J Biosci (2024)49:41
DOI: 10.1007/s12038-023-00410-x

Finding identical sequence repeats in multiple protein sequences: An algorithm

Vikas Kumar Maurya, Madhumathi Sanjeevi, Chandrasekar Narayanan Rahul, Ajitha Mohan, Dhanalakshmi Ramachandran, Rashmi Siddalingappa, Roshan Rauniyar and Sekar Kanagaraj*

Department of Computational and Data Sciences, Indian Institute of Science, Bengaluru 560 012, India

**Corresponding author (Email, sekar@iisc.ac.in)*

MS received 31 October 2022; accepted 16 October 2023

In recent years, several experimental evidences suggest that amino acid repeats are closely linked to many disease conditions, as they have a significant role in evolution of disordered regions of the polypeptide segments. Even though many algorithms and databases were developed for such analysis, each algorithm has some caveats, like limitation on the number of amino acids within the repeat patterns and number of query protein sequences. To this end, in the present work, a new method called the internal sequence repeats across multiple protein sequences (ISRMPS) is proposed for the first time to identify identical repeats across multiple protein sequences. It also identifies distantly located repeat patterns in various protein sequences. Our method can be applied to study evolutionary relationships, epitope mapping, CRISPR-Cas sequencing methods, and other comparative analytical assessments of protein sequences.

Keywords. Repeat identification; pattern searching algorithm; protein sequences; computer programs; evolutionary conservation; domain region

1. Introduction

Knowledge of a protein's structure, and function could be gained from its primary sequence's hidden patterns (for example, amino acid sequence repeats, hereafter sequence repeats) (Luo and Nijveen 2014). Due to the advent of high-power digital computing and data mining techniques, the essential hidden features could be fetched easily within a short period of time to address various biological problems. A protein sequence with five or more continuous amino acid residues is known as a sequence motif. If this motif occurs more than once in a single or multiple protein sequence, then it is referred to as an identical repeat. These repeats are vital in various biological processes, like regulating a protein's function and evolutionary trajectory. Further, the repeat patterns in the domain of a protein sequence play a significant role as they are indispensable in regulating most of the biological processes like transcription and translation. Notably, such repeats within a domain can fold independently (greater than 50 amino acid residues), and are evolutionarily conserved (Rajathei *et al.* 2019). Also, due to diploid chromosomes, the number of such repeats in the eukaryotic proteome is higher than in the prokaryotic proteome (Marcotte *et al.* 1999). Further, experimental studies suggest that sequence repeats are more involved in disease conditions; amino acid repeats are more closely linked to many neurodegenerative disorders like Parkinson's disease (Klein and Westenberger 2012).

Contemporary research emphasises the functional role of repeats in both coding as well as non-coding regions (Uthayakumar *et al.* 2012). However, the origin of repeats remain unclear, as it may be either due to gene duplication or by chance. Over the last decade,

Supplementary Information: The online version contains supplementary material available at https://doi.org/10.1007/s12038-023-00410-x.

Published online: 28 February 2024

several articles reported that interactions of proteins with other biomolecules are constantly regulated by repeats such as ankyrin, leucine-rich repeats (LRRs), toroid repeats, etc. On the contrary, a few repeats, like 'gate-keeper' patterns, restrain inter-domain interaction (Luo and Nijveen 2014). Thus, repeats have dual functions in regulating biological processes. As mentioned earlier by Vetting and co-workers (Vetting *et al.* 2006), these studies prompted and motivated us to develop an efficient methodology to identify such repeats in multiple protein sequences of biological importance.

To this end, there are algorithms such as TRUST (Szklarczyk and Heringa 2004), RADAR (Heger and Holm 2000), REPPER (Gruber *et al.* 2005), CENSOR (Kohany *et al.* 2006), SWELFE (Abraham *et al.* 2008), RPS (Babu *et al.* 2011), FAIR (Senthilkumar *et al.* 2010), and RepEx (Michael *et al.* 2019) available in the literature which primarily focus on identifying repeats within individual protein sequences. Some of these algorithms also identify repeats in nucleotide sequences. However, they predominantly focus on detecting sequence repeats of more than 15 residues and might not effectively capture shorter repeats. A notable limitation of these algorithms lies in the input sequence length, with the maximum number of amino acid residues at 6000. While a few algorithms might not impose such a limitation, they only analyse a single amino acid sequence at a time. Furthermore, a notable drawback arises when RADAR and TRUST are capable of analysing multiple protein sequences. These algorithms treat multiple sequences as a single entity, potentially compromising the accurate detection of repeats within individual sequences (Nirjhar *et al.* 2008). Although FAIR and RepEx are trained in detecting direct, inverted/palindrome, mirror repeats within various sequences, they cannot detect repeated patterns that exist across multiple sequences. This underscores the existing algorithms' drawbacks, particularly in terms of repeat length, input sequence length, and handling multiple protein/nucleotide sequences. Our proposed algorithm, internal sequence repeats across multiple protein sequences (ISRMPS) seeks to address these limitations by offering enhanced capabilities in detecting repeats, regardless of repeat length, longer sequences, and effectively analysing multiple protein sequences simultaneously. Based on the present research and to the best of our knowledge, no algorithm or method is available to identify a particular sequence motif (for example, RADHASEKAR) present in a set of protein sequences (for example, 1000 protein sequences). Thus, we have developed a new method to identify a sequence motif available in multiple protein sequences.

2. Materials and methods

The primary objective of this work was to extract identical repeats across multiple protein sequences. To ensure that the proposed approach determines the best results compared to the existing methods, in the present research paper, we have addressed the problem using three different algorithms. The ensemble of these procedures to solve the problem is put forward through an iterative investigation, encompassing multi-angulated and inter-operable capabilities. Initially, the user feeds an input file with multiple protein sequences. The proposed method extracts the identical repeats, their number of occurrences, and their corresponding positions in given sequences. The problem was tackled using the following algorithmic approaches. The only input for these algorithms are protein sequences and the minimum number of amino acids in an identical repeat.

2.1 *Brute-force approach (BFA)*

This straightforward, exhaustive search approach uses multiple for-loops to keep track of sequence repeats. The outer loop considers all sequences; the inner loop is initiated for each sequence, and the innermost loop checks the repeat present in the preceding sequences. If a motif is present in these sequences, it is a repeat and is stored in another list. This process is carried out until the end of the sequence is reached. The output is produced from the appended lists. A detailed explanation of the algorithm is shown in the supplementary file 1. This approach performed well for a limited number of sequences with smaller sequence lengths. The algorithm's time complexity is $O(n^3)$, where n is the length of the most extended sequence. The time complexity of the BFA algorithm is described in figure 1. BFA stores all sequence patterns in the computer's memory, resulting in slow execution and thus leading to a memory error. By applying this approach, the sample input and output results are shown in figure 2.

2.2 *Suffix tree-based approach (STBA)*

This approach uses the concept of a suffix tree. We constructed a suffix tree using Ukkonen's algorithm (Ukkonen 1995) in linear time. After extracting all

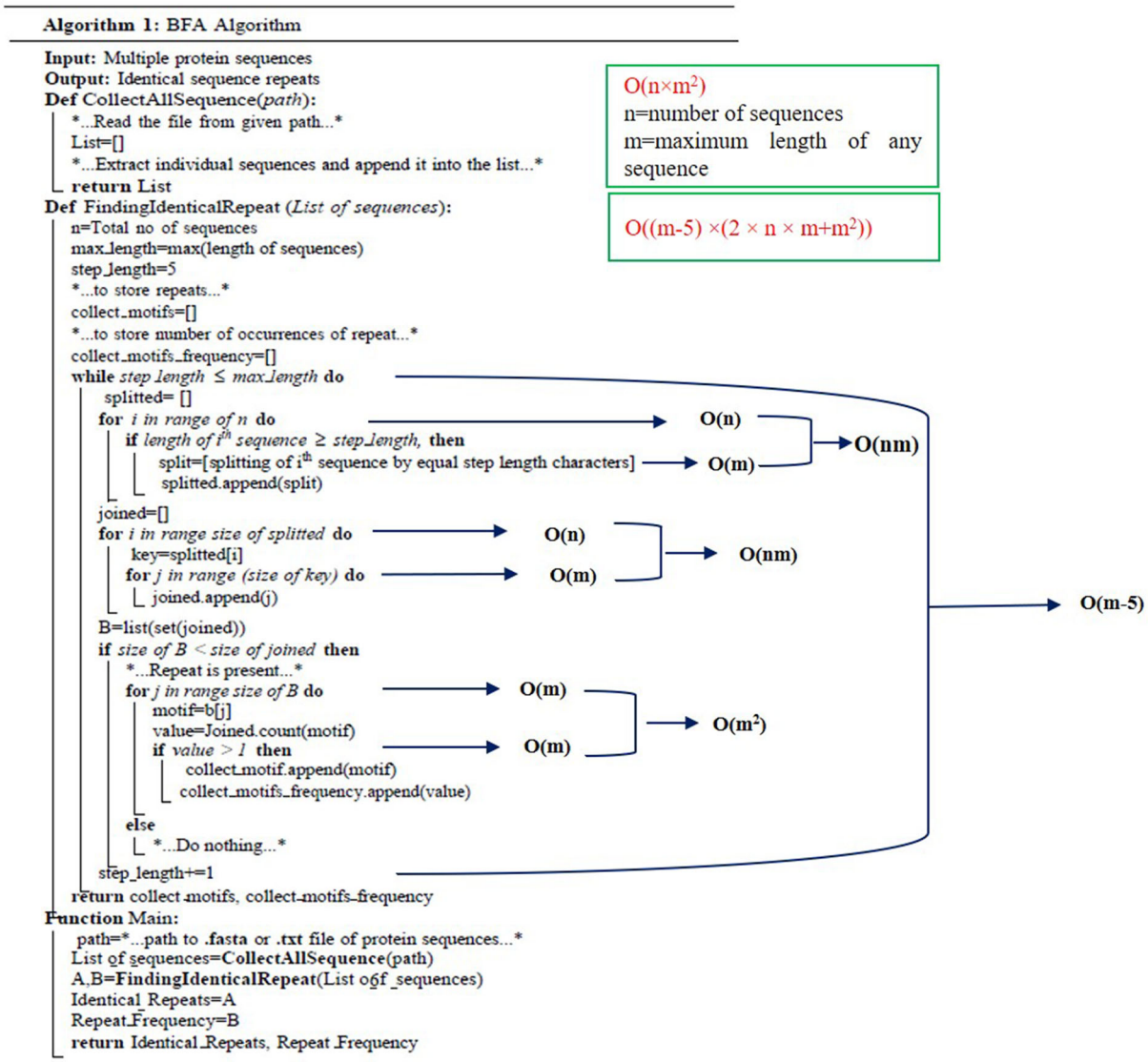

Figure 1. Time complexity of the BFA algorithm.

biological sequences from the input, they were concatenated using unique special characters. This resulted in a long string (sequence) called *CombinedSeq*, which was used to construct the suffix tree. All possible leaf nodes (a node with no child) corresponding to each internal node were counted. Later, a list of all potential internal nodes was generated. The path from the root to this internal node will be repeating, and residues along this path will be an identical repeat. The number of leaves attached to each internal node will be the number of occurrences of that particular repeat. Further, the breadth-first search (BFS) approach was used to reach each internal node from the root and save its path. The residue associated with this path will be the corresponding repeat. Four unique special characters join sequences in this *CombinedSeq* [{#, @, &, !}-MYSFVSTGTNSVLLGTCAY#MYSFVSTXX@LLVTLMYSFVTGTCAYM&SFYVGTCAYMYSMYSFV!]. The suffix tree construction of *CombinedSeq* is shown in figure 3.

2.2.1 *Performance analysis of STBA:* The bottleneck of STBA is space consumption. Further, the time complexity increases exponentially if the subset removal step is considered. O(n) time is needed for a suffix tree construction using Ukkonen's algorithm (Ukkonen 1995). Here, 'n' is the length of the combined protein sequences. O(n) time is needed for BFS to count the number of leaves attached to each internal node. O(n) time is needed for traversing from the root node to each internal node root. Thus, the overall time complexity is O(n×num), where 'num' is the number

(a) Sample Input Sequence

Seq1 – MYSFVSTGTNSVLLGTCAY

Seq2 – MYSFVSTXX

Seq3 – LLVTLMYSFVTGTCAYM

Seq4 – SFYVGTCAYMYSMYSFV

(b) Sample Output Repeats

	1	2	3	4
Repeat	*MYSFVST*	*GTCAYM*	*GTCAY*	*MYSFV*
No. of Amino Acids in a repeat	7	6	5	5
No. of Occurrences	2	2	3	4
Positions	Seq1 [1,7] Seq2 [1.7]	Seq3 [12,17] Seq4 [5.10]	Seq1 [15,19] Seq3 [12.16] Seq4 [5,9]	Seq1 [1,5] Seq2 [1.5] Seq3 [6,10] Seq4 [13.17]

Figure 2. (**a**) Screenshot of the sample input sequences submitted to ISRMPS. (**b**) Sample results retrieved from ISRMPS using BFA.

of all internal nodes. Since the value of num can be as large as n–1, the worst-case time complexity of the algorithm is still quadratic, $O(n^2)$.

2.3 *Rabin–Karp based method (RKBM)*

The Rabin–Karp algorithm is a pattern-searching algorithm developed by Richard M. Karp and Michael O. Rabin (Karp and Rabin 1987). This approach uses a hashing method to locate an exact match for a pattern in a given string. A rolling hash prunes the text positions that do not match the pattern. Further, it is rolled to the next position, looking for a match. This process is continued until the end of the given text. The RKBM uses the concept of hash and sliding windows. The size is fixed for the sliding window and slides over the input text. For each slide, the substring is hashed to numeric values efficiently. The hashing technique is used here to have a significantly lower chance of hash collision. Hash values are calculated by sliding the window one step ahead based on the previous hash values, and all values are stored further. Searching is halted once these values are collected in a complete text for a given window size (pattern length). The same hash value and their index position in a given text were searched in the search step. When the number of distinct hashes, i.e, the number of patterns available, was found to be more than two in a given text, simultaneously a distinct index position was retrieved for that unique hash value. The overview of the proposed methodology is illustrated in figure 4. The user feeds a fasta format file containing multiple protein sequences. The sequences and the IDs are extracted and concatenated with special characters (a list of special characters is pre-defined). Hashing is performed in the next step, assigning a numeric value for each concatenated sequence. Later, the longest repeated sequence motif and the number of such sequence motifs are captured using the Rabin–Karb algorithm. The minimum motif length considered here is five since to form a regular secondary structure, a minimum of five amino acid residues is required (Worsfold *et al.* 2019). Thus, if a motif length is less than 5, the sequence is not considered for further comparison in subsequent iterations. However, the motif is stored if the length is greater than 5. Further, duplicates of a particular motif are eliminated by retaining only one occurrence. In the final stage, the sequence of all repeats, their frequency of occurrence, and their position in the original sequence are retrieved as the final output. A detailed explanation of the algorithm is shown in supplementary file 1.

2.3.1 *A detailed step-wise analysis of RKBM:*

Step 1. Extraction of input protein sequences and their sequence ID.

The fasta format file containing multiple protein sequences is the input. All protein sequences and IDs are collected as vector strings. The sequence IDs are assigned a number in increasing order.

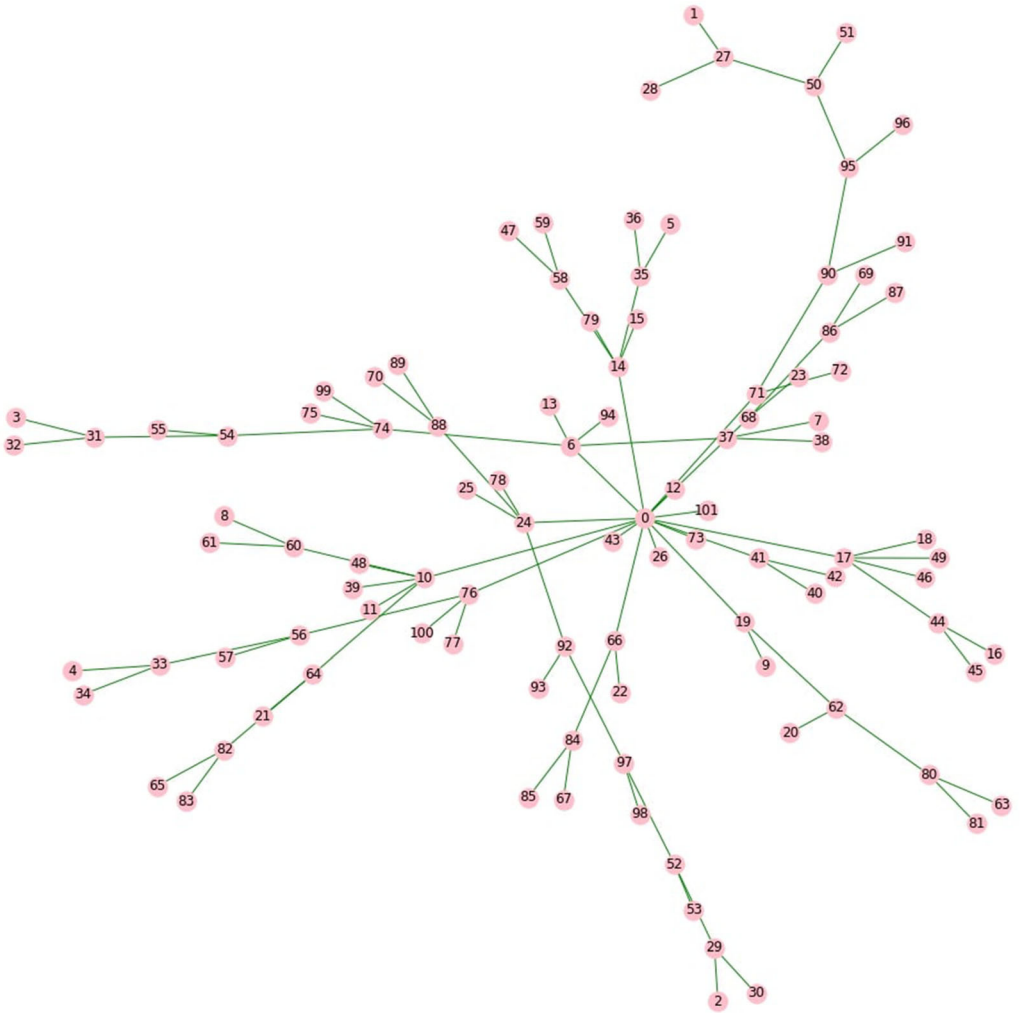

Figure 3. Representation of the suffix tree.

Step 2. Concatenation of input protein sequences.

All protein sequences are concatenated using special characters. For example, 10 distinct special characters concatenate 10 protein sequences.

Suppose the number of protein sequences exceeds the number of unique characters available. A combination of three distinct characters is created from the list of predefined special characters. For example, we have 20 protein sequences, but the number of unique characters is 15. We combine three special symbols and use them to concatenate input protein sequences once all special characters are utilized.

Step 3. Hashing of each character present in a concatenated sequence with a numeric number.

Protein sequences are made of 20 amino acids, each representing one character of the english alphabet (A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y and V). Each alphabet is hashed with its ASCII code (a unique number). Once all the characters are hashed, the special characters are hashed, increasing from the highest ASCII number of any amino acid alphabet.

Step 4. Find the longest duplicated string in a concatenated string using RKBM.

Step 5. Extract all the repeated strings using recursion.

Once the longest repeated string and its total number of occurrences are known, the old concatenated string is recursively modified.

2.3.2 *Subset removal using recursion:* For example, consider the longest repeated string found ‘n’ times in the original concatenated string. We have modified this concatenated string by removing the first ‘n–1’ occurrences of the repeated string and joined this residual string with distinct special characters. This newly formed concatenated string is used in the next step for extracting the longest repeated string using RKBM.

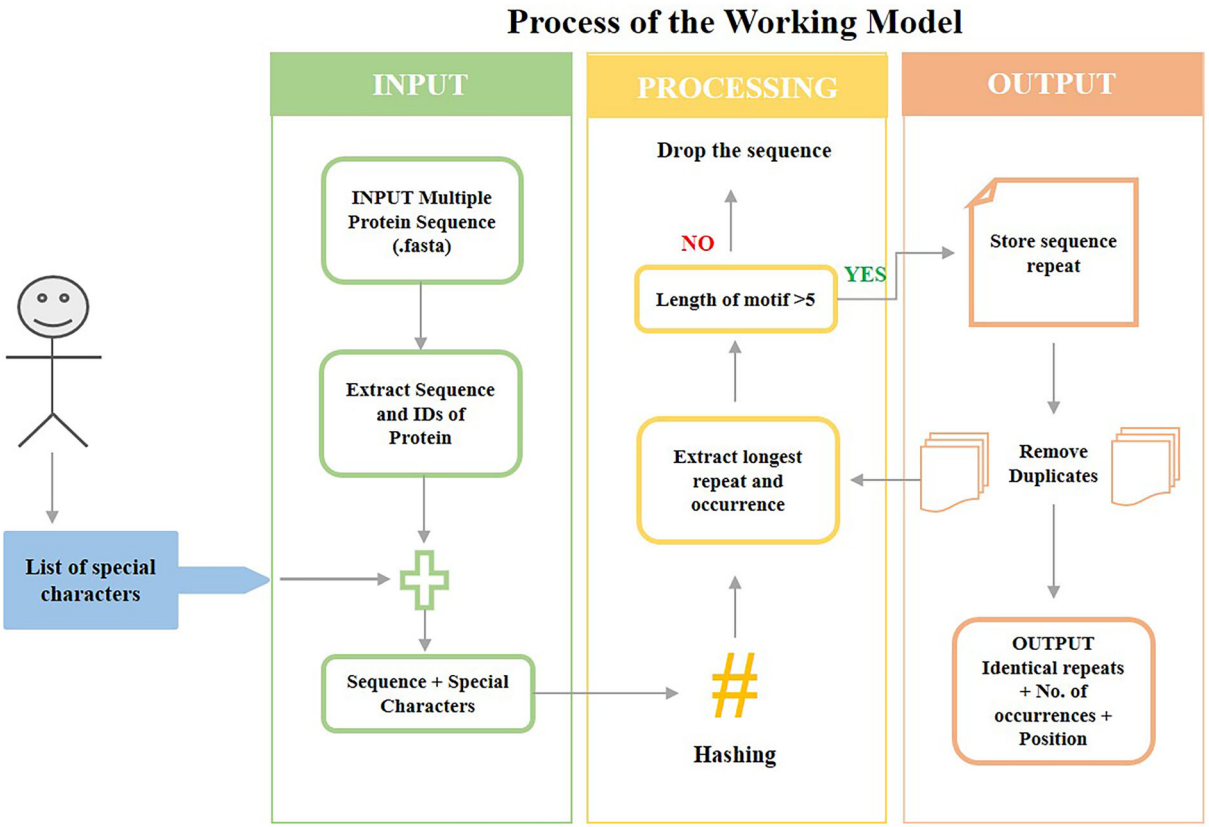

Figure 4. Workflow of the RKBM approach adopted in the present study.

This process is done recursively until the repeated string length obtained is greater than or equal to 5. Lists of 2500 special characters are predefined for this research purpose. The number of total possible three-character combinations

$$2500C3 = \frac{2500!}{3!(2500-3)!} = 2.601 \times 10^9$$

Hence, there is no limit on the number of sequences fed to the system, and it is scalable to extract repeats across a virtually infinite number of protein sequences.

2.3.3 *Performance analysis of RKBM:* Let us have k input protein sequences of varying lengths (n1, n2, n3, n4, n5, n6, n7, ... nk). After concatenating with distinct special characters, its combined sequence length (N) becomes as follows in eq. (1),

$$N = \sum_{i=1}^{k} n_i + k \tag{1}$$

The average time complexity could be calculated using eq. (2),

$$T(N) = N\log N + \sum_{j=1}^{m}\left\{N - \left(\sum_{i=1}^{n_{jr}} L_j(F_{ji}-1)\right) * \log\left(N - \left(\sum_{i=1}^{n_{jr}} L_j(F_{ji}-1)\right)\right)\right\} \tag{2}$$

where, N is concatenated sequence length, L_j is length of identical sequence repeat [L_j>4; ∀j], n_{jr} is total number of distinct repeats having length L_j, and F_{ji} is total number of occurrences of ith repeat having length L_j. The code is written in Python language 3.8.0. The performance measurement and execution of the algorithm are done on the Windows Operating system (specifications: Processor-11th Gen Intel(R) Core (TM) i5-1135G7 @ 2.40 GHz 1.38 GHz, RAM-8GB, 64bit OS) as well as on Linux (specifications: processor-Intel(R) Core (TM) i7-6700 CPU 3.40 GHz8, Memory-31.3 GB, 64bit OS) operating systems. The ISRMPS algorithm is available to any user worldwide by downloading the files in tar format. Interested researchers may write to K Sekar to obtain the source code of this algorithm. Also, users are requested to cite this article in their research publications.

3. Results and discussion

To the best of our knowledge, the ISRMPS algorithm developed in the present study is the first reported method to date on repeat searches across multiple protein sequences. Hence, to validate the usefulness of our ISRMPS method, we compared it with the existing similarity search algorithms such as BLAST and CLUSTALW for protein sequences. The local similarity among two or more query protein sequences can be identified using BLAST's iterative pairwise alignment approach. However, in the case of large multiple protein queries, it displays a CPU (processing time) limit exceeded error if it exceeds 1 h. To overcome such errors, the user should reduce the number of query protein sequences (Altschul *et al.* 1990). Likewise, the user can deploy the CLUSTALW to identify the conservation across sequences (Thompson *et al.* 1994). Specifically, it uses the progressive alignment method to identify similar patterns only in vicinity regions, not distant regions (Mansour 2008). For example, two proteins (Q9BT76 and Q96M86) were submitted to BLASTP and ISRMPS to find the common local pattern. The corresponding results are shown in figure 5(a) and 5(b), respectively. The results indicate that ISRMPS successfully identifies the presence of repeats ('PGPGP') across two protein sequences even though they are located in distant regions. Specific instances are reported where protein sequences are aligned using CLUSTALW and manually corrected to evaluate repeats across protein sequences (Tanabe *et al.* 2012). In such cases, our method directly identifies repeats without any manual involvement. This is further discussed in case study 2 (later section). The ISRMPS method is better than BLAST and CLUSTALW considering the above-mentioned limitations.

3.1 *Complexity analysis of the ISRMPS algorithm*

In this section, we assessed the efficiency of the proposed algorithm through a time complexity analysis, a widely accepted method in computational biology and related fields for evaluating algorithmic performance. Our evaluation involved a dataset of 550 protein sequences randomly selected from the UniProtKB database, divided into 10 sets with varying sequence lengths. We conducted experiments on three distinct processors: P1 with an Intel(R) Core (TM) i7-6700 CPU @ 3.40 GHz, 24 GB of RAM, and running a 64-bit system; P2 equipped with an Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20 GHz, 32 GB of RAM, and a 64-bit system; and P3 featuring an AMD Ryzen 9 3900X 12-Core, 64 GB of RAM, and a 64-bit system. Results of our experiments, including the elapsed time for each processor are summarized in table 1. Additionally, we visualized the runtime of concatenated sequence length derived from the dataset shown in figure 6. Notably, as the number of concatenated sequences increases, the runtime exhibits quadratic growth. The maximum repeat length discovered among the concatenated sequences of length 51,148 is characterized by a repeat length of 140 residues and occurs twice across 100 sequences. This particular repeat was identified within the protein sequences P0DI81 and P0DI82 from *Homo sapiens*. In summary, based on our analysis, the algorithm's performance is closely tied to two main factors: (i) the size of the input dataset and (ii) the availability of computational resources. We have observed a clear quadratic relationship between the number of concatenated sequences and runtime, emphasizing the significance of these factors in determining efficiency. Additionally, we identified a substantial repeat pattern within the dataset. This study adds value to the algorithm's utility and sheds light on specific biological insights.

3.2 *Case study 1: Tuberculosis*

Tuberculosis is caused by *Mycobacterium tuberculosis (M.tb)*. The genome of *M.tb* encodes a unique protein family known as the PGRS family, with largely unexplored functions (Meena 2015). Evidence suggests that PE-PGRS proteins promote bacterial survival and modulate host immunity, metabolism, cell death, and autophagy (Meena 2015). The PGRS domain, glycine, and alanine are frequently found as 'GGAGGX' and 'GGNGGX' patterns repeated in the protein sequence (glycine-rich motifs, X represents any amino acid). The glycine-rich motif repeats in the PGRS domain are implicated in forming a Ca^{2+} binding structure. This is a parallel β-helix structure, typical of calcium-binding proteins. Meena (2015) highlighted the potential of PE-PGRS as novel targets of anti-mycobacterial intervention for TB control.

Our method can extract all the repeat motifs across all protein sequences of the PE-PGRS multigene family of *M.tb*. A summary of the result is represented in table 2. The detailed results can be retrieved from the supplementary file 2.

3.3 *Case study 2: Malaria*

Malaria protein, *Plasmodium falciparum* serine-repeat antigen (SERA) is a potential vaccine candidate

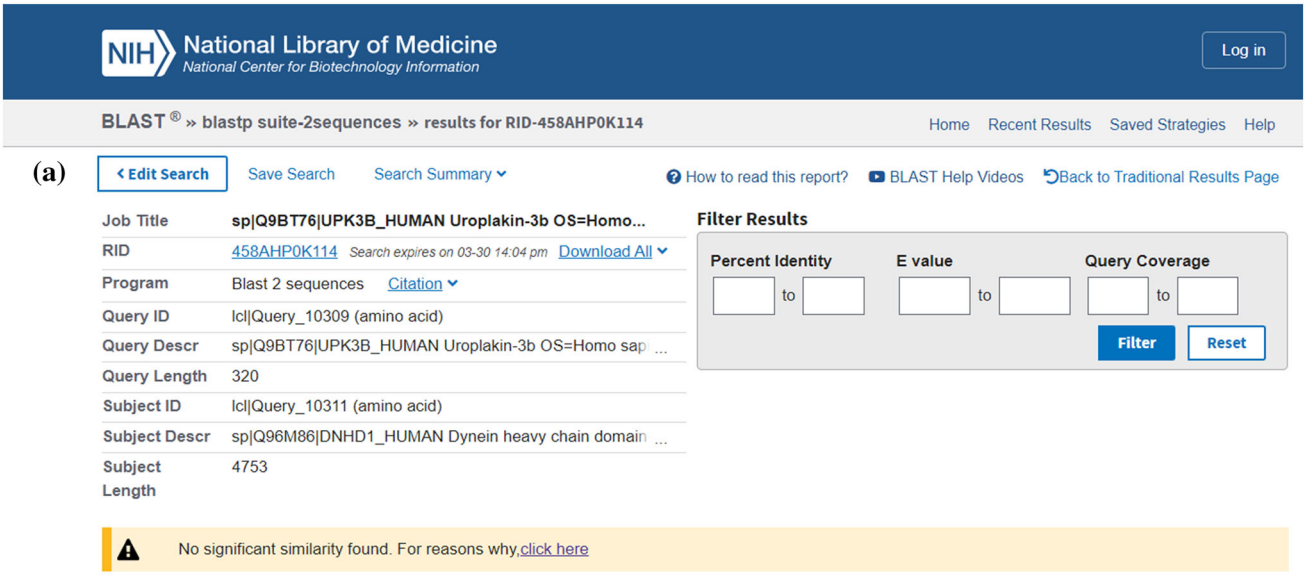

Figure 5. (**a**) Results retrieved from BLASTP and (**b**) Output of ISRMPS using an ensemble approach.

Table 1. Time elapsed for protein sequences in different processors (P1, P2, and P3)

Set number	Number of protein sequence	Concatenated sequence length	Number of repeats found	Repeat max length	Time elapsed (min)		
					P1	P2	P3
1	10	5477	23	12	0.3953	0.2232	0.1418
2	20	15871	209	20	2.47125	3.0675	1.9709
3	30	13814	109	15	0.8091	1.000	0.66686
4	40	21027	213	44	4.02466	4.9670	3.24286
5	50	27923	420	12	6.39008	7.9100	5.3207
6	60	34598	540	18	13.2342	16.3943	10.9252
7	70	45134	828	21	23.1954	27.6213	18.7087
8	80	54256	1289	21	30.9767	38.18919	26.5200
9	90	39991	902	61	25.4335	31.8467	22.0980
10	100	51148	1186	140	47.7358	59.4116	41.0092

representing antigenic variations across protein sequences collected from various geographical locations worldwide (Tanabe *et al.* 2012). The PfSERA5 protein length in standard reference strain (3D7) is 997 amino acids. Nevertheless, the protein length varies from 915 aa to 1047 aa in protein sequences collected from field isolates represented from various geographical locations worldwide, like Africa, South-east Asia, and South America. This diversity is mainly due to repeat variations observed in two regions in the N-terminal 47 kDa domain. One of the repeat regions represent repeats of eight amino acids called octamer repeats. Variations within octamer repeats and repeat numbers have been identified (Tanabe *et al.* 2012). The

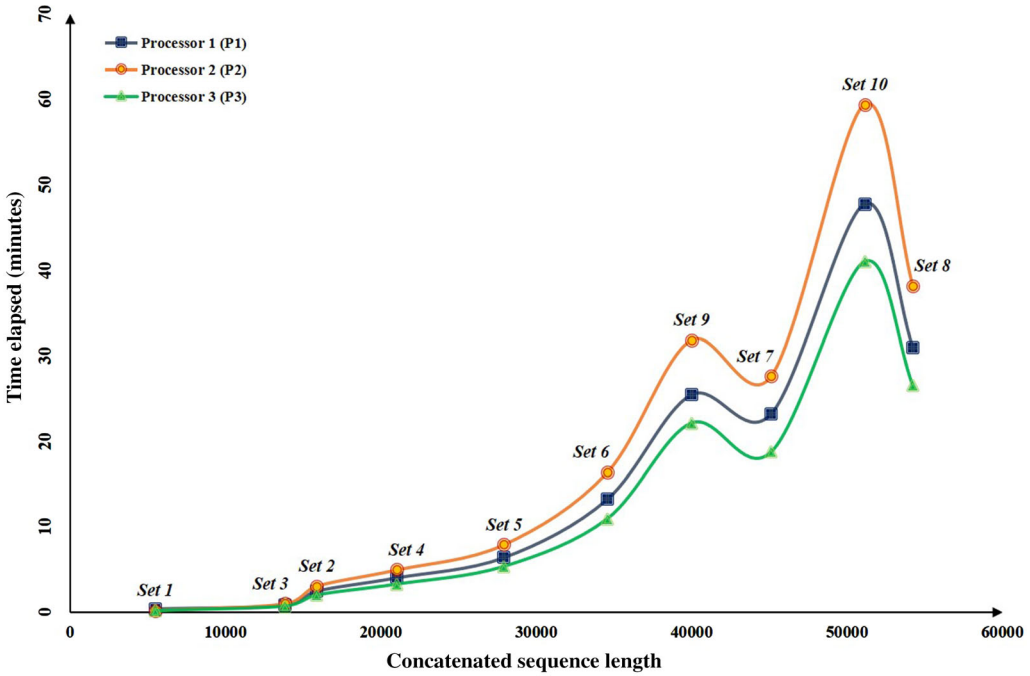

Figure 6. Time complexity cure for ISRMPS algorithm in different processors (PI, P2, and P3).

Table 2. Output obtained from the protein sequences associated with tuberculosis

Dataset Info	Value
Proteins sequence	*Mycobacterium tuberculosis*
Total number of input protein sequences	60
Total number of repeats found	1688
Maximum number of amino acids in the repeat	375
Minimum number of amino acids in the repeat	5
Maximum number of occurrences	1538
Minimum number of occurrences	2

Table 3. Output obtained from protein sequences associated with malaria

Dataset Info	Value
Proteins sequence	Malaria protein sequences
Total number of input protein sequences	36
Total number of repeats found	141
Maximum number of amino acids in the repeat	977
Minimum number of amino acids in the repeat	5
Maximum number of occurrences	97
Minimum number of occurrences	2

most varying octamer repeat type is 'PQGSTGAS'. We reappraised this sequence analysis using our method. Our method could identify all the haplotypes with this 'PQGSTGAS' repeat type variation. A summary of the result is provided in table 3. Detailed results can be retrieved from supplementary file 3.

3.4 *Case study 3: Ovarian cancer*

A list of 326 ovarian cancer proteins was retrieved from the UniProt and given as input to the ISRMPS method. Results indicate that across all the cancer proteins, hexamer repeat 'GSTAPP' was found 42 times in mucin cancer proteins. The alpha subunit of such protein is proven to have a significant role in protecting the epithelial cells against various types of microbial infection due to their unique adhesive property. Also, several studies demonstrated that they are promising cancer markers as their abnormal expression leads to the pathogenic effect. The detailed results can be retrieved from the supplementary file 4.

3.5 *Case study 4: Breast cancer*

Inherited mutations in *BRCA1* and *BRCA2* genes are responsible for the onset of breast cancer. To identify

the repeats found across all the 1789 breast cancer proteins, they are given as input to the ISRMPS method. The results indicate that the 'TGEKP' pentapeptide occurs 159 times across all the breast cancer proteins. This pentapeptide occurs near the zinc finger motif in most protein sequences. A summary can be retrieved from supplementary file 5.

4. Conclusion

The present study proposes an efficient method (ISRMPS) to find identical sequence repeats across multiple protein sequences. As stated above, the input for our proposed method is only the protein sequences and the minimum number of amino acid residues present in an identical repeat. Identifying such repeats in the proteome aids in discovering various significant conditions like microsatellites in the coding regions, disordered regions, and positive selection pressure in the locus. In addition, our method can be effectively used in the comparative study of orthologous, paralogous, and analogous protein sequences. Thereby, the hidden patterns of the evolutionary process could be revealed. Similarly, our method can also be deployed to extract repeats within the domain region, as they play an important role in diseases. Eventually, it could be applied to predict the functional annotation of a hypothetical protein sequence. This work will be extended to identify repeats across multiple DNA sequences.

Acknowledgements

KS thanks the ICMR for funding the project 'Do protein sequence repeats play a role in biological process and disease conditions' (ISRM/12(34)/2020). KS, RS and DR thank the Center for Development of Advanced Computing (CDAC) for funding the project 'An Indian Initiative on setting up a high-fidelity structural data archival/retrieval system for Life Sciences-(PDBi)'. RS thanks the Department of Science and Technology-Science and Engineering Research Board (DST-SERB), New Delhi, India, for providing research grant and postdoctoral fellowship (PDF/2019/000254). AM thanks Dr D S Kothari Postdoctoral Fellowship (BL/18-19/0320), funded by the University Grants Commission (UGC). All the authors thank the Department of Computational and Data Sciences, Indian Institute of Science, Bengaluru, India, for providing the necessary support.

Author contributions

Prof. SK conceptualized the study. VKM and RS devised the methodology. AM, MS, RS and DR assessed the algorithm, methods. CNR and AM contributed to case studies. AM and RR wrote the manuscript. MS and CNR reviewed the manuscript.

Declarations

Conflict of interest The authors acknowledge that there is no conflict of interest related to financial and research interest related to this manuscript.

References

Abraham A-L, Rocha EPC and Pothier J 2008 Swelfe: a detector of internal repeats in sequences and structures. *Bioinformatics* **24** 1536–1537

Altschul SF, Gish W, Miller W, *et al*. 1990 Basic local alignment search tool. *J. Mol. Biol.* **215** 403–410

Babu V, Uthayakumar M, Kirti Vaishnavi M, *et al*. 2011 RPS: Repeats in protein sequences. *J. Appl. Crystallogr.* **44** 647–650

Gruber M, Söding J and Lupas AN 2005 REPPER—repeats and their periodicities in fibrous proteins. *Nucleic Acids Res.* **33** W239–W243

Heger A and Holm L 2000 Rapid automatic detection and alignment of repeats in protein sequences. *Proteins* **41** 224–237

Karp R and Rabin MO 1987 Efficient randomized pattern-matching algorithms. *IBM J. Res. Dev.* **31** 249–260

Klein C and Westenberger A 2012 Genetics of Parkinson's disease. *Cold Spring Harb. Perspect. Med.* **2** a008888

Kohany O, Gentles AJ, Hankus L, *et al*. 2006 Annotation, submission and screening of repetitive elements in Repbase: RepbaseSubmitter and Censor. *BMC Bioinform*. **7** 474

Luo H and Nijveen H 2014 Understanding and identifying amino acid repeats. *Brief. Bioinform.* **15** 582–591

Mansour A 2008 ClustalW©: Widespread Multiple sequences alignments program. *J. Cell Mol.* **7** 81–82

Marcotte EM, Pellegrini M, Yeates TO, *et al*. 1999 A census of protein repeats. *J. Mol. Biol.* **293** 151–160

Meena LS 2015 An overview to understand the role of PE _ PGRS family proteins in *Mycobacterium tuberculosis* H 37 R v and their potential as new drug targets. *Biotechnol. Appl. Biochem.* **62** 145–153

Michael D, Gurusaran M, Santhosh R, *et al*. 2019 RepEx: A web server to extract sequence repeats from protein and DNA sequences. *Comput. Biol. Chem.* **78** 424–430

Nirjhar B, Chidambarathanu N, Daliah M, *et al*. 2008 An Algorithm to find all identical internal sequence repeats. *Curr. Sci.* **95** 188–195

Rajathei DM, Parthasarathy S and Selvaraj S 2019 Identification and analysis of long repeats of proteins at the domain level. *Front. Bioeng. Biotechnol.* **7** 250

Senthilkumar R, Sabarinathan R, Hameed BS, *et al*. 2010 FAIR: a server for internal sequence repeats. *Bioinformation* **4** 271–275

Szklarczyk R and Heringa J 2004 Tracking repeats using significance and transitivity. *Bioinformatics* **20** (Suppl 1) i311–i317

Tanabe K, Arisue N, Palacpac NM, *et al*. 2012 Geographic differentiation of polymorphism in the *Plasmodium falciparum* malaria vaccine candidate gene SERA5. *Vaccine* **30** 1583–1593

Thompson JD, Higgins DG and Gibson TJ 1994 CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* **22** 4673–4680

Ukkonen E 1995 On-line construction of suffix trees. *Algorithmica* **14** 249–260

Uthayakumar M, Benazir B, Patra S, *et al*. 2012 Homepeptide repeats: implications for protein structure, function and evolution. *Genom. Proteom. Bioinform.* **10** 217–225

Vetting MW, Hegde SS, Fajardo JE, *et al*. 2006 Pentapeptide repeat proteins. *Biochemistry* **45** 1–10

Worsfold P, Townshend A, Poole CF, *et al.* 2019 *Encyclopedia of analytical science*, 3rd edition (Elsevier)

Corresponding editor: Deepesh Nagarajan