Figure 9 (**a**) best TM-score and (**b**) their RMSD, (**c**) TM-score average of the five best predictions, (**d**) RMSD average of the five best predictions, (**e**) GDT-TS average.

Figure 10 compares the second group of 16 to 30 amino acids with the best and the five best obtained using the TM-score metric and their RMSD, and the GDT-TS average. In this comparison, we added the second group of instances' results of QUARK; Rosetta was omitted because it is unable to predict most of the instances of this group.

In Figure 10a we observe very similar behavior among GRSA2-SSP, PEP-FOLD3, I-TASSER, and Rosetta. Note in this figure, GRSA2-SSP and PEP-FOLD3 obtain the best prediction. In Figure 10c, when the best five predictions are compared, I-TASSER obtains the best results, followed by PEPFOLD3 and GRSA2-SSP. Additionally, when the RMSD average is compared (Figure 10d), I-TASSER is the best, followed by PEP-FOLD3 and GRSA2-SSP. Finally, in Figure 10e, when GDT-TS is compared, GRSA2-SSP has a similar performance to PEP-FOLD3, I-TASSER, and QUARK. According to this figure, GRSA2-SSP and I-TASSER obtained a similar average.

Figure 11 compares the third group of 31 to 49 amino acids with the five best results obtained using the TM-score metric and their RMSD y GDT-TS. This comparison added the Rosetta approach because it can process the number of aa in this group. As we observe, the best algorithm is I-TASSER, followed by Rosetta, QUARK, PEP-FOLD3, and finally GRSA2-SSP.

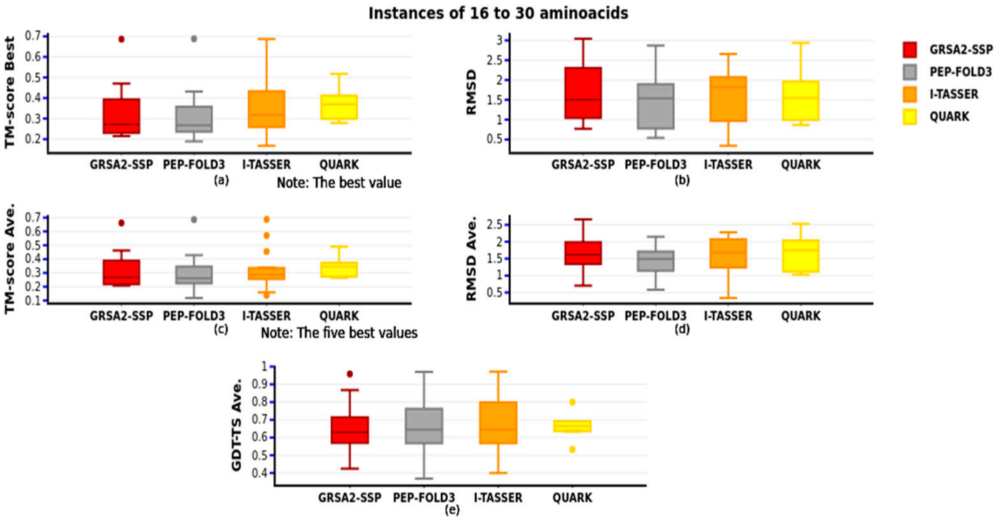


Figure 10. Comparison of GRSA2-TBM, PEP-FOLD3, and I-TASSER by TM-score (16 to 30 amino acids). Figure 10 (**a**) best TM-score, and (**b**) their RMSD, (**c**) TM-score average of the five best predictions, (**d**) RMSD of the five best predictions, and (**e**) GDT-TS average of the five best predictions.

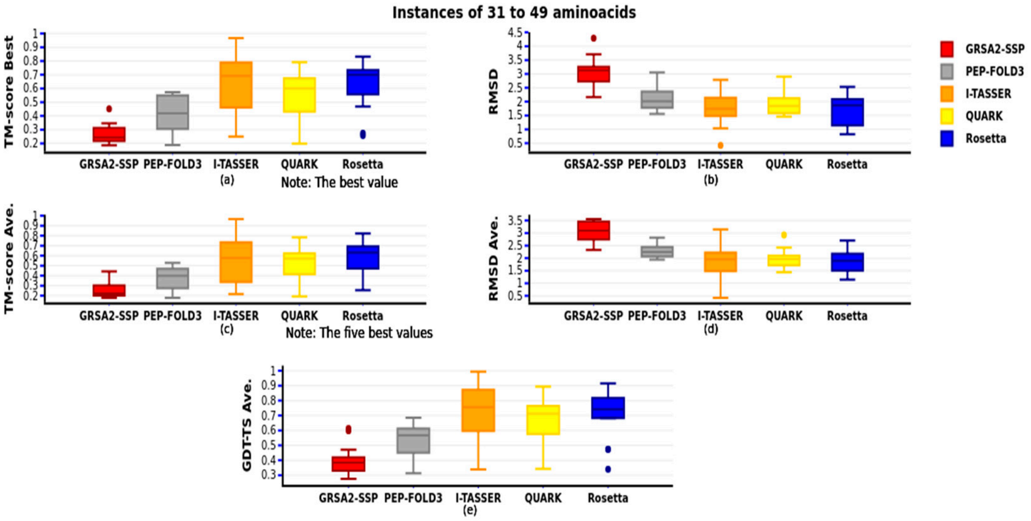


Figure 11. Comparison of GRSA2-SSP, PEP-FOLD3, I-TASSER, QUARK, and Rosetta by TM-Score (31 to 49 amino acids). Figure 11 (**a**) best TM-score, and (**b**) their RMSD, (**c**) TM-score average of the five best predictions, (**d**) RMSD average of the five best predictions, and (**e**) GDT-TS average of the five best predictions.

The 45 instances evaluated in the below experimentation show the application of the secondary structure results and refine them with the GRSAX algorithms, enhancing the performance in energy, RMSD, and TM-score. Specifically, when GRSA2-SSP is compared with PEP-FOLD3, I-TASSER, QUARK, and Rosetta, we observed that our algorithm performs well in small instances (Group 1 and 2). Nevertheless, in the largest instances, our algorithm is not the best, but it is competitive.

We carried out a second experimentation with six mini-proteins (5wll, 5lo2, 5up5, 5uoi, 2ki0, and 2kik) presented in Table 3. The mini-proteins come from the de novo protein design field [74–78]. This data set was proposed to observe the behavior of our best algorithm in these kinds of instances.

Table 3. Mini-proteins.

Instances				
N°	**PDB Code**	**aa**	**Number of Variables (Torsion Angles)**	**SS**
1	5wll	26	174	A
2	5lo2	34	192	A
3	2ki0	36	214	N
4	5up5	40	266	N
5	5uoi	43	282	A
6	2kik	48	306	A

Note: alpha-helix (A) and none (N) for secondary structure.

We applied the same evaluation of all the algorithms, as in the first experimentation, using RMSD, TM-score, and GDT-TS metrics. Table 4 shows the results of all the algorithms in this data set. Evaluating them with TM-score and GDT-TS, we observe that the best algorithms were Rosetta, I-TASSER, and GRSA2-SSP, where the number of times the best results were achieved 3, 2, and 1, respectively. Additionally, evaluating with the RMSD, the best algorithms were again Rosseta, I-TASSER, and GRSA2-SSP, but this time they obtained the best results in two instances, which were (5uoi, 2kik), (2ki0, 5up5), and (5wll, 5lo2), respectively. As a result, we can say that Rosetta is the best algorithm, followed by I-TASSER, and GRSA2-SSP.

Table 4. Average metrics results of Mini-proteins.

Approaches	**Instances**								
	5wll			**5lo2**			**2ki0**		
	RMSD	**TM-Score**	**GDT-TS**	**RMSD**	**TM-Score**	**GDT-TS**	**RMSD**	**TM-Score**	**GDT-TS**
GRSA2-SSP	0.656 *	0.642 *	0.944 *	1.504 *	0.501	0.649	2.172	0.354	0.504
PEP-FOLD3	1.074	0.526	0.892	1.922	0.532	0.769	2.422	0.466	0.697
I-TASSER	0.823	0.530	0.737	1.734	0.608	0.776	0.620 *	0.899 *	0.986 *
QUARK	0.897	0.565	0.788	1.848	0.527	0.713	2.228	0.450	0.688
Rosetta	N/A	N/A	N/A	1.552	0.694 *	0.849 *	2.146	0.460	0.710
Approaches	**Instances**								
	5up5			**5uoi**			**2kik**		
	RMSD	**TM-Score**	**GDT-TS**	**RMSD**	**TM-Score**	**GDT-TS**	**RMSD**	**TM-Score**	**GDT-TS**
GRSA2-SSP	2.234	0.277	0.403	3.194	0.192	0.340	2.756	0.339	0.508
PEP-FOLD3	2.512	0.372	0.541	2.516	0.481	0.629	2.282	0.395	0.597
I-TASSER	1.390 *	0.782 *	0.900 *	2.565	0.512	0.664	2.187	0.448	0.557
QUARK	1.880	0.614	0.778	2.022	0.633	0.777	2.028	0.462	0.627
Rosetta	1.716	0.692	0.838	1.642 *	0.753 *	0.871 *	1.968 *	0.665 *	0.785 *

Note: The asterisk (*) represents the best result in each column.

5. Conclusions

In this paper, we present the methodology GRSA-SSP for Protein Folding Problem applied to peptides. The objective of this problem is to predict the functional tridimensional protein structure. The algorithms developed with this methodology are GRSA0-SSP, GRSA1-SSP, GRSAE-SSP, and GRSA2-SSP. The main relevance of the algorithm GRSA2-SSP, developed with this methodology, is that it produces very good results in the case of peptides; specifically, it is similar or better than the algorithms Rosetta, PEP-FOLD3, QUARK, and I-TASSER for the small and medium peptides, according to the experimentation presented. The last algorithms have traditionally been among the best of the CASP competition; besides, they use modern machine learning techniques like artificial neural networks.

We compared the algorithms developed with the original algorithms GRSA0, GRSA1, GRSAE, and GRSA2; we used a data set of 45 instances for this comparison. We showed that the hybrid algorithms produced with the GRSA-SSP methodology outperform the original ones. For this comparison, we used the metrics Energy, RMSD, TM-score, and execution time. We observed that the best of all these algorithms is GRSA2-SSP formulated with the proposed methodology.

We made a second evaluation comparing the GRSA2-SSP algorithm with the best state-of-the-art algorithms (we used the same data set of 45 instances). We selected for this comparison PEP-FOLD3, I-TASSER, QUARK, and Rosetta. We used a data set of forty-five instances divided into three groups, from small to large peptides. The experimentation shows that for groups 1 and 2, GRSA2-SSP performs as well as these algorithms. We observe that for the first group PEP-FOLD3 was the best, followed by GRSA2-SSP, while in the second group, the best algorithm was I-TASSER followed by GRSA2-SSP and PEP-FOLD3. Finally, in the third group, the best algorithm was Rosseta, followed by I-TASSER. Additionally, we present an analysis of GRSA2-SSP results for each type of secondary structure, obtaining a better behavior with alpha structures.

Furthermore, we assessed GRSA2-SSP with a second data set of six instances named mini proteins. The GRSA2-SSP results were compared with PEP-FOLD3, I-TASSER, QUARK, and Rosetta. The best algorithms in this data set were Rosetta, I-TASSER, and GRSA2-SSP because the number of times the best TM-score and GDT-TS were 3, 2, and 1, respectively. However, each of the three achieved two times the first place when RMSD was evaluated. As a result, the best of these algorithms for this data set is Rosetta, followed by I-TASSER and GRSA2-SSP.

We conclude that GRSAX-SSP algorithms enhance the original GRSA algorithms. The best of them is GRSA2-SSP which achieves very good results, surpassing the best state-of-art for peptides up to thirty amino acids. Finally, we note that the main advantage of our methodology is that it is simpler than the most powerful approaches of the literature.

Author Contributions: J.F.-S. and J.P.S.-H. contributed equally to the development of this paper. Conceptualization, J.P.S.-H., D.A.S.-M. and J.F.-S.; methodology J.F.-S., D.A.S.-M., J.P.S.-H., and J.J.G.-B.; Software J.P.S.-H., D.A.S.-M. and F.G.M.-N.; validation, J.P.S.-H. and J.F.-S.; formal analysis, D.A.S.-M., F.G.M.-N., J.J.G.-B., and G.C.-V.; writing—original draft J.F.-S., J.P.S.-H., and D.A.S.-M.; writing—review and editing, J.F.-S., D.A.S.-M. and J.P.S.-H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors would like to acknowledge with appreciation and gratitude CONACYT and TecNM/Instituto Tecnológico de Ciudad Madero. Also, we acknowledge Laboratorio Nacional de Tecnologías de la Información (LaNTI) for the access to the cluster.

Conflicts of Interest: The authors declare that they have no competing interests.

References

1. Uhlig, T.; Kyprianou, T.; Martinelli, F.G.; Oppici, C.A.; Heiligers, D.; Hills, D.; Calvo, X.R.; Verhaert, P. The emergence of peptides in the pharmaceutical business: From exploration to exploitation. *EuPA Open Proteom.* **2014**, *4*, 58–69. [CrossRef]
2. Agyei, D.; Danquah, M.K. Industrial-scale manufacturing of pharmaceutical-grade bioactive peptides. *Biotechnol. Adv.* **2011**, *29*, 272–277. [CrossRef]
3. Patel, L.N.; Zaro, J.L.; Shen, W.-C. Cell Penetrating Peptides: Intracellular Pathways and Pharmaceutical Perspectives. *Pharm. Res.* **2007**, *24*, 1977–1992. [CrossRef]
4. Danquah, M.; Agyei, D. Pharmaceutical applications of bioactive peptides. *OA Biotechnol.* **2012**, *1*. [CrossRef]
5. Fosgerau, K.; Hoffmann, T. Peptide therapeutics: Current status and future directions. *Drug Discov. Today* **2015**, *20*, 122–128. [CrossRef] [PubMed]

6. Vetter, I.; Davis, J.L.; Rash, L.D.; Anangi, R.; Mobli, M.; Alewood, P.F.; Lewis, R.J.; King, G.F. Venomics: A new paradigm for natural products-based drug discovery. *Amino Acids* **2010**, *40*, 15–28. [CrossRef] [PubMed]
7. Craik, D.J.; Fairlie, D.P.; Liras, S.; Price, D. The Future of Peptide-based Drugs. *Chem. Biol. Drug Des.* **2013**, *81*, 136–147. [CrossRef] [PubMed]
8. Stalmach, A.; Johnsson, H.; McInnes, I.B.; Husi, H.; Klein, J.; Dakna, M. Identification of urinary peptide biomarkers associated with rheumatoid arthritis. *PLoS ONE* **2014**, *9*, e104625.
9. Gautam, A.; Kapoor, P.; Chaudhary, K.; Kumar, R.; Raghava, G.P. Tumor homingpeptides as molecular probes for cancer therapeutics, diagnostics and theranostics. *Curr. Med. Chem.* **2014**, *21*, 2367–2391. [CrossRef] [PubMed]
10. Li, Z.J.; Cho, C.H. Peptides as targeting probes against tumor vasculature for diagnosis and drug delivery. *J. Transl. Med.* **2012**, *10* (Suppl. S1). [CrossRef]
11. Lau, J.L.; Dunn, M.K. Therapeutic peptides: Historical perspectives, current development trends, and future directions. *Bioorgan. Med. Chem.* **2018**, *26*, 2700–2707. [CrossRef] [PubMed]
12. Vlieghe, P.; Lisowski, V.; Martinez, J.; Khrestchatisky, M. Synthetic therapeutic peptides: Science and market. *Drug Discov. Today* **2010**, *15*, 40–56. [CrossRef] [PubMed]
13. Kendrew, J.C.; Bodo, G.; Dintzis, H.M.; Parrish, R.G.; Wyckoff, H.; Phillips, D.C. A three-dimensional model of the myoglobin molecule obtained by X-ray analysis. *Nature* **1958**, *181*, 662–666. [CrossRef] [PubMed]
14. Perutz, M.F.; Rossmann, M.G.; Cullis, A.F.; Muirhead, H.I.L.A.R.Y.; Will, G.; North, A.C.T. Structure of hemoglobin. *Brookhaven. Symp. Biol.* **1960**, *13*, 165–183. [PubMed]
15. Anfinsen, C.B. Principles that Govern the Folding of Protein Chains. *Science* **1973**, *181*, 223–230. [CrossRef]
16. Hart, W.E.; Istrail, S. Robust Proofs of NP-Hardness for Protein Folding: General Lattices and Energy Potentials. *J. Comput. Biol.* **1997**, *4*, 1–22. [CrossRef] [PubMed]
17. Levinthal, C. Are There Pathways for Protein Folding. *J. Chim. Phys.* **1968**, *65*, 44–45. [CrossRef]
18. Li, Z.; Scheraga, H.A. Monte Carlo-minimization approach to the multiple-minima problem in protein folding. *Proc. Natl. Acad. Sci. USA* **1987**, *84*, 6611–6615. [CrossRef] [PubMed]
19. Morales, L.B.; Garduño-Juárez, R.; Romero, D. Applications of Simulated Annealing to the Multiple-Minima Problem in Small Peptides. *J. Biomol. Struct. Dyn.* **1991**, *8*, 721–735. [CrossRef]
20. Frausto, J.; Román, E.F.; Romero, D.; Soberon, X.; Liñán, E. Analytically Tuned Simulated Annealing Applied to the Protein Folding Problem. In Proceedings of the 7th International Conference on Computational Science, Beijing, China, 27–30 May 2007; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4488, pp. 370–377.
21. Frausto, J.; Sánchez, J.P.; Sánchez, M.; García, E.L. Golden Ratio Simulated Annealing for Protein Folding Problem. *Int. J. Comput. Methods* **2015**, *12*, 1550037. [CrossRef]
22. Maldonado, F.; Frausto, J.; Sánchez, J.; González, J.; Liñán, E.; Castilla, G. Evolutionary GRSA for Protein Structure Prediction. *Int. J. Comb. Optim. Probl. Inform.* **2016**, *7*, 75–86.
23. Frausto, J.; Sánchez, J.P.; Maldonado, F.; González, J.J. GRSA Enhanced for Protein Folding Problem in the Case of Peptides. *Axioms* **2019**, *8*, 136. [CrossRef]
24. Hiranuma, N.; Park, H.; Baek, M.; Anishchenko, I.; Dauparas, J.; Baker, D. Improved protein structure refinement guided by deep learning based accuracy estimation. *Nat. Commun.* **2021**, *12*, 1–11. [CrossRef]
25. Xu, D.; Zhang, Y. Toward optimal fragment generations for ab initio protein structure assembly. *Proteins* **2012**, *81*, 229–239. [CrossRef] [PubMed]
26. Wang, D.; Geng, L.; Zhao, Y.-J.; Yang, Y.; Huang, Y.; Zhang, Y.; Shen, H.-B. Artificial intelligence-based multi-objective optimization protocol for protein structure refinement. *Bioinformatics* **2020**, *36*, 437–448. [CrossRef] [PubMed]
27. Senior, A.W.; Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Žídek, A.; Nelson, A.W.R.; Bridgland, A.; et al. Protein structure prediction using multiple deep neural networks in the 13th Critical Assessment of Protein Structure Prediction (CASP13). *Proteins Struct. Funct. Bioinform.* **2019**, *87*, 1141–1148. [CrossRef] [PubMed]
28. Lamiable, A.; Thévenet, P.; Rey, J.; Vavrusa, M.; Derreumaux, P.; Tufféry, P. PEP-FOLD3: Faster de Novo Structure Prediction for Linear Peptides in Solution and in Complex. *Nucleic Acids Res.* **2016**, *44*, W449–W454. [CrossRef]
29. Zhang, Y.; Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins* **2004**, *57*, 702–710. [CrossRef]
30. Yang, J.; Yan, R.; Roy, A.; Xu, D.; Poisson, J.; Zhang, Y. The I-TASSER Suite: Protein structure and function prediction. *Nat. Methods* **2015**, *12*, 7–8. [CrossRef] [PubMed]
31. Yang, J.; Zhang, Y. I-TASSER server: New development for protein structure and function predictions. *Nucleic Acids Res.* **2015**, *43*, W174–W181. [CrossRef] [PubMed]
32. Rohl, C.A.; Strauss, C.E.; Misura, K.M.; Baker, D. Protein Structure Prediction Using Rosetta. *Oncogene Tech.* **2004**, *383*, 66–93. [CrossRef]
33. Xu, D.; Zhang, Y. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins* **2012**, *80*, 1715–1735. [CrossRef]
34. Dill, K.A.; Maccallum, J.L. The Protein-Folding Problem, 50 Years On. *Science* **2012**, *338*, 1042–1046. [CrossRef] [PubMed]
35. Dill, K.A. Dominant forces in protein folding. *Biochemistry* **1990**, *29*, 7133–7155. [CrossRef] [PubMed]
36. Ponder, J.W.; Case, D.A. Force Fields for Protein Simulations. *Accessory Fold. Proteins* **2003**, *66*, 27–85. [CrossRef]

37. Brooks, B.R.; Bruccoleri, R.E.; Olafson, B.D.; States, D.J.; Swaminathan, S.; Karplus, M. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.* **1983**, *4*, 187–217. [CrossRef]
38. Momany, F.A.; McGuire, R.F.; Burgess, A.W.; Scheraga, H.A. Energy parameters in polypeptides. VII. Geometric parameters, partial atomic charges, nonbonded interactions, hydrogen bond interactions, and intrinsic torsional potentials for the naturally occurring amino acids. *J. Phys. Chem.* **1975**, *79*, 2361–2381. [CrossRef]
39. Eisenmenger, F.; Hansmann, U.H.; Hayryan, S.; Hu, C.-K. [SMMP] A modern package for simulation of proteins. *Comput. Phys. Commun.* **2001**, *138*, 192–212. [CrossRef]
40. Jiang, P.; Xu, J. RaptorX: Exploiting structure information for protein alignment by statistical inference. *Proteins* **2011**, *79*, 161–171.
41. Zhou, H.; Pandit, S.B.; Skolnick, J. Performance of the Pro-sp3-TASSER server in CASP8. *Proteins* **2009**, *77*, 123–127. [CrossRef]
42. Konstantin, A.; Lorenza, B.; Jürgen, K.; Torsten, S. The SWISS-MODEL workspace: A web-based environment for protein structure homology modelling. *Bioinformatics* **2006**, *22*, 195–201.
43. Schmitt, S.; Kuhn, D.; Klebe, G. A New Method to Detect Related Function among Proteins Independent of Sequence and Fold Homology. *J. Mol. Biol.* **2002**, *323*, 387–406. [CrossRef]
44. Lemer, C.M.-R.; Rooman, M.J.; Wodak, S.J. Protein structure prediction by threading methods: Evaluation of current techniques. *Proteins* **1995**, *23*, 337–355. [CrossRef]
45. Dorn, M.; e Silva, M.B.; Buriol, L.S.; Lamb, L.C. Three-dimensional protein structure prediction: Methods and computational strategies. *Comput. Biol. Chem.* **2014**, *53*, 251–276. [CrossRef]
46. Zhang, Y. Interplay of I-TASSER and QUARK for template-based and ab initio protein structure prediction in CASP10. *Proteins* **2013**, *82*, 175–187. [CrossRef]
47. Bhardwaj, G.; Mulligan, V.K.; Bahl, C.D.; Gilmore, J.M.; Harvey, P.J.; Cheneval, O.; Buchko, G.W.; Pulavarti, S.V.S.R.K.; Kaas, Q.; Eletsky, A.; et al. Accurate de novo design of hyperstable constrained peptides. *Nature* **2016**, *538*, 329–335. [CrossRef] [PubMed]
48. Harada, R.; Nakamura, T.; Shigeta, Y. A Fast Convergent Simulated Annealing Algorithm for Protein-Folding: Simulated Annealing Outlier FLOODing (SA-OFLOOD) Method. *Bull. Chem. Soc. Jpn.* **2016**, *89*, 1361–1367. [CrossRef]
49. Zhang, L.; Ma, H.; Qian, W.; Li, H. Protein structure optimization using improved simulated annealing algorithm on a three-dimensional AB off-lattice model. *Comput. Biol. Chem.* **2020**, *85*, 107237. [CrossRef] [PubMed]
50. Zhang, L.; Ma, H.; Qian, W.; Li, H. Sequence-based protein structure optimization using enhanced simulated annealing al-gorithm on a coarse-grained model. *J. Mol. Model.* **2020**, *26*, 1–13. [CrossRef]
51. Mitra, P.; Shultis, D.; Brender, J.R.; Czajka, J.; Marsh, D.; Gray, F.; Cierpicki, T.; Zhang, Y. An Evolution-Based Approach to De Novo Protein Design and Case Study on Mycobacterium tuberculosis. *PLoS Comput. Biol.* **2013**, *9*, e1003298. [CrossRef]
52. Banerjee, A.; Pal, K.; Mitra, P. An evolutionary profile guided greedy parallel replica-exchange Monte Carlo search algorithm for rapid convergence in protein design. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2019**, *18*, 489–499. [CrossRef] [PubMed]
53. Wu, S.; Zhang, Y. LOMETS: A local meta-threading-server for protein structure prediction. *Nucleic Acids Res.* **2007**, *35*, 3375–3382. [CrossRef]
54. Zheng, W.; Zhang, C.; Wuyun, Q.; Pearce, R.; Li, Y.; Zhang, Y. LOMETS2: Improved meta-threading server for fold-recognition and structure-based function annotation for distant-homology proteins. *Nucleic Acids Res.* **2019**, *47*, W429–W436. [CrossRef] [PubMed]
55. Yang, J.; Roy, A.; Zhang, Y. BioLiP: A semi-manually curated database for biologically relevant ligand–protein interactions. *Nucleic Acids Res.* **2012**, *41*, D1096–D1103. [CrossRef]
56. De Oliveira, S.; Law, E.C.; Shi, J.; Deane, C.M. Sequential search leads to faster, more efficient fragment-based de novo protein structure prediction. *Bioinformatics* **2017**, *34*, 1132–1140. [CrossRef] [PubMed]
57. Jones, D.T. Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.* **1999**, *292*, 195–202. [CrossRef] [PubMed]
58. Faraggi, E.; Kloczkowski, A. Accurate Prediction of One-Dimensional Protein Structure Features Using SPINE-X. In *Prediction of Protein Secondary Structure*; Methods in Molecular Biology; Humana Press: New York, NY, USA, 2017; Volume 1484, pp. 45–53.
59. Jones, D.T.; Singh, T.; Kosciolek, T.; Tetchner, S. MetaPSICOV: Combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. *Bioinformatics* **2015**, *31*, 999–1006. [CrossRef] [PubMed]
60. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [CrossRef]
61. Černý, V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *J. Optim. Theory Appl.* **1985**, *45*, 41–51. [CrossRef]
62. Frausto, J.; Martinez, F. Golden Ratio Annealing for Satisfiability Problems Using Dynamically Cooling Schemes. In *Foundations of Intelligent Systems*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 4994, pp. 215–224.
63. Frausto, J.; Martinez, F. Golden annealing method for job shop scheduling problem. In *MACMESE'08: Proceedings of the 10th WSEAS International Conference on Mathematical and Computational Methods in Science and Engineering*; World Scientific and Engineering Academy and Society (WSEAS): Stevens Point, WI, USA, 2008; pp. 374–379.
64. Frausto, J.; Liñán, E.; Sánchez, J.P.; González, J.J.; González, C.; Castilla, G. Multiphase Simulated Annealing Based on Boltzmann and Bose–Einstein Distribution Applied to Protein Folding Problem. *Adv. Bioinform.* **2016**, *2016*, 7357123.
65. Martinez, F.; Frausto, J. A simulated annealing algorithm for the satisfiability problem using dynamic Markov chains with linear regression equilibrium. Simulated Annealing. *InTechOpen* **2012**, *21*, 281–285.
66. Lam, A.Y.S.; Li, V.O.K. Chemical Reaction Optimization: A tutorial. *Memetic Comput.* **2012**, *4*, 3–17. [CrossRef]

67. Altschul, S.F.; Madden, T.L.; Schäffer, A.A.; Zhang, J.; Zhang, Z.; Miller, W.; Lipman, D.J. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.* **1997**, *25*, 3389–3402. [CrossRef]
68. Maupetit, J.; Derreumaux, P.; Tuffery, P. PEP-FOLD: An online resource for de novo peptide structure prediction. *Nucleic Acids Res.* **2009**, *37* (Suppl. S2), W498–W503. [CrossRef]
69. Shen, Y.; Maupetit, J.; Derreumaux, P.; Tufféry, P. Improved PEP-FOLD approach for peptide and miniprotein structure pre-diction. *J. Chem. Theory Comput.* **2014**, *10*, 4745–4758. [CrossRef] [PubMed]
70. Zhang, Y.; Skolnick, J. TM-align: A protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.* **2005**, *33*, 2302–2309. [CrossRef] [PubMed]
71. Munte, C.E.; Vilela, L.; Kalbitzer, H.R.; Garratt, R.C. Solution structure of human proinsulin C-peptide. *FEBS J.* **2005**, *272*, 4284–4293. [CrossRef]
72. Luitz, M.P.; Bomblies, R.; Zacharias, M. Comparative Molecular Dynamics Analysis of RNase-S Complex Formation. *Biophys. J.* **2017**, *113*, 1466–1474. [CrossRef]
73. Zemla, A.; Moult, J.; Fidelis, K. Processing and evaluation of predictions in CASP4. *Proteins* **2001**, *45*, 13–21. [CrossRef]
74. Lombardi, A.; Pirro, F.; Maglio, O.; Chino, M.; DeGrado, W.F. De Novo Design of Four-Helix Bundle Metalloproteins: One Scaffold, Diverse Reactivities. *Accounts Chem. Res.* **2019**, *52*, 1148–1159. [CrossRef]
75. Liang, H.; Chen, H.; Fan, K.; Wei, P.; Guo, X.; Jin, C.; Zeng, C.; Tang, C.; Lai, L. De novo design of a beta alpha beta motif. *Angew. Chem. Int. Ed. Engl.* **2009**, *48*, 3301–3303. [CrossRef] [PubMed]
76. Baker, E.G.; Bartlett, G.J.; Goff, K.L.P.; Woolfson, D.N. Miniprotein Design: Past, Present, and Prospects. *Accounts Chem. Res.* **2017**, *50*, 2085–2092. [CrossRef] [PubMed]
77. Rocklin, G.J.; Chidyausiku, T.M.; Goreshnik, I.; Ford, A.; Houliston, S.; Lemak, A.; Carter, L.; Ravichandran, R.; Mulligan, V.K.; Chevalier, A.; et al. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science* **2017**, *357*, 168–175. [CrossRef] [PubMed]
78. Zhang, S.-Q.; Chino, M.; Liu, L.; Tang, Y.; Hu, X.; DeGrado, W.F.; Lombardi, A. De Novo Design of Tetranuclear Transition Metal Clusters Stabilized by Hydrogen-Bonded Networks in Helical Bundles. *J. Am. Chem. Soc.* **2018**, *140*, 1294–1304. [CrossRef] [PubMed]