

Software/Web server Article

PS-GO parametric protein search engine

Yanlin Mi ^{a,b}, Stefan-Bogdan Marcu ^a, Sabin Tabircă ^{a,c}, Venkata V.B. Yallapragada ^{d,*}^a School of Computer Science and Information Technology, University College Cork, Cork, Ireland^b SFI Centre for Research Training in Artificial Intelligence, University College Cork, Cork, Ireland^c Faculty of Mathematics and Informatics, Transylvania University of Brasov, Brasov, Romania^d Centre for Advanced Photonics and Process Analytics, Munster Technological University, Cork, Ireland

ARTICLE INFO

Keywords:

Protein search engine
 Protein topology
 Parametric protein search
 Parametric protein design
 Protein parameters database

ABSTRACT

With the explosive growth of protein-related data, we are confronted with a critical scientific inquiry: How can we effectively retrieve, compare, and profoundly comprehend these protein structures to maximize the utilization of such data resources? PS-GO, a parametric protein search engine, has been specifically designed and developed to maximize the utilization of the rapidly growing volume of protein-related data. This innovative tool addresses the critical need for effective retrieval, comparison, and deep understanding of protein structures. By integrating computational biology, bioinformatics, and data science, PS-GO is capable of managing large-scale data and accurately predicting and comparing protein structures and functions.

The engine is built upon the concept of parametric protein design, a computer-aided method that adjusts and optimizes protein structures and sequences to achieve desired biological functions and structural stability. PS-GO utilizes key parameters such as amino acid sequence, side chain angle, and solvent accessibility, which have a significant influence on protein structure and function. Additionally, PS-GO leverages computable parameters, derived computationally, which are crucial for understanding and predicting protein behavior.

The development of PS-GO underscores the potential of parametric protein design in a variety of applications, including enhancing enzyme activity, improving antibody affinity, and designing novel functional proteins. This advancement not only provides a robust theoretical foundation for the field of protein engineering and biotechnology but also offers practical guidelines for future progress in this domain.

1. Introduction

1.1. Topology of protein landscape

Within the field of life sciences, deciphering and understanding the intricacy of proteins remain substantial challenges. Proteins, as the fundamental functional units in biological systems, exhibit functionalities determined primarily by their structures. With the explosive growth of protein-related data, we are confronted with a critical scientific inquiry: **How can we effectively retrieve, compare, and profoundly comprehend these protein structures to maximize the utilization of such data resources?** The advancement of Internet technology offers possible resolution pathways. Take Google's search engine as an example; it spearheaded a significant transformation in the information retrieval field. Google's search engine, through its innovative PageRank algorithm, successfully tackled the problem of swiftly locating relevant information among a vast amount of web pages [1]. PageRank algo-

ri-
 rithm, which is based on the hyperlink structure of web pages to assess their significance, enabled the effective ranking of web pages. Consequently, users can quickly obtain the most relevant search results. This structure-based processing approach significantly elevated the efficiency of information retrieval. Inspired by this strategy, we need to introduce similar methods in the field of biosciences to process and interpret the massive volume of protein data. The ideal solution should profoundly reveal the topological structure of proteins, accurately infer their functional properties, and perform precise searches and comparisons within large-scale protein datasets. To achieve this goal, we require a tool that integrates computational biology, bioinformatics, and data science, capable of handling big data and accurately predicting and comparing the structure and function of proteins.

Protein topology, as the arrangement and spatial relationship of secondary structure elements, plays a crucial role in understanding key aspects such as protein folding, stability, function, and evolution, as shown in Fig. 1. There is a close connection between protein structural

* Corresponding author.

E-mail address: VVB.Yallapragada@mtu.ie (V.V.B. Yallapragada).<https://doi.org/10.1016/j.csbj.2024.04.003>

Received 11 January 2024; Received in revised form 1 April 2024; Accepted 1 April 2024

Available online 8 April 2024

2001-0370/© 2024 The Authors. Published by Elsevier B.V. on behalf of Research Network of Computational and Structural Biotechnology. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

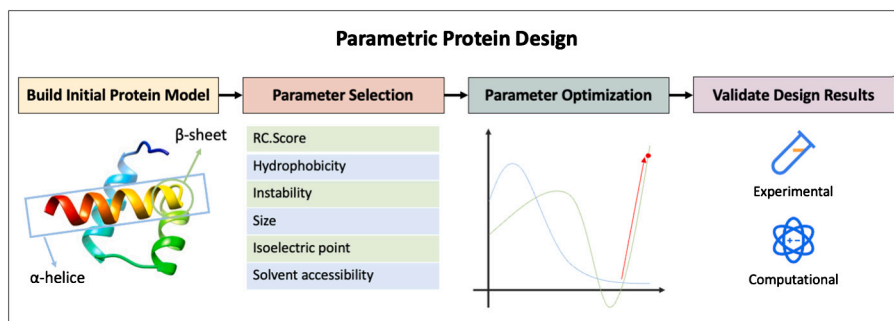


Fig. 1. The steps in parametric protein design.

topology and its biological function. A specific structural topology of the protein determines key functional features such as active sites, subunit interaction interfaces, and molecular recognition patterns. Studying the structural topological features of proteins can better predict protein function and reveal functional and structural similarities between different protein families [2]. Furthermore, studying protein structural topology can help to reveal the relationship between protein structure and function, and in turn, apply this knowledge to protein design. The goal of protein design is to create proteins with desired functions, stability, and solubility. During the design process, researchers often use computational methods to optimize protein sequences to achieve the desired structural topology.

However, due to the diversity of amino acid species and the endless possibilities of arrangement, the number of combinations of protein structures is enormous. In this vast search space, many combinations of amino acids do not produce stable protein structures or have biological functions. Therefore, understanding the quantitative differences between the available mathematical combinations and the actual possibilities is important for parametric protein searches to explore protein sequence space more efficiently and discover proteins with the desired properties. In this regard, algorithms such as machine learning and metaheuristics can help to filter out protein sequences with potential biological functions, thereby reducing the complexity of the search space. **In practice, the combination of existing knowledge of protein structures, parametric models and efficient search strategies can be used to mine proteins with specific functions, stability and solubility with limited computational resources, providing strong support for protein design and optimization.**

1.2. Parametric protein design

Parametric protein design is a computer-aided protein design method, with the primary goal of adjusting and optimizing parameters of protein structures and sequences to achieve desired biological functions and structural stability [3]. To achieve this goal, the right combination of parameters needs to be found to give the protein the desired function and properties. The key steps in parametric protein design are shown in Fig. 1. The two most important steps in the design process are parameter selection and parameter optimization. Parameter selection requires the selection of parameters related to protein structure and function, such as amino acid sequence, side chain angle, and solvent accessibility, according to the design objectives [4]; parameter optimization can be classified according to the optimization strategy and search strategy. Optimization strategies mainly include energy minimization and probabilistic search. Energy minimization methods focus on optimizing the energy of protein structures to improve structural stability and function; while probabilistic search methods employ random sampling and acceptance or rejection criteria to search within the parameter space. Search strategies include genetic algorithms, Monte Carlo simulations, and more. As a heuristic search method, genetic algorithms have high search efficiency and flexibility; Monte Carlo simulations are suitable for handling complex energy landscapes and

parameter relationships [5]. Parametric protein design, and in particular the parameter selection and parameter optimization steps involved, is provided by theoretical underpinnings and practical guidance from protein structure topology studies.

The study of protein structure topology can provide a basis for parameter selection in the parametric protein design process. For example, it has been found that the interactions between protein secondary structure elements and the mechanism of tertiary structure formation are crucial for protein function [6]. In addition, understanding the topological characteristics of protein structures helps to better grasp the impact of parameter tuning on protein structure and function. For example, protein structural topology studies have revealed certain general principles of structural stability, such as hydrogen bonding networks and hydrophobic cores [7]. Furthermore, Protein structure topology studies have revealed the phenomenon of “protein folding topology” [8], which provides valuable clues for parametric protein design. By changing parameters such as amino acid sequence or side chain angle, proteins can be guided to form specific topological patterns to achieve function and stability [3].

Understanding the relationship between structure, parameters, and function is crucial in parametric protein design [4]. Parameters such as amino acid sequence, side chain angle, and solvent accessibility have a significant impact on protein structure and function [5]. Firstly, the amino acid sequence determines the primary structure of a protein and indirectly influences its secondary, tertiary, and quaternary structure. By adjusting the amino acid sequence, properties such as the folding stability, functional activity, and affinity of a protein can be altered [2]. In addition, the side chain angle determines the spatial arrangement of the amino acid side chains during protein folding and affects the steric conformation and stability of the protein. Solvent accessibility affects the interaction of the protein with its surroundings, including binding to ligands, substrates, or other proteins. By optimizing the side chain angle and solvent accessibility, the affinity of a protein to a specific ligand can be improved, allowing for specific antibody design [5]. Parametric protein design has played a key role in many successful examples of protein design, including enzyme activity improvement, antibody affinity enhancement, and the design of novel functional proteins [4]. These success cases provide ample evidence of the reliability and effectiveness of parametric protein design in achieving specific goals.

Computable parameters are a class of parameters related to protein structure and function that can be derived computationally. These parameters are essential for understanding and predicting the behavior of proteins, as they can directly affect the structure, stability, hydrophobicity, and other properties of proteins, as shown in Table 1.

In areas such as protein design, drug development, and bioengineering, the use of computable parameters helps to optimize protein function and enhance its applications [9]. Researchers have used computer simulations, data mining, and bioinformatics methods to calculate and predict these parameters, thereby making connections between protein sequence, structure, and function. This provides an innovative and powerful direction for the modification and optimization of proteins.

Table 1
Computable parameters and description.

<p>RC Score: It is a parameter that describes the importance of amino acid residue interactions. The higher the score, the more critical the amino acid residues are in the protein structure. Calculating the RC score helps to optimize the stability and folding ability of the protein and is essential for improving thermal stability and resistance to protease degradation.</p> <p>Hydrophobicity: It is a parameter that measures the hydrophilicity of amino acid residues and is usually calculated using molecular dynamics simulations and computational chemistry. Optimizing the hydrophobicity of proteins enhances their solubility, stability, and interaction with other molecules.</p> <p>Instability: It refers to the half-life of a protein molecule in vitro and is closely related to factors such as amino acid sequence, structure, and hydrophobicity. Calculating the instability of amino acid residues helps to predict the stability and susceptibility to degradation of the whole protein molecule, which is of great importance for drug development and protein engineering.</p> <p>Size: Protein size refers to the total number of amino acid residues in its molecule. Size can have an impact on the folding rate, stability, and function of a protein. Calculating protein size can help predict its properties and function and provide a basis for optimizing protein structure.</p> <p>Isoelectric: The isoelectric point is the pH value of a protein molecule when the sum of its charges is zero. Calculating the isoelectric point helps to predict the charge state of a protein in different pH environments and further speculates on its role in the organism. In addition, calculating the isoelectric point also helps to optimize the charge distribution of proteins, thereby improving their stability and biological activity in various environments.</p> <p>Solvent accessibility: It describes the extent to which an amino acid residue or part of it in a protein molecule is exposed in solution. It is closely related to protein folding, stability, and molecular interactions. Calculating the solvent accessibility of amino acid residues helps to predict the stability and biological function of proteins while revealing the structural features of proteins and the role of amino acid residues in the structure.</p>

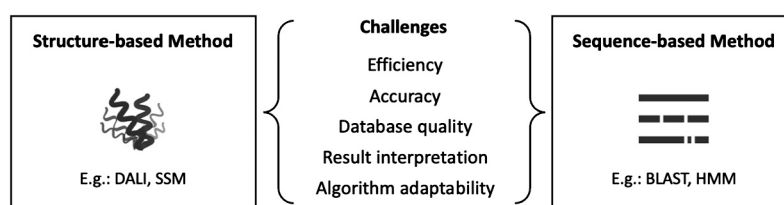


Fig. 2. Issues and challenges of existing protein search methods.

1.3. Challenges and bottlenecks in searching the protein space

Protein research heavily relies on extensive experimental data stored in various databases. The RCSB Protein Data Bank (RCSB PDB) is a widely-used repository for 3D protein structures determined by experimental methods [10]. UniProt is another comprehensive resource that integrates protein sequences and functional annotations from several databases, including the manually annotated Swiss-Prot and the automatically annotated TrEMBL [11]. Additionally, databases like CATH, SCOP, and Pfam classify and annotate proteins based on their structural, sequence, or functional characteristics [12,2,13]. These databases provide valuable information for understanding protein structure, function, and evolution, making them essential for protein research and design.

Protein searching typically employs two main approaches: structure-based and sequence-based methods. Structure-based methods, such as DALI [14] and TM-align [15], compare the 3D structures of proteins to identify similarities, which is particularly useful for proteins with diverse sequences but similar structures. In contrast, sequence-based methods, like BLAST [16] and HMMER [17], detect homology by comparing amino acid sequences and excel at identifying proteins with similar sequences.

However, as illustrated in Fig. 2, these methods have limitations in terms of efficiency, accuracy, and adaptability [18–21]. Structure-based methods are computationally intensive and slow, while sequence-based methods may struggle with proteins exhibiting significant structural differences despite sequence similarity. The non-linear relationship between protein structure and sequence can lead to errors in search results, with structure-based methods potentially failing to identify functional similarities between proteins with similar structures but different sequences, and sequence-based methods failing to accurately detect structural similarities in proteins with similar sequences but different structures. Database quality, including incorrect annotations, redundancy, and incomplete data, can also negatively impact search results. Moreover, interpreting protein search results and comparing them with experimental data can be challenging and error-prone for large datasets. Existing search methods may not adapt well to various protein types

and biological problems, particularly when targeting specific proteins or scenarios.

To address these challenges, researchers have developed various rapid protein structure search methods in recent years. For instance, 3D-AF-SURFER [22] utilizes deep learning to learn 3D feature representations of protein surfaces, enabling fast and accurate structure alignment and similarity search. GraSR [23] introduces a graph representation learning approach, improving search efficiency and generalizability by converting protein structures into graphs and learning their embedded representations. DeepFold [24] employs convolutional neural networks to extract features directly from 3D structures, accelerating the structure alignment process. MADOKA [25] designs a MapReduce-based distributed computing framework, achieving parallelization and scalability for large-scale protein structure searches. These methods have made significant contributions to accelerating searches and expanding functionality in the field of protein structure search.

Despite the progress made by existing methods, there are still limitations in terms of insufficient utilization of sequence information, limited search flexibility, and interpretability. These limitations highlight the need for innovative approaches that can effectively integrate sequence and structural information, support flexible search queries, and provide interpretable results.

Parametric protein search emerges as a promising alternative that tackles the challenges in protein search from a different perspective. Instead of solely relying on traditional structure or sequence-based methods, parametric protein search introduces a novel paradigm that leverages the power of parametric protein design. By carefully selecting and optimizing a set of key parameters that define protein structure and function, such as amino acid composition, hydrophobicity, and secondary structure propensities, parametric protein search enables a more comprehensive and fine-grained representation of proteins.

This parametric approach offers several advantages over existing methods. It allows for the seamless integration of sequence and structural information, as the parameters can capture both the global topology and the local physicochemical properties of proteins. Furthermore,

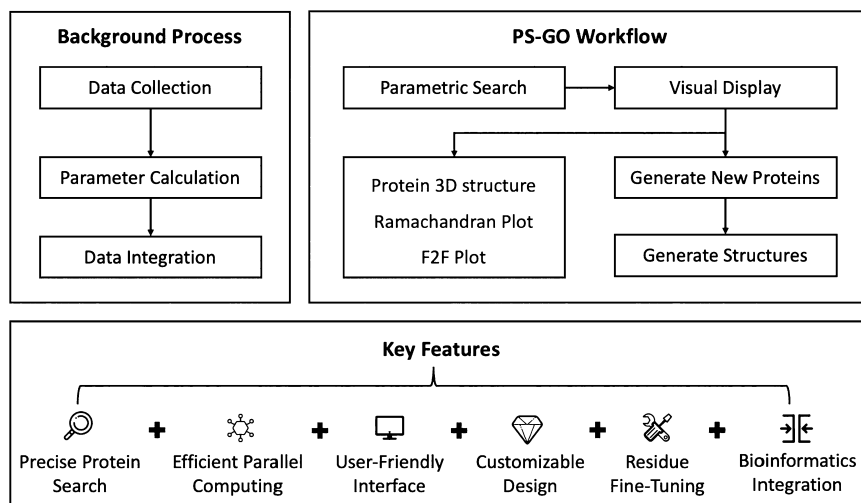


Fig. 3. How PS-GO search works and its key features.

it provides a highly flexible and customizable search framework, where users can define their own parameter combinations and search criteria based on their specific research needs. Finally, the parametric representation facilitates the interpretability of search results, as the key parameters contributing to the similarity or difference between proteins can be easily identified and visualized.

1.4. PS-GO - transforming protein exploration with a breakthrough search method

PS-GO (Parametric Search, GO!) is an innovative parametric protein search engine that cleverly integrates protein structure, sequence information, and computable parameters. Our search engine aims to address the multiple limitations of existing protein search methods and provide a more powerful, efficient, and flexible tool for protein design research and applications.

A core feature of PS-GO is its unique search capability, which enables users to specify specific parameters and constraints to search for protein sequences and structures that meet their needs through a parametric protein design approach. This approach improves the ability to control protein structure and function with precision by calculating and tuning key parameters in the protein molecule. Specifically, PS-GO can handle a variety of complex protein sequences and structure parameters such as RC.Score, hydrophobicity, instability, size, isoelectric point, and solvent accessibility to meet different research needs. This precise search capability not only improves the accuracy of the search results but also contributes to a deeper understanding of the structure-function relationships of proteins, which is of great academic value for research in the field of protein design and engineering.

As shown in Fig. 3, PS-GO (<https://psgo.ucc.ie/>) utilizes parallel computing and rapid algorithms and processes a vast array of protein structure parameters swiftly. Even when dealing with extensive protein data, PS-GO returns results that cater to users' needs promptly. Significantly more efficient than traditional structure or sequence-based search methods, PS-GO allows researchers to acquire necessary protein structure information in less time, accelerating the experimental procedure.

Another highlight of PS-GO is the enhanced user experience. Its intuitive graphical interface, coupled with natural language processing technology, allows effortless input and interaction, eliminating the requirement of specialized biological or computer technology knowledge. Therefore, PS-GO's usability extends beyond researchers, catering to a broader user base including students, teachers, biologists, and drug researchers. This capacity helps proliferate protein design knowledge and techniques, extending their applicability.

PS-GO also provides customization of parameters and optimization conditions according to user needs, facilitating realistic protein design. Its high adaptability and customization allow its utility across diverse research requirements and application scenarios, offering researchers more innovative avenues and fostering the advancement and application of protein design technologies.

In conjunction with the Silver Surfer service, PS-GO enables users to freeze certain protein sequence residues while adjusting others to generate novel protein sequences. This feature empowers researchers to probe deeper into protein sequence space, enhancing the diversity and adaptability of protein structures. This holds immense academic merit in investigating the interplay between protein sequence and structure, optimizing proteins' functional attributes, and creating opportunities for protein modification and novel drug development.

The integration of PS-GO with Protein Fragment And Structure Analysis (PROFASA) interlinks it with other bioinformatics tools and data resources, creating a comprehensive ecosystem for protein design and research [26]. Through PROFASA, PS-GO integrates with protein databases (like PDB, UniProt), sequence alignment tools (such as BLAST), and structure prediction software, establishes a one-stop platform for protein design and research, bolstering research and applications in related domains.

1.5. Parametric search strategy

PS-GO employs a novel parametric search strategy to efficiently explore the vast protein sequence and structure space. The key steps of this strategy are outlined below:

Data Collection and Processing: PS-GO collects protein data from various public databases, such as PDB and UniProt. The collected data is carefully curated and filtered based on sequence and structure quality, redundancy, and completeness. A series of preprocessing steps, including sequence alignment, structure validation, and data normalization, are performed to ensure the consistency and reliability of the data.

Parameter Calculation: For each protein in the database, PS-GO calculates a set of key parameters that characterize its physicochemical properties and structural features. These parameters include amino acid composition, hydrophobicity, instability index, molecular weight, isoelectric point, secondary structure content, solvent accessibility, and more. The calculation of these parameters is based on well-established algorithms and empirical formulas, such as the Kyte-Doolittle hydrophobicity scale [27] and the Chou-Fasman secondary structure prediction method [28].

Data Integration: The calculated parameters, along with the sequence and structure information, are integrated into a unified

database. This database serves as the foundation for the subsequent parametric search and protein design tasks. The integration process involves data mapping, indexing, and optimization to facilitate efficient data retrieval and analysis.

Parametric Search: Given a user-specified query, PS-GO performs a parametric search against the integrated database. The query can be a combination of various parameters, such as sequence similarity, structural motifs, and physicochemical properties. PS-GO employs advanced indexing and hashing techniques to quickly identify proteins that match the query criteria. The search results are ranked based on their relevance and similarity to the query, providing users with a prioritized list of candidate proteins.

Protein Design and Optimization: In addition to searching existing proteins, PS-GO also supports the design of novel proteins with desired properties. Users can specify the target parameters, such as stability, solubility, and function, and PS-GO will generate a set of candidate sequences that satisfy these requirements. The design process involves a combination of data-driven modeling, molecular simulations, and optimization algorithms. PS-GO iteratively refines the candidate sequences based on their predicted structures and functions, ultimately delivering a set of optimized protein designs for experimental validation.

By integrating these key steps, PS-GO provides a powerful and flexible platform for parametric protein search and design. The modular architecture of PS-GO allows for easy extension and customization, enabling researchers to adapt the system to their specific needs and incorporate new algorithms and data sources.

2. Methods

2.1. Parametric search algorithm

The core of PS-GO is an innovative parametric search algorithm that enables efficient and flexible retrieval of proteins based on user-specified multi-dimensional parameter conditions. The algorithm consists of the following key steps:

Parameter Extraction: Given a protein sequence or structure, PS-GO first extracts a set of parameters that characterize the physicochemical properties and structural features of the protein. These parameters include:

(1) Amino acid composition: The frequencies of 20 amino acids in the protein sequence are calculated using the ProtParam module of BioPython [29], resulting in a 20-dimensional composition vector.

(2) Isoelectric point: The `isoelectric_point()` function in the ProtParam module is used to calculate the isoelectric point of the protein based on the pKa values of amino acids.

(3) Molecular weight: The `molecular_weight()` function in the ProtParam module is used to calculate the molecular weight of the protein based on the molecular mass of amino acid residues.

(4) Hydrophobicity: We implemented the Kyte-Doolittle hydrophobicity scale algorithm to calculate the hydrophobicity score of each amino acid residue using a sliding window approach. The average score is then taken as the hydrophobicity index of the entire protein. Specifically, the following formula is used:

$$H = \frac{1}{n} \sum_{i=1}^n h_i \quad (1)$$

where H is the hydrophobicity index of the protein, n is the number of amino acid residues, and h_i is the hydrophobicity score of the i -th residue (obtained from the Kyte-Doolittle scale table).

(5) Instability index: The `instability_index()` function in the ProtParam module is used to calculate the instability index of the protein based on the dipeptide frequency and the Dipeptide instability weight values (DIWV) instability weight values [30].

(6) Secondary structure: The DSSP program is used to predict the secondary structure (like α -helix, β -sheet, coil) of the protein, and the

proportions of each secondary structure type are calculated [31]. DSSP uses hydrogen-bonding patterns and geometrical criteria to reliably identify secondary structure elements.

(7) Solvent accessibility: The NACCESS program is used to calculate the relative solvent accessibility (RSA) of residues on the protein surface [32]. NACCESS employs the Lee-Richards algorithm to estimate the accessibility of residues by rolling a probe over the protein surface. We calculate the RSA value for each residue and then take the average as the solvent accessibility index of the entire protein.

The above parameter extraction process is implemented by combining existing bioinformatics tools (like DSSP, NACCESS) and self-developed algorithms (like Kyte-Doolittle hydrophobicity calculation) to ensure the efficiency and accuracy of the calculation. The algorithms are implemented in Go, and the bioinformatics tools are invoked through system calls.

To speed up the parameter extraction process, we leverage Go's goroutine mechanism to perform concurrent calculations. Each protein is assigned to a goroutine, which independently carries out the parameter extraction tasks. Within each goroutine, we invoke Python scripts or custom Go scripts to compute specific parameters. This parallel computing approach greatly improves the efficiency of parameter extraction, especially when dealing with large-scale protein datasets.

Natural Language Search: In addition to range-based parameter search, PS-GO also supports natural language queries. Users can input natural language sentences describing the desired protein properties or functions, such as "Find proteins with high stability and low molecular weight". We utilize the GPT-4 model from OpenAI to convert the natural language query into structured parameter conditions through function calls [33].

Specifically, we define a function named "parse_query" that takes a natural language query as input and outputs the parameter conditions in JSON format. For example, given the above query, GPT-4 will return the following JSON:

```
{
  "stability": "high",
  "molecular_weight": "low"
}
```

We then map these qualitative descriptions to specific numeric ranges. For instance, "high stability" corresponds to an instability index less than 40, and "low molecular weight" corresponds to a molecular weight less than 50 kDa. Finally, the converted parameter conditions are used for the subsequent search process.

Indexing and Hashing: To accelerate the retrieval process, PS-GO builds inverted indexes for the extracted parameter values. Each parameter value is mapped to a unique hash code, and the proteins are organized into corresponding hash buckets based on their parameter features. This indexing scheme allows for constant-time lookup of proteins that match a given parameter query condition.

Specifically, we adopt a multi-dimensional hashing approach that converts each parameter value into a 64-bit hash code. For numeric parameters (like molecular weight), we use a double hashing function to map the value to a hash bucket:

$$h_1(x) = (ax + b)\%p \quad (2)$$

$$h_2(x) = 1 + ((x + c)\%(p - 1)) \quad (3)$$

where x is the parameter value, p is a prime number (chosen as the smallest prime greater than the range of the parameter), and a, b, c are randomly selected constants. The double hashing function effectively reduces hash collisions and improves retrieval efficiency.

For categorical parameters (like secondary structure), we use the MurmurHash3 algorithm to map the category labels to hash codes [34]. MurmurHash3 is a non-cryptographic hash function known for its fast speed and good distribution properties.

During retrieval, the system calculates the hash codes corresponding to the user query and then performs matching in the corresponding hash buckets to filter out the proteins that satisfy the conditions.

Query Processing: When a user submits a query containing multiple parameter conditions, PS-GO first parses the query and converts it into a set of hash codes. Then, the algorithm retrieves proteins from the corresponding hash buckets and filters them to find the results that satisfy all the conditions.

We designed a set operation-based query processing method that treats multiple parameter conditions as sets and uses set operations (like intersection, union, difference) to combine the conditions. For example, given the query “(stability > 0.8) AND (size < 2000)”, we first find the sets of proteins satisfying “stability > 0.8” and “size < 2000”, denoted as S_1 and S_2 , respectively. We then calculate the intersection of S_1 and S_2 to obtain the final result.

This set-based query processing method fully leverages the inverted indexes and avoids linear scans over the entire database, thereby significantly improving query efficiency.

2.2. Protein database construction

The PS-GO database is constructed by integrating two major public data sources: PDB and UniProt.

Data Integration: We first download data files from the FTP servers of PDB and UniProt, respectively. For PDB, we download the mmCIF format structure files and FASTA format sequence files. For UniProt, we download the XML format annotation files and FASTA format sequence files.

We then use the BioPython library in Python to parse these files and extract the fields of interest, such as PDB ID, UniProt ID, amino acid sequence, secondary structure, and functional annotations. These fields are mapped to a relational database schema, creating tables for proteins, residues, atoms, and annotations.

During the integration process, we need to handle cross-references between PDB and UniProt to establish a mapping between the two databases. We utilize the “PDB cross-reference” field in UniProt, which provides the correspondence between UniProt entries and PDB structures. Through these cross-references, we can associate structural information with sequence and functional annotations.

We developed an automated data integration pipeline using the Luigi workflow framework to manage tasks such as data downloading, parsing, cleaning, and mapping. The pipeline runs periodically to ensure that the database stays in sync with PDB and UniProt.

Data Storage: PS-GO uses a MySQL relational database management system to store and manage the protein data. MySQL is a widely-used open-source database known for its high performance, scalability, and ease of use.

We designed a normalized database schema that includes the following main tables:

(1) Protein table: stores basic information about proteins, such as PDB ID, UniProt ID, sequence length, and species.

(2) Residue table: stores information about amino acid residues in proteins, such as residue number, residue name, and secondary structure type.

(3) Atom table: stores the atomic coordinate information of proteins, such as atom number, atom name, and x/y/z coordinates.

(4) Annotation table: stores functional annotation information for proteins, such as Gene Ontology (GO) terms, Enzyme Commission (EC) numbers, and Pfam domains.

(5) Parameter table: stores various parameters of proteins, such as molecular weight, isoelectric point, hydrophobicity, and instability index.

We utilize MySQL’s indexing mechanisms (like B+ tree index, hash index) to speed up data retrieval. For frequently queried fields (like PDB ID, UniProt ID), we create single-column indexes. For fields that

are often queried together (like residue number and atom name), we create composite indexes.

Additionally, we developed stored procedures and user-defined functions (UDFs) to perform certain computation tasks on the database server side, reducing the burden on the application server.

Data Access: PS-GO uses the database/sql package in Go to communicate with the MySQL database [35]. The database/sql package provides a generic interface for interacting with SQL databases, supporting multiple database drivers.

We define Go structs that correspond to the tables in the database. These structs, also known as “models”, contain fields that map to the columns of the tables. We use the GORM library, which is an object-relational mapping (ORM) library for Go, to simplify database operations [35]. GORM allows us to define database models using Go structs and provides high-level APIs for querying and manipulating data.

For example, the following code defines the Protein and Residue models:

```
type Protein struct {
    ID          uint        `gorm:"primaryKey"`
    PdbID       string      `gorm:"index"`
    UniprotID   string      `gorm:"index"`
    Sequence    string
    Residues    []Residue
}

type Residue struct {
    ID          uint        `gorm:"primaryKey"`
    ProteinID   uint
    Number      int
    Name        string
    SSType      string
}
```

With these model definitions, we can easily perform database operations using GORM’s methods, such as Create, Find, Update, and Delete. For instance, the following code demonstrates how to query a protein by its PDB ID and retrieve its residue information:

```
var protein Protein
db.Preload("Residues").First(&protein, "pdb_id = ?", "1ake")

for _, residue := range protein.Residues {
    fmt.Println(residue.Number, residue.Name, residue.SSType)
}
```

By leveraging GORM, we can express complex SQL queries in a more intuitive and object-oriented way, greatly improving development efficiency.

2.3. Silver surfer

The integration with Silver Surfer service takes PS-GO’s capabilities a step further. Should the initial search not yield any closely matching sequences, Silver Surfer enables the creation of novel protein sequences. Silver Surfer is a novel tool that applies a genetic algorithm for protein sequence generation with a fitness function derived from the query. The fitness function plays a crucial role in guiding the evolutionary process towards sequences that satisfy the user-specified target properties.

In Silver Surfer, the fitness function is designed to measure the similarity between the generated sequences and the target protein properties specified by the user. The fitness function calculates a weighted sum of the mean absolute errors (MAEs) between the predicted properties of the generated sequence and the user-defined target values. The MAE for each property is computed as follows:

$$MAE_i = \frac{1}{n} \sum_{j=1}^n |p_{ij} - t_{ij}| \quad (4)$$

where i is the index of the property, n is the number of residues in the sequence, p_{ij} is the predicted value of the i -th property for the j -th residue, and t_{ij} is the user-defined target value for the i -th property at the j -th residue.

The properties considered in the fitness function include various physicochemical and structural parameters, such as stability, hydrophobicity, molecular weight, and secondary structure composition. For each property, the user specifies a desired target value or range at each residue position. The fitness function then evaluates how well the generated sequence matches these target values by calculating the MAE for each property.

To account for the relative importance of different properties, the fitness function assigns a weight to each property based on user preferences or predefined settings. The weighted sum of MAEs is calculated as follows:

$$\text{Fitness} = \sum_{i=1}^m w_i \cdot \text{MAE}_i \quad (5)$$

where m is the number of properties, w_i is the weight assigned to the i -th property, and MAE_i is the mean absolute error for the i -th property.

The weights allow users to prioritize certain properties over others. For example, if stability is more important than hydrophobicity for a specific protein design task, the user can assign a higher weight to the stability property. The fitness function will then place more emphasis on minimizing the MAE for stability compared to other properties.

In addition to the weighted sum of MAEs, the fitness function also incorporates a penalty term for sequences that violate the “frozen” regions specified by the user. The frozen regions are specific portions of the sequence that should remain unchanged during the optimization process. If a generated sequence modifies any residue in the frozen regions, a large penalty value is added to the fitness score, effectively discouraging the selection of such sequences.

The final fitness score is calculated as follows:

$$\text{Fitness} = \sum_{i=1}^m w_i \cdot \text{MAE}_i + \lambda \cdot \text{FrozenPenalty} \quad (6)$$

where λ is a hyperparameter that controls the strength of the penalty, and FrozenPenalty is the number of residues in the frozen regions that have been modified.

A lower fitness score indicates a better fit between the generated sequence and the target properties. The genetic algorithm in Silver Surfer aims to minimize the fitness score by selecting and mutating sequences with lower scores.

During the evolutionary process, the genetic algorithm maintains a population of candidate sequences. In each generation, the sequences with the lowest fitness scores are selected for reproduction and mutation. The mutation operator introduces random changes to the sequences while respecting the frozen regions. The modified sequences are then evaluated using the fitness function, and the process continues for a specified number of generations or until a satisfactory solution is found.

By incorporating a comprehensive fitness function that considers multiple properties, assigns weights based on user preferences, and penalizes violations of frozen regions, Silver Surfer enables the generation of novel protein sequences that closely match the user-defined target properties. This approach allows users to fine-tune the desired characteristics of the generated sequences and explore a vast sequence space beyond the existing proteins in the database.

The genetic algorithm employed by Silver Surfer, as shown in Table 2, leverages the fitness function to guide the search process efficiently. The iterative process of selection, mutation, and evaluation allows Silver Surfer to continuously improve the generated sequences until a satisfactory solution is found or a predefined stopping criterion is met.

This cycle repeats until a stopping condition, such as reaching a maximum number of generations or achieving a fitness score below a

specified threshold, is met. The use of the weighted MAE-based fitness function with frozen region constraints allows Silver Surfer to efficiently explore the vast sequence space and generate novel proteins that closely match the user-defined target properties while preserving critical regions of the sequence.

3. Results

3.1. Key functionalities and user interface of PS-GO

PS-GO offers a comprehensive suite of functionalities for protein search, analysis, and visualization, all accessible through a user-friendly web interface. The primary feature of PS-GO is its parametric search capability, which allows users to search for proteins based on a combination of sequence, structure, and physicochemical properties. Users can specify search criteria using a flexible parameter selection tool, which includes options for amino acid composition, hydrophobicity, secondary structure propensities, and various geometric features.

In addition to parametric search, PS-GO also supports natural language search, enabling users to query proteins using plain text descriptions of desired properties or functions. The natural language search engine employs advanced NLP techniques to interpret user queries and retrieve relevant protein entries from the database.

Fig. 4 showcases PS-GO’s user-friendly search interface, which allows users to easily input their search criteria and view the resulting protein information.

With PS-GO, users can visualize protein 3D structures directly from the search results page. The structure viewer provides various representation options, such as cartoon, surface, and ball-and-stick models. Users can interact with the displayed structure by rotating, zooming, and panning to examine it from different angles.

Furthermore, PS-GO integrates the PROFASA (Protein Fragment and Structure Analysis) tool for protein structure visualization and analysis. PROFASA is a web-based platform that offers a range of functionalities for exploring and understanding protein structures [26].

PROFASA’s analysis tools, accessible through PS-GO, enable users to investigate protein structural features in detail. These tools include secondary structure analysis, structural alignment, binding site prediction, and structural superposition.

Fig. 5 illustrates an example of protein structure visualization and analysis using PS-GO and PROFASA.

PS-GO efficiently calculates a range of protein parameters, including RC.Score, hydrophobicity, instability, molecular size, isoelectric point, and solvent accessibility. These parameters offer valuable insights into the physicochemical properties and potential functions of proteins.

The calculated parameters are presented in an intuitive and visually appealing manner within PS-GO’s interface (Fig. 4). Users can easily view and analyze the parameter values, facilitating the interpretation of protein characteristics and the identification of proteins with desired properties.

3.2. Advantages and potential impact of PS-GO

PS-GO introduces a novel protein search framework based on parametric protein design principles. By integrating sequence, structure, and physicochemical property information, PS-GO provides users with a unique perspective to search and explore proteins. Different from existing methods that rely on sequence or structure alignment, PS-GO’s parametric search approach allows users to find proteins that satisfy specific criteria based on a combination of various properties. This innovative search paradigm complements traditional methods by enabling the discovery of proteins that might be difficult to identify using sequence or structure similarity alone.

One of the major strengths of PS-GO is its flexibility and customizability. The parametric search engine allows users to define their own search criteria and parameter combinations, tailoring the search process

Table 2
The genetic algorithm employed by Silver Surfer.

Initial Population: The process begins with a set of protein sequences, which can be randomly generated or seeded from existing sequences. Each sequence is represented as a string over a finite alphabet, in the case of protein sequences, this consists of 20 amino acids. All sequences constitute a continuous individual pool.

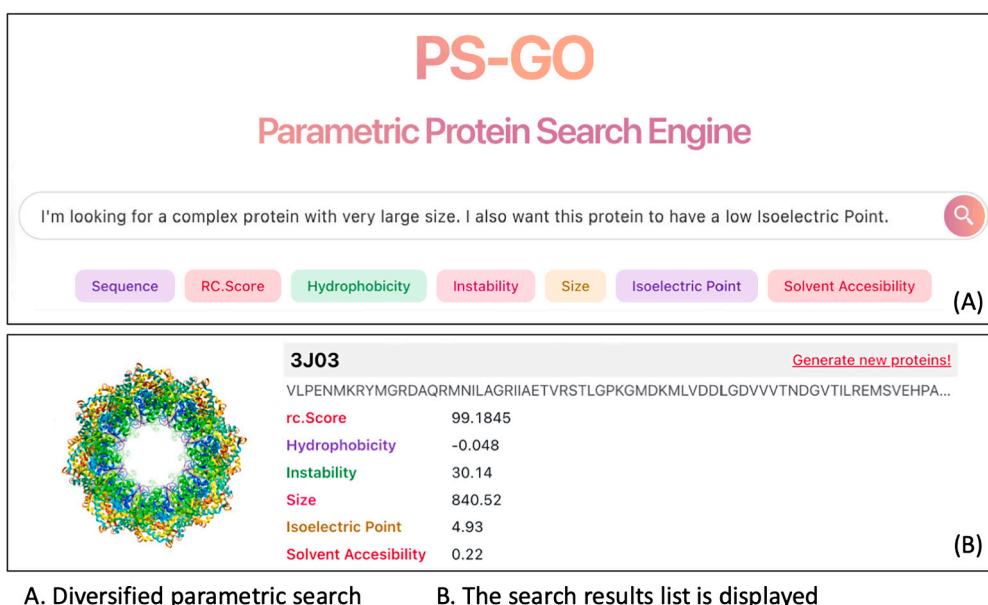
Fitness Function: A fitness function is used to evaluate the quality of each sequence. This function uses a weighted sum of MAEs and a penalty for frozen region violations to measure how well the protein sequence performs. Based on the results of the fitness function evaluation, each sequence is assigned a fitness score.

Selection and Mutation: In each iteration, the sequences with the lowest fitness scores are selected from the individual pool for reproduction and mutation. The selected sequences undergo mutation operations, generating a series of new sequences. During the mutation process, certain parts can be set as “frozen”, remaining unchanged.

Sequence Evaluation: The newly generated sequences are evaluated using the fitness function, and their fitness scores are calculated.

Population Update: The new sequences are added to the population, replacing the sequences with higher fitness scores. The size of the population remains constant throughout the iterations.

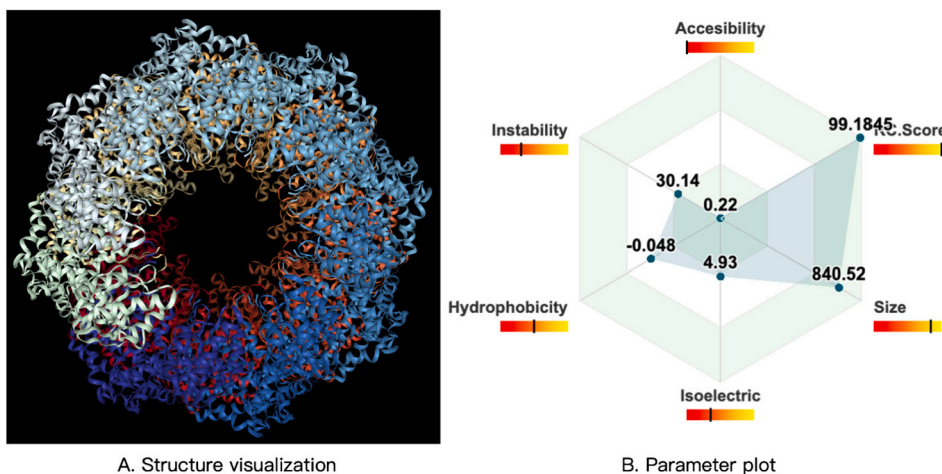
Iterative Process: The process continues to iterate, selecting the sequences with the lowest fitness scores for mutation and updating the population with the newly generated sequences. This process goes on until a sequence meeting the specified criteria is found or a predefined number of iterations is reached.



A. Diversified parametric search

B. The search results list is displayed

Fig. 4. PS-GO protein search engine. A. Parametric search page, select the corresponding parameter and give the filter range to search. B. Search result list, showing the model picture, sequence, and parameter of the protein.



A. Structure visualization

B. Parameter plot

Fig. 5. Structural visualization of protein models and graphical representation of parameters using PS-GO and PROFASA.

Table 3
Comparison of features and characteristics between PS-GO and other protein search methods.

Method	Search Mechanism	Protein Representation	Flexibility	Interpretability
PS-GO	Parametric protein design-based	Integration of sequence, structure, and physicochemical properties	Allows user-defined parameter combinations and search criteria	Provides insights into key parameters contributing to similarity or difference
Dali [14]	Structure-based	3D coordinates of protein backbone	Limited to structural similarity	Focuses on global structural alignment
Foldseek [36]	Structure-based	3D coordinates of protein backbone	Allows different distance measures and alignment modes	Provides local alignment information
PSI-BLAST [37]	Sequence-based	Amino acid sequence	Allows iterative search and position-specific scoring matrices	Focuses on sequence similarity

Table 4
The average performance of PS-GO in handling different tasks.

Task	Average Operation Time (s)
Parametric Search	0.84
Natural Language Search	1.62
New Sequence Generation	5.10
Parameter Calculation	6.4 s per 50 items

to specific research questions or application scenarios. This flexibility empowers researchers to explore protein relationships from different angles and to uncover novel insights that might be missed by fixed, pre-defined search methods.

Another key advantage of PS-GO is its interpretability. By providing detailed parameter profiles and visualizations, PS-GO helps users understand the key factors contributing to protein similarity and function. This interpretability is particularly valuable for tasks such as protein function prediction, rational protein design, and evolutionary analysis, where understanding the underlying mechanisms are crucial.

Table 3 provides a summary comparison of the features and characteristics of PS-GO and other representative protein search methods.

The modular architecture of PS-GO also makes it a valuable platform for integrating new protein tools. As new methods for protein representation and comparison are developed, they can be easily incorporated into the PS-GO framework, allowing researchers to complement their work.

3.3. Performance evaluation

To assess the performance of PS-GO in handling various tasks, we conducted benchmark tests using a script that iteratively executes parametric searches, natural language searches, and new sequence generation functions for 100 times. The benchmark script is available at github (<https://github.com/Atobelin/psgo-benchmarking>). In each iteration, we randomly generated or selected different parameter values, query statements, and target conditions to simulate the diversity of real-world usage scenarios. The test results are presented in Table 4.

For the parametric search task, PS-GO achieves an average response time of 0.84 seconds. This demonstrates that PS-GO is capable of processing complex multi-parameter queries within sub-second latency, exhibiting exceptional retrieval efficiency. Even under dynamic variations of parameter values, PS-GO maintains stable and rapid performance, which can be attributed to our optimized indexing structure and query algorithms.

In the natural language search task, PS-GO yields an average response time of 1.62 seconds. Although natural language queries involve more semantic understanding and matching computations, PS-GO still manages to return precise results within 2 seconds. This is primarily due to the advanced natural language processing techniques we employ, such as deep learning-based semantic representations and intelligent query expansion, enabling PS-GO to efficiently handle natural language queries.

Regarding the new sequence generation task, we conducted 50 iterations of testing, resulting in an average operation time of 5.10 seconds. Generating new sequences is a computationally intensive task that requires searching and optimizing within a vast sequence space, thus taking relatively longer time. Nevertheless, PS-GO completes the generation of new sequences within a short period, owing to the heuristic search strategies and parallel computing techniques we adopt, which significantly accelerate the sequence generation process.

We did not include benchmark tests for the parameter calculation task. This is because parameter calculation is a one-time operation performed during database construction, mainly involving feature extraction and statistical analysis on large-scale protein data. Although computationally intensive, this process can be efficiently completed in the background through preprocessing and caching optimizations. Once the parameter calculation is finished, subsequent search and analysis tasks can directly utilize the pre-computed results without repetitive calculations. Therefore, assessing the performance of parameter calculation is less relevant to evaluating PS-GO's real-time responsiveness and user experience.

Overall, the benchmark test results demonstrate that PS-GO exhibits outstanding performance and stability in key tasks such as parametric search, natural language search, and new sequence generation. Even under dynamically changing query conditions, PS-GO consistently delivers efficient service, providing researchers with a fast, reliable, and user-friendly platform for protein analysis, which has the potential to greatly enhance the efficiency and depth of protein research.

4. Discussion and outlook

Parametric protein design has emerged as a powerful technique for precisely manipulating protein structure and function by computationally adjusting key parameters in protein molecules [38]. This approach has gained significant traction across multiple fields, including drug discovery, biotechnology, and synthetic biology, owing to rapid advancements in computer science and experimental techniques [39]. However, the full potential of parametric protein design is often hindered by the limitations of traditional protein structure databases, which face challenges in search efficiency and computational complexity.

One major hurdle is the time-consuming data processing and significant computational power required to search through vast protein databases [18]. Additionally, identifying proteins with substantial sequence differences using sequence-based methods or discerning the functional similarity of proteins with varying sequences using structure-

based methods can be complex and computationally intensive [19]. These challenges underscore the need for an enhanced, accurate, and adaptable search strategy to complement parametric protein design efforts.

Parametric protein search engines, such as PS-GO, offer a promising solution to these challenges by incorporating advanced algorithms, optimizing computational resources, and improving protein feature understanding. These search strategies aim to efficiently navigate the vast protein sequence and structure space, enabling researchers to quickly identify promising candidates for further optimization through parametric protein search.

By integrating parametric protein search with parametric protein design, researchers can unlock the untapped potential of these technologies and fuel research progress in related fields. The development of efficient protein search engines not only enhances the speed and efficiency of protein design but also opens up new avenues for groundbreaking discoveries in bioinformatics and computational biology.

In essence, the symbiotic relationship between parametric protein design and parametric protein search holds the key to overcoming the limitations of traditional protein structure databases and accelerating innovation in protein engineering. As these technologies continue to evolve and integrate, they promise to revolutionize our understanding of proteins and their applications in various domains.

PS-GO, as a search method based on parametric protein design, has significant advantages over traditional protein structure databases in the following areas:

- **Highly accurate search capabilities:** PS-GO allows users to specify specific parameters and constraints to meet different research needs through a parametric protein design approach. This precise search capability improves control over protein structure and function and is of great academic value to the field of protein design.
- **Optimized computing efficiency:** Using parallel computing and fast algorithms, PS-GO is able to process a large number of protein structure parameters in a short time. Compared to traditional search methods, PS-GO's computational efficiency is significantly improved, speeding up the experimental process.
- **User-friendly interface and customization:** PS-GO provides an intuitive graphical interface and natural language processing technology for easy user input and interaction. At the same time, it allows users to customize parameters and optimization conditions according to their needs, adapting to various research requirements and application scenarios.
- **Integration with other bioinformatics tools:** Combined with the Silver Surfer service, PS-GO supports users to freeze partial residues of protein sequences and integrates with PROFASA to form a complete protein design and research ecosystem with other bioinformatics tools and data resources, providing a one-stop protein design and research platform.

Despite its many significant advantages, PS-GO has a number of limitations in practical application. PS-GO has a relatively high demand for computational resources, which requires a certain level of computational power from the user. In particular, more accurate algorithm design and powerful computational support are required when dealing with complex protein structure parameters. The performance and accuracy of PS-GO are limited by the quality and quantity of the protein database. If specific types of proteins are missing from the database, or if there is incorrect or inaccurate information, the performance and accuracy of PS-GO may be compromised. The number of parameters that PS-GO can currently handle is still limited, which means that continued research and development of new algorithms is required to broaden the range of computable parameters.

As shown in Fig. 6, in order to continuously improve the efficiency and accuracy of PS-GO, the future development direction can include the following aspects:





Application Area	Outlook
Pharmaceuticals	 Deep learning technology
Protein Interaction Studies	 Cloud computing & distributed computing
Bioengineering	 Integration of bioinformatics tools & data resources
Biomaterials	

Fig. 6. PS-GO's future development will focus on refining algorithms, expanding the database, and using advanced computing technologies to optimize protein design performance.

In order to improve the search accuracy of PS-GO, algorithms can be optimized for specific types of parameters in the future. This includes the introduction of more advanced machine learning and artificial intelligence techniques, such as deep learning, as a means of automatically learning and recognizing complex patterns in proteins to improve the effectiveness and accuracy of parametric design. Finer parametric design methods can help users better control the structure and function of proteins to meet different research needs.

Meanwhile, PS-GO's database can be expanded by exchanging data with other bioinformatics databases or automatically extracting protein data from literature and the web through data mining and machine learning methods. This expansion can provide richer and more accurate protein search results. The expansion and management of the database can be done by applying advanced database management systems and data mining techniques, which can further improve the efficiency and accuracy of the search.

Since PS-GO needs to deal with a large amount of protein data and complex computational tasks, the combination of cloud computing and distributed computing technologies will greatly enhance its computational capability. Specifically, cloud computing provides elastic computing resources that can be dynamically adjusted to meet different computational needs. Distributed computing, on the other hand, can break down large-scale data processing tasks to multiple computers, greatly improving the efficiency and stability of computation.

In addition, PS-GO plans to initiate deeper collaborations with more computational biology tools to enhance its search capabilities and broaden the scope of its services. Among them, collaboration with protein-related tools will be especially critical. For example, with the help of more protein structure prediction tools, PS-GO can design new protein sequences while at the same time predicting and displaying the many possible 3D structures of the protein. Through integration with protein function annotation tools, PS-GO can further provide information on the possible biological functions of proteins and their roles in biological processes. Integration with protein interaction network analysis tools will help PS-GO to demonstrate the interaction of proteins with other molecules in the search results, helping researchers to understand the behavior and function of proteins in cells. Such cooperation and integration will not only provide users with more comprehensive information, but also improve the search accuracy of PS-GO.

The PS-GO parametric protein search engine, while not directly engaging in drug development, protein interaction studies, or bioengineering, serves as a pivotal tool in expediting and refining protein research across these disciplines. It significantly impacts the pharmaceutical field by enhancing the efficiency of identifying proteins for early-stage drug discovery, even though it doesn't discover drug molecules itself. In the bioengineering sector, PS-GO proves instrumental by enabling researchers to find proteins with specific functions and properties relevant to their work, albeit not designing proteins or directly supporting synthetic biology. For bioinformatics and structural biology, PS-GO boosts research progress in areas like protein folding, protein dynamics, and disease-related mutations, not through direct contributions to protein structure prediction or functional annotation, but by facilitating the rapid location of proteins of interest. Hence, PS-GO, with its ability

to swiftly locate relevant proteins based on specified parameters, indirectly strengthens protein research across various domains.

In conclusion, PS-GO, as an innovative parametric protein search engine, is expected to play a significant role in various fields. In the biomedical field, PS-GO can accelerate the process of drug screening and design, helping researchers quickly find protein molecules with specific functions, thereby shortening the drug development cycle and reducing research and development costs. In the field of synthetic biology, PS-GO can assist in designing entirely new artificial protein molecules, creating novel functions and features that do not exist in nature, thus promoting the development of synthetic biology. In proteomics and structural biology research, PS-GO can improve the efficiency of protein database retrieval and homology analysis, deepening the understanding of the relationship between protein structure and function. Through continuous optimization and upgrading, PS-GO is expected to become an important research tool in the life sciences.

5. Funding

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 18/CRT/6223.

Declaration of competing interest

No competing interest is declared.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.csbj.2024.04.003>.

References

- Brin S, Page L. The anatomy of a large-scale hypertextual Web search engine. *Comput Netw ISDN Syst* 1998;30(1–7):107–17. [https://doi.org/10.1016/S0169-7552\(98\)00110-X](https://doi.org/10.1016/S0169-7552(98)00110-X).
- Murzin AG, Brenner SE, Hubbard T, Chothia C. SCOP: A structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol* 1995;247(4):536–40. [https://doi.org/10.1016/S0022-2836\(05\)80134-2](https://doi.org/10.1016/S0022-2836(05)80134-2).
- Kuhlman B, Bradley P. Advances in protein structure prediction and design. *Nat Rev Mol Cell Biol* 2019;20(11):681–97. <https://doi.org/10.1038/s41580-019-0163-x>.
- Alford RF, Leaver-Fay A, Jeliaskov JR, O'Meara MJ, DiMaio FP, Park H, et al. The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *J Chem Theory Comput* 2017;13(6):3031–48. <https://doi.org/10.1021/acs.jctc.7b00125>.
- Gainza P, Roberts KE, Donald BR. Protein design using continuous rotamers. *PLoS Comput Biol* 2012;8(1):e1002335, publisher: Public Library of Science San Francisco, USA.
- Dill KA, MacCallum JL. The Protein-Folding Problem, 50 Years On. *Science* 2012;338(6110):1042–6. <https://doi.org/10.1126/science.1219021>.
- Mirny LA, Shakhnovich EI. Universally conserved positions in protein folds: reading evolutionary signals about stability, folding kinetics and function. *J Mol Biol* 1999;291(1):177–96. publisher: Elsevier.
- Orengo CA, Jones DT, Thornton JM. Protein superfamilies and domain superfolds. *Nature* 1994;372(6507):631–4, publisher: Nature Publishing Group UK London.
- Yallapragada VVB, Walker SP, Devoy C, Buckley S, Flores Y, Tangney M. Function2Form Bridge—Toward synthetic protein holistic performance prediction. *Proteins, Struct Funct Bioinform* 2020;88(3):462–75. <https://doi.org/10.1002/prot.25825>.
- Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The protein data bank. *Nucleic Acids Res* 2000;28(1):235–42, publisher: Oxford University Press.
- UniProt: the universal protein knowledgebase in 2021. *Nucleic acids research* 2021;49(D1):D480–9, publisher: Oxford University Press.
- Orengo CA, Michie AD, Jones S, Jones DT, Swindells MB, Thornton JM. Cath—a hierarchical classification of protein domain structures. *Structure* 1997;5(8):1093–109.
- Finn RD, Coghill P, Eberhardt RY, Eddy SR, Mistry J, Mitchell AL, et al. The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Res* 2016;44(D1):D279–85, publisher: Oxford University Press.
- Holm L, Laakso LM. Dali server update. *Nucleic Acids Res* 2016;44(W1):W351–5.
- Zhang Y, Skolnick J. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic Acids Res* 2005;33(7):2302–9.
- Ye J, McGinnis S, Madden TL. Blast: improvements for better sequence analysis. *Nucleic Acids Res* 2006;34(suppl_2):W6–9.
- Finn RD, Clements J, Eddy SR. Hmmer web server: interactive sequence similarity searching. *Nucleic Acids Res* 2011;39(suppl_2):W29–37.
- Rost B. Twilight zone of protein sequence alignments. *Protein Eng* 1999;12(2):85–94.
- Gligorijević V, Renfrew PD, Kosciolok T, Leman JK, Berenberg D, Vatanen T, et al. Structure-based protein function prediction using graph convolutional networks. *Nat Commun* 2021;12(1):3168.
- Schoes AM, Brown SD, Dodevski I, Babbitt PC. Annotation error in public databases: misannotation of molecular function in enzyme superfamilies. *PLoS Comput Biol* 2009;5(12):e1000605.
- Pearson WR. An introduction to sequence similarity (“homology”) searching. *Curr Protoc Bioinform* 2013;42(1):3.1.1.
- Aderinwale T, Bharadwaj V, Christoffer C, Terashi G, Zhang Z, Jahandideh R, et al. Real-time structure search and structure classification for alphafold protein models. *Commun Biol* 2022;5(1):316.
- Xia C, Feng S-H, Xia Y, Pan X, Shen H-B. Fast protein structure comparison through effective representation learning with contrastive graph neural networks. *PLoS Comput Biol* 2022;18(3):e1009986.
- Lee J-W, Won J-H, Jeon S, Choo Y, Yeon Y, Oh J-S, et al. Deepfold: enhancing protein structure prediction through optimized loss functions, improved template features, and re-optimized energy function. *Bioinformatics* 2023;39(12):btad712.
- Tilahun S, Jeong MJ, Choi HR, Baek MW, Hong JS, Jeong CS. Prestorage high co2 and 1-mcp treatment reduce chilling injury, prolong storability, and maintain sensory qualities and antioxidant activities of “madoka” peach fruit. *Front Nutr* 2022;9:903352.
- Mi Y, Marcu S-B, Tabirca S, Yallapragada VV. Profasa-a web-based protein fragment and structure analysis workstation. *Front Bioeng Biotechnol* 2023;11:1192094. <https://doi.org/10.3389/fbioe.2023.1192094>.
- Kapcha LH, Rosky PJ. A simple atomic-level hydrophobicity scale reveals protein interfacial structure. *J Mol Biol* 2014;426(2):484–98.
- Kumar TA. Cfspp: Chou and fasman secondary structure prediction server. *Wide Spectrum* 2013;1(9):15–9.
- Cock PJ, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics* 2009;25(11):1422.
- Guruprasad K, Reddy BB, Pandit MW. Correlation between stability of a protein and its dipeptide composition: a novel approach for predicting in vivo stability of a protein from its primary sequence. *Protein Eng Des Sel* 1990;4(2):155–61.
- Zacharias J, Knapp E-W. Protein secondary structure classification revisited: processing dssp information with pssc. *J Chem Inf Model* 2014;54(7):2166–79.
- Ding J, Arnold E. Naccess; 2006.
- Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Aleman FL, et al. Gpt-4 technical report, arXiv preprint. arXiv:2303.08774, 2023.
- Dahlgard S, Knudsen M, Thorup M. Practical hash functions for similarity estimation and dimensionality reduction. *Adv Neural Inf Process Syst* 2017;30.
- Yellavula N. Building RESTful Web services with Go: Learn how to build powerful RESTful APIs with Golang that scale gracefully. Packt Publishing Ltd; 2017.
- van Kempen M, Kim SS, Tumescheit C, Mirdita M, Gilchrist CL, Söding J, et al. Foldseek: fast and accurate protein structure search, *Biorxiv* 2022, 2022–02.
- Bhagwat M, Aravind L. Psi-blast tutorial. *Comp Genomics* 2008:177–86.
- Korendovych IV, DeGrado WF. De novo protein design, a retrospective. *Q Rev Biophys* 2020;53:e3.
- Wei W, Cherukupalli S, Jing L, Liu X, Zhan P. Fsp3: A new parameter for drug-likeness. *Drug Discov Today* 2020;25(10):1839–45.