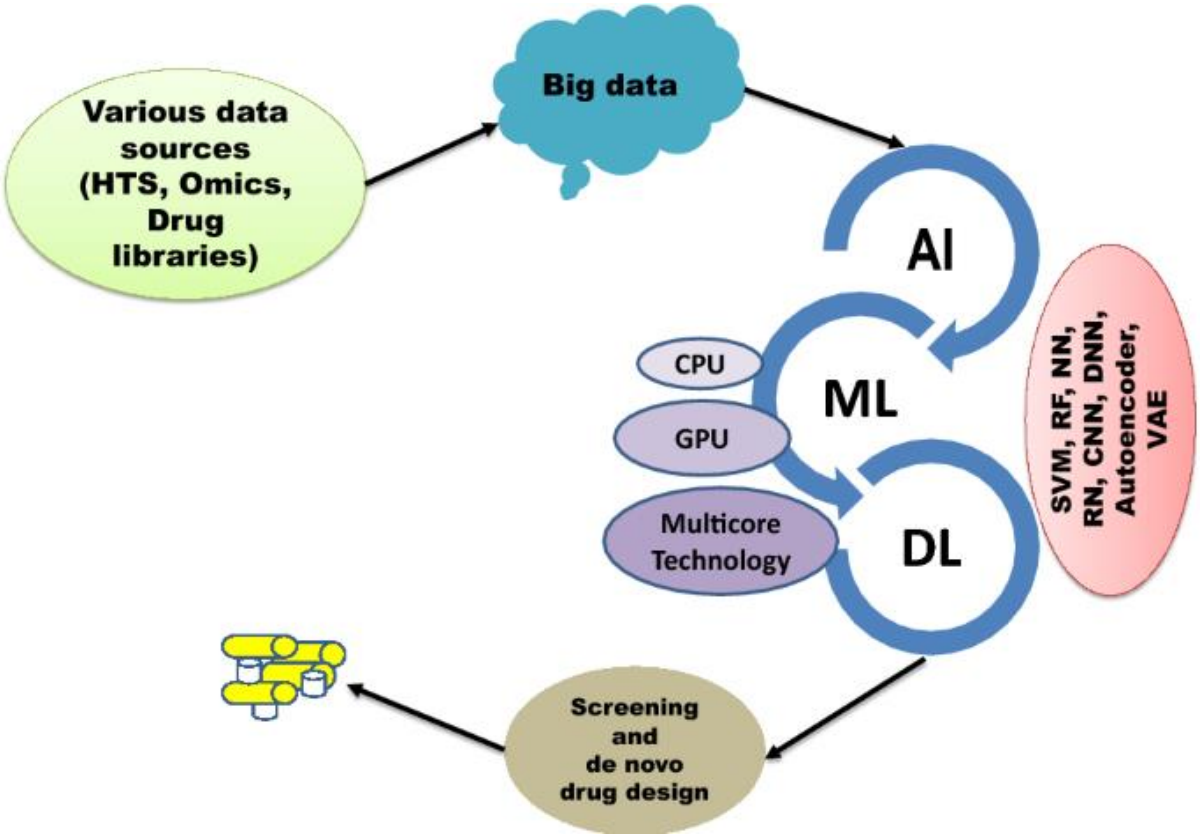
Biological sequence alignment and drug design using soft computing techniques

In bioinformatics, a sequence alignment is a way of arranging the sequences of DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences.

The accumulation of massive data in the plethora of Cheminformatics databases has made the role of big data and artificial intelligence (AI) indispensable in drug design. This has necessitated the development of newer algorithms and architectures to mine these databases and fulfil the specific needs of various drug discovery processes such as virtual drug screening, de novo molecule design and discovery in this big data era. The development of deep learning neural networks and their variants with the corresponding increase in chemical data has resulted in a paradigm shift in information mining pertaining to the chemical space. The present review summarizes the role of big data and AI techniques currently being implemented to satisfy the ever-increasing research demands in drug discovery pipelines.



Application of soft computing becomes relevant for solving some Bioinformatics and molecular biology problems. Protein classification leads to identification and proper functional assignment of uncharacterized proteins with a final goal towards finding

homologies and drug discovery. Again, structure based ligand design is one of the crucial steps in rational drug discovery, where a small molecule is designed by targeting the structure and biochemical properties of the target. The application of soft computing offers an on promising approach to achieve efficient and reliable heuristic solution. On the other side the continuous development of high quality biotechnology, e.g. micro-array techniques and mass spectrometry, which provide complex patterns for the direct characterization of cell processes, offers further promising opportunities for advanced research in bioinformatics. So One important sub-discipline within bioinformatics involves the development of new algorithms and models to extract new, and potentially useful information from various types of biological data including DNA(nucleotide sequences) and proteins (amino acid sequences). Analysis of these macromolecules is performed both structurally and functionally using the major components of soft computing like Fuzzy Sets (FS), Artificial Neural Networks (ANN), Evolutionary Algorithms (EAs) (including genetic algorithms (GAs), genetic programming (GP), evolutionary strategies (ES)), Support Vector Machines (SVM), Wavelets, Rough Sets (RS), Simulated Annealing (SA), Swarm Optimization (SO), Memetic Algorithms (MA), Ant Colony Optimization (ACO) and Tabu Search (TS).

Need for Soft Computing techniques in Bioinformatics

The different tasks involved in the analysis of biological data include Sequence alignment, genomics, proteomics, DNA and protein structure Prediction, gene/promoter identification, phylogenetic analysis, analysis of Gene expression data, protein Folding, docking and molecule and Drug design. Data analysis tools used earlier in bioinformatics were mainly based on statistical techniques like regression and estimation. Soft computing in bioinformatics can be used in handling large, complex, inherently uncertain, data sets in biology in a robust and computationally efficient manner thus fuzzy sets (soft computing technique) can be used as a natural framework for analyzing them. Most of the bioinformatic tasks involve search and optimization of different criteria (like energy, alignment score, overlap strength), while requiring robust, fast and close approximate solutions. Evolutionary and other soft computing search algorithms like TS, SA, ACO, PSO etc. provide powerful searching methods to explore huge and multi-model solution spaces.

Artificial intelligence and machine learning methods have been used successfully in analyzing sequence data and have played an important role in elucidating many biological functions, such as protein functional classification, active site recognition, protein structural features identification, and disease prediction outcomes.

3.1 SCMs applications on Sequence alignment

Sequence alignment is a common task in bioinformatics. It plays an essential role in detecting regions of significant similarity among a collection of primary sequences of nucleic acids or proteins. If they are highly similar, then they have similar 3D structures or share similar functions. Given a family $S = (S_1,..., S_N)$ of N sequences, the problem can formally be represented as a set of sequences, and each sequence has its own length. The characters of sequences are defined over an alphabet Σ including a gap symbol denoted by '–', which is a molecular biology term, indel (insertion or deletion). The indels indicate that some parts of a sequence are inserted or deleted. The sequence is either a DNA, ribonucleic acid (RNA), or amino acid (protein) sequence. The nucleotide bases are adenine (A), cytosine (C), guanine (G), thymine (T), and uracil (U). The alphabet is *{*A, C, G, T*}* and *{*A, C, G, U*}* for DNA and RNA, respectively. The sequence alignment problem has two computational approaches: local alignment and global alignment. Global alignment is used Needleman-Wunch algorithms. Local alignment is used Smith-Waterman algorithms. In global alignment, sequences are aligned as a whole, whereas in local sequence alignment, similarities detected locally between sequences are aligned [50]. Assume that 2 DNA sequences are given as S1 = *{GCTG*AACG*}* and S2 = *{CTATAATC}* with lengths |S1| and |S2|, respectively. This pair of sequences can be aligned as shown in Figure 2.

An alignment without gaps: GCTGAACG
CTATAATC

An alignment with gaps: GCTGA--A--CG
--CT--ATAATC

Figure 2. Sequence alignment of 2 DNA sequences

Gap is sequence of g missing characters inserted in a string to achieve alignment. Gaps are assigned with two kinds of negative scores: Gap-open penalty: negative score associated with the initiation of a gap (i.e., with the first missing character), and Gap-extension penalty: negative score associated with each additional missing character.

For reasons of computational complexity, sequence alignment is divided into two categories:

- Pairwise alignment (i.e., the alignment of two sequences).
- Multiple-sequence alignment (i.e., the alignment of three or more sequences).

Pairwise alignment problems have exact solutions by using dynamic programming. Multiple-sequence alignment problems have approximate (heuristic) solutions. The function of sum-of-pairs is the most popular scoring method for evaluation of the quality of the alignment. The goal of general multiple sequence alignment algorithms is to find out the alignment with the highest sum-of-pairs [50]. There are numerous existing methods for sequence alignment. The efficiency of an alignment is assessed by the application of SCMs.

Neural networks, which are one of the systems employed in AI, are used to identify chemical structures that can have medical relevance. Successful training of neural networks must be preceded by the acquisition of relevant information about chemical compounds, functional groups, and their possible biological activity. In general, a neural network requires a large set of training data, which must contain information about the chemical structure–biological

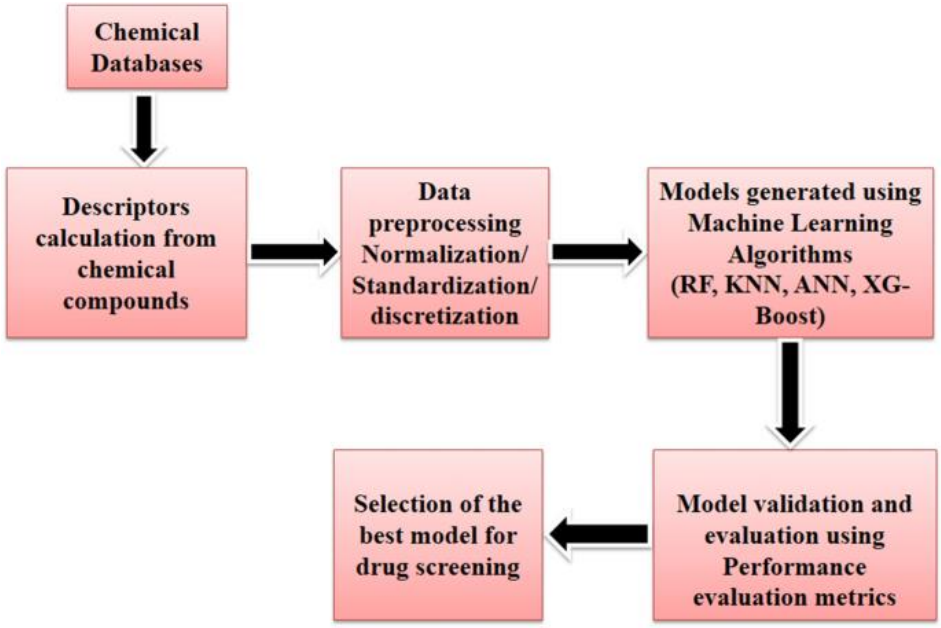
activity relationship. The data can come from experimental measurements, but can also be generated using appropriate quantum models.

Artificial neural networks are abstract models that mimic the complex structure and functioning of the brain. They are widely being used in the fields of systems biology and drug discovery for resolving complexity associated with mathematical models, virtual screening of compounds, deciphering quantitative structure–activity relationships, estimation of pharmacokinetic and pharmacodynamics properties, and during formulation development. With a number of variables deciding the outcome, neural networks are excellent in deciphering nonlinear relationships among the variables and predicting the outcome way before in the drug discovery process. The traditional drug discovery process often proves to be lengthy, expensive, and difficult. The present chapter discusses the possibility of using artificial neural networks to improve the efficiency and speed of therapeutic discovery.

The advancement of ANN, called deep neural network (DNN), is now gaining attention for its successful application in drug discovery-related areas such as, to generate novel molecules, predicting the biological activity as well as the absorption, distribution, metabolism, excretion and toxicity (ADMET) properties of the drug candidate molecules. Like the ML approach, deep learning method was found to be effective in building the QSAR/QSAP models.

Advent of AI in drug design

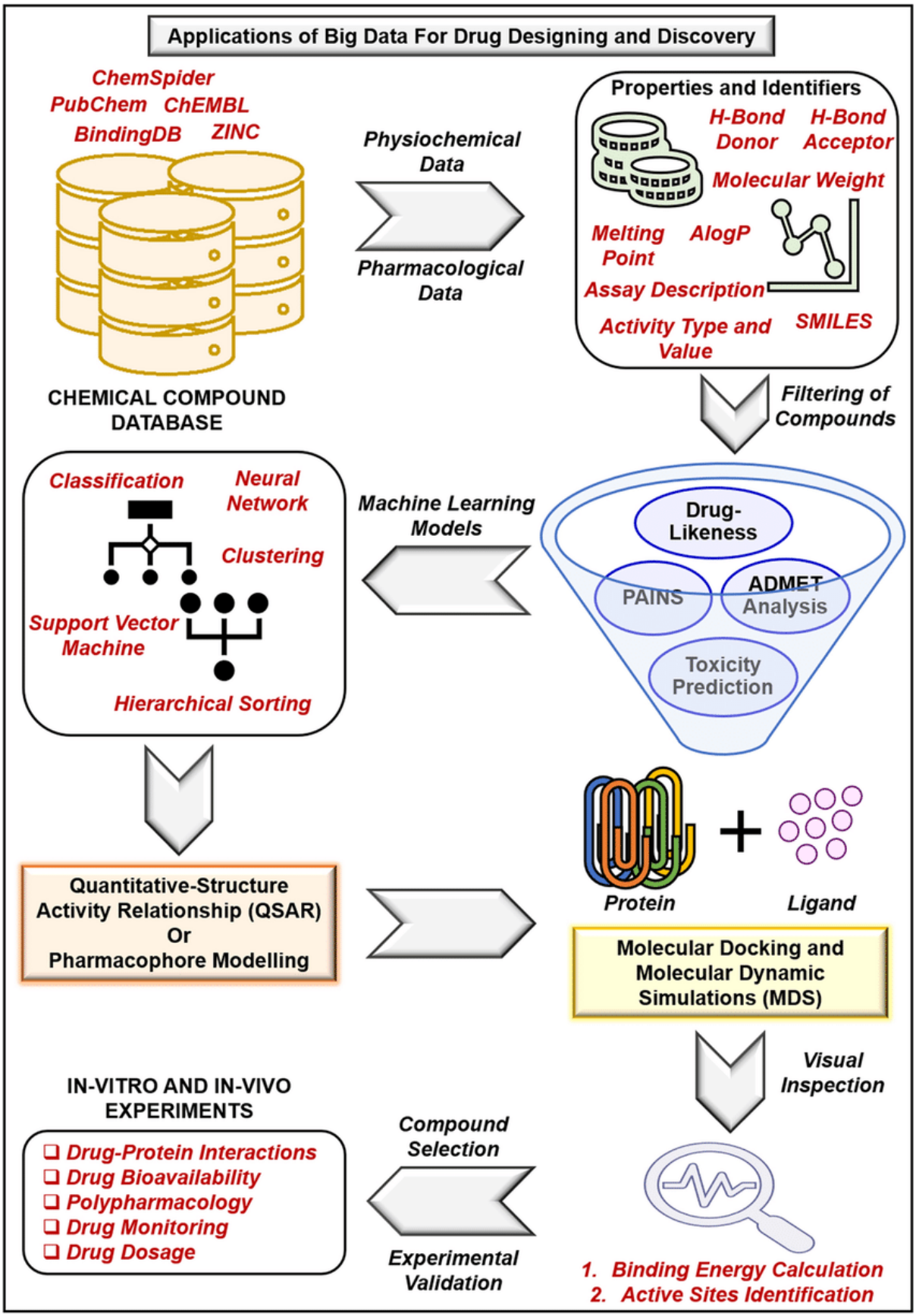
Drug discovery is a complex and lengthy venture which requires a multidisciplinary approach. A drug molecule to reach the market passes through multiple defined stages, wherein each step has its challenges, timeline and cost. Despite numerous advancements in the understanding of biological systems, identifying a novel drug molecule for therapeutic purposes still remains largely a lengthy, costly and complicated process . The human genome project (HGP) has facilitated several advancements in drug development, including precision medicine and target identification for a disease. Compared to the traditional approach, both in vitro and in silico methods have a greater propensity to lower drug discovery costs. These computational approaches in the early stages of drug development also minimize the time span to distinguish a drug candidate with suitable therapeutic effects by excluding compounds exhibiting complex side effects. The modern drug discovery pipelines integrate hierarchical steps that engage various phases such as target identification, target validation, screening of lead candidates against the desired target, optimization of identified hits to increase the affinity, selectivity, metabolic stability, and oral bioavailability. Once a lead molecule is recognized and evaluated, it undergoes preclinical and clinical trials. Finally, the identified molecule that complies with all these investigations moves forward for approval as a drug.



The typical steps for implementing machine learning-based prediction methods consist of data preprocessing, model learning, and evaluation. The data preprocessing steps comprise preparing the data suitable for the various machine learning algorithms, such as discretization and standardization. The model learning phase constitutes the actual implementation of the machine learning algorithms. The final phase involves performance evaluation methods and metrics to assess the numerous trained machine learning models.

Recurrent neural network (RNN) and deep neural network algorithms are widely exploited in target modelling studies. Alpha fold, an AI tool that relies on DNN, is widely used to predict the 3D structure from its primary sequence. The feature extraction potential of deep learning makes it a promising method to predict the secondary structure, backbone torsional angle and residue contacts in protein. Thus, protein folding study can be determined from its sequences with the help of AI methods. DN-fold is another deep learning network method widely used for protein folding and can efficiently predict the structural fold of the protein. With the growth of protein sequence data, AI methods also significantly contribute in predicting the protein–protein interaction studies by using the DNN called DeepPPI, which outperforms (prediction accuracy 80.82%) the traditional ML-based approach (prediction accuracy 65.80%), as the latter approach is faced with the problem of manual feature extraction.

Apart from protein modelling, AI has a role in drug screening, where it reduces the time to identify a drug-like compound. ML algorithms such as nearest neighbour classifiers, RF, extreme learning machines, SVMs, and DNNs are used for the drug molecule's virtual screening and synthetic feasibility. ML-based drug screening has been successfully applied to identify drug-like molecules against various diseases such as cancer and neurogenerative disorders.


Applications of Big Data For Drug Designing and Discovery
ChemSpider
PubChem
ChEMBL
BindingDB
ZINC
CHEMICAL COMPOUND DATABASE
Physiochemical Data
Pharmacological Data
Properties and Identifiers
H-Bond Donor
H-Bond Acceptor
Molecular Weight
Melting Point
AlogP
Assay Description
Activity Type and Value
SMILES
Filtering of Compounds
Drug-Likeness
PAINS
ADMET Analysis
Toxicity Prediction
Machine Learning Models
Classification
Neural Network
Clustering
Support Vector Machine
Hierarchical Sorting
Quantitative-Structure Activity Relationship (QSAR) Or Pharmacophore Modelling
Protein
Ligand
Molecular Docking and Molecular Dynamic Simulations (MDS)
Visual Inspection
1. Binding Energy Calculation
2. Active Sites Identification
Compound Selection
Experimental Validation
IN-VITRO AND IN-VIVO EXPERIMENTS
Drug-Protein Interactions
Drug Bioavailability
Polypharmacology
Drug Monitoring
Drug Dosage

5. APPLYING FUZZY LOGIC

Fuzzy logic is a form of multi-valued logic that deals with reasoning that is approximate rather than fixed and exact. In contrast with "crisp logic" i.e. Boolean logic, where binary sets have two-valued logic: true or false, fuzzy logic variables may have a truth value that ranges in degree between 0 and 1 [18]. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false. It is based on the fuzzy-set theory proposed by L.A. Zadeh in 1965.

In a fuzzy system, the values of a fuzzified input execute all the rules in the knowledge repository that have the fuzzified input as part of their premise. This process generates a new fuzzy set representing each output or solution variable. Defuzzification creates a value for the output variable from that new fuzzy set [13]. So, in order to apply fuzzy logic to an application, first the inputs must be fuzzified so that their value is in the range 0 to 1, then the rules defined by the application are applied, and after this, the results derived from various rules are combined using an aggregation function. Finally, the aggregated results are defuzzified by using an inference function. The evaluations of the fuzzy rules and the combination of the results of the individual rules are performed using fuzzy set operations. The operations on fuzzy sets are different than the operations on non-fuzzy sets [14]. The operations for OR and AND operators are max and min, respectively. For complement (NOT) operation, NOT(A) is evaluated as (1-A).

The second matching technique being discussed uses three input variables – match-count (#match), mismatch-count (#mismatch), and calculated-score (#score – calculated using substitution matrix). These inputs are then fuzzified using following membership functions:

μ(match) = { 0, if #match=0

1, if #match=lenSeq - (1)

[0,1] (1 - #match / lenSeq)

}

μ(mismatch)= { 0, if #mismatch=0

1, if #mismatch=lenSeq - (2)

[0,1] (1 - #mismatch / lenSeq)

}

μ(score) = { 0, if #score <=0

1, if #score = perfectScore - (3)

[0,1] #score / perfectscore

}

In these equations, lenSeq is the length of the shorter sequence of the two sequences being matched, and perfectScore is the score of matching the two candidate sequences, if there are no indels or replacements.

SAGA: Sequence Alignment by Genetic Algorithm

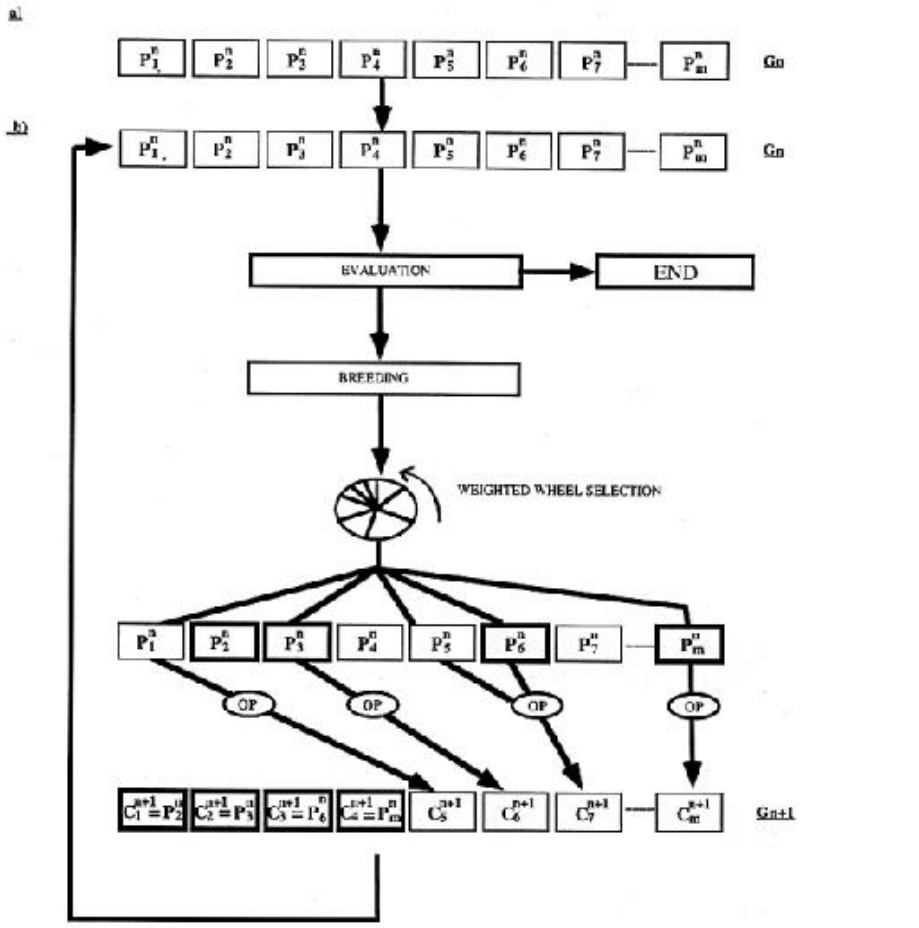
To align protein sequences, we designed a multiple sequence alignment method called SAGA. SAGA is derived from the simple genetic algorithm described by Goldberg. It involves using a population of solutions which evolve by means of natural selection. The overall structure of SAGA is shown in Fig . The population we consider is made of alignments. Initially, a generation zero (G_0) is randomly created. The size of the population is kept constant. To go from one generation to the next, children are derived from parents that are chosen by some kind of natural selection, based on their fitness as measured by the OF (i.e. the better the parent, the more children it will have). To create a child, an operator is selected that can be a crossover (mixing the contents of the two parents) or a mutation (modifying a single parent). Each operator has a probability of being chosen that is dynamically optimised during the run.

These steps are repeated iteratively, generation after generation. During these cycles, new pieces of alignment appear because of the mutations and are combined by the crossovers. The selection makes sure that the good pieces survive and the dynamic setting of the operators helps the population to improve by creating the children it needs.

Following this simple process, the fitness of the population is increased until no more improvement can be made. All these steps, shown in Figure , can be summarised by the following pseudo-code: Initialisation . The first step of the algorithm (Fig. a) is the creation of a random population. This generation zero consists of a set of alignments containing only terminal gaps. A population size of 100 was used in all of the results presented here. To create one of these alignments, a random offset is chosen for all the sequences (the typical range being from 0 to 50 for sequences 200 residues long) and each sequence is moved to the right, according to its offset.The sequences are then padded with null signs in order to have the same length, L. The alignments of generation zero will be the parents of the children used to populate generation one.

Initialisation	1. create G_0
Evaluation	2. evaluate the population of generation n (G_n)
	3. if the population is stabilised then END
	4. select the individuals to replace
	5. evaluate the expected offspring (EO)
Breeding	6. select the parent(s) from G_n
	7. select the operator
	8. generate the new child
	9. keep or discard the new child in G_{n+1}
	10. goto 6 until all the children have been successfully put into G_{n+1}
	11. n = n+1
	12. goto EVALUATION
End	13. end

Evaluation . To give birth to a new generation, the first step is the evaluation of the fitness of each individual. This fitness is assessed by scoring each alignment according to the OF. The better the alignment, the better its score, and thus the higher its fitness. If the purpose is to minimise the OF, as is the case for OF1 and OF2, then the scores are inverted to give the fitness. The expected offspring (EO) of an alignment is derived from the fitness. It is typically a small integer. The method we used to derive it is known as remainder stochastic sampling without replacement. In the case of OF1 and OF2 the typical values of the EO are between 0 and 2, which can be considered as an acceptable range



Reference 1:

https://www.researchgate.net/publication/225980687_AlineaGA_A_Genetic_Algorithm_for_Multiple_Sequence_Alignment