Optimizing Global Network Alignment in Protein-Protein Interaction Networks

Dissertation zur Erlangung des Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)
am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

von

ERHUN GİRAY TUNCAY

Berlin
2022

Betreuer:	Prof. Dr. Tim CONRAD
Erstgutachter:	Prof. Dr. Tim CONRAD
Zweitgutachter:	Prof. Dr. Tolga CAN
Tag der Disputation:	26 / 06 / 2023

Abstract

Global Network Alignment in Protein-Protein Interaction Networks is an NP-complete problem due to the contradicting nature of its biological and topological alignment objectives. There have been several aligners developed focusing on various priorities and objectives of the problem. However, none of these alignment heuristics provide exact solutions, despite the fact that they achieve problem objectives up to a certain extent. For this reason, the research question of uniting stronger aspects of dissimilar Network Alignment heuristics is quite promising. In this thesis, it is aimed to improve the methods to scan the search space of this problem by managing the simultaneous use of several heuristics and two novel population-based meta-heuristic methods are proposed for this purpose.

The first one of these methods (SUMONA) is a supervised genetic algorithm approach that is an extension to the computationally demanding multi-objective memetic algorithm called OptNetAlign. This method intends to accelerate and guide the alignment process by modifying the crossing-over mechanism of the genetic algorithm with inputs from other aligners/heuristics while preventing premature convergence by randomizing the usage of these inputs. The algorithm is based on a generic procedure that generates several alignments with changing heuristics and input parameters, classifies the generated alignments, establishes a randomized alignment selection mechanism from the classified alignments for cross-over and finally adjusts global and local search parameters. It is possible to achieve better running time performance, prioritize certain objectives upon others and also optimize the secondary objectives with this method.

The second method (PERSONA) is a particle swarm inspired collaborative approach that orchestrates several aligners to share their partial solutions continuously while they progress. These aligners jointly constitute a particle swarm that searches for multi-objective solutions of the alignment problem in a reactive actor environment. Within the swarm, the leading or prominent actors send the stronger portion of their solution as a subgraph to other actors and receive the stronger subgraphs of the counter party back upon evaluation of those partial solutions. The individual alignment heuristics were also developed within the scope of the same research and they were implemented based on alternatives such as seed-and-extend approaches with various centrality and sequence seeds, cluster mapping approach and node similarity prioritization. Both the population-based meta-heuristic tasks and the individual heuristic tasks were implemented in a non-deterministic fashion in order to improve flexibility and preventing to be trapped in locally optimal solutions. The results achieved with this method are remarkably optimized and balanced for both topological and node similarity objectives.

Acknowledgements

First and foremost, I am deeply grateful to my supervisor Tim CONRAD for encouraging me to do research in such a remarkable field and providing me inspiring feedback. I would also like to warmly thank Tolga CAN for helping me to discover and adopt a open research field question of systems biology that made me progress in challenging techniques and for setting me an example with his excitement in science. Furthermore, I am grateful to Rıza Cenk ERDUR who oriented me to use Actor Systems as a both lightweight and proactive infrastructure for a collaborative meta-heuristics design of my research problem. Moreover, I would like to express my special thanks to Bernhard RENARD for giving me the opportunity to work in his inspiring research group at Robert Koch Institute. I would additonally like to thank my previous doctoral committee member Belgin ERGENÇ and master's supervisor Tatyana YAKHNO for their constructive feedback on my ideas.

I see PhD as a humbling journey reminding an individual that learning is a never ending process and there is much to comprehend both technically and spiritually from everyone and everything. Several valuable friends and colleagues including but not limited to dear departed Sezer, Oğuzhan, Kafa Onur, Adil, Rasim, Ahmet Emre, Güneş, Tuğba, Işıl, Ethem, Candaş, Gürkan, Derya, Müş, Cemal, Hakan, Seko, Akın, Baran, Onur Vargün, Deniz, Şakir, Cumhur, Yücel Teyze, Gözde, Serkan, Hakan Bal, Seán, Miyo, Caner, Tobias, Christine, Camilo and Pietro inspired, motivated and cheered me up throughout this challenging journey. Moreover, I would like to thank all my friends in Bornova Anatolian High School (in particular my classmates in AFK), İzmir Institute of Technology, Alsancak (Muzaffer İzgü Sokak in particular), Kumamoto, FU and MPM since we embraced each other as part of a memorable community. Plus, I express my gratitude to my father Suavi, mother Gülhan and sister Güliz Seray for their trust in me at all times. Finally, I would like to thank my wife Tuna for always being there with her support and patience during the hard work and my daughter Roza Umay for teaching me how to love and smile!

Contents

List of Figures

List of Tables

List of Algorithms

Terminology and Abbreviations

Terminology

Heuristics vs Meta-heuristics

The term heuristic is used for a singular problem-dependent deterministic solution procedure whereas the term meta-heuristic is used for representing a problem-independent heuristic orchestration framework throughout the thesis.

Node Mapping vs Network Alignment

The term mapping is used as the building block of an alignment throughout the thesis. A network alignment is the product of a network comparison between two or more organisms and it is composed of several node mappings associated with predicted functions.

BitScoreSum vs Biological Sequence Similariy

A BitScoreSum is a particular implementation of Biological Sequence Similarity.

Abbreviations

FU Berlin	Freie Universität Berlin
EC	Edge Consistency
ICS	Induced Conserved Structure
S^3	Symmetric Substructure Score
LCCS	Largest Common Connected SubGraph
GO	Gene Ontology
GOC	Gene Ontology Consistency
GOE	Gene Ontology Enrichment
BS	Biological Sequence Similarity
AFS	Average of Functional Similarity
FC	Functional Consistency
BLAST	Basic Local Alignment Search Tool
PPI	Protein-Protein Interaction
GNA	Global Network Alignment
LNA	Local Network Alignment
UPMX	Uniform Partially Matched Crossover
GA	Genetic Algorithm
REST	Representational State Transfer
JSON	JavaScript Object Notation
WSM	Weighted Sum Model
WPM	Weighted Product Model
CPU	Central Processing Unit
OWL	Web Ontology Language

1 Introduction

1.1 PPI Networks for Function Prediction

One major goal of bioinformatics is to identify the preserved functions across different species or organisms. Theoretically, such shared functions can be investigated based on evolutionary information through the homology and specifically orthology concepts that constitute 'inheritance through homology' methods as the most common approach in protein function prediction compared to the alternative "de novo sequencing" methods [1]. Fitch [2] defines homology broadly as the relationship of any two characters that have descended and probably diverged from a common ancestral character. The review emphasizes that there are structural and functional characters that need to be distinguished within this definition. In this scope, Lee, Redfern and Orengo [1] describe orthology as a type of homology that takes place due to a speciation event and paralogy as another type of homology that is the product of a gene duplication event. According to the same study, these descriptions are essential in function prediction since orthologues tend to indicate the preserved function in different species while paralogues tend to evolve new functions despite their considerably high sequence similarities in several cases. Traditional orthologue discovery methods involve tasks such as homology search with Basic Local Alignment Search Tool (BLAST), discarding non-homologue protein sequences and exhaustive search against reference proteomes or Expressed Sequence Tag (EST) databases [3] if the network interactions of proteins are not taken into account.

It has been revealed in some studies that sequence and network information complement each other in orthology detection [4]. Consequently, orthology of a pair of proteins from different organisms can be identified thoroughly considering their interactions with other proteins in their own environments. Kuchaiev and Przulj [5, 6] emphasize that proteins cooperate among each other by forming physical bonds to be able to perform their particular function and for this reason huge interaction networks emerge to carry out all the essential cellular activities. The interactome within a cell require detailed and multi dimensional representations as a complex problem domain. Pavlopolulos [7] states that biological processes within a cell can be represented using Protein-Protein Interaction (PPI) Networks as a high level and static description composed of binary interactions [8]. The study reports that the interaction data can be detected via large-scale and high-throuhput techniques such as the pull down assays

[9], tandem affinity purification (TAP) [10], yeast two-hybrid (Y2H) [11], mass spectrometry [12], microarrays [13] and phage display [14].

There are several databases that store single species interaction data such as the Yeast Proteome Database (YPD) [15], Human Protein Reference Database (HPRD) [16], Human Protein Intaraction Database HPID [17], Human Annotated and Predicted Protein Interactions (HAPPI) [18], Drosophila Interactions Database (DroID) [19] along with databases storing multiple species interaction data such as the the Munich Information Center for Protein Sequences (MIPS) [20], the Molecular Interactions (MINT) database [21], the IntAct database [22], the Database of Interacting Proteins (DIP) [23], the Biomolecular Interaction Network Database (BIND) [24], the BioGRID database [25], SNAPPI-DB [26] and 3did [27]. Besides, some meta-databases or integrated data sources such as Stitch [28], String [29], Mintact [30] and APID [31] intend to facilitate consistent extraction of aggregate data. Fig. 1.1 represents the PPI Network of Caenorhabditis Elegans with binary interactions in the 2016 Version of Intact Database visualized using Cytoscape [32].

The records in all the above-mentioned databases are verified based on their particular choices of verification such as curation from literature, user submission of experimental results, extraction from structural databases of proteins or prediction. It should be noted that prediction based verification methods require more attention among others in terms of the confidence they provide. Gonzalez and Kann [34] summarizes several computational interaction prediction methods that evaluate empirical information based on experimental data by using the relative frequency of interacting domains [35], maximum likelihood estimation of domain interaction probability [36, 37], co-expression [38], domain architecture [39] or parsimony [40] along with theoretical approaches that evaluate the fact that interacting proteins co-evolve to preserve their function [41, 42], conserve gene order [43] or are fused in some organisms [44, 45].

According to Mosca, Pons, Céol, Valencia and Etol [46], PPIs annotated with molecular functions will play a key role in fields such as clinics, pharmacology, descriptive biology and synthetic biology on condition that the data is curated and interaction types are distinguished according to certain conventions such as the BioPAX standard [47] defined by the Protein Standard Initiative-Molecular Interaction (PSI-MI) consortium [48, 49]. Bandyopadhyay, Sharan and Ideker [50] emphasize that many research studies [51, 52] focus on annotating functions to subject proteins based on their sequence similarity to better characterized proteins in molecular biology domain. The same study argues that functional annotation becomes harder in the presence of several similar paralogous proteins since there is not

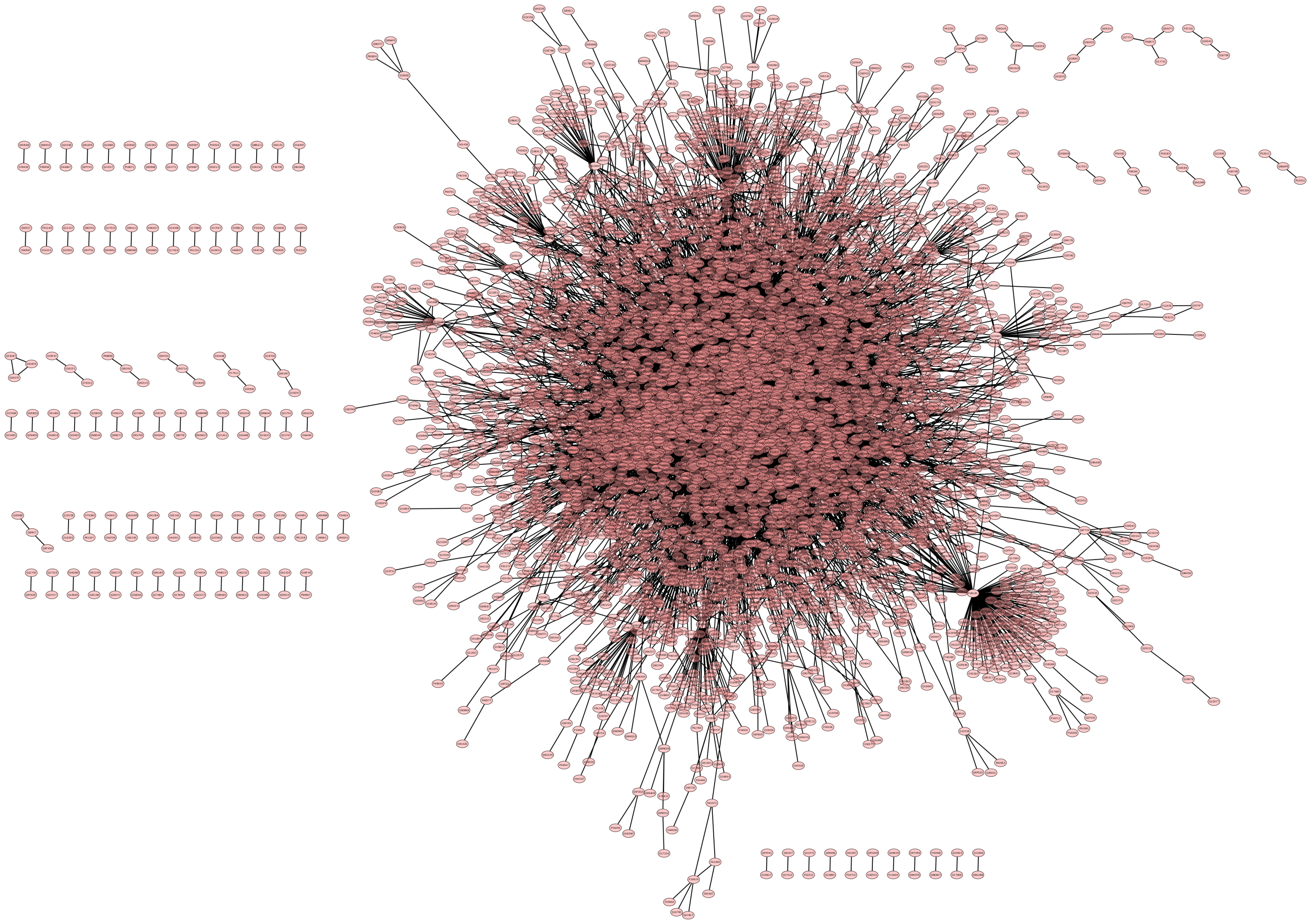

Figure 1.1: PPI Network of Caenorhabditis Elegans in the 2016 Version of Intact Database. The figure shows that a rare number of proteins interact with very few others for performing extremely distinct functions in contrast to the majority of the proteins that are involved in the same cellular processes and form clusters with each other so that they incorporate into a large molecule for performing the fundamental cellular functions [33].

enough evidence to identify the true ortholog among them [53]. On the other hand, several experimental and computational methods have been proposed to predict functions of proteins based on their interactions with other biomolecules [7].

There are several sources that can be annotated to PPIs for protein function prediction. The Gene Ontology (GO) project is the most prominent one of these sources and it is a product of an effort to describe the attributes of genes, gene products and sequences consistently through controlled vocabularies and ontologies for collaborative community use [54]. The project was founded in 1998 and its vocabularies and functional annotations are expanding ever since within the directions explored by novel biological research [54, 55]. Lee et al. [1] mentions the GO project 4 as the most prominent functional annotation source among other sources such as ENZYME [56], Swiss-Prot [57], FunCat [58], KEGG [59], MetaCyc [60] and Reactome [61] especially due to its structured vocabularies that describe gene products in terms of their associated biological processes, cellular components and molecular functions in a machine readable way. The project is releasing monthly updates with a DOI unique identifier and has recently updated its representation model of gene functions with a framework called GO-Causal Activity Modeling (GO-CAM) to be based on Web Ontology Language (OWL) for combining GO annotation data with larger vocabularies. Despite this radical change, backward compatibility for the standard GO annotations (GAF format) is still ensured by the project [55].

1.2 PPI Network Comparison Methods

Bandyopadhyay et al. [50] mentions network comparison as an alternative to gene and protein sequence similarity approaches for functional orthology detection due to the significance of conserved or common interactions possessed by a protein pair in comparison [62]. Network comparison is a challenging task since the similarity or difference of the compared networks has several dimensions. Tantardini, Ieva, Tajoli and Piccardi [63] state that it is necessary to define a well-balanced distance measure for being able to compare networks. The study further claims that such a task requires a compromise among factors such as effectiveness of results, computational efficiency and interpretability depending on the problem domain. The most well-known techniques for network comparison are graph kernel based methods and network alignment that handle these factors in a completely different way as discussed below.

Computational efficiency can be an important factor that effects convenience of use for preferring a network comparison method. Shen and Guda [64] discuss that the alignment-free graph kernels produce more summarized network comparison results with a single value of an inner product and relatively shorter processing time despite lacking a detailed view of results unlike Network Alignment methods. The study lists various graph kernel types based on random walks, paths, subtrees, graphlets or edges that were used for network comparison in the literature. The annotations of protein functions can be transferred across species through graph kernels of PPI data [65] as well as other biologically representative data such as co-expression, genetic interaction, metabolic pathways, domain structure, and sequence [66–68]. Fan et al. [65] also propose creating unified multi-species protein representations through homologous landmark nodes for achieving a multi-network kernel comparing several PPI networks.

On the other hand, Network Alignment methods produce more detailed results that enable comparing proteins with respect to multiple biological and topological similarity measures. According to Pan, Wang and Li [69], Network Alignment addresses the problem of identifying similar and dissimilar regions in the PPI Networks of two or more species. The book reports that Network Alignment is widely used in detecting conserved subnetworks that may have common protein functional modules [6, 70, 71]. As a remarkable example of another application field, Ma and Liao [72] discusses that Network Alignment across biological networks can directly be used to transfer the biological knowledge of well-studied species to others. The same study states that network alignment enables discovering system-wide relationships between different species. PathBLAST [73] is the first proposed method that aims to compare a pair of organisms by aligning their Protein-Protein Interaction Networks locally based on BLAST [74] e-values of protein pairs in order to identify protein pathways and complexes conserved by evolution and to detect the reliability of protein interactions. This tool was followed by other local network alignment algorithms such as NetworkBLAST [75] that scores mappings according to the density of their respective protein complexes, MaWish [76] that evaluates evolutionary similarity between graph structures based on match, mismatch, and duplication concepts and Graemlin [77] that extends alignment seeds based on the phylogenetic trees.

According to Seah, Bhowmick and Dewey [78], the motivation behind Local Network Alignment (LNA) is the impossibility of achieving high-confidence mappings in all regions of a network and the flexibility to identify unrelated conserved regions between the input networks so that known functional components can be found in a new species. The study explains that it is this motivation that drives LNA approaches to discard low-confidence mappings totally in contrast to global network alignment that aims to map all proteins in the compared organisms. On the contrary, the idea behind Global Network Alignment (GNA) is that the fundamental set of biological functions taking place within the cells of any compared organisms should map to each other regardless of the complexity of the process. Besides, it is likely that there can be inconsistencies while merging multiple LNA results with each other for yielding a complete picture [79]. Consequently, it is assumed that the PPI Network of a less complex organism can be subsumed by the other compared network in terms of functionality. For this reason, the most prominent low-confidence mappings are also aligned in Global Network Alignment methods in order to have a complete alignment picture. Guzzi and Milenkovic [80] discusses that GNA results could be refined to achieve higher functional quality without compromising topological quality by removing functionally insignificant mappings. The study complements this discussion by proposing that LNA mapping regions could drastically be increased to improve topological quality without much compromise in functional quality. In this regard, DualAligner [78] can be seen as an exemplary attempt to fill the gap between local and global alignment with a two-step approach that initially aligns high confidence protein pairs and subsequently performs functional region-to-region alignment for protein pairs that do not achieve high confidence by employing soft subgraph pair mapping constraints and using GO annotated PPI networks as inputs.

Global Network Alignment in PPI Networks can be defined as an NP-Complete Network Comparison problem based on the node similarity of protein pairs along with the topological similarity of their neighborhoods. Proteins are represented as the nodes of a network, while the interactions between them are represented as edges that form the PPI network for modeling the compared organisms in the GNA problem. The goal of the problem is to align each node in the smaller network to a particular node in the larger networks [81] assuming that a pairwise comparison is performed. Node similarity is usually measured by biological sequence similarity and by the common experimentally verified functional annotations. On the other hand, topological similarity is measured by various metrics that evaluate the mapped edges of the compared networks in different ways. Every alignment approach tries to map the most similar nodes of the compared organisms to each other with its particular alignment scoring function based on a combination of these similarity measurements.

IsoRank [82] is the first Global Network Alignment algorithm that intends to map all the input nodes of the compared networks based on both pairwise node similarity and neighborhood topology similarity of the mapped node pairs. This algorithm formulates the trade-off between network overlap and node similarity as an eigenvalue problem and extracts high-scoring and mutually consistent mappings. IsoRank uses BLAST scores as the node similarity measure but it also encourages usage of additional measures such as synteny-based scoring and functional similarity for the same purpose. The algorithm was later extended to solve the global multiple-network alignment with a computationally efficient and error tolerant method called IsoRankN [70].

There are various Network Aligners that evaluate the similarity of motifs in the compared networks. GRAAL [83] is the first purely topological network aligner that evaluated a graphlet vector of several possible network motif combinations for describing the local neighborhood topology of each node to be aligned. This algorithm was extended by several sequels such as H-GRAAL [84] that apply a Hungarian algorithm based approach instead of the traditional seed-and-extend approach, Mi-GRAAL [6] that integrates multiple types of node and topological similarity metrics, C-GRAAL [85] employing neighborhood density in addition to graphlet degree signatures, and L-GRAAL [86] that combines the seed-and-extend approach with graphlet degree signatures by Lagrangian relaxation technique. It is also important to be able to identify clusters as well as network motifs for a better topological comparison across networks. Apart from the GRAAL series, another significant motif focused aligner is GHOST [87] that uses spectral signatures for comparing topological similarity followed by a local search. Sarich, Conrad, Bruckner, Conrad and Schütte [88], discuss the sparse nature of information in networks and the necessity of identifying modules or clusters from them to achieve more compact information. The study further evaluates the choices of allowing overlapping clusters by assigning nodes to multiple modules as well as allowing some nodes not to be assigned to any module at all. Such choices effect the format of the compact meta-data to be stored as network properties for further analyses.

Another remarkable category of Global Network Aligners is seed-and-extend aligners that propagate their most significant seed mappings to a complete alignment around them. SPINAL [89] and HubAlign [90] are prominent examples of this category both of which will be discussed in detail in the following chapter while GRAAL [83] and GHOST [87] can be categorized as both motif focused and seed-and-extend aligners. A final category in terms of method can be search based aligners. GNA algorithms in this category particularly employ heuristics, meta-heuristics or swap based operations for refining alignments. These aligners evaluate alignment objectives simultaneously while they progress to be able to perform optimization among them. Some prominent examples of this class are PISWAP [91] that iteratively refines the initial alignments of custom heuristics with topological information while compromising sequence information and OptNetAlign [92] that will be discussed in detail in the following chapter. Many algorithms utilize these generalized approaches in a hybrid fashion and can be categorized as members of more than one category.

Mapping approach is a distinguishing feature among Network Aligners depending on the analyses they will be involved with. Ma and Liao [72] explain that the node mapping approach of network alignment algorithms can be categorized as one-to-one, one-to-many and many-to-many based on their mapping constraint. The study argues that one-to-one and one-to-many network alignments may correlate better with topological metrics such as edge correctness or the number of conserved edges due to their more simplistic assumptions about protein interactions whereas many-to-many alignments are able to represent protein complexes due to their more realistic and flexible assumptions about the evolutionary processes. GNA and LNA definitions do not explicitly include a mapping approach but it is more common in GNA to use one-to-one mapping since it aims to complete an alignment by leaving no unmapped nodes in the smaller networks and allowing low-confidence mappings. LNA usually focuses on regions with high-confidence mappings and several studies has adopted the many-to-many approach to capture such regions. Accordingly, Elmsallati, Clark and Kalita [81] argue that LNA results are more difficult to interprete by researchers despite being more consistent with biological theory due to the assumption that a protein may have multiple orthologs through duplication or other means. This statement holds true only for the comparison of commmonly used many-to-many LNA and one-to-one GNA. However, there are various LNA and GNA algorithms that do not adopt their respective mainstream mapping approach and violate this statement.

Another distinguishing feature among Network Aligners is their capability of performing Multiple Network Alignment as opposed to being restricted to a pairwise comparison. Multiple Network Alignment can be considered as a combination of several pairwise network alignment tasks aligned with each other. As the number of aligned networks increase, the number of consistent mappings may decrease whereas their validity may increase since they are proven iteratively with the new aligned networks. Some prominent examples of multiple network aligners are Graemlin 2.0 [93] that automatically learns high scoring matches from a training set of known network alignments and phylogenetic information, SMETANA [94] that employs a semi-Markov random walk for computing node correspondence scores used to construct the maximum expected alignment, NetCoffee [95] based on a set of weighted bipartite graphs used in maximizing a target global alignment function, and BEAMS [96] that performs the alignment upon backbones of possible cliques from a k-partite sequence similarity graph. These aligners all allow many-to-many mappings and have a global network scope except for Graemlin 2.0 that focuses on local mappings.

1.3 Population Based Meta-heuristics

Blum and Roli [97] summarize meta-heuristics algorithms as abstract approximation strategies that guide the search process by exploring the search space efficiently for optimal solutions. The study also highlights that these algorithms may include elements that aim to avoid getting trapped in locally optimal solutions and store previous search experience. It is a good idea to develop problem independent meta-heuristic approaches when it is not possible to combine different problem-specific heuristics directly into a single robust method. Meta-heuristic algorithms do not ensure a globally optimal solution and they usually employ random variables for traversing a search space with flexibility. The search scope of a meta-heuristic algorithm can vary from local to global in the solution space and there are several methods that perform both local and global search.

Evolutionary computation techniques are a special subset of meta-heuristic algorithms that exhibit a remarkable adaptation for optimizing results based on nature-inspired methods. Gandibleux and Ehrgott [98] define Evolutionary Algorithms as methods to produce a population of solutions that progress both individually and in cooperation among themselves by exchanging information or other means to achieve a maturated approximation. The study further explains the usual components of these algorithms as an exchange mechanism, a solution randomizing approach, an elite solutions archive, a fitness measure, a penalty strategy for infeasible solutions as well as ranking, guiding and clustering methods. The study also claims that population-based methods are very attractive tools for solving multi-objective problems due to the contribution of the whole population to the evolutionary process and the parallelism of the generation mechanism along the members of the population. It is also worth noting that evolutionary algorithms consist of an implicit or explicit component that tries to learn the correlations between decision variables in order to identify high quality regions of a search space in an optimization problem as discussed by Blum and Roli [97]. The study further emphasizes that this class of meta-heuristics focus on high quality regions of the search space by biased sampling and recombination of results.

There are various search based Network Aligners developed based on meta-heuristics such as SANA [99] or particularly based on evolutionary algorithms such as OptNetAlign [92] and MAGNA [100]. Search based aligners can either produce a population of alignment results or a single result. The search tasks should efficiently be distributed among multiple computational entities to benefit from a population behavior in Network Alignment. Consequently, it becomes essential to utilize a concurrent or a parallel infrastructure to implement a population that performs a meta-heuristic search of optimum alignments against multiple contradicting topological and biological metrics. Population based behavior can be useful in establishing a hybrid meta-heuristics approach that orchestrates a variety of heuristics [101]. The memetic algorithm of OptNetAlign [92] can be regarded as an example of such a hybrid meta-heuristics approach in which population members perform local search heuristics in addition to the global Genetic Algorithm (GA) behavior. A population may perform collaborative tasks given that its members are minimally capable of exhibiting a reactive behavior upon receiving messages from other members. The extensive search task of the Network Alignment problem can efficiently be distributed by using such a population.

It is essential to utilize a convenient parallelization architecture for establishing a collaborative or synchronized behavior in meta-heuristic algorithms. According to Cung [102], meta-heuristic algorithms can either be parallelized by performing parallel best neighbor search at each iteration of a single walk or by investigating multiple trajectories in parallel with different processors. In addition, Talbi [103] lists the major parallel models of population-based algorithms mentioned originally in [104] as the master/slave model in which crucial operations are split between master and slave processes, the distributed island model that employs multiple meta-heuristics simultaneously to orchestrate their respective sub-populations and the cellular model that focuses on interactions of individuals in their small neighborhoods. The choice among these models mostly depends on the delegation of tasks among entities and the search scope of the meta-heuristic algorithm in question.

Another aspect of a collaborative meta-heuristic algorithm is the concept of proactivity that characterizes the extent of self-initiated behavior for its individual entities to solve the problem in hand. Such proactive entities can play a key role in designing a flexible and efficient optimization strategy. The need for reactive or autonomous proactive behavior depends on the interactivity of entities required by an optimization method. However, there is a trade-off between autonomous behavior of the entities in a swarm and the computational overhead caused by autonomy. For this reason, an individual entity should be given autonomous behavior only when it is going to have a significant contribution to the whole population-based meta-heuristic model. The level of autonomy can be characterized by the additional behavior brought to the entities of a population with the respective design paradigms of actors or agents.

An entity can proactively contribute to a whole population by directly undertaking a certain part of the computation or by serving another informative or communicative role that notifies other entities. Similarly, Idzik, Byrski, Turek and Kisiel-Dorohinicki [105] emphasize that agents or actors may carry out computational tasks and communicative driver tasks that request execution according to the environment coupled together in a population of a particular evolutionary meta-heuristic algorithm. Consequently, these concepts can be used as building blocks for designing behaviors of population members in a swarm. Most swarm intelligence approaches are built upon agent based concepts since the agent model is very convenient for describing the parallel relations of entities in meta-heuristics approaches such as Particle Swarm Optimization [106, 107]. Furthermore, Krzywicki, Turek, Byrski and Kisiel-Dorohinicki [108] discuss the significance of asynchronous modeling for agent-oriented and concurrent actor systems performing evolutionary computation by referring to the complex structure of biological system as their main source of inspiration. In this sense, it should be noted that a concurrent meta-heuristics or evolutionary design introduces another degree of freedom beyond parallelism by allowing partially overlapping time periods for executing tasks of individual entities in a population.

According to Burgin [109], a computational actor model intends modeling, analyzing and organizing concurrent digital computations of several entities by formalizing an arbitrary receipt event with the concepts of a message and messenger. The study further emphasizes that an actor can formally be defined according to the set of its properties and relations in a certain environment, the sets of possible actions it performs and it is exposed to as well its reaction and proaction functions that characterize its behavior. Practically, actors in Computational Actor Models can create new actors, send messages and receive messages upon receiving a message whereas the actors in the more general system actor model can perform any action defined in its environment [106, 109]. Burgin [109] additionally describes an agent as an actor that acts to fulfill the goals of another system or actor regardless of the scientific domain that it is used in. Agents inherit the message passing concurrency model feature based on parallel computation from actors [106] in order to perform the goals of another computational entity in a proactive fashion. However, such decentralized goal definitions are computationally more demanding compared to central goal definitions that orchestrate a population of lightweight members in a meta-heuristic problem. For this reason, the actor paradigm can be the choice of minimal proactivity with efficient concurrency functionality and less computational overhead compared to the agent paradigm that offer more autonomous intelligence with more overhead. Thus, it may be concluded that the actor paradigm becomes an ideal candidate for establishing a population-based collaborative meta-heuristic for the Global Network Alignment problem provided that the collaboration procedures are neatly defined.

1.4 Thesis Outline

In this thesis, I present two new algorithmic approaches for the multi-objective optimization of Global Network Alignment problem that is NP-complete due to the contradicting nature of its biological, annotational and topological alignment objectives. The main goal of the thesis is to improve the methods to scan the search space of the problem by managing the simultaneous use of several heuristics. This task is crucial due to the limitations of the existing network alignment algorithms such as their inability to effectively search for mappings due to excessively strong assumptions, rigid procedures or being restricted to basic objectives as well as the computationally demanding nature of the alternative meta-heuristics based alignment algorithms that perform a thorough search with weak assumptions. For this purpose, I propose two population-based meta-heuristic solutions to tackle the limitations of this problem, the first one being a supervised genetic algorithm approach while the second one being a particle swarm inspired collaborative method. Throughout the projects, I was initially co-supervised by Tolga CAN and Rıza Cenk ERDUR and later on I was supervised by Tim CONRAD as my main supervisor.

In Chapter 2, I introduce SUMONA that is an extension to the computationally demanding multi-objective memetic algorithm called OptNetAlign. SUMONA accelerates and guides the alignment process by modifying the crossing-over mechanism of the genetic algorithm with inputs from other aligners/heuristics while preventing premature convergence by randomizing the usage of these inputs. In this scope, I present a generic procedure that generates several alignments with changing heuristics and input parameters, classifies the generated alignments, establishes a randomized alignment selection mechanism from the classified alignments for cross-over and finally adjusts global and local search parameters. SUMONA was implemented with TBB and Boost libraries for parallelization. SUMONA was tested with both real and synthetic datasets to evaluate its performance in various topological, biological and functional objectives. I wrote the resulting manuscripts below with valuable contributions from my co-author Tolga CAN.

Tuncay EG, Can T. *SUMONA: A Supervised Method for Optimizing Network Alignment.* Computational Biology and Chemistry, 2016, doi: 10.1016/j.compbiolchem.2016.03.003

Tuncay EG, Can T. *SUMONA: A Supervised Method for Optimizing Network Alignment*, Proceedings of the Fourteenth Asia Pacific Bioinformatics Conference (APBC 2016), January 11-13, 2016. http://www.sfasa.org/apbc2016/content/APBC2016_ProgramBook.pdf

In Chapter 3, I introduce PERSONA that orchestrates several aligners to share their partial solutions continuously while they progress. These aligners jointly constitute a particle swarm that searches for multi-objective solutions of the alignment problem in a reactive actor environment. Within the swarm, the leading or prominent actors send the stronger portion of their solution as a subgraph to other actors and receive the stronger subgraphs of the counter party back upon evaluation of those partial solutions. The individual alignment heuristics were also developed within the scope of the same research and they were implemented based on alternatives such as seed-and-extend approaches with various centrality and sequence seeds, cluster mapping approaches and node similarity prioritization approaches. Both the population-based meta-heuristic tasks and the individual heuristic tasks were implemented in a non-deterministic fashion in order to improve flexibility and preventing to be trapped in locally optimal solutions. PERSONA was tested with datasets from prominent PPI databases and compared with several multi-objective Global Network Aligners in order to be evaluated from different perspectives. The application implemented with Akka concurrency architecture and Neo4J Graph Database that particularly enables to employ several alignment improving heuristics by matching complicated network patterns. I wrote the resulting manuscript below with helpful input of from my co-authors Tim CONRAD and Rıza Cenk ERDUR as well as inspiring consultation from Tolga CAN. The most recent experiment results can be seen from the journal.

Tuncay EG, Erdur RC, Conrad T. *Parallel Exchange of Randomized Sub-Graphs for Optimization of Network Alignment: PERSONA*. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2022, doi: 10.1109/TCBB.2022.3231489

Chapter 4 gives a summary of the thesis and provides an outlook for potential future developments in the field of Multi-Objective Optimization of Global Network Alignment in PPI Networks.

2 SUMONA: A Supervised Method for Optimizing Network Alignment

This chapter focuses on improving the multi-objective memetic algorithm for protein-protein interaction (PPI) network alignment, Optimizing Network Aligner - OptNetAlign, via integration with other existing network alignment methods such as SPINAL, NETAL and HubAlign. The output of this algorithm is an elite set of aligned networks all of which are optimal with respect to multiple user-defined criteria. However, OptNetAlign is an unsupervised genetic algorithm that initiates its search with completely random solutions and it requires substantial running times to generate an elite set of solutions that have high scores with respect to the given criteria. In order to improve running time, the search space of the algorithm can be narrowed down by focusing on the most desired criteria and trying to optimize other relevant criteria on a more limited set of solutions. The method presented in this chapter improves OptNetAlign in a supervised fashion by utilizing the alignment results of different network alignment algorithms with varying parameters that depend upon user preferences. Therefore, the user can prioritize certain objectives upon others and achieve better running time performance while optimizing the secondary objectives.

2.1 Background

Network Alignment is the most prominent method for comparing two or more protein interaction networks to identify their regional similarities and differences. Network alignment can be used to infer several biological facts about protein interactions such as predicting functions and functional annotations of proteins or protein complexes, predicting and validating protein interactions, detecting orthologies and reconstructing phylogenetic trees [69]. The recent demand about comparing protein interaction networks of different organisms has resulted in a number of network alignment algorithms to be developed. The heuristics employed by these algorithms greatly vary due to the NP-Completeness of the GNA problem. Some studies [81, 110] state that the NP-Completeness of the Network Alignment problem is due to its equivalence to a graph-matching problem and particularly a subgraph isomporphism problem that does not have an exact solution and needs to be optimized in terms of a similarity scoring, cost or distance function [111–113].

All network alignment algorithms allow inexact matching since they aim to discover the similarities and differences between the aligned organisms. Allowing inexact matching introduces the concepts of gap, mismatch, node deletion and insertion as approximation approaches by penalizing the performance of these algorithms [69, 110]. Network alignment algorithms utilize different techniques and each of them has characteristic advantages and disadvantages which makes them more applicable to some datasets while being inapplicable for others. However, the performance of these network aligners are still far from the desired level and they generate significantly different alignments with common aligned pairs as low as 5% of the total number of matches. Therefore it becomes necessary for the user to run many different aligners in order to choose the most applicable alignment among all possible alternatives [92, 114].

Performance of Network Alignment algorithms can be evaluated with various quality objectives. These objectives evaluate alignment quality with respect to measures such as sequence similarity, topological fit, matching ratio, neighborhood similarity, density matching and the GO annotations between aligned proteins. While some algorithms favor only one parameter over the others, some algorithms allow a user-defined prioritization among them. However, these alignment goals usually contradict each other and most of these algorithms can produce significant results with just a few of these objectives by ignoring the others. The obvious reason for this fact is that most aligners rely on alignment approaches that improve particular objectives at the expense of others. Nevertheless, it is possible to perform multi-objective optimization with methods such as OptNetAlign that minimize the compromise of objective criteria [92]. This particular method produces a pareto front of diverse and elite set of good quality alignments regarding all the conflicting objectives by employing a Uniform Partially Matched Crossover (UPMX) genetic algorithm during this optimization process. However, OptNetAlign is computationally very expensive although it has the potential to find the optimal equilibrium among the desired parameters. This is due to the genetic algorithm of OptNetAlign starting with an initial population of completely random alignment solutions. Besides, one of the most significant drawbacks of OptNetAlign is that it cannot prioritize the given alignment objectives among themselves.

Limitations in computationally demanding meta-heuristic search operations can be tackled by reinforcing such tasks with more direct heuristics. Thus, it becomes a beneficial strategy to utilize existing heuristics coupled with the flexible search power of optimization to yield better results in a computationally efficient time. There are some aligners that utilize existing heuristics such as BoNA [115] that is only capable of refining an existing alignment by capturing its interactive information until convergence and PISwap [91] and MAGNA [100] that are both capable of creating new alignments and refining the ones created by existing aligners. To this end, this study proposes a number of different starting procedures for OptNetAlign that facilitate the aligner reach significant levels among some of the desired parameters before proceeding with the regular optimization process. These procedures mainly include employing the outputs of different aligners such as NETAL [116], HubAlign [90], and SPINAL [89] instead of one of the random parents in each cross-over cycle of OptNetAlign. The outputs of the other aligners are generated with different parameters, frequencies, and intervals for diverse results. The results, hence the performance, change dramatically in relevant objectives due to the characteristic advantages and adaptability of each aligner on different datasets.

On the other hand, the effects of supervising the cross-over operation is also investigated in this study. For this reason, output alignments of various network aligners are utilized as one of the parent inputs of each cross-over iteration with a randomized pattern. Randomization is achieved by introducing the inputs with a cyclic method or random selection from predefined classes. Thus, the input alignment changes dynamically in a randomized way by using the output of a different aligner with different parameters in every iteration. The main motivation behind continuously changing one of the input alignments during the cross-over process is to prevent local optimal solutions. Different input alignments are expected to have alternative results stronger with respect to different objectives that provide diverse solutions. Moreover, a final optional step of supervised single objective hill climbing is applied in each cycle for every objective that scores below a given threshold to figure out its effect on the optimization process.

2.2 Methods

The method proposed as Supervised Method for Optimizing Network Alignment (SUMONA) is based on the Multi Objective Memetic OptNetAlign algorithm. OptNetAlign creates a population of elite alignments using UPMX, mutation and local search operations based on an efficient swap method. The algorithm, archives elite alignments into a Pareto dominated set with respect to various topological and biological objectives, exploring the trade-off among them during execution. The Pareto dominated set has a constant size and is dynamically updated whenever a new elite alignment is found by replacing it with the weakest member. The user can determine the cross-over and mutation rate as well as the probability of swap, single objective hill climbing, and multi-objective hill climbing at runtime. As a

result, a diverse set of high-quality candidate alignments are generated in a single run. The resulting set of elite alignments are composed of diverse solutions that indicate an optimized compromise between topological and biological match quality [92].
SUMONA is an improvement over the OptNetAlign methodology and its main contribution is increasing the performance of achieving multiple alignment objectives by supervising the optimization process and prioritizing some objectives above others. Most existing network aligners follow an approach of levelization between the similarity measures that they employ. Therefore various input alignments can be generated by using them. SUMONA approach uses yet another generated alignment as input of OptNetAlign at each iteration. So, the input of every iteration becomes random up to a limit and provides the potential to optimize the solution with respect to the desired objectives. Consequently, when a significant improvement is observed in some objectives, the dynamical optimization environment can be fed with other generated input alignments that are stronger in the unimproved objectives in order to gain performance for more optimized results. The performance of SUMONA depends on many factors such as alignment objectives, network characteristics of the aligned species and quality of input data that is generated by other prominent aligners.

2.2.1 Alignment Objectives

Alignment objectives may be conflicting in practice and usually, achieving all alignment objectives simultaneously is not possible in most network aligners. Some quality measures for the GNA problem can briefly be summarized as follows [90, 114]:

- Edge Consistency (EC) is the fraction of the conserved edges in the two aligned networks to the total number of edges in the first network.
- Induced Conserved Structure (ICS) and Symmetric Substructure Score (S^3) are based on penalizing an aligned pair if it aligns a node from a sparse region of its network to another one that is located in denser regions. ICS and S^3 are EC refinements designed to take inter-connectivity into account and all these measures focus on topological similarity of an alignment rather than node similarity.
- Largest Common Connected SubGraph (LCCS) is the size of the largest connected component of the aligned part of the networks that is a factor of topological significance. The size is specified as number of edges.

- Functional Consistency (FC) is the ratio of aligned proteins that have more than k common Gene Ontology annotations to the size of the smaller network. It is also known as Gene Ontology Consistency (GOC) score.
- Average of Functional Similarity (AFS) is related with FC and it takes the distance of the terms in the GO term hierarchy into account as their semantic similarity.
- Biological Sequence Similarity (BS) is usually calculated by the BLAST bit score sum. For this reason, it can also be called BitScoreSum.

BS is inherently a major quality measure among others in this set since it is a direct indicator of homology. However, one point to consider for homology is that it is not always a sign of orthology showing similar functions since a it may also be a sign of paralogy showing new evolved functions. Another point to mention is that high FC and AFS scores can be no coincidence as long as they are evaluated based on experimentally verified or high precision in-silico annotations. Thus they may be considered more reliable than the topological measures EC, LCCS and S^3 that rely on the assumption of associating similar interactions with similar functions. However, the degree of impact for quality measure relevance is a matter of debate since it requires extensive correlation analyses with known true mappings in alignments beyond a rough reliability estimate. Therefore quality measure prioritization cannot be done universally.

On the other hand, some of these fundamental measures used as GNA objectives also have other improved derivatives for measuring performance in more detail. Most of these objectives and their derivatives are used as optimization objectives except for LCCS, which is computed post-execution, in the OptNetAlign algorithm. However, one apparent drawback of OptNetAlign is that it cannot prioritize its optimization objectives and it does the optimization in parallel for all the defined objectives concurrently. Besides, the objectives are in conflict among themselves for most alternatively generated solutions. Eventually, the computation time increases drastically while searching the non-conflicting and fully optimized alignments throughout the search space.

2.2.2 Network Aligners Employed in Supervised Optimization

The method proposed in this study aims to improve the performance of OptNetAlign by supervising the optimization process and prioritizing some optimization objectives above others so that it becomes easier to optimize the remaining objectives. In order to accomplish

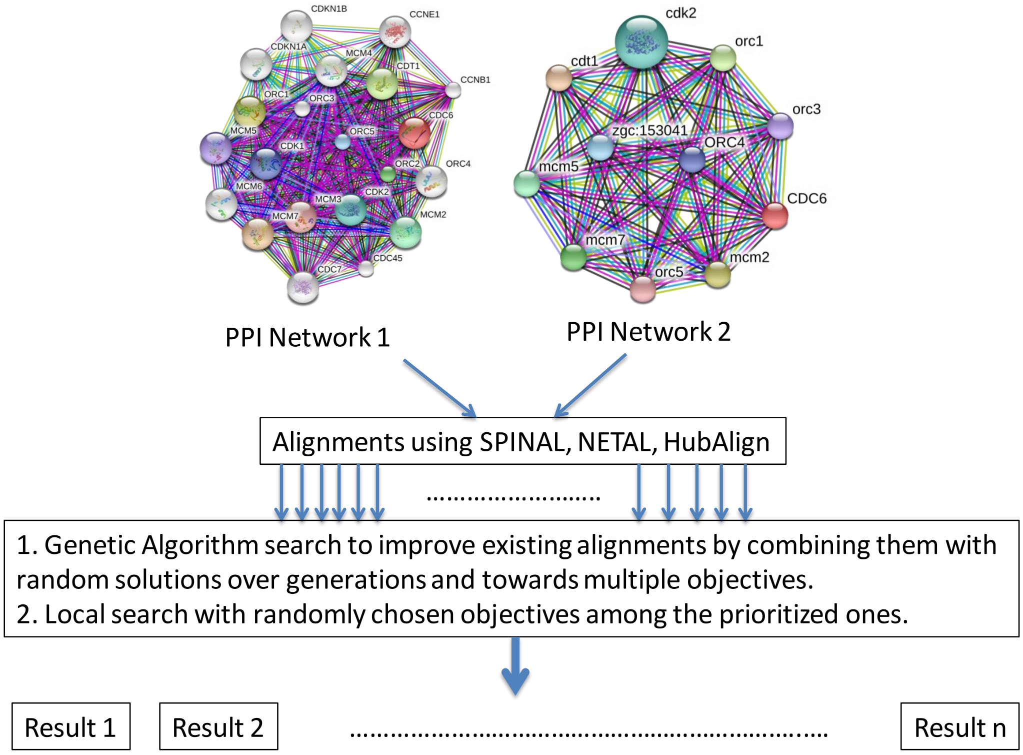
CDKN1B
CCNE1
CDKN1A
MCM4
CDT1
CCNB1
ORC1
ORC3
CDC6
MCM5
CDK1
ORC5
ORC2
ORC4
MCM6
MCM3
CDK2
MCM7
MCM2
CDC45
CDC7
cdk2
orc1
cdt1
orc3
zgc:153041
ORC4
mcm5
CDC6
mcm7
mcm2
orc5
PPI Network 1
PPI Network 2
Alignments using SPINAL, NETAL, HubAlign
1. Genetic Algorithm search to improve existing alignments by combining them with random solutions over generations and towards multiple objectives.
2. Local search with randomly chosen objectives among the prioritized ones.
Result 1
Result 2
Result n

Figure 2.1: Summarized Illustration of SUMONA. First, the supervision input set is diversified by executing other alignment algorithms multiple times with varying parameters. Then, GA and local search tasks are supervised by using results of those alignment algorithms as supervision inputs.

this task, the outputs of certain network alignment algorithms are used as inputs of the GA and local search tasks as presented in Figure 2.1. These input alignments are provided by the outputs of the network aligners HubAlign [90], NETAL [116], and SPINAL [89], with distinctive characteristics. These algorithms are chosen mainly because of their different optimization approaches successful in different objectives and they are explained below in detail:

HubAlign [90] is a GNA algorithm especially useful in detecting functionally similar proteins. Its main idea is starting an alignment from the hub or bottleneck proteins that are topologically more relevant. The algorithm initially predicts the topological importance of a protein from global network topology information with a minimum-degree heuristic approach. In the next step, it starts the alignment from topologically important proteins and then extends it gradually to the whole network. The objectives that are used to assess HubAlign's performance consider both topological and sequence homology information.

SPINAL (Scalable Protein Interaction Network Alignment) algorithm [89] is composed of two main steps. In the first step, initial similarity scores for all possible aligned pairs are calculated based on pairwise local neighborhood matchings. The second step generates a final one-to-one mapping with a seed and extend approach by improving a solution subset locally via iterative swaps. Both steps utilize the neighborhood bipartite graphs and a set of contributors for computing maximum weight matching of edges as a common primitive.

NETAL [116] is a fast GNA algorithm that employs a greedy approach to produce significant results with respect to EC, LCCS and the number of common Gene Ontology terms between the aligned proteins. The algorithm generates an Alignment Score Matrix using two other matrices of similarity scores and interaction scores in the first step. In this context, similarity scores are calculated as the weighted sum of topological and biological scores and interaction scores are dynamically calculated based on the number of conserved interactions of each node. In the second step, the global alignment is searched with a greedy approach by using the Alignment Score Matrix.

2.2.3 The Structure of the Genetic Algorithm

The structure of the genetic algorithm used in SUMONA is based on OptNetAlign [92]. SUMONA briefly intends to supervise the optimization strategy of OptNetAlign by focusing on the search space around many input alignments provided by other aligners. On the other hand, OptNetAlign performs a complete optimization process covering the whole search space with its basic GA search which is comparably a time consuming task. The structure of the GA employed by OptNetAlign is based on a multi-objective memetic approach that tries to discover a representative set of non-dominated alignments with respect to all desired objectives using swap-based operations of cross-over, mutation and hill climbing. Each cycle of the algorithm is initiated by searching random members of the population and performing cross-over and mutation on these alignments, followed by a local search performed with hill climbing. The hill climbing operation is randomly chosen either to make moves that do not worsen any objective or to improve one objective while ignoring the others. The achieved alignment is stored in a special fixed size archive data structure if it is not Pareto dominated by any other alignment and it is among the top ranking solutions. All non-dominated alignments in this archive are presented as the outputs of the application after a certain number of cycles specified by the user [92].

OptNetAlign [92] employs the permutation encoding scheme by labeling the nodes of networks with representative integers. These integers start from 0 and end with n-1 for both networks in which n denotes the number of nodes. Accordingly, a single alignment is represented with an array using the index numbers for the first network and index values for the other so that permutation based search operators can be used. The cross-over operator adopted in OptNetAlign is UPMX that is based on swapping the ith elements in two alignment arrays of the same size and same node number. The first swap operation causes there to be duplicate values in both arrays if the swapped values are not equal, therefore another swap operation is performed between the resulting duplicate values in both arrays . For local search, the algorithm involves two different swap based hill climbing approaches. The first one, hillClimbNonDominated, performs random swaps repetitively and rolls back a swap if any of the given objectives worsen. The second one, hillClimbOneObj, totally ignores the unspecified objectives by focusing on only one objective and performing a swap unless the given objective is worsened. The first hill climbing approach is used as a basic step to improve the archived alignments, and the second one is used to ensure a level of variance for the population in objective search space, so that it will be more possible to find stronger representatives to be archived. As a final remark, it is worth mentioning that fitness values are efficiently evaluated based on the difference after each swap in all cross-over, mutation and hill climbing operations since there is no other means to modify the permutations [92].

2.2.4 Supervised Optimization Method

The generic workflow of *SUMONA: A Supervised Method for Optimizing Network Alignment* can be described in detail as follows:

1. Input Alignment Generation: Alignments for two specific organisms are generated with aligners HubAlign, NETAL, and SPINAL. These network alignment algorithms generate a diverse set of outputs that are used as the initial population of the GA search of OptNetAlign. Each aligner is executed numerous times with different parameter values within an interval in order to generate a diverse and extensive set.

2. Classifying Input Alignments: Every alignment in the set of generated alignments has typical advantages with respect to different objectives. It is required to identify

the strong aspects of each and every alignment and classify them with respect to their common strengths in this step. For this reason, they are tested against all desired objectives and classified as strong alignments with respect to the objectives in which their alignment scores are above a predefined threshold. The blend of the final population can be controlled by using classified alignments in given proportions.

3. Establishing a Supervision Strategy: This step first focuses on a selection method for choosing an input alignment in a randomized way to be used as one of the parents of OptNetAlign cross-over process in every generation. The input alignments are used instead of one of the random parents either for a given period leaving the rest of the inputs as totally random or throughout all execution period. The reason of randomization is the necessity to preserve diversity for covering more search space. Diversity is provided by choosing a different member of the input alignments set with a characteristic rule in each generation. An applicable supervision strategy involves using the classified input alignments in given proportions and frequencies with a certain order so as to supervise the optimization process and allow prioritization. The desired prioritization may include a levelization between objectives which can simply be ensured by allowing more cycles for achieving more desired objectives. Another essential part of an supervision strategy is the final step of local search that is applicable for desired objectives such as the ones that score below a predefined threshold in each cycle. It's better to perform the local search with a proper single objective hill climbing method since its alternatives have a relatively high performance cost.

4. Reviewing the Execution Parameters: The effects of certain parameters such as the number of generations (cycles), number of hill climbing iterations and output population size are investigated in this step while the optimization process is executed. The execution can be repeated with different parameter values in case the optimization results are not successful enough.

The workflow of SUMONA can be performed with a few modifications in its main procedure. We designed and implemented two SUMONA variants with custom modifications in the course of this study. SUMONA1 procedure starts with inserting all the alignments that score above a given threshold in any of the predefined objectives to a common list. SUMONA1 does not classify the input alignments, but instead it uses the alignments from the input list one by one in each cycle of the Genetic Algorithm. Then it restarts again from the first list element in a cyclic fashion, when all the alignments of the input list has already been used. SUMONA1 does not perform a special single objective hill climbing operation but it rather performs the regular single objective hill climbing operation which randomly picks one of the objectives to improve likewise the original OptNetAlign algorithm. On the other hand, SUMONA2 first classifies the input alignments with respect to the given objectives by inserting the ones that score above a certain threshold value to the respective classes. A single alignment is included in more than one alignment class if its strength is not limited to one objective during this step. In the next step, the algorithm picks a random member of a randomly chosen class as one of the parents in each cycle of the Genetic Algorithm. This step ensures a balance among the user-defined objectives. Finally, SUMONA2 performs single objective hill climbing for all objectives that score below a predefined threshold in the hill climbing step of the algorithm. If all objectives score above their respective predefined thresholds, then single objective hill climbing is performed once for a random objective among the user-defined ones. This modification aims to focus on the easiest local improvements for the optimization process. SUMONA1 and SUMONA2 are described in detail with the following pseudo-codes in Algorithm 2.1 and 2.2. It may easily be seen from the pseudo-codes that the backbone of the algorithm stays the same in both variants apart from certain modifications.

The key variables in the algorithms and their explanations are the number of hill climb iterations (niters), cross-over rate (cxrate), swap probability at each index for cross-over (cxswappb), mutation rate (mutrate), swap probability at each index for mutation (mutswappb), probability of single objective hill climbing (oneobjrate), list of threshold values for objectives (trObjs). The most convenient values of these variables for efficient optimization results differ for each data set. For this reason, they need to be configured for each case. Besides, the success of network alignment algorithms depends on the quality of data as well as the evolutionary distance between species. Most network aligners have better results with synthetic data since most experimentally derived protein interaction networks are incomplete and noisy. Furthermore, the evolutionary distance between the aligned species should be small, if significant results are sought [114]. Finally, data format conversion between various data sources and network aligners is a practical problem to be solved when dealing with different data sets and network alignment methods. Some components of this problem are entity mapping (node, annotation, function, etc.), document structure, similarity ratio and scores and parsing data fields. In this study, we developed simple conversion tools for compatibility between all the aligners used in SUMONA.

Algorithm 2.1 SUMONA1 Variant

```
procedure SUMONA1($Inputs, Objectives, PopSize$)
    for all $input \in Inputs$ do
        if $InputStrongForAtLeastOneObjective$ then
            $insert(input, InputList)$
        end if
    end for
    $it \leftarrow 0$
    while time limit not reached do
        if $ArchiveSize \leq PopSize/2$ then
            $parent1 \leftarrow RandomAlignment(random)$
            $parent2 \leftarrow InputList(it \bmod NoofInputs)$
            $it \leftarrow it + 1$
        else
            $parent1 \leftarrow ArchivedAlignment(random1)$
            $parent2 \leftarrow ArchivedAlignment(random2)$
        end if
        $child \leftarrow initializeAlignment(parent1, parent2)$
        if $probability \geq cxrate$ then
            $child \leftarrow crossover(cxswappb, parent1, parent2)$
        end if
        if $probability \geq mutrate$ then
            $child \leftarrow mutate(mutswappb, child)$
        end if
        if $probability \geq oneobjrate$ then
            $randObj \leftarrow RandomlyChosenObjective$
            $child \leftarrow hillClimb(randObj, niters, child)$
        else
            $child \leftarrow hillClimbNonDominated(niters, child)$
        end if
        $child \leftarrow hillClimbNonDominated(niters, child)$
        $insert(child, Archive)$
    end while                                                  ▷ Per Thread
    return $ParetoAlignments$
end procedure
```

Algorithm 2.2 SUMONA2 Variant

```
procedure SUMONA2($Inputs, Objectives, trObjs, PopSize$)
    for all $Obj \in Objectives$ do
        for all $input \in Inputs$ do
            if $trObj \geq ObjScore$ then
                $insert(input, ObjectiveClass)$
            end if
        end for
    end for
    while time limit not reached do
        if $ArchiveSize \leq PopSize/2$ then
            $parent1 \leftarrow RandomAlignment(random)$
            $rand1 \leftarrow RandomlyChosenClass$
            $rand2 \leftarrow RandomlyChosenItem$
            $parent2 \leftarrow ObjectiveClass(rand1, rand2)$
        else
            $parent1 \leftarrow ArchivedAlignment(random1)$
            $parent2 \leftarrow ArchivedAlignment(random2)$
        end if
        $child \leftarrow initializeAlignment(parent1, parent2)$
        if $probability \geq cxrate$ then
            $child \leftarrow crossover(cxswappb, parent1, parent2)$
        end if
        if $probability \geq mutrate$ then
            $child \leftarrow mutate(mutswappb, child)$
        end if
        if $probability \geq oneobjrate$ then
            for all $Obj \in Objectives$ do
                if $trObj \geq ObjScore$ && $\neg hillClimbed$ then
                    $child \leftarrow hillClimb(Obj, niters, child)$
                    $hillClimbed \leftarrow true$
                end if
            end for
            if $\neg hillClimbed$ then
                $randObj \leftarrow RandomlyChosenObjective$
                $child \leftarrow hillClimb(randObj, niters, child)$
            end if
        else
            $child \leftarrow hillClimbNonDominated(niters, child)$
        end if
        $child \leftarrow hillClimbNonDominated(niters, child)$
        $insert(child, Archive)$
    end while                                                    ▷ Per Thread
    return $ParetoAlignments$
end procedure
```

2.3 Results

In this section, we discuss the performance and scores of SUMONA1 and SUMONA2 variants with respect to real and synthetic datasets. Besides, we test and analyze the running time of our experiments in order to compare SUMONA variants with OptNetAlign.

2.3.1 Tests Based on Real Data

Tests of real data are performed by aligning *Saccharomyces Cerevisiae* and *Drosophila Melanogaster* datasets included in Isobase [117] to optimize the prominent objectives of OptNetAlign including EC, GOC, S^3 and BitScoreSum. The population size parameter for the pareto front is 100 for all tests. Four possible supervision strategies are tested in this section. The first three supervision strategies are all based on SUMONA1 while the fourth one is based on SUMONA2. The optimization objectives used for the first three strategies are GOC, S^3 and BitScoreSum in the same order and these strategies use the default niters parameter of the OptNetAlign application as 0. Besides, all strategies use the default parameters cxrate=0.7, cxswappb=0.1, mutrate=0.1, mutswappb=0.005 and oneobjrate=0.1 respectively. Statistical significance of the comparison in each supervision strategy is evaluated with an unpaired t-test score by rejecting the null hypothesis that two experiments produce equal results if p-value<0.05. Furthermore, Bonferonni Correction is applied as a multiple testing methodology for comparing several objective scores of each experiment. This method is preferred since it does not require the tests to be independent. Bonferonni Correction is simply carried out by dividing p-value by the number of hypotheses to yield the statistical significance threshold for discovering familywise error rate. Therefore, the statistical significance threshold is calculated as 0.05/8 = 0.00625.

Supervision Strategy 1: This supervision strategy employs SUMONA1 and it assigns a different SPINAL result with a particular *Normalized Sequence Similarity Score* ratio as one of the parent inputs of the cross-over process in each cycle. This parent input changes by assigning the next SPINAL result from a previously prepared set in a cyclic fashion. This is performed by incrementing the Normalized Sequence Similarity Score ratio by 0.1 from 0 to 1 and restarting from 0 after all possible values are tried. This cyclic behaviour goes on throughout the execution of the optimization process.

The test results shown in *Table 2.1* compare one-parent cyclic SPINAL inputs with totally random cross-over parents. The tests were performed with 100, 200, 400, 600, 800, and

Table 2.1: Cyclic SPINAL vs. Totally Random

SPINAL	Average	Maximum	Minimum
EC	0,114	0,118	0,111
ICS	0,178	0,183	0,174
S^3	0,0746	0,0769	0,0725
GOC	115,55	123,04	103,47
BitScoreSum	70810,76	76996,5	60212,5
Size	5378,52	5389	5369
ICS×EC	0,02033	0,0214	0,0192
S^3Variant	0,128	0,133	0,124
Totally Random	**Average**	**Maximum**	**Minimum**
EC	0,0116	0,0117	0,01161
ICS	0,0309	0,0311	0,0307
S^3	0,00853	0,00859	0,00847
GOC	137,99	138,22	137,66
BitScoreSum	16000,22	16186,5	15758
Size	5499	5499	5499
ICS×EC	0,00036	0,000365	0,000355
S^3Variant	0,0118	0,0119	0,0117

Table 2.2: T-Test for Cyclic SPINAL vs. Totally Random

Objective	Test Score
EC	$3,22 \times 10^{-164}$
ICS	$1,36 \times 10^{-174}$
S^3	$5,87 \times 10^{-166}$
GOC	$3,50 \times 10^{-49}$
BitScoreSum	$2,10 \times 10^{-92}$
Size	2×10^{-132}
ICS×EC	5×10^{-145}
S^3Variant	2×10^{-159}

1000 cycles (generations) for observing the improvement in the results. Finally, the results for 1000 cycles are compared to realize the contribution of the generated inputs. The t-test

Table 2.3: Cyclic HubAlign with Different Cycles

1000 cycles	**Average**	**Maximum**	**Minimum**
EC	0,0187	0,0208	0,0142
ICS	0,0326	0,0358	0,0255
S^3	0,01203	0,0134	0,00919
GOC	365,21	367,58	362,28
BitScoreSum	373538,08	377282	370990
Size	5499	5499	5499
ICS×EC	0,000618	0,000747	0,000361
S^3Variant	0,0191	0,0213	0,0144
2000 cycles	**Average**	**Maximum**	**Minimum**
EC	0,0196	0,0226	0,0140
ICS	0,0368	0,0419	0,0266
S^3	0,013	0,0149	0,00925
GOC	369,48	374,97	361,59
BitScoreSum	362453,21	370640	354411
Size	5499	5499	5499
ICS×EC	0,000736	0,00095	0,000372
S^3Variant	0,0200	0,0231	0,0142

results in *Table 2.2* show that cyclic SPINAL has significantly better results with respect to all objectives. Besides, the maximum and minimum results achieved by Cyclic SPINAL are significantly higher as well except for Size objective.

Supervision Strategy 2: This strategy is similar to Supervision Strategy 1 in a sense that it uses inputs with a cyclic order while its only difference is using cyclic HubAlign results as inputs instead of SPINAL results. The parameter α, which determines the effect of sequence information against topological similarity, is incremented by 0.1 from 0 to 1 throughout each cycle. The parameter λ, which determines the effect of the edge weight against node weight, stays constant in the whole process with a value of 0.1.

The test results shown in *Table 2.3* compare the progress of one-parent cyclic HubAlign inputs of 2000 cycles with the same inputs of 1000 cycles. The success of cyclic HubAlign inputs is much lower than cyclic SPINAL inputs in most topological objectives while still being better than the original OptNetAlign algorithm. Nevertheless, cyclic HubAlign inputs

Table 2.4: T-Test for Cyclic HubAlign with Different Cycles

Objective	Test Score
EC	0,00425
ICS	$3,58 \times 10^{-11}$
S^3	$2,87 \times 10^{-5}$
GOC	$5,39 \times 10^{-20}$
BitScoreSum	$8,72 \times 10^{-48}$
Size	Undefined
ICS×EC	$3,07 \times 10^{-7}$
S^3Variant	0,00409

are far more better than cyclic SPINAL inputs in terms of functional objectives such as GOC and BitScoreSum. Additionally, HubAlign inputs show improvement in all topological parameters from 1000 cycles to 2000 cycles as shown by the similarity results of t-test in *Table 2.4*. Therefore, we may conclude that the progress in certain objectives do not take place in the expense of others for this strategy.

Supervision Strategy 3: The third supervision strategy we evaluated is a combination of cyclic one-parent SPINAL and NETAL inputs. The parameter α, which controls the weight of similarity and interaction scores in NETAL, varies from 0 to 1 and other parameters are fixed with default values. The SPINAL cyclic behavior is the same with Optimization Strategy 1. There are two versions of this strategy. In the first one, cyclic NETAL inputs are crossed over after three cycles of cyclic SPINAL inputs. In the second one, cyclic SPINAL inputs follow three cycles of NETAL inputs.

The test results of *Table 2.5* and *Table 2.6* show that the strategy of starting with cyclic SPINAL inputs has slightly better performance in topological objectives and oppositely the other one has slightly better performance in functional objectives when both strategies are run for 1000 cycles. Both strategies have better topological results compared to the previous tests. This fact confirms the evident performance of NETAL in topological objectives.

Supervision Strategy 4: The last supervision strategy is based on SUMONA2 unlike the others above. In this sense, this strategy is the most generic one compared to the others that take the order of inputs into account. The input data is the collection of SPINAL, NETAL and HubAlign data of the previous supervision strategies. There is no need to specify the order of inputs to be used in this strategy since SUMONA2 picks a random item from a

Table 2.5: Cyclic NETAL Following Cyclic SPINAL

SPINAL → NETAL	Average	Maximum	Minimum
EC	0,151	0,174	0,11
ICS	0,235	0,269	0,174
S^3	0,102	0,118	0,0722
GOC	93,62	113,33	77,1
BitScoreSum	48090,13	73304	27631,5
Size	5461,26	5499	5371
ICS×EC	0,0364	0,0469	0,0191
S^3Variant	0,179	0,211	0,123
NETAL → SPINAL	**Average**	**Maximum**	**Minimum**
EC	0,138	0,172	0,101
ICS	0,213	0,265	0,161
S^3	0,0916	0,116	0,0662
GOC	96,93	122,33	62,41
BitScoreSum	54243,67	83343	12418
Size	5418,28	5499	5330
ICS×EC	0,0303	0,0455	0,0163
S^3Variant	0,161	0,208	0,112

Table 2.6: T-Test for SPINAL → NETAL vs. NETAL → SPINAL

Objective	Test Score
EC	0,000111
ICS	$3,4 \times 10^{-5}$
S^3	$6,77 \times 10^{-5}$
GOC	0,102
BitScoreSum	0,0319
Size	$1,93 \times 10^{-7}$
ICS×EC	$5,17 \times 10^{-5}$
S^3Variant	0,000103

randomly chosen class in each cycle. On the other hand, an additional optimization objective is used in the respective experiments of this strategy in order to achieve balance between the number of topological and functional objectives. The user-specified objectives are EC, S^3, BitScoreSum and GOC in the same order. One other used parameter is niters which has been tried with values 0, 500, 1000, 1500 and 2000 in order to observe the trend in its effect. It is concluded that this parameter has limited effect if it is increased beyond a certain value and its most effective values are 1000 and 1500 among all the trials in terms of execution time and achieved results. As a result, the test results of 1500 and 0 hill climb iterations are compared with each other to observe the effect of hill climbing. Additionally, the treObjs parameters of SUMONA2 are determined as the average score of the inputs for each objective. Finally, the effects of establishing an Unclassified Class for inputs that are not inserted into any of the objective classes is observed by giving it the same probability to be chosen with all the Objective Classes in each cycle of this supervision strategy. However, it is realized that the Unclasified Class has no significant positive effect on the optimization results due to its limited number of items. This modification does not yield a significant diversity for optimization, since there are few alignments in the Unclassified class. For this reason, the Unclassified Class is not used in the experiment.

Test results of *Table 2.7* and *Table 2.8* show that the performance of this supervision strategy significantly improves in all objectives except Size. However, there is a significant decline in Size objective due to the SPINAL inputs used during cross-over. Apart from that, the overall performance of this strategy is much better compared to the other ones when it is run for 1000 cycles likewise the other tests. This fact can simply be observed by comparing the average and maximum values of all the objectives that each strategy achieves. Finally, Supervision Strategy 4 with hill climbing surprisingly achieves the best scores of GOC and ICS among all strategies, although it is expected to achieve the optimum values considering its input data.

Table 2.9 shows the individual performance of HubAlign, NETAL and SPINAL algorithms for comparison. The performance is evaluated with respect to the specific 11 sample alignments for each algorithm that are mentioned in the previous scenarios.

If the output alignment sets are examined, it will be seen that 1000 cycles of SUMONA2 (Supervision Strategy 4) with 1500 “hill climb iterations” generate alignments that surpass the maximum values of other aligners in GOC and ICS objectives in which those aligners are most successful. The individual alignments generated by SUMONA2 are also significantly better in other objectives compared to the performance of the respective aligners. Therefore, we may conclude that our algorithm SUMONA generates alignments that pareto dominate other well known aligners with 1000 cycles which is so short in the scale of optimization

Table 2.7: All inputs with 1500 vs. 0 Hill Climb Iterations

1500 iterations	**Average**	**Maximum**	**Minimum**
EC	0,0975	0,169	0,0594
ICS	0,22	0,337	0,138
S^3	0,073	0,127	0,0435
GOC	366,15	510,97	141,96
BitScoreSum	170974,74	310214	10785
Size	4206	4561	4042
ICS×EC	0,0237	0,0570	0,00829
S^3Variant	0,11	0,204	0,0632
no iterations	**Average**	**Maximum**	**Minimum**
EC	0,076	0,158	0,00903
ICS	0,123	0,247	0,0142
S^3	0,0502	0,107	0,00556
GOC	145,27	330,22	45,73
BitScoreSum	121881,96	344004	613
Size	5407,46	5499	5351
ICS×EC	0,0127	0,039	0,000129
S^3Variant	0,085	0,188	0,00911

Table 2.8: T-Test for 1500 vs. 0 Hill Climb Iterations

Objective	**Test Score**
EC	0,000161
ICS	$1,32 \times 10^{-18}$
S^3	$6,85 \times 10^{-8}$
GOC	$7,02 \times 10^{-31}$
BitScoreSum	0,000649
Size	$7,45 \times 10^{-124}$
ICS×EC	$4,53 \times 10^{-8}$
S^3Variant	0,000329

processes.

Table 2.9: Inputs of HubAlign, NETAL and SPINAL

HubAlign	**Average**	**Maximum**	**Minimum**
EC	0,0205	0,0252	0,012
ICS	0,0305	0,0352	0,0174
S^3	0,0124	0,0149	0,00716
GOC	307,75	393,05	49,55
BitScoreSum	327699,55	419155	313
Size	5499	5499	5499
ICS×EC	0,00064	0,000886	0,000209
S^3Variant	0,0209	0,0259	0,0122
NETAL	**Average**	**Maximum**	**Minimum**
EC	0,216	0,237	0,0171
ICS	0,303	0,333	0,0242
S^3	0,146	0,161	0,0101
GOC	46,92	47,06	46,66
BitScoreSum	737,09	798	403
Size	5499	5499	5499
ICS×EC	0,0711	0,0792	0,000415
S^3Variant	0,282	0,311	0,0174
SPINAL	**Average**	**Maximum**	**Minimum**
EC	0,163	0,169	0,134
ICS	0,233	0,246	0,211
S^3	0,106	0,11	0,0893
GOC	105,39	119,56	43,56
BitScoreSum	63875,05	76669,5	267,5
Size	5241,82	5389	4481
ICS×EC	0,0379	0,0406	0,0283
S^3Variant	0,194	0,204	0,155

2.3.2 Tests Based on Synthetic Data

A sample dataset from NAPAbench based on the crystal growth (CG) [118] model is also tested as an example with synthetic data [119]. The reason that CG is chosen among other well known alternatives is that it tries to emulate network topology and characteristic age distribution features of real protein interaction networks so that a meaningful alignment can be achieved with respect to network evolution.

The chosen dataset from NAPAbench CG datasets is the Pairwise Family 2 dataset which contains 3000 nodes for the synthetic organism A and 4000 nodes for the synthetic organism B. The Functional Orthology definitions of NAPAbench were used in place of the Functional Annotations in the experiments. The input alignments of SUMONA2 were generated with SPINAL, HubAlign and NETAL in the same fashion with Supervision Strategy 4 except for NETAL that uses a biological similarity coefficient (b) of 0.5 and a neighbor contribution coefficient (c) of 0.5 for 11 different values of α from 0 to 1. The user-specified objectives are EC, S^3, BitScoreSum and GOC in the same order.

As a result, 33 inputs were generated 13 of which could surpass none of the user-specified threshold values that are the average score of inputs for each optimization objective respectively. Conversely, the remaining 20 input alignments were successful with respect to all the user-specified objectives. Therefore, the classification step became simply a filtering process for eliminating weaker inputs.

Both OptNetAlign and SUMONA2 were performed for 1000 cycles with niters=1500 hill climb iterations in each cycle. The threshold conditions of the post hill climbing step were determined by multiplying the original threshold values of EC and S^3 by 1.4 and multiplying the original threshold values of GOC and BitScoreSum by 1.6 for better performance. As a result, SUMONA2 outperformed OptNetAlign in every user-specified objective as shown in *Table 2.10* with the respective t-test in *Table 2.11*. Besides, SUMONA2 reached the most significant scores in ICS and S^3 objectives. It may also be observed that there is not enough diversity in the results of most objectives of SUMONA2. This fact is due to the lack of proper classification and very similar performance of various input alignments in the previous phase.

2.3.3 Optimization Trends in Real Datasets

Fig. 2.2, 2.3, 2.4 and 2.5 show the improvement of several user-specified objectives as the GA cycles proceed. A total of 1300 cycles with 1500 hill climb iterations are run using SUMONA2 in order to analyze the trends in the user-specified EC, S^3, BitScoreSum and GOC objectives respectively. The dataset used is all the alignments generated by SPINAL, HubAlign and NETAL exactly in the same way with Supervision Strategy 4. The trends in all user-specified objectives are observed for intervals of 100 cycles periodically. All the data

Table 2.10: SUMONA2 vs. 0ptNetAlign with Synthetic Data

SUMONA2	Average	Maximum	Minimum
EC	0,736	0,737	0,735
ICS	0,879	0,881	0,878
S^3	0,668	0,67	0,667
GOC	2195,61	2197	2195
BitScoreSum	518833,93	521091	516494
Size	2640,69	2643	2639
ICS×EC	0,647	0,649	0,645
S^3Variant	2,79	2,8	2,77
OptNetAlign	**Average**	**Maximum**	**Minimum**
EC	0,298	0,317	0,279
ICS	0,446	0,475	0,423
S^3	0,217	0,234	0,202
GOC	391,23	454	275
BitScoreSum	122594,59	137102	92132,9
Size	3000	3000	3000
ICS×EC	0,133	0,15	0,118
S^3Variant	0,424	0,464	0,386

Table 2.11: T-Test for SUMONA2 vs. 0ptNetAlign with Synthetic Data

Objective	Test Score
EC	$8,1 \times 10^{-168}$
ICS	$6,25 \times 10^{-147}$
S^3	$2,25 \times 10^{-178}$
GOC	$1,06 \times 10^{-161}$
BitScoreSum	$2,72 \times 10^{-152}$
Size	8×10^{-171}
ICS×EC	$3,59 \times 10^{-186}$
S^3Variant	$2,4 \times 10^{-304}$

is collected from one single experiment for performing consistent observations.

The optimization objectives EC and S^3 are contradictory to BitScoreSum and GOC since

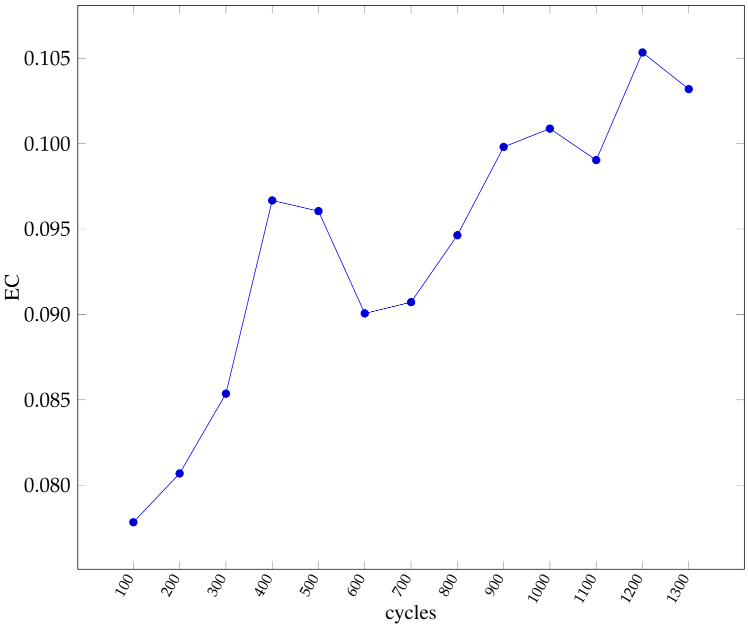

Figure 2.2: Trend of EC Score vs. Cycles. A cycle is the execution of the cross-over, mutation and local search steps of the GA for once in a population.

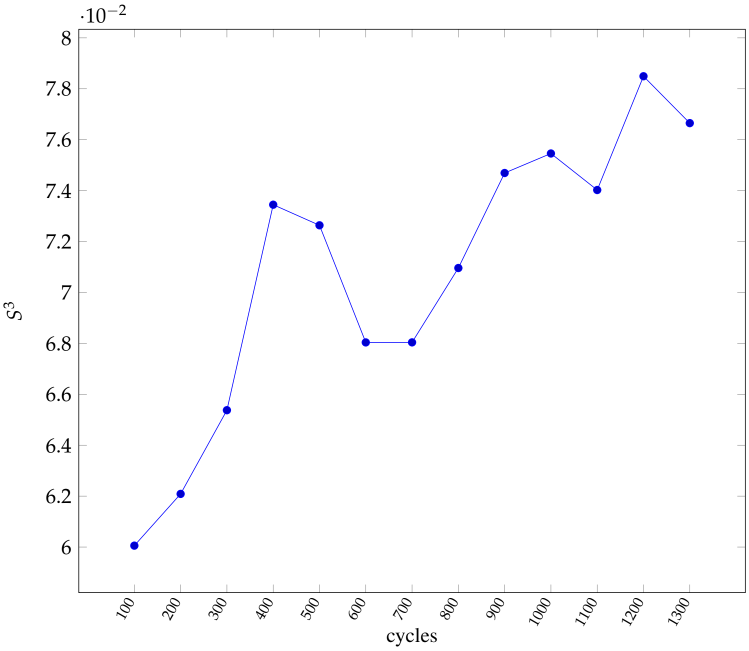

Figure 2.3: Trend of S^3 Score vs. Cycles. A cycle is the execution of the cross-over, mutation and local search steps of the GA for once in a population.

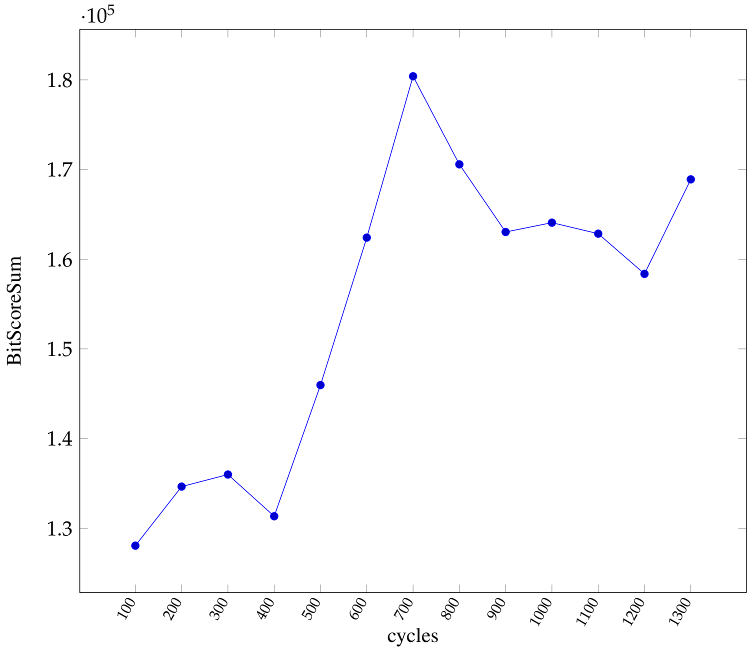

Figure 2.4: Trend of BitScoreSum Score vs. Cycles. A cycle is the execution of the cross-over, mutation and local search steps of the GA for once in a population.

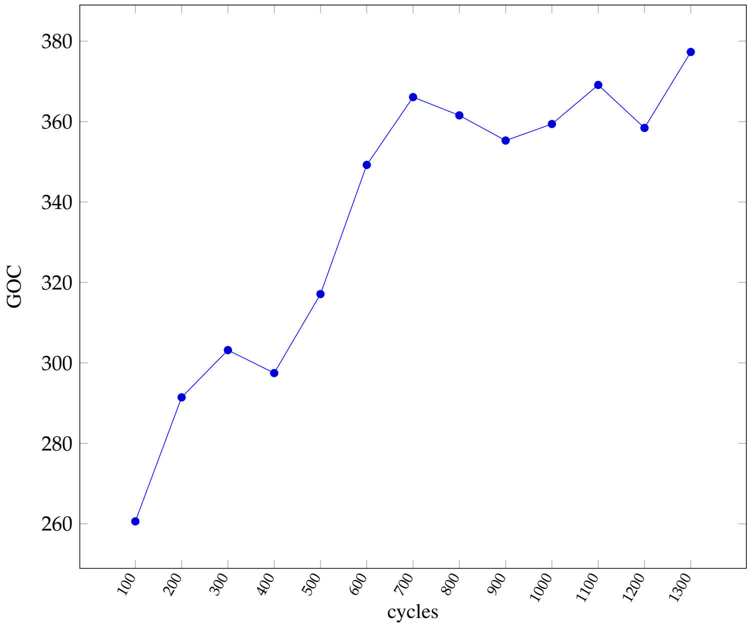

Figure 2.5: Trend of GOC Score vs. Cycles. A cycle is the execution of the cross-over, mutation and local search steps of the GA for once in a population.

the first two are topological and the latter are functional objectives. That's why optimizing these objectives becomes a relatively slow process compared to optimizing a consistent set of objectives. Nevertheless, it is obviously seen that all objectives are improved in the long run, apart from fluctuations in the objective score of each objective. *Fig. 2.2* and *Fig. 2.3* show that EC and S^3 trends have similar curves with each other while it is seen from *Fig. 2.4* and *Fig. 2.5* that BitScoreSum and GOC have relatively less similar trends.

As a final remark, none of the trends of the user-specified optimization objectives indicate a sign of pre-mature convergence in our observations. The approach that prevents the algorithm to be trapped into a certain local optimum is initiating the algorithm with a diverse and randomized set of data in order to cover the search space as much as possible. Besides, our method encourages using the result sets of aligners other than SPINAL, HubAlign and NETAL with various parameters for improving the diversification upon user preference.

2.3.4 Running Time Analysis of Real Datasets

SUMONA achieves better optimization results compared to OptNetAlign in 1000 cycles as a standard which approximately means the same running period for both methods. In other words, OptNetAlign would achieve similar optimization results in a longer running period. Time and Space Complexity of SUMONA are almost similar to OptNetAlign, since our modifications in the backbone of OptNetAlign do not include complicated inner loops and/or extra memory consuming data structures and our additional computational terms has negligible effect in terms of time and space complexity. Besides, best and average case scenarios of SUMONA have an advantage in terms of time and memory due to supervised data. Running time of our algorithm is analyzed with the *time* command used in most Linux and Unix systems. Running time analysis include usage of user and system time resources assuming that the Central Processing Unit (CPU) usage is 100%. The user time mainly corresponds to the internal operations of an application while the system time is the time that the application spends for system calls to the operating system.

Fig. 2.6 shows the time comparison of SUMONA1 and OptNetAlign with 0 and $3\times500=1500$ iterations of hill climbing where 500 is the coefficient used for conversion of iterations among various local search algorithms in the original algorithm. The Cyclic SPINAL inputs of Supervision Strategy 1 are used for the respective SUMONA1 tests. The optimized objectives are GOC, S^3 and BitScoreSum respectively.

The test results indicate that SUMONA1 is 36.09 CPU seconds behind OptNetAlign when

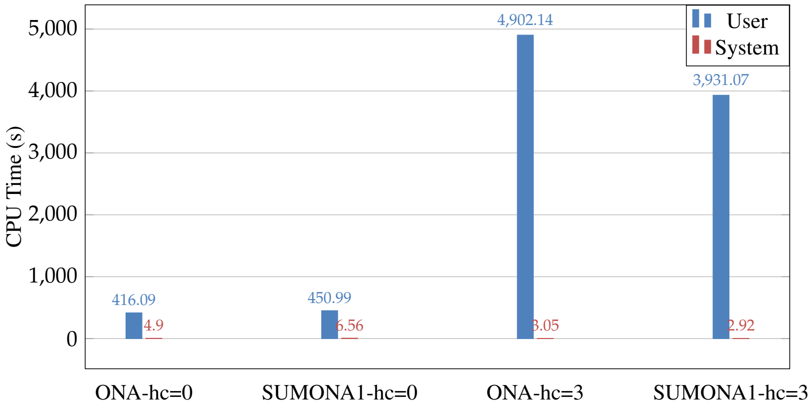

Figure 2.6: Running Time Comparison of SUMONA1 and OptNetAlign (ONA). The objectives of the experiments are GOC, S^3 and BitScoreSum. The comparison was carried out twice with different number of Hill Climbing Iterations (hc). The number of Hill Climbing Iterations (hc) are multiplied by an embedded coefficient of 500.

there is no post hill climbing step at all. This small gap may be due to the insertion of source alignments from SPINAL outputs before initiating the optimization. On the other hand, SUMONA1 surpasses OptNetAlign by 971.2 CPU seconds when the test is performed with 3×500=1500 iterations of post hill climbing step at each cycle. Since the cross-over operations of SUMONA1 produce more successful results, it becomes easier to perform successful optimization results in the following local search phase. Therefore, the successful optimizations during the computationally expensive hill climbing step yield a significantly better result in terms of time.

Fig. 2.7 shows the time comparison of SUMONA2 and OptNetAlign with 0 and 3×500=1500 iterations of hill climbing. The source alignments are all of the SPINAL, HubAlign and NETAL alignments used in Supervision Strategy 4. The optimized objectives are EC, S^3, BitScoreSum and GOC respectively.

The test results indicate that OptNetAlign performs 2.26 CPU seconds better than SUMONA2 when there is no post hill climbing step at all. The reason for this fact is the balance between functional and topological objectives in the source data of SUMONA2 and optimization objectives of the test. On the other hand, SUMONA2 performs 639.68 CPU seconds better than OptNetAlign with 3×500=1500 iterations of post hill climbing search. The difference is very significant but it is slightly less than the difference of SUMONA1-hc=3 and ONA-hc=3 shown in *Fig. 2.2*. The decline in the difference is due to the fact that SUMONA2 focuses on

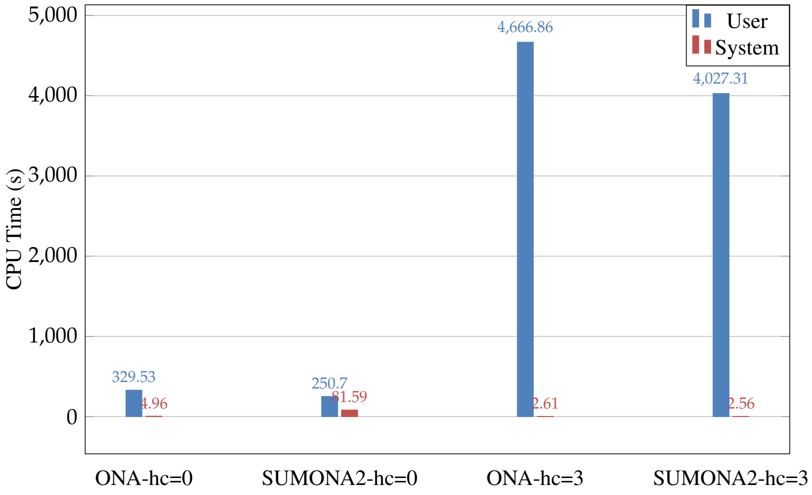

Figure 2.7: Running Time Comparison of SUMONA2 and OptNetAlign (ONA). The objectives of the experiments are EC, S^3, BitScoreSum and GOC. The comparison was carried out twice with different number of Hill Climbing Iterations (hc). The number of Hill Climbing Iterations (hc) are multiplied by an embedded coefficient of 500.

less successful objectives in the hill climbing step.

2.4 Discussion of Results and Future Work

SUMONA has been supervised by a set of alignment inputs generated by HubAlign, SPINAL and NETAL and then subsequently compared to its main competitor OptNetAlign with respect to multiple objectives throughout this study. In this scope, we carried out various multi-objective alignment tasks with both of the competitors and then evaluated the significance of the difference achieved by the superior aligner with a t-test for each objective. Besides, we performed SUMONA experiments with a variety of supervision strategies that were explained in Section 2.3 as well as a variety of alignment objectives, execution cycles and input alignments. The implementation details and experiment parameters are further explained in Appendix A.1. As a result, it was observed that SUMONA achieved superior results than totally random OptNetAlign especially in objectives that the respective supervision inputs are strong. It was additionally observed that supervision with SUMONA positively effected the time frame that significant results are achieved. Besides, one key finding was that it is possible to achieve more significant results with a more diverse set of input alignments. In addition, temporal trends of alignment results showed that the diversity of input alignments has a positive effect on premature convergence since contradictory objectives display continuous progress with diverse input alignments. Another key finding was the positive effect of supervised hill climbing in terms of the rate and significance achieved in the designated objectives. This finding majorly proves the necessity of employing local search or individual heuristics along with a global meta-heuristics approach for this problem. Finally, the experiments showed that the cyclic order of the supervising input alignments had a statistically significant effect on the results of each objective which might hint the importance of initialization, or in other words, the core set of mappings in alignment tasks.

One possible improvement of SUMONA method can be dynamically choosing one of the random cross-over inputs of the next cycle according to the progress direction of each member of the population. The next input can be some other generated alignment which is more successful with respect to the objectives that the particular population member is not good at. However, the drawback of this improvement might be that the input selection may be in a continuous shift between two uniform sets of alignments that have opposite strengths. Premature convergence due to focusing on a constant objective set should also be tackled in such an improvement.

Other possible extensions to SUMONA are:

- Employing different supervision strategies in dedicated threads can provide alternative solutions for different data sets during parallelization. Unsuccessful dedicated threads can be killed and successful ones can be prioritized in the meantime for better performance.

- Development of a multi-agent approach can lead to more proactive and dynamic decisions in a changing optimization environment. Possible productive alignments can be exchanged between agents in such an environment.

- Semi-global alignment inputs can be generated for OptNetAlign so that the search space is narrowed for randomizing and optimizing the remaining nodes.

- Setting LCCS as an optimization objective might be another improvement for OptNetAlign. A dynamic calculation strategy can be proposed for this purpose.

- Cluster similarity and cluster alignment of functionally orthologous proteins can be used as a new objective,

- The biological relevance of the produced alignments can be interpreted by examining how well the aligners align orthologous proteins [114].

3 Parallel Exchange of Randomized SubGraphs for Optimization of Network Alignment: PERSONA

The aim of Network Alignment in Protein-Protein Interaction Networks is discovering functionally similar regions between compared organisms. One major compromise for solving a network alignment problem is the trade-off among multiple similarity objectives while applying an alignment strategy. An alignment may lose its biological relevance while favoring certain objectives upon others due to the actual relevance of unfavored objectives. One possible solution for solving this issue may be blending the stronger aspects of various alignment strategies until achieving mature solutions. This chapter proposes a parallel approach called PERSONA that allows aligners to share their partial solutions continuously while they progress. All these aligners pursue their particular heuristics as part of a particle swarm that searches for multi-objective solutions of the same alignment problem in a reactive actor environment. The actors use the stronger portion of a solution as a subgraph that they receive from leading or other actors and send their own stronger subgraphs back upon evaluation of those partial solutions. Moreover, the individual heuristics of each actor takes randomized parameter values at each cycle of parallel execution so that the problem search space can thoroughly be investigated. The results achieved with PERSONA are remarkably optimized and balanced for both topological and node similarity objectives.

3.1 Background

Network Alignment generates node mappings between networks of organisms in question in order to compare them functionally. Alignment results can be used in various areas such as predicting functions of unannotated proteins, revealing mechanisms of certain diseases and reproducing a rooted phylogenetic tree based on the discovered evolutionarily conserved pathways or protein complexes and detected functional orthologs across species [120]. Most Global Network Alignment algorithms rely upon the assumption that the functions of smaller networks map one-to-one to the functions of bigger networks homologously unlike most Local Network Alignment algorithms that focus on overlapping highly conserved subnetworks by allowing many-to-many node mappings [121–123].

One of the most significant drawbacks of the existing one-to-one Global Network Alignment algorithms is the lack of homogeneity across the mappings of an alignment in terms of quality. This problem arises since the main heuristic of an alignment algorithm is applicable only for a certain proportion of mappings. This means it may be beneficial to alter the alignment heuristic during the progression of an alignment process according to its current performance and it is worth evaluating the contributions of every new set of mappings to an alignment individually. Such an interactive approach can be achieved by means of a querying mechanism capable of classifying the contributions of a particular set of mappings in terms of various topological similarity and node similarity metrics as well as various alignment heuristics that prioritize different metrics. The interactivity can further be extended with a collaborative infrastructure that orchestrates a population of aligners and enables exchanging significant mappings among population members that are strong in different metrics.

This chapter proposes a hybrid approach that combines several fundamental alignment heuristics and meta-heuristic search tools to design a custom multi-objective optimization work-flow and a population of custom designed aligners that interact with each other collaboratively for solving the Global Network Alignment problem against multiple objectives. The most significant traits of PERSONA are adaptation to new data sets, adjustment among objectives with high precision and providing mature alignments that are balanced with respect to all objectives. This chapter is structured as follows: Firstly, globally recognized performance objectives of the Global Network Alignment problem are introduced. Later on, the multi-objective optimization approach of PERSONA is introduced by explaining the role of employing various heuristics in the process, the execution steps of the algorithm and the architecture of the framework. Following the methodology, performances of state-of-the-art multi-objective aligners are compared with respect to various data sets. Finally, the results are discussed and the achieved highlights were summarized as a conclusion.

3.2 Methods

The PERSONA methodology is designed to succeed in multiple objectives that make most sense for a global Protein-Protein Interaction Network comparison. However, it is not currently possible to make a prioritization among these objectives or identify robust aggregate objectives out of the existing ones. For this reason, we represent the Global Network Alignment problem as a multi-objective optimization problem in order to achieve optimal performance in each of these objectives. In this context, we propose the collaborative methodology that depends on exchanging essential subgraph information throughout this chapter. Besides, we also introduce a heuristics suite enabling to design a custom singular alignment to generate multiple aligners with different characteristics. Subsequently, we explain the graph specific persistency infrastructure that stores the essential network characteristics and alignments in progression as well as the concurrent computation architecture that handles the collaboration tasks among the entities of the whole system for implementing the methodology.

3.2.1 Performance Objectives

One major compromise for solving a network alignment problem is the trade-off among multiple alignment objectives that evaluate alignment performance. Alignment performance can be computed with respect to topological similarity and node similarity metrics. Topological similarity objectives aim to address the functional similarities of the organisms in terms of their network structure and interaction patterns and evaluates node pairs that contribute to similar interaction patterns from both of the organisms to be aligned. On the other hand, node similarity objectives aim to address the functional similarity of node pairs from both of the compared organisms individually without considering their network structures. In this study, alignment quality is evaluated with several well known topological and node similarity objectives used in other Global Network Alignment studies [90, 92, 99, 116, 124] as summarized below:

EC is an early topological similarity objective computed as the ratio of the edges in the smaller network mapped to the edges in the bigger network to all the edges in the smaller network. Mathematically $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ represents two graphs or networks while $g : V_1 \rightarrow V_2$ represents the alignment between two networks. Therefore, the mathematical representation of EC becomes [83]

$$\frac{|\{(u,v) \in E_1 : (g(u), g(v)) \in E_2\}|}{|E_1|} \times 100 \tag{3.1}$$

$\boldsymbol{S^3}$ is another topological similarity objective derived from EC that penalizes unaligned edges in both the larger and the smaller network. In $\boldsymbol{S^3}$, G[V] is represented as an induced subnetwork of G with the node set V, while E(G) is represented as the edge set of G. If we denote $f(E_1) = \{(g(u), g(v)) \in E_2 : (u,v) \in E_1\}$ and $f(V_1) = \{g(v) \in V_2 : v \in V_1\}$, then $\boldsymbol{S^3}$ becomes [100]

$$\frac{|f(E_1)|}{|E_1| + |E(G_2[f(V_1)])| - |f(E_1)|} \tag{3.2}$$

LCCS is the number of aligned node pairs in the largest connected subgraphs of an alignment that exist in both networks with an exact copy. The necessity for this measure is due to the significance of aligning large and contiguous subgraphs rather than a number of small disconnected network regions that would not give an equivalent insight into common topology of two networks [6]. For computing the aligned LCCS, first LCCS is calculated for the aligned nodes of both networks seperately and then their intersection is computed by considering the member node pairs. The ratio of these node pairs to the size of the whole alignment gives the corresponding normalized score for this objective.

GOC is a node similarity objective in which every aligned protein pair contributes to the overall GOC score by adding it the ratio of the size of the intersection of their common GO terms to the size of the union of their GO terms. The overall GOC score is computed by summing up these ratios coming from all aligned protein pairs. The Jaccard-Index based formula below represents the contribution of a single aligned node pair to the overall GOC score [89]:

$$GOC(u,v) = \frac{GO(u) \cap GO(v)}{GO(u) \cup GO(v)} \tag{3.3}$$

Gene Ontology Enrichment (GOE) is the percentage or number of aligned protein pairs that share at least one GO term annotation. This metric is a node similarity objective that is more adapted to the Global Network Alignment problem [125].

Resnik Similarity is one of the most popular semantic similarity measures [126]. The method computes the similarity between two terms as the Information Content of the Most Informative Common Ancestor in a particular Annotation Corpus [127]. The Information Content of a concept can be considered as its likelihood particularly in the Gene Ontology Annotation Database based on the annotation frequency [127, 128]. Resnik method is a pairwise term semantic similarity measure that is not directly applicable to genes and proteins and it can be adapted to proteins with a mixing strategy based on the average or maximum of all term pairwise similarities as well as the average of similarity between best matching terms [126].

BS is a node similarity measure calculated by summing up the BLAST bitscores of the aligned protein pairs [92, 114]. The biological sequence similarity for all the possible node pairs from both of the aligned organisms can be measured with BLASTP [129, 130] application using the corresponding FASTA texts of the proteins with user defined values for parameters such as word-size and e-value.

The exact biological relevance of all these objectives are yet to be discovered. For this reason, they can not precisely be prioritized among each other and it becomes mandatory to review all objective scores as a whole to get a better meaning. Nevertheless, it is possible to establish a particular aggregate objective function that evaluates multiple objectives intuitively in order to yield decisive results. We used such intuitive objective functions in Section 3.3.4 for efficient evaluation of our experiments.

3.2.2 Collaborative Method

Global Network Alignment is inherently a multi-objective optimization problem since the prioritization of the objectives among each other is a grey area despite certain definitions that highlight the prior significance of node similarity objectives over topological ones due to the verifiable evolutionary conservation information [110]. Since there is no obvious prioritization among objectives, it makes sense to preserve a set or population of alignments that are stronger in different objectives and able to learn from each other. PERSONA is inspired by the concept of Particle Swarm Optimization [131] as a meta-heuristics approach that intends collaboration among a population of aligners to optimize multiple objectives.

The original definition of Particle Swarm Optimization problem needs to be revised by concepts of multi-objective optimization as well as domain-specific properties for defining Global Network Alignment as a multi-objective problem. The most essential multi-objective optimization definition required for this revision is that a sample solution of a problem is considered pareto optimal [132] or non-dominated if improving the score of one of the objectives would require worsening other objective scores in the current solution space. Additionally, the concept of pareto dominance is also important for comparing two solutions. According to this definition, solution x pareto dominates another solution y, if it is strictly better than y with respect to at least one objective and is at least equivalent to y with respect to the remaining objectives [92]. Based on these definitions, a possible domain-specific revision is defining multiple swarm leaders out of the non-dominated solutions to map the fair number of problem objectives. Yet, it is possible that excessive number of non-dominated solutions exist and some of them are repetitive since they reside within the same neighborhood or in close distance. As a result, a leader selection methodology becomes an essential part of the process [133]. The selected leaders among the non-dominated solutions heavily effect the convergence rate and diversity of solutions. On the other hand, some algorithms restrict the number of stored non-dominated solutions by filtering a relatively useless set of them in order to improve performance. This is not an easy task since an exact quantification is not possible among multiple objectives and it may lead to a compromise in the diversity of the population [134].

There is a need to establish a robust trade-off mechanism in Particle Swarm Optimization in order to achieve a balanced pareto optimality among objectives while preserving the diversity in the population for preventing premature convergence. PERSONA uses a straightforward strategy of assigning each objective a dynamic leader with maximum objective performance to limit the number of global leaders in the swarm. Furthermore, PERSONA stores all the historical leader solutions in an archive in order to be capable of recalling their distinct characteristics for diversity. Additionally, PERSONA performs a parallel execution where each aligner of its population is scheduled to periodically execute its individual heuristics within its particular alignment strategy while the collaboration tasks are also scheduled to interactively perform amongst all aligners. The aligner population is diversified by focusing each member to primarily different objectives so that distinct regions of the search space can be scanned by the population. The meta-heuristic search tools described below are the flexible building blocks of the collaborative methodology:

Broadcasting and Following Leading Aligners of Each Objective: The global best aligner of each objective broadcasts its corresponding partial solution as a subgraph to the remaining follower aligners periodically (see Appendix B.2).

Exchanging Partial SubGraphs with respect to Objective Score comparison: Every aligner sends their scores in each objective to all other aligners for pairwise multi-objective comparison and check whether the receiver or the sender pareto dominates the other by being superior in all the objective scores. If this is the case, then the pareto dominated party receives the whole alignment of the other one as a subgraph. Otherwise, the original sender sends one of its significant subgraphs by random objective selection (see Appendix B.1).

Removing Low Scoring Mappings for Unprogressive Objective Scores: Low scoring mappings are removed by random objective selection when their respective alignment does not improve for a long period. This operation is useful for avoiding local maximizations.

Random Search: A number of random mappings are occasionally added to each alignment so that divergent solutions can be searched in the search space.

Recalling Historically Significant Partial Solutions: A random instance from The Global Best Scores History is broadcasted periodically to serve as the social memory or in other words experience of the aligner population.

Periodically Calling Various Heuristics with Random Parameters and Certain Probabilities: Every aligner of the population periodically performs its particular alignment approach based on certain heuristics with random parameter values in certain boundaries and probabilities of occurrence as part of its individual behavior. Further explanation can be found in Section 3.2.3 about possible heuristics that each aligner may perform individually.

The crucial collaboration tasks that these tools perform are based on extracting and exchanging the most significant subgraphs as a solution subset for each of the above mentioned objectives. The exchanged subgraph entities are extracted via special queries for each objective. These queries are explained in Appendix B.3 in detail. Apart from that, actors that may be defined as objects that encapsulate a state or behavior [135] are used as individual aligners in PERSONA due to their capability of exchanging messages with others and storing individual state information. In this context, a Tail-Chopping algorithm based scheduling approach [135] is used for choosing the next available aligner while performing interactive tasks such as exchanging results between aligners or broadcasting leading alignments that are shown in Algorithm 3.1 and 3.2 :

3.2.3 Aligner Heuristics Suite

PERSONA proposes various heuristics that can compose the characteristic behaviors of an individual aligner. Each aligner may have multiple heuristics as part of their behavior and each heuristic may have a user defined probability of occurrence at each execution cycle. These heuristics have been implemented with the Cypher Query Language [136] of the Neo4J Graph Database [137] infrastructure in order to make the search operations based on explicit reasoning. The Alignment Heuristics Suite developed as part of the study is described in three main groups below and it is further explained in Appendix B.4:

Seed and Extend Approaches: In this group of heuristics, the significant seeds are identified with respect to topological centrality scores of seeding pairs in addition to their node similarity thresholds. The alignment is intended to propagate to neighboring edges after identifying the central node pairs with a chosen centrality approach from this group. The topological centrality approaches present in this group are described by several studies [138–141] as follows:

- Page Rank: is the iteratively accumulated transitive influence or connectivity of each node distributed over its neighbors. The influence is computed by counting the frequency of hitting each node during a random traversal.

Algorithm 3.1 Send and Receive Policy for Individual Comparison and Exchange. ***SG=SubGraph, LOBO= List of Better Objectives, RSO = Randomly Selected Objective*

(a) Send Policy

```
procedure SEND($router$)
    for all $a \in aligners$ do
        $receiver \leftarrow scheduleNextReceiver(router))$
        $sendObjectiveScores(a, receiver, SG(message))$
    end for
end procedure
```

(b) Receive Policy

```
procedure ONRECEIVE($message$)
    $senderScores \leftarrow receiveScores(message)$
    if $ParetoDominate(senderScores, receiverScores)$ then
        $addAllPossibleMappings(sender, receiver)$
    else if $ParetoDominate(receiverScores, senderScores)$ then
        $sendAllPossibleMappings(SG(receiver), sender)$
else
    $LOBO \leftarrow listBetterObjectives(sender, receiver)$
    $RSO \leftarrow randomlySelectOneObjective(LOBO)$
    $sendBackSubGraph(SG(RSO), sender)$
end if
end procedure
```

Algorithm 3.2 Send and Receive Policy for BroadCast and Following Leading Aligners

(a) Send Policy for Broadcast

```
procedure BROADCAST
    for all $o \in objectives$ do
        $alignerID \leftarrow findAlignerWithBestScores(o)$
        $key \leftarrow markSubGraph(o, alignerID)$
        $broadcastBestAlignersSubGraph(key, noSender)$
    end for
end procedure
```

(b) Receive Policy for Following Leading Aligners

```
procedure ONRECEIVE($message$)
    $key \leftarrow receiveSubGraphMark(message)$
    $addSubGraphToAlignment(key)$
end procedure
```

- Betweenness Centrality: is the frequency that a node acts as an intermediary node between other nodes on a shortest path. This metric indicates the global importance of a node in terms of providing access and connectivity to other nodes.
- Closeness Centrality: is the average distance of a node to all other nodes in a network based on the shortest path.
- Harmonic Centrality: is a variant of closeness centrality that is based on the inverse of the distances of all other nodes rather than their distances.
- Connectivity Degrees: is the number of local duples, triples or quadruples that a node is a part of.

Cluster Mapping Approach: The assumption behind this approach is that the reciprocal clusters of the compared organisms have similar interaction patterns that might indicate common functionalities as proposed by previous studies [79]. Therefore aligning interaction clusters having significant node similarity would also yield topologically strong mappings. The interaction clusters are identified either by Louvain Modularity or Label Propagation algorithms [138] as part of each heuristic. Focusing on mappings within clusters also contribute to improve the LCCS objective performances of the alignments.

Prioritization of node pairs with significant node similarity: The heuristics designed by this approach are simply based on focusing on node pairs with high BS, GOC and GOE but starting with edges or favoring edges wherever possible.

It is technically possible for a domain expert to generate a complete and ideal alignment of an organism pair by using a combination of the above mentioned heuristics with proper parameter values intuitively. Besides, a domain expert may easily improve an existing aligner by simply removing the ineffective mappings and adding effective mappings with these heuristics. Alternatively, the interactivity required for collaboration has been achieved with actors that exchange partial solutions in order to make the process more generic and expertise independent.

3.2.4 Alignment Process

The multi-objective optimization process of PERSONA is constructed upon a scheduling mechanism that calls the above mentioned meta-heuristic tools in a particular workflow. The user is provided some flexibility to build a custom workflow with a different order of steps and different execution periods of the meta-heuristic tools. The exact workflow designed and tested as part of this study is presented in Fig. 3.1:

Additionally, the workflow of the whole alignment process can be summarized as four main steps that are explained below in further detail:

1. Definition: Off-line jobs such as the cluster, connectivity and centrality computations are performed initially due to their computationally demanding nature. Subsequently, certain network characteristics are discovered and stored for the alignment strategy to be network independent. These characteristics include summary statistics and percentile based meta-data of centrality scores, biological sequence similarities and shared gene ontology terms between node pairs. Next, the number of aligners and the periodical schedule of the specific alignment jobs of aligners and exchange jobs are set for execution. In this scope, certain heuristics are scheduled as part of each alignment behavior while global and pairwise exchange policies are also defined and scheduled.

2. Initialization: Later on, some characteristic heuristics are used to initialize each aligner with proper similarity thresholds in order to propagate the alignment up to a mature state. Any available heuristic can be used for this purpose but the seed-and-extend heuristics are the most powerful candidates for the maturation intended in this step since they can identify topologically central mappings that also have the highest possible node similarity values initially. These heuristics can further be repeated by gradually lowering their node similarity thresholds. Alternatively, incomplete external alignments may be used to initialize each aligner. As a result, aligners achieve their initial partial solutions before starting the consecutive interactive phase of the application.

3. Collaboration: This step is the collaborative step where all aligners in the system act with a constant frequency of self improvement and a constant frequency of exchange that is defined and scheduled in the *Definition* step. Each aligner uses its own set of heuristics with randomized occurrence probabilities and parameter values as part of its behavior at every cycle. The parameter values are randomly assigned to each heuristic regarding their value interval during this process. Moreover, most significant mappings of leading aligners of each objective are periodically sent to other aligners and are also marked for future use during an alignment process. Finally, mappings with no contribution are removed as well as the ones that violate the one-to-one mapping restriction in each cycle. Meanwhile, a counter counts the number of cycles that an

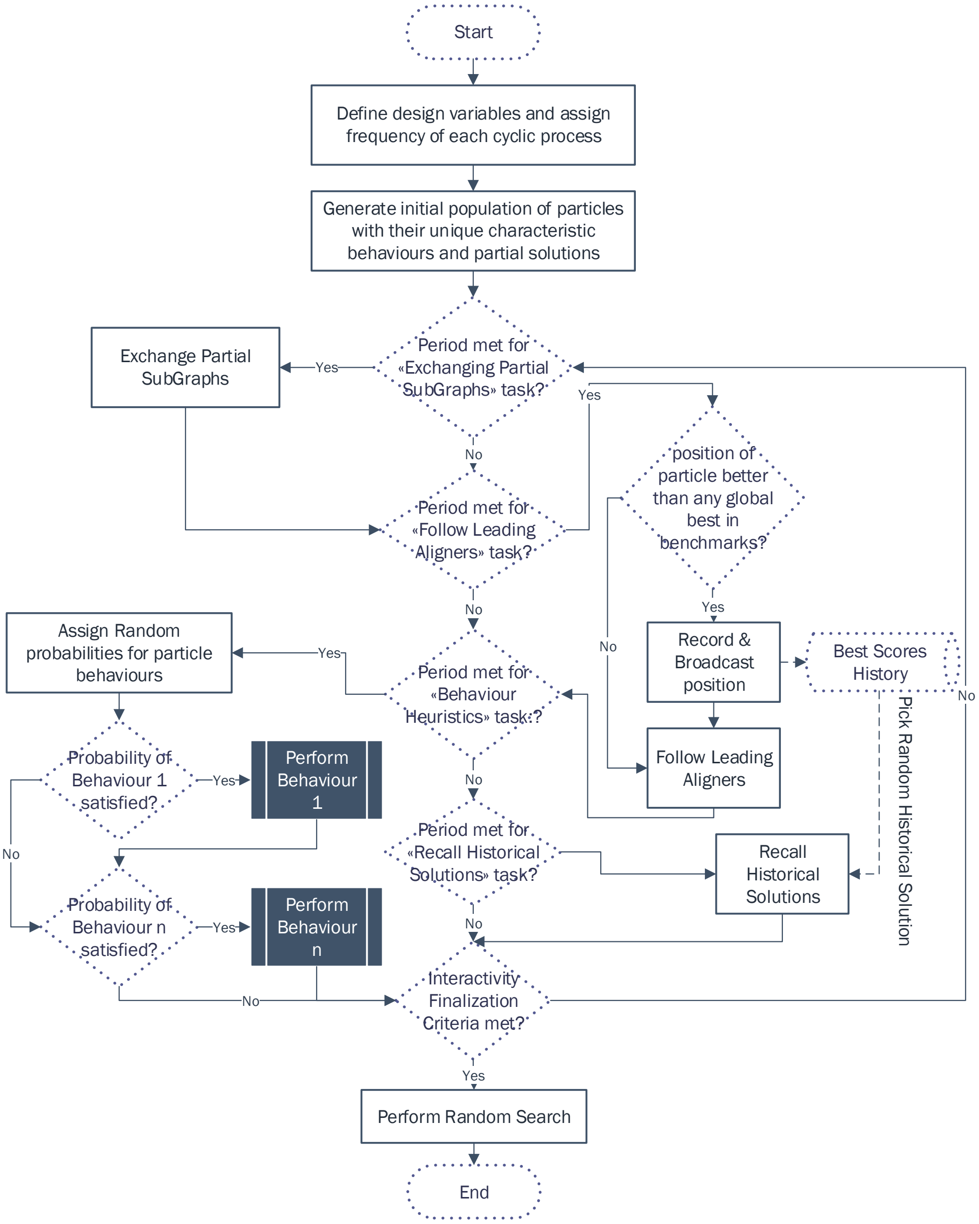
Start
Define design variables and assign frequency of each cyclic process
Generate initial population of particles with their unique characteristic behaviours and partial solutions
Period met for «Exchanging Partial SubGraphs» task?
Yes
Exchange Partial SubGraphs
No
Yes
Period met for «Follow Leading Aligners» task?
position of particle better than any global best in benchmarks?
Yes
No
No
Record & Broadcast position
Best Scores History
Follow Leading Aligners
Assign Random probabilities for particle behaviours
Yes
Period met for «Behaviour Heuristics» task?
No
Probability of Behaviour 1 satisfied?
Yes
Perform Behaviour 1
No
Period met for «Recall Historical Solutions» task?
Recall Historical Solutions
Pick Random Historical Solution
No
Probability of Behaviour n satisfied?
Yes
Perform Behaviour n
No
No
Interactivity Finalization Criteria met?
Yes
Perform Random Search
End

Figure 3.1: Workflow of a Particle from the Swarm. The main steps are *Definition*, *Initialization*, *Collaboration* and *Post-Processing*. The schedule of *Collaboration* and *Post-Processing* steps in the workflow depend on their initially designated execution period. In this sense, these steps are executed until a stopping criterion is satisfied. The stopping criterion may either be a repetitive failure of alignment progress or a specific number of scheduled cycles depending on the design of the respective experiment.

aligner has been unprogressive for and when the counter of an aligner reaches the threshold value, then the system randomly removes a limited number of minimally contributing mappings.

4. Post-Processing: This step is used for fine-tuning the alignments achieved through the PERSONA step. The standard procedure of this step is completing the alignment with random search for finding random mappings with positive effect. The standard procedure is applied by adding a limited number of random mappings and removing the ones with no positive effect in a loop until the alignment is complete. The pareto dominated alignments are filtered as a final task.

Generally, all optimization techniques require adaptations for preventing premature convergence and improving access to different regions of the search space. In PERSONA, every aligner specializes in particular set of heuristics and improves its alignment mostly with them. On the other hand, aligners can not reach certain regions of the search space if they are restricted with their own heuristics and that's why they need to collaborate to optimize their solutions. The collaboration prevents their convergence to a premature solution.

3.2.5 Architectural Design

PERSONA is implemented with the Typed Actor paradigm of Akka Concurrency Framework [135] utilizing tools such as Future Messages for concurrent completion of tasks, Typed Actors for message receiving patterns, Routers and Schedulers for determining the subsequent message recipients as well as a Neo4J graph database infrastructure for persistence of aligner states and exchanged subgraphs. Fig. 3.2 shows the flow of information among all essential entities of PERSONA methodology (see Appendix B.5 for more information about the whole design).

3.3 Results

Since PERSONA aims to achieve a balanced blend of results, we chose its competitors among aligners that possess a potential to achieve balance in multiple objectives. We performed the experiments with four distinctly extracted real world data sets for detailed interpretation. This chapter explains the process of these tasks in detail.

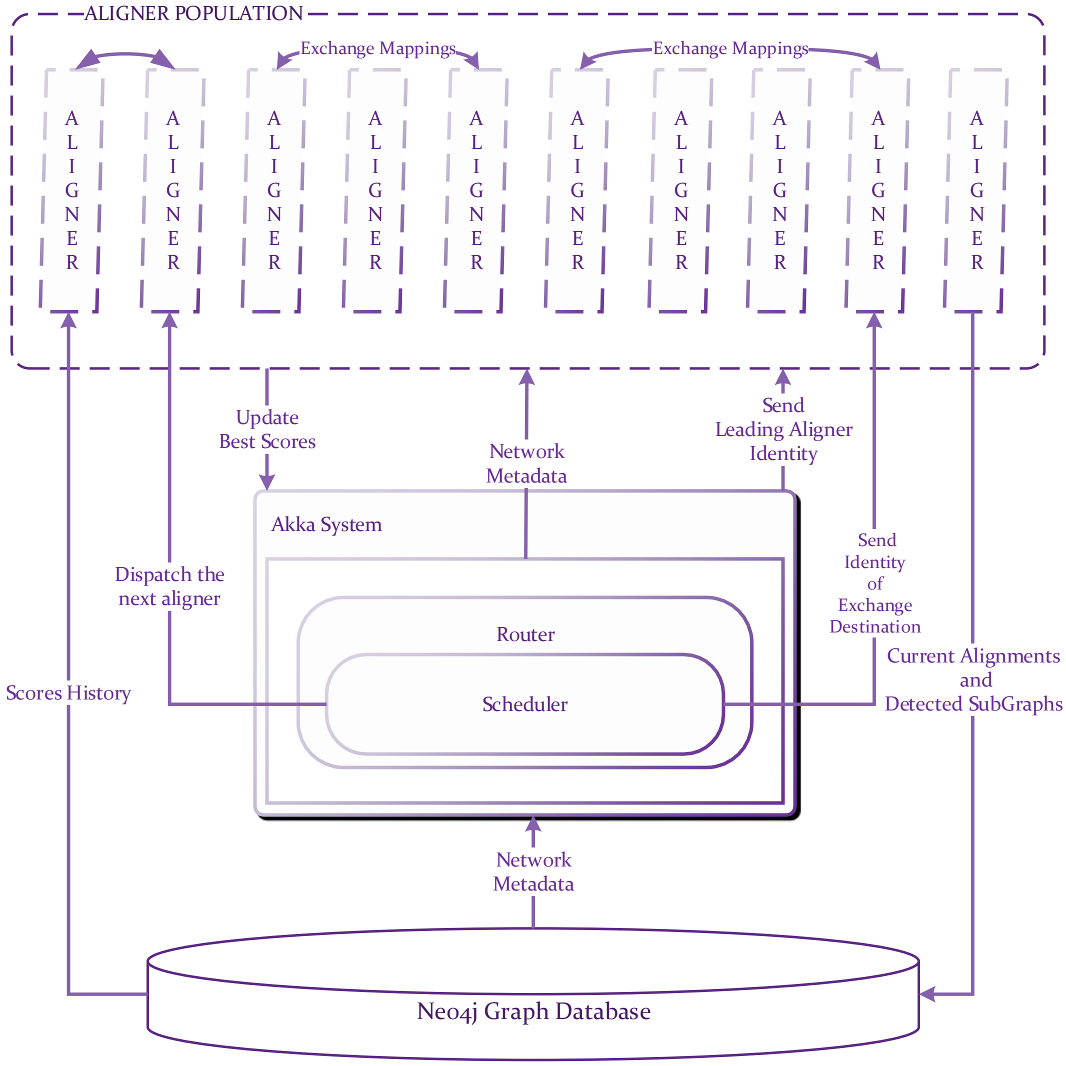

Figure 3.2: Cooperative Architecture of the Swarm. This architecture demonstrates the information exchange infrastructure as a basis of cooperation among the particles in the swarm. Each particle in the swarm is implemented as a Typed Actor with standard send and receive policies for group and individual behavior formalized in Algorithm 3.1 and 3.2. The architecture mainly relies on a Graph Based Persistence Model that stores individual aligner states and exchanged subgraphs along with a Concurrent Computation Framework that schedules execution order of sender and receiver aligners by means of a Scheduler and Router. The scheduler and router identifies and assigns upcoming senders and receivers based on their identity that is used in retrieving their latest state and significant subgraphs. Best Scores in each objective is updated after each mapping task and notified to the whole swarm.

3.3.1 Implementation Characteristics of PERSONA Population

We have implemented PERSONA with a special experimental setup that consists of a population of maximum ten customly designed aligners with various different and complementary characteristics. We designed each member of the population by using a combination of items from the previously mentioned Aligner Heuristics Suite as part of its behavior. Thus, each member may be regarded as an individual alignment algorithm implemented for this study. The distinguishing feature for each aligner was its primary heuristic. Some aligners relied mainly on a unique selection of a centrality detection algorithm while the remaining ones either had a main behavior of node pair prioritization based on node similarity or a cluster mapping approach based on a unique clustering technique from our Aligner Heuristics Suite. We intended to create diversity in the population by dedicating a single clustering or centrality detection heuristic to a particular aligner of the population. The respective aligner names in the population based on their primary heuristic is listed below:

- Page Rank Seeding Aligner,
- Betweenness Centrality Seeding Aligner,
- Closeness Centrality Seeding Aligner,
- Harmonic Centrality Seeding Aligner,
- Connectivity Degrees Seeding Aligner,
- Cluster Mapping Aligner with Label Propagation,
- Cluster Mapping Aligner with Louvain Modularity,
- Sequence Similarity Prioritizing Aligner,
- Sequence Similarity Seeding Aligner,
- Hybrid Aligner of Centrality and Sequence Similarity

All the above mentioned aligners in our experiments performed their primary heuristics with high probabilities as part of their main behavior in each interaction cycle. Besides, all aligners performed secondary or complementary heuristics from the heuristics suite in lower probabilities than their primary heuristics. The main reason for executing heuristics with non-standard probabilities was to increase randomness and flexibility to search for optimized solutions. Additionally, we also randomized the output of each heuristic by random value assignment for the parameters that it requires. The secondary heuristic that we used most frequently was "Heuristic for Forming Edge Pair Mappings from Existing Node Pair Mappings" since it enables to propagate with edges of particular node similarity from seed mappings. Besides, we also used "Heuristic for Removing Inductive Mappings" and "Heuristic for Removing Low Scoring Mappings" frequently for especially opening up search space productively for every aligner. Since each aligner was composed of a combination of heuristics, its probability of achieving a balance among the favored objectives is improved remarkably. Finally, we employed the Post-Processing step of PERSONA with an individual alignment approach that starts with Biological Similarity Seeding. The propagation after seeding was maintained by forming edge pairs with an incremental constraint relaxation strategy in each cycle.

3.3.2 Competitors & Simulation Environment

NETAL [116], SPINAL [89], PISwap [91] and HubAlign [90] may be defined as a particular class of competitors for PERSONA since they are all deterministic aligners that evaluate topological and biological inputs. NETAL [116] uses a greedy method by evaluating an alignment scoring matrix. HubAlign [90] is based on preliminarily evaluating and scoring the topological and biological importance of proteins to identify hub nodes to align and then assigning alignment scores to protein pairs by considering sequence similarity and the importance score. PISwap [91] is a method that iteratively refines the initial alignments of custom heuristics with topological information while compromising sequence information achieved by the well-known Hungarian algorithm [142]. On the other hand, SPINAL [89] performs a fine-grained conflict resolution and following a coarse-grained construction of estimate scores. We executed these aligners iteratively in their most powerful range of each application parameter for producing significant sets of results.

SANA [99], PROPER [143], MAGNA++ [144] and PINALOG [145] form another class of competitors for PERSONA since they employ non-deterministic optimization by evaluating topological and biological inputs in order to generate a single alignment. SANA [99] follows a simulated annealing based optimization approach. PROPER [143] generates a seed of high sequence similarity protein pairs based on percolation matching and then progresses only with structural mapping. MAGNA++ [144] is an improvement version over the original MAGNA [100] method that combines existing ‘parent’ alignments into superior ‘children’ alignments and then evolves this process over multiple generations. It enables maximization of a node conservation measure simultaneously with the chosen edge conservation measure and provides automatical utilization of all available resources by means of parallelization. Finally, PINALOG [145] method combines information from protein sequence, function and network topology information and it consists of 3 fundamental steps starting with preliminary detection of communities with CFinder [146], followed by community mapping with respect to similarities and finalized with extension mapping of proteins in the neighbourhood of the core protein pairs.

We finally evaluated other unique alignment algorithms such as GEDEVO [147] and OptNetAlign [92] for the purpose of comparing a diverse set of approaches. GEDEVO [147] is a graph comparison tool that generate a single alignment based on the so-called Graph Edit Distance (GED) model where one graph is to be transferred into another one with a minimal number of edge insertions and deletions. The optimization methodology of this tool relies mainly on topological information but it can also be extended to utilize biological similarity. Conversely, OptNetAlign [92] performs multi-objective optimization with respect to functional, biological and topological inputs based on a genetic algorithm that employs Uniform Partially Matched Crossover and hill climbing on a population of pareto optimal alignment results. PERSONA generates a population of alignments similar to OptNetAlign [92] but it rather performs the pareto optimality check as a final step. Nevertheless, it manages to generate alignments that are stronger in various objectives due to the different nature of aligner behaviors in its population.

We principally chose most of these competitors due to their two-sided nature that compromise between node similarity and topological similarity measures. Another reason for choosing them was their applicability due to their existing documentation and source codes. We intended to compare our method with aligners such as IBNAL [148] and SSAlign [149] that assume annotational, biological and topological inputs simultaneously along with OptNetAlign [92] and PINALOG [145] due to their capability of mapping experimentally verified annotations but their source code was unavailable. We executed all the competitors of PERSONA with the recommended modes and parameter values as well as other performance related instructions of their authors. More information is provided in Appendix B.6 about the usage parameters of the competitors. The source code, application and execution instructions of PERSONA is also available on the github repository https://github.com/giraygi/ppi-alignment.

3.3.3 Datasets

We evaluated competitor algorithms along with PERSONA by comparing C. Elegans (CE), S. Cerevisiae (SC), M. Musculus (MM), H. Sapiens (HS) with D. Melanogaster (DM) based on their protein, network, pairwise biological sequence similarity and annotation data. We initially used the earliest benchmark data set Isobase [117] since it was tested by several algorithms in the literature. Additionally, some distinctive data sets extracted by recent aligners such as PROPER [143] and SANA [99] were also used in evaluation. The network data of PROPER [143] was retrieved from Intact 2016 [30] and integrated with sequence similarities from UniProt [150] as well as experimentally verified terms denoted by “EXP”, “IDA”, “IMP”, “IGI”, “IEP” and “IPI” codes in Gene Ontology Annotation (UniProt-GOA) [151]. The data set of SANA was retrieved from BioGRID 2017 [25] database with a complete list of protein-protein interactions, experimentally verified GO terms and sequence similarities. Finally, we evaluated the competing alignment algorithms with the 12.07.2021 dated release of Mentha [152] data set that integrates experimentally curated PPI data of several molecular interaction databases by providing automated access with the Proteomics Standard Initiative Common Query Interface (PSICQUIC) [153, 154] in compliance with International Molecular Exchange (IMEx) [155] policies. We further integrated the sequence similarities computed by BLASTP from FASTA texts in UniProt and annotations from UniProt-GOA into Mentha networks in a similar fashion with the Intact data set.

As a notable remark, PROPER [143] has used Intact 2016 data set with a considerably high threshold for BS that ignores lower similarity values and complicating its competitors in achieving high BS scores. For this reason, the respective data set will be referred as "Trimmed Intact 2016" in the following text. On the other hand, we generated the respective Annotation Corpus of each data set and used the FastSemSim python library (https://pypi.python.org/pypi/fastsemsim) for computing Resnik Similarity of GO terms of protein pairs with the same maximum best match mixing strategy chosen by SANA [99]. Additionally, we removed the repeating gene ontology terms in BioGRID 2017 data set to extract another gene ontology instance for using the data set conveniently with the more general objective GOC. We tested PERSONA with the exact form of these data sets used by the aforementioned aligners for keeping the comparison conditions uninfluenced. Table 3.1 represents the number of nodes, edges and sequence similarity links along with the average number of common annotations among possible node pairs for each organism pair evaluated in Section 3.3.4

Table 3.1: Network Sizes of Data Sets. The first compared organism in the first column is denoted as "Left", whereas the second compared organism is denoted as "Right" in the other columns. The "Similarity Links" column denotes the number of available BS links throughout all possible node pairs between the first and second compared organisms. "Average Common Annotations" column denotes the average number of common annotations throughout all possible node pairs between the first and second compared organisms.

	Nodes Left	Nodes Right	Edges Left	Edges Right	Similarity Links	Average Common Annotations
DM-CE Intact	8532	4950	26289	11550	5669	9.07
DM-SC BioGRID	7937	5831	34753	77149	132007	1.64
HS-DM Isobase	9633	7518	36386	25830	97172	1.26
DM-MM Mentha	10827	9674	45706	31577	178415	5.93

Most of the data sets used by the competitors were not compatible due to the different input varieties they required. For this reason, we carried out complex transformation procedures for being able to test all aligners with data sets mentioned in this study along with additional ones for future experiments. We stored resulting data sets in a github repository https://github.com/giraygi/ppi-alignment-data with some of the transformation procedures in https://github.com/giraygi/ppi-alignment-converters/. We tested the consequent data sets with all previously listed aligners.

3.3.4 Results

The alignments generated with the prominent multi-objective aligners are mostly not able to pareto dominate each other since they are not completely superior than one another in all desired objectives. Therefore, it becomes necessary to compare aligners based on an aggregate score that is a projection of all the desired objectives. For this reason, it makes sense to group resembling objectives and to make an assumption about the significance of each objective and each group of objectives. As part of a meaningful classification of objectives, the edge similarity based topological measures can be grouped with each other while the functional measures based on shared annotations can be proposed as another group. We can then include the undoubtedly meaningful primitive measure of edge similarity as well as the more developed S^3 that is able to penalize the dense-to-sparse mappings into the edge similarity based topological measures group. On the other hand, it is also meaningful to include GOC and GOE into the functional measures based on shared annotations group since they interpret the functional coverage by directly using the same annotational data. Thus, it becomes easier to assign weights of importance to groups rather than individual objectives.

Consequently, we can consider five groups of objectives being Edge Similarity (EC and S^3), Global Topological Coverage (LCCS), Annotational Coverage (GOC and GOE), BS and Semantic Similarity (Resnik). By assigning a coefficient of 1 to each group and simply dividing the coefficients within the groups to group sizes, EC, S^3, GOC and GOE get coefficients of 0.5 whereas LCCS, Resnik and BS get coefficients of 1. Therefore, it is possible to conclude a simplistic multi-criteria decision making model for either a Weighted Sum Model (WSM) or a Weighted Product Model (WPM) [156] as an abstract aggregated objective function in order to be able to choose between different pareto optimal solutions without giving priority to any objective group or individual objective other than their designated coefficient. The WSM approach would also require the results to be comparable so that they can be used as part of the same equation. We normalized all objectives defined in Section 3.2.1 with Total Sum Scaling and Z-Score Normalization techniques in order to obtain comparable results for this purpose. We employed the formulas (3.4) and (3.5) for interpreting the alignment results of all competitor algorithms.

$$WSM = \frac{EC}{2} + \frac{S^3}{2} + LCCS + \frac{GOC}{2} + \frac{GOE}{2} + BS + Resnik \tag{3.4}$$

$$WPM = \sqrt{EC \times S^3 \times GOC \times GOE} \times LCCS \times BS \times Resnik \tag{3.5}$$

Subsequently, the formula (3.6) represents the relative performance of Algorithm K to Algorithm L since all the objectives are benefit criteria and the higher values of them represent better performance accordingly. The normalized performance of the Algorithm K on the j_{th} objective is represented as a_{K_j} in the formula. Besides, W is the vector of the same objective coefficients used in the formulas (3.4) and (3.5).

$$P(A_K/A_L) = \prod_{j=1}^{n}(a_{K_j}/a_{L_j})^{W_j}, for K, L, = 1, 2, 3, ..m \tag{3.6}$$

Average performances of algorithms in the full set of objectives are represented in Fig. 3.3, 3.4, 3.5 and 3.6. Z-Score Normalized Results of each column sum up to 0 and Total Sum Scaled Results of each column sum up to 1 for each objective in these figures. For aggregating average objective performances into a single alignment score, both WSM and

Table 3.2: Aggregate Scores on DM-CE Trimmed Intact 2016. The score columns are sorted in a decreasing order for WSM and in an increasing order for WPM from left to right.

Competitor	$WSM_{TotalSum}$	$WSM_{Z-Score}$	$WPM_{TotalSum}$
PERSONA	0.57	3.66	1.00
PROPER	0.55	3.00	1.26
OptNetAlign	0.53	1.64	2.30
PINALOG	0.52	2.67	1.90
HubAlign	0.44	0.97	4.35
SPINAL1	0.44	0.67	4.98
PISwap	0.42	-0.58	70.18
SANA	0.38	-0.97	17.88
SPINAL2	0.37	-0.40	10.19
GEDEVO	0.28	-2.61	11,122.49
MAGNA++	0.26	-4.32	56.67
NETAL	0.24	-3.72	71,859.42

WPM were applied for Total Sum Scaled data whereas only WSM was applied for Z-Score Normalized data since it includes negative values. Higher aligner scores represent higher performance in both $WSM_{Z-Score}$ and $WSM_{TotalSum}$ columns of Tables 3.2, 3.3, 3.4 and 3.5. On the contrary, higher scores represent the superiority of PERSONA over the compared aligner in the column $WPM_{TotalSum}(PERSONA)/(Other)$ computed with the relative performance formulation above.

We carried out all experiments with an Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz processor and 16 GB RAM on a 64 bit Linux platform. One exception was PINALOG [145] since we alternatively carried out its execution with the supercomputer at Freie Universität Berlin (FU Berlin) called CURTA [157] after it endangered the safety of the personal computer due to an execution period of almost one day and severe heating problems without termination. The CPU time required by each algorithm on DM-CE Trimmed Intact 2016 Data Set was shown in Table 3.6. We terminated time consuming algorithms after 3 hours of application except PINALOG [145] that did not present any intermediary result after 3 hours of execution. We distributed the time consumption proportionally to the number of produced alignments where applicable.

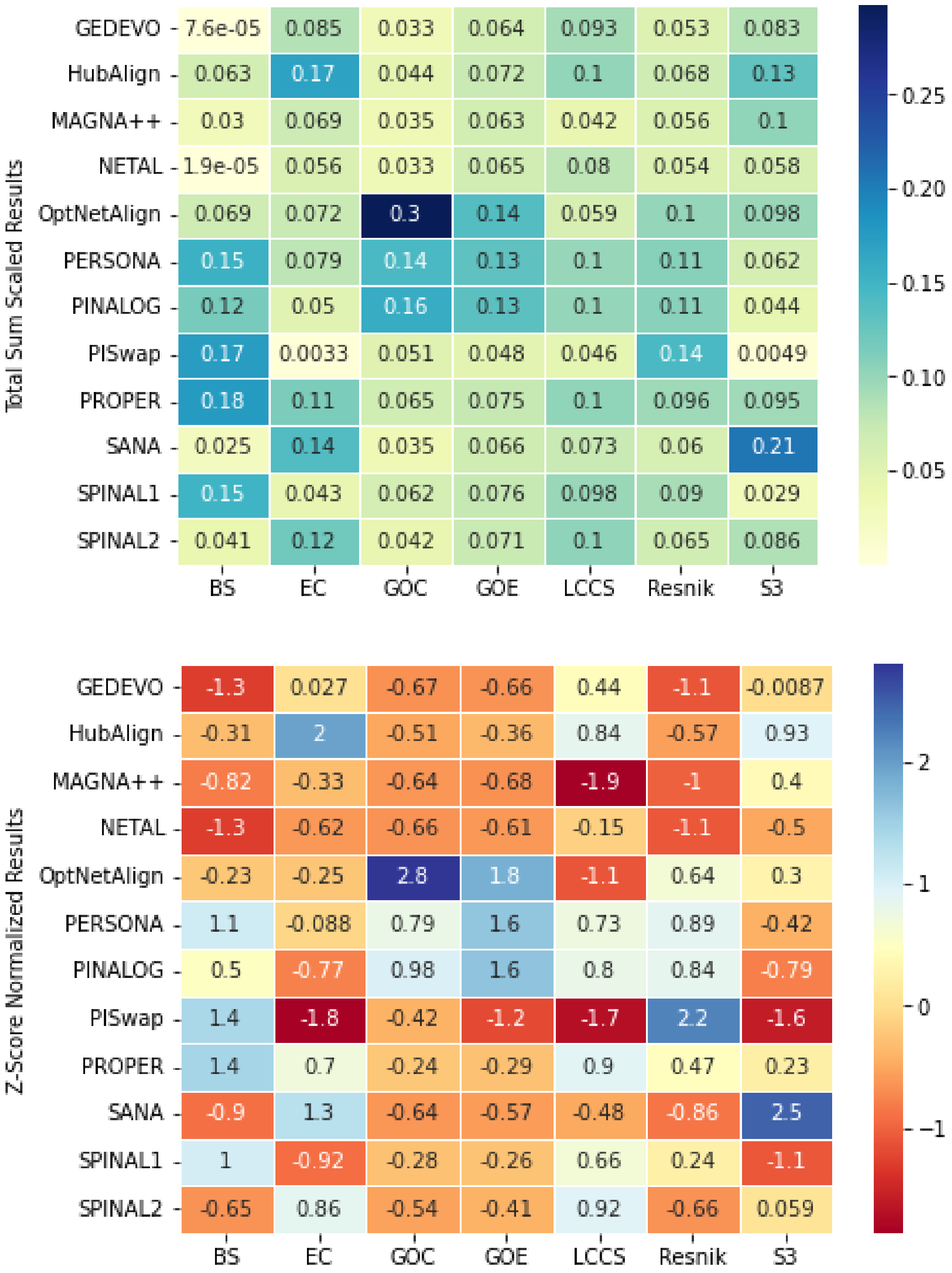

Figure 3.3: Objective Scores on DM-CE Intact 2016 Data Set. Each row includes an array of the object specific scores of a particular alignment algorithm, whereas each column corresponds to a particular objective. The heatmap scales of Total Sum Scaled Results (on top) and Z-Score Normalized Results (on bottom) are displayed on the right hand side.

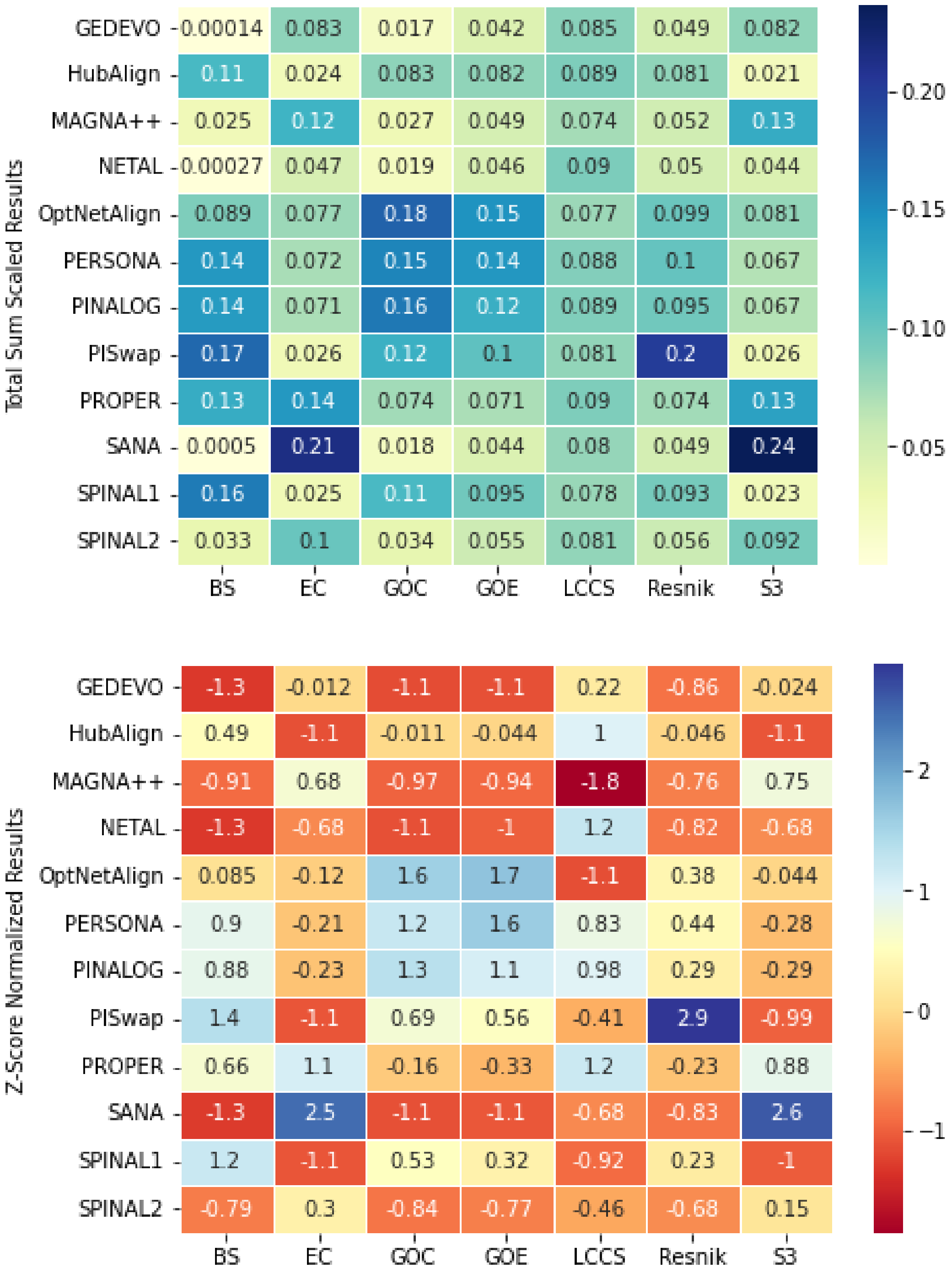

Figure 3.4: Objective Scores on DM-SC BioGRID 2017 Data Set. Each row includes an array of the object specific scores of a particular alignment algorithm, whereas each column corresponds to a particular objective. The heatmap scales of Total Sum Scaled Results (on top) and Z-Score Normalized Results (on bottom) are displayed on the right hand side.

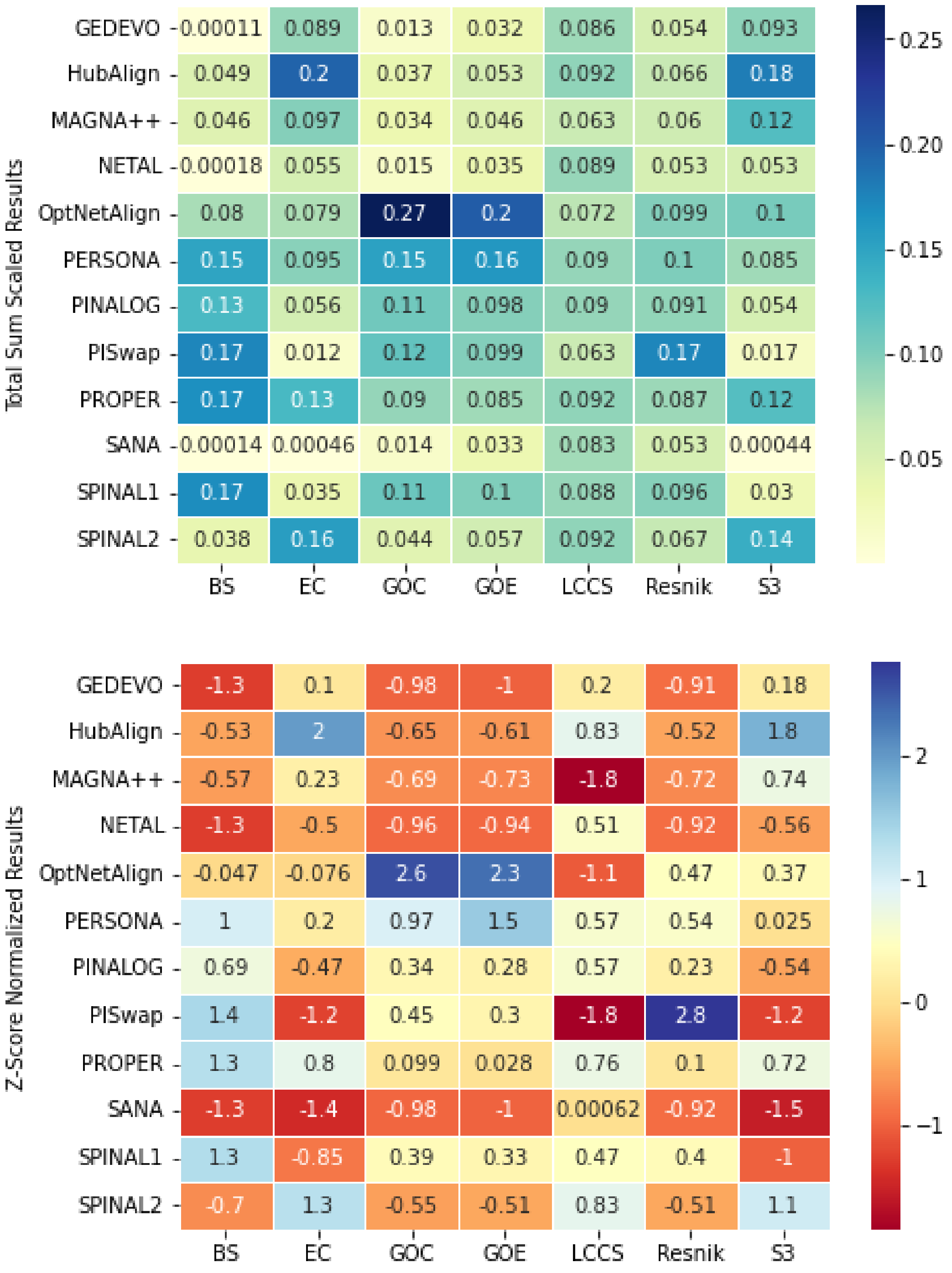

Figure 3.5: Objective Scores on HS-DM Isobase Data Set. Each row includes an array of the object specific scores of a particular alignment algorithm, whereas each column corresponds to a particular objective. The heatmap scales of Total Sum Scaled Results (on top) and Z-Score Normalized Results (on bottom) are displayed on the right hand side.

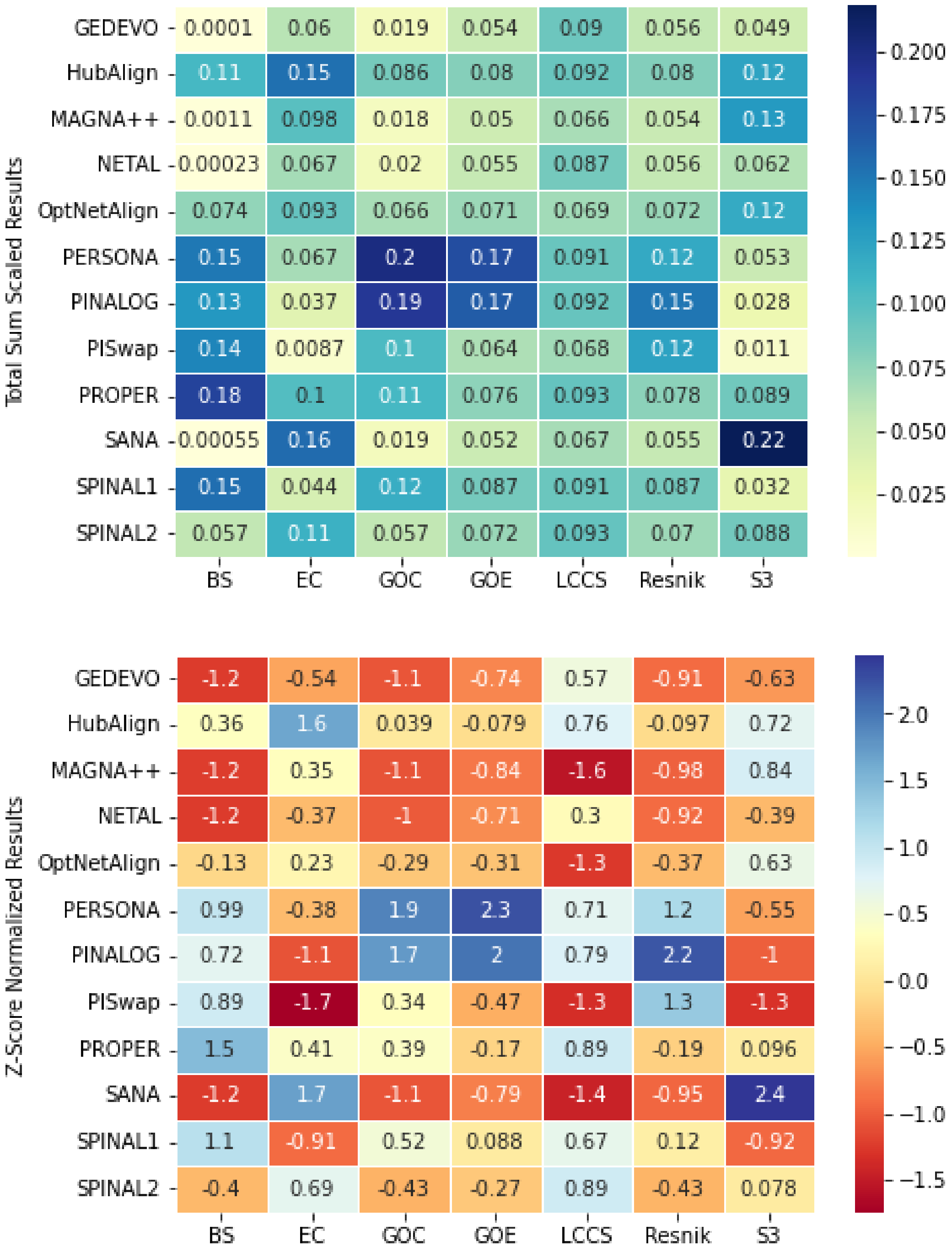

Figure 3.6: Objective Scores on DM-MM Mentha 2021 Data Set. Each row includes an array of the object specific scores of a particular alignment algorithm, whereas each column corresponds to a particular objective. The heatmap scales of Total Sum Scaled Results (on top) and Z-Score Normalized Results (on bottom) are displayed on the right hand side.

Table 3.3: Aggregate Scores on DM-SC BioGRID 2017. The score columns are sorted in a decreasing order for WSM and in an increasing order for WPM from left to right.

Competitor	$WSM_{TotalSum}$	$WSM_{Z-Score}$	$WPM_{TotalSum}$
PISwap	0.59	3.43	1.60
PERSONA	0.55	3.34	1.00
PINALOG	0.54	3.11	1.13
PROPER	0.50	2.34	1.55
OptNetAlign	0.50	0.89	1.51
SPINAL1	0.46	-0.13	4.39
HubAlign	0.39	0.33	8.60
SANA	0.39	-1.36	1,044.58
MAGNA++	0.31	-3.75	30.57
SPINAL2	0.31	-2.51	21.11
GEDEVO	0.24	-3.07	10,733.68
NETAL	0.22	-2.62	8,165.79

Table 3.4: Aggregate Scores on HS-DM Isobase. The score columns are sorted in a decreasing order for WSM and in an increasing order for WPM from left to right.

Competitor	$WSM_{TotalSum}$	$WSM_{Z-Score}$	$WPM_{TotalSum}$
PERSONA	0.59	3.48	1.00
OptNetAlign	0.58	1.95	1.59
PROPER	0.56	2.96	1.30
PISwap	0.53	1.44	6.46
SPINAL1	0.49	1.62	3.88
PINALOG	0.47	1.30	3.19
HubAlign	0.44	1.02	7.62
SPINAL2	0.40	0.26	10.84
MAGNA++	0.32	-3.36	25.09
GEDEVO	0.25	-2.84	19,445.10
NETAL	0.22	-3.16	17,764.47
SANA	0.16	-4.67	3,268,998.10

3.4 Discussion of Results and Future Work

Our approach for comparing multi-objective performance of PERSONA with the competitor GNA algorithms in this study is based on an abstract aggregated objective function as it is

Table 3.5: Aggregate Scores on DM-MM Mentha 2021. The score columns are sorted in a decreasing order for WSM and in an increasing order for WPM from left to right.

Competitor	$WSM_{TotalSum}$	$WSM_{Z-Score}$	$WPM_{TotalSum}$
PERSONA	0.61	4.51	1.00
PINALOG	0.58	4.56	1.76
PROPER	0.54	2.52	1.61
HubAlign	0.50	2.18	2.00
SPINAL1	0.47	1.24	3.92
PISwap	0.43	-0.69	18.90
OptNetAlign	0.39	-1.64	6.81
SPINAL2	0.39	0.10	7.59
SANA	0.34	-2.48	1,557.46
MAGNA++	0.27	-4.13	1,384.43
NETAL	0.24	-3.11	7,410.44
GEDEVO	0.24	-3.07	20,590.75

Table 3.6: Elapsed Time of Competitor Algorithms on DM-CE Trimmed Intact 2016. Elapsed time is in CPU Seconds and it is sorted in an increasing order.

Competitor	Elapsed Time
PROPER	12.1
PISWAP	13.40
NETAL	57.84
PERSONA	240
HubAlign	247.06
SANA	339.11
Spinal	579.14
OptNetAlign	2457.42
MAGNA++	21625.09
GEDEVO	40991.52
PINALOG	89082.05

explained in Section 3.3.4. For this purpose, we have assigned a coefficient of significance to each alignment objective and we have used aggregate WSM and WPM scores as the primary source of comparison since it is very hard to interpret the relative performance of each competitor algorithm in each individual objective. The aggregate scores in Tables 3.2, 3.3, 3.4 and 3.5 show that PERSONA is able to achieve a consistently high performance whereas other aligners occasionally achieved better results in certain data sets. On top of the aggregate scores, individual objective scores in Figures 3.3, 3.4, 3.5 and 3.6 also gave some remarkable insights about the alignment methodologies. Furthermore, dual interpretations of objectives also revealed the trade-offs or compromises inherently carried within each competitor algorithm. In this context, PERSONA achieved a considerably high performance on GOC, GOE, Resnik and BS. Additionally, it achieved a relatively good performance of LCCS. However, the collaboration strategy of PERSONA increased the probability of leaving unaligned edges due to trying to keep the balance for other objectives. Consequently, it achieved relatively poor results in S^3 as the unaligned edges are penalized.

One key finding was about the effects of some data sets over alignment performance. Accordingly, the major drawback of Trimmed Intact 2016 Data Set was that it ignored pairwise BS scores less than 150 bits by trimming an essential part of the respective data. Consequently, the performance of other competitor algorithms were effected drastically due to their higher sensitivity to the missing part of the data. Since PROPER focuses only on top ranking sequence similarities by default, it gains an advantage against competitor algorithms that are able to consider low similarity node pairs in their alignment approach. Despite this fact, PERSONA managed to generate competent alignments with PROPER by either its regular approach or by starting the alignment in the post-processing phase so that it could map larger chunk of node pairs in each cycle with higher finalMappingFactor inputs. The latter approach achieved higher BS scores than PROPER. Besides, PROPER has scored low GOC and GOE values despite its relatively high performance in EC and comparable performance in BS.

There were other key findings about the performances of competitors. For instance, PISwap [91] consistently achieved very high Resnik Similarity Scores in all data sets despite its moderate scores in GOC and GOE. It also achieved partial best aggregate scores on DM-SC BioGRID 2017 data set. On the other hand, PINALOG [145] performed better with annotational inputs and consequently its results without them were discarded. It also performed comparably better in functional and biological objectives despite its lower performance in topological objectives. Furthermore, HubAlign [90] did not perform well in the pairwise comparison of DM-SC on primarily the topological similarity objectives in either of the data sets despite its relatively better performance on the other pairwise comparisons. This shows that the algorithm is not capable of adapting certain topological network characteristics of the respective organism pair.

It is also worth noting that the first mode of SPINAL [89] called SPINAL1 generated better aggregate results than its second mode SPINAL2 despite its relatively lower performance on topological similarity objectives. This result was mainly due to the fact that topological improvement of SPINAL2 has resulted in a more significant compromise in the node similarity objectives and SPINAL1 has managed to generate more balanced alignments. Another interesting finding was that the second mode of OptNetAlign [92] called OptNetAlign2 was able to perform well in GOC and GOE whereas it performed very poorly in BS in all data sets. On the contrary, OptNetAlign1 achieved an overall balance in all the objectives but it did not demonstrate any distinguishing difference in any of them.

Some competitor algorithms were able to produce meaningful results only in topological objectives. For instance, the performance of SANA [99] in the widely accepted GOC, GOE and BS objectives were far from satisfactory due to its different interpretations of these metrics. As another example, NETAL [116] did not offer any input methodology for node similarity data although stated otherwise in its documentation. For this reason, it did not generate any positive results in BS and it could generate barely positive results in GOC and GOE. Similarly, GEDEVO [147] was not able to produce positive results in the functional and biological objectives although it was provided an optional biological or annotational similarity input. Finally, MAGNA++ [144] was able to achieve moderately positive results on BS and yet it was not able to achieve positive results in the functional objectives of GOC, GOE and Resnik.

The aim of this study is developing a generic and balanced approach that would perform well on every kind of network generated through various data sources. On the meta-heuristic level, the application harnesses the differences between concurrently executing aligners by comparing their current results in multiple objectives and sending the superior subgraphs to low scoring aligners so that the population achieves a balanced performance. It is also possible that some exchanged mappings may not contribute to the receiving aligner since some of its node pairs are already occupied with other mappings. In such cases, only the possible mappings of the superior subgraph are used in alignment but then it becomes possible that the remaining mappings have mostly lost their superiority characteristics. Fig. 3.7 illustrates such a topological similarity loss upon removal of the central mapping between proteins *A* and *W* from a subgraph. PERSONA employs a removing policy in each execution cycle in order to filter such remaining mappings that has no contribution. However, it may also be possible to predict the compromises or trade-offs beforehand and prevent sending useless subgraphs with a more precise conflict resolution policy. Yet, such a policy may conversely create some overhead due to alignment score computation of intersecting subgraphs beforehand.

The current version of PERSONA does not filter pareto dominated solutions until the final step. Filtering them during the Collaboration step may also make sense in terms of dealing with an elite and useful set of alignments. On the contrary, pareto dominated solutions may still include significant subgraphs to be evaluated by a superior alignment with collaboration as a remark of partial performance. The significant partial alignments generated throughout the alignment process of PERSONA may also be used as building blocks for building a superior global alignment or shed light to a local alignment solution. In this scope, strong partial alignments achieved by PERSONA or other aligners can easily be merged with each other by PERSONA infrastructure. One key feature introduced by PERSONA for this task is the accurate measurement of partial performances in each objective.

PERSONA would also demonstrate a remarkable performance in a hypothetical objective that unifies biological, topological and annotational similarity objectives since it enables querying individually and topologically significant node pairs simultaneously as part of a single heuristic. Individually significant node pairs are individual mappings that possess significant biological and annotational similarity with each other regardless of their interactions with other nodes. On the other hand, topological significance refers to multiple node pairs that form a remarkable number of aligned edges when they are mapped with each other. Individually and topologically significant node mappings identified by this feature may compose the core of an alignment since they would require minimal or no compromise in any objective. Subgraphs composed of such mappings may also be stored as a regional benchmark in the collective memory to be evaluated by each population member for merging with its individual alignment. Apart from that, these significant subgraphs may further be merged with themselves in different combinations in order to build superior alignments with minimal compromise in most objectives.

Another future improvement in the data model could be modeling the Dynamic Network Alignment problem [158] mentioned in DynaWAVE [159] on top of the current data model of PERSONA. The current data model may easily be adapted to the Dynamic Network Alignment problem that addresses the temporal component of protein-protein interactions since it only requires node properties for activation and deactivation times of the interactions. Last but not the least, a many-to-many local or global alignment solution may also be incorporated into this application by removing the one-to-one mapping restriction from its built-in search tools and heuristics libraries. Such a solution might either be used as part of a

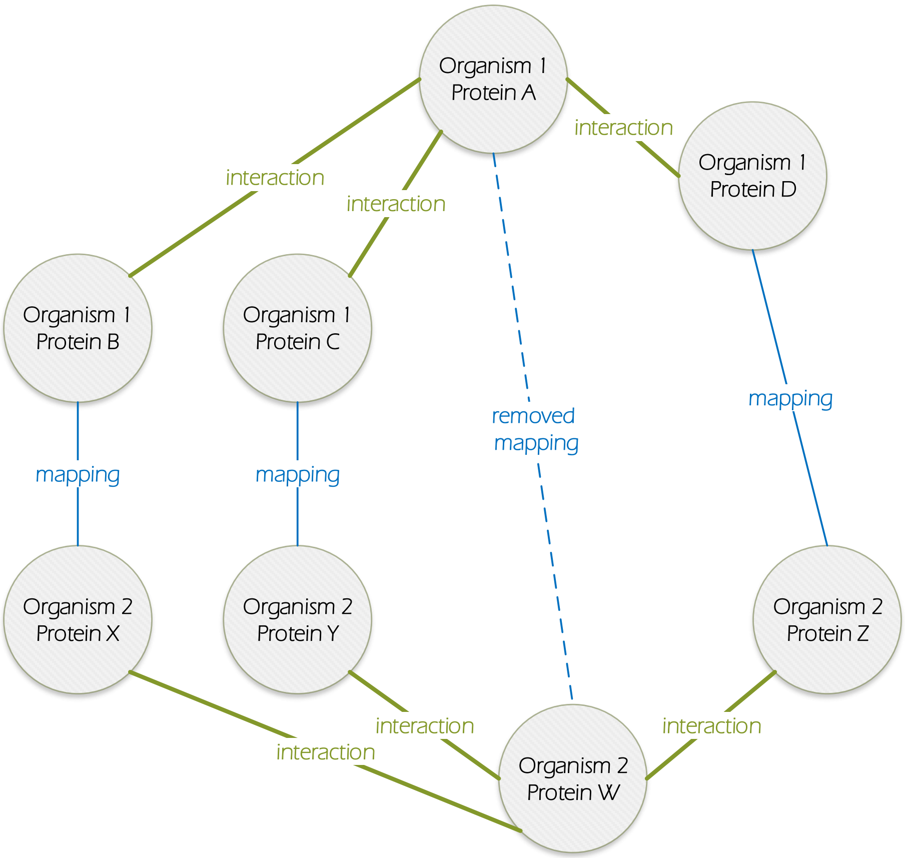

Figure 3.7: Removed Mapping Effecting Topological Similarity. The removed mapping demonstrated with a dashed line has a critical effect upon topological similarity since it forms 3 aligned edges contributing to topological similarity. However, the remaining mappings cannot form any aligned edges without it.

complete alignment procedure or an aggregation procedure of external one-to-one alignments in the same fashion with Ulign [160].

PERSONA was developed with the recently deprecated Typed Actors technology of Akka [135]. Nevertheless, implementing the aligners with the active object pattern has the advantage of benefiting from both object-oriented and Typed Actor paradigms. Neo4j Graph Database technology [137] was mainly employed in order to be able to utilize the flexible Cypher Graph Querying Language [136]. Besides, Neo4j embedded-mode that would theoretically provide slightly better performance with Java platform was compromised with the driver-mode to yield visualization on the browser. As a result of this decision, it is always possible to intervene the alignment process by accessing it from other platforms and applications with the driver mode so that the alignment process can be performed on a cluster environment or can be monitored to intervene on special cases. However, there are various techniques that would yield a disk independent solution so that it is possible to query the graph database from memory instead of the disk. Disk independency may be achieved by employing and configuring Neo4j Graph Data Science Library [161] with in-memory mode or installing the database to the memory rather than the disk manually or via the ImpermanentDatabase feature primarily intended for testing Neo4j applications. Such a configuration and implementation would drastically improve performance and enable employing bolder meta-heuristic optimization rules within the application.

One final future improvement for PERSONA can be publishing all its heuristics and subgraph exchange methods through a Representational State Transfer (REST) [162, 163] interface for facilitating a decentralized architecture conveniently. Consequently, population members residing in different platforms will be able to perform their computations locally and store their actual alignment status in a remote platform that hosts the central graph database. The respective distributed architecture will be designed in a fault tolerant fashion that allows occasional disconnections and ensures the stability of the graph database. It will be possible to improve the level of proactivity and collaboration in such an architecture since it will be handled by the host platforms of the population members. It will further be possible to orchestrate multiple populations for pairwise comparisons or multi organism comparisons for Multiple Network Alignment by introducing multiple population central nodes that store the graph database. In this context, it will be necessary to design and implement population members in a way that processes JavaScript Object Notation (JSON) format messages since it is the most common message exchange format of REST interfaces for interoperability [164].

4 Summary and Outlook

Global Network Alignment is an informative network comparison method that aims to predict orthologous proteins with equivalent cellular functions in the PPI Networks of the compared organisms. GNA results are primarily used to validate PPI Networks conserved across multiple organisms and detect evolutionarily conserved pathways or protein complexes [165–167] so that each protein of a particular organism is mapped to an equivalent protein with known behaviors such as binding the same ligand and similar DNA sites in the compared organisms [168]. GNA is performed with a one-to-one mapping approach when it is assumed that each protein represents a single cellular function and the functions of a smaller PPI Network are a subset of a larger PPI Network during their comparison. As it is mentioned in Section 1.2, the performance of one-to-one GNA is measured with respect to several biological, topological and annotational similarity objectives since there is no single model that explains orthology completely. As a result, it becomes complicated to search for the most accurate mappings that form an alignment due to the contradicting nature of these objectives.

The search operation to find acceptable solutions to Network Alignment problems can mainly be performed by heuristics or meta-heuristics based approaches. Heuristics based approaches usually intend to find an ideal alignment in a deterministic fashion in contrast to the non-deterministic meta-heuristics based approaches that intend to find a remarkable approximation. Meta-heuristics based approaches play a key role for achieving flexibility in GNA search results since a purely deterministic procedure relies on questionably strong assumptions that may not be universal. Stützle [169] states that meta-heuristics algorithms rely on biased and intelligent randomness that originate from probabilistic decisions made during search in contrast to pure random search. For this reason, it becomes pertinent to achieve the biased and intelligent randomness in question by randomizing the behavior of a given set of heuristics based on input parameter values. In this regard, a set of heuristics procedures with diverse solution sets may further be used for biased and intelligent randomization of a meta-heuristics search procedure in order to achieve flexibility without compromising computational performance. Furthermore, we may effectively design and implement such a meta-heuristics search procedure in a population based fashion for GNA as it is discussed in Section 1.3 due to advantages such as the inherent ability to escape local optima, adapting multiple objectives, handling problem constraints, adjusting resolution of results by population size and performance on NP-hard problems where exhaustive search is impractical [170].

This thesis proposes two population based meta-heuristics approaches that utilize a set of heuristics simultaneously to scan the search space of the Global Network Alignment problem optimally against multiple contradicting objectives. The results show that populations in cooperation can bring the strong sides of different heuristics together provided that aligners break away from local maximum solutions with randomization and global memory in order to prevent premature convergence. The thesis also brings up an extensive library implementing several fundamental tasks such as supervision, partial solution exchange, local search and opening search space to be used as the building blocks of a customly defined population based meta-heuristic approach for the GNA problem. In addition, the thesis puts forward a suite of heuristics that might be used in various combinations to build a custom singular alignment method. The outcomes of the two methods are evaluated, compared with each other and concluded as follows:

4.1 Evaluation of SUMONA

The results achieved with SUMONA prove that it is possible to benefit from the characteristic advantages of different aligners while trying to optimize for better solutions in network alignment. The performance comparisons of SUMONA1, SUMONA2 and the original OptNetAlign algorithm show that the optimization can be less costly if the process is supervised. The comparisons are performed with different inputs and supervision strategies for better observation. Plus, it is seen that certain objectives can be prioritized above others using proper input data and supervision strategy. Eventually, the diverse and elite set generated throughout the process, is composed of non dominated alignments and improves the applicability of the results under different conditions by domain experts [92]

On the other hand, the results of both SUMONA variants show that more randomized search achieves better results with respect to multiple objectives since it provides the chance to cover more alternative solutions. Plus, classification of input data becomes useful for achieving the desired level of randomization and proper blend of results. SUMONA2 can be used in classifying alignments with significantly different characteristics. Therefore it will be more successful when there are several contradicting optimization objectives specified by the user. Moreover, the supervised hill climbing methodology of SUMONA2 provides significant improvement in most desired objectives while resulting in decline in some others. The decline may be eliminated by applying an alternative less greedy method such as Steepest Ascent Hill Climbing or Random-Restart Hill Climbing as the final local search step in each GA cycle.

It is also important to choose reasonable alignment parameters with the proper supervision method and input data carefully for performing a high quality optimization. The contradictions and the balance among objectives due to their nature, is an important consideration for the optimization process. Finally, the performance and quality of source alignments obtained from other aligners is a key factor in choosing the most effective supervision strategy. This fact can be observed by the performance of SUMONA2 which indicates that it is better to use a diverse and yet consistent set of input data for supervising the optimization process.

4.2 Evaluation of PERSONA

PERSONA has been tested with a number of data sets that belong to different years to demonstrate the evolution of available data in time with new findings. Consequently, it enables an in-depth analysis of alignment performance with a temporal component. In this context, the results of this study prove that PERSONA is generating high performance alignments with all data sets thanks to its collaborative approach. Thus, it becomes obvious that it follows a network independent and robust methodology compared to other aligners that demonstrate an unstable performance on various different data sets. Therefore, it may be concluded that the particle swarm inspired meta-heuristic optimization approach of PERSONA is able to adapt the characteristics of evolving networks due to its collaboration strategies as a swarm, individual alignment heuristics as behaviors, interpretation of network meta-data, randomized discovery in the search space and so on.

PERSONA is able to handle local maximums by removing some of the minimally contributing mappings of an alignment for opening search space in each execution cycle. In order not to repeat weak mappings, certain tasks of the algorithm such as assigning parameter values in alignment heuristics, removal of minimally contributing mappings, search in the post-processing phase, selecting historically significant partial solutions or selection of partial alignments to be exchanged in each cycle are implemented in a randomized fashion. Such randomized tasks enabled access to different regions in the search space so that stronger mappings could be discovered without converging prematurely. It is also essential to decide how to initialize the particles of the swarm since the initial mappings heavily effect the final alignments. The reason is that, initial mappings narrow down the solution space and they may also be exchanged among the concurrently progressing aligners even if some of them are removed from an alignment in the further cycles of execution.

This study showed that PERSONA is able to achieve remarkable results among multiple objectives of the Global Network Alignment problem. The performance of the PERSONA and the competitor aligners were evaluated with multiple criteria decision making tools of WPM and WSM to reach a conclusion in this sense. On the contrary, balance among multiple objectives may still be compromised for achieving superiority on some of the objectives based on user priorities. User priorities can always be targeted by altering aligner behaviours. However, it should be noted that most aligners make more significant compromises in the objectives that they ignore compared to the superior results that they achieve in the objectives that they focus. Therefore, the necessity of storing and further utilization of partial solutions with significant topological similarity and node similarity proves to be true for preventing huge compromises in contradicting objectives. Proper PERSONA methods should be employed for further utilization of such partial solutions.

4.3 Comparison of Methods and Concluding Remarks

PERSONA and SUMONA are both population based meta-heuristics methods that intend to minimize computational time and complexity by narrowing down the search space of GNA problem effectively. The main reason that these methods employ population based approaches is that they aim to generate a diverse set of candidate solutions from various regions of the search space. Each population member in both methods is assigned as an individual aligner that aims to generate its own alignment by collaborating with other members of the population as discussed in Section 1.3. The collaboration in question is carried out by periodically exchanging reliable solution subsets with respect to the meta-heuristics policy of the population. Both algorithms allow adjusting the exchange frequency through their input parameters. In addition to collaboration, each aligner perform individual behavior that distinguishes itself from the rest of the population. In both algorithms, execution proceeds until termination criteria of either cycler or CPU time are met. The population based results were aggregated and compared with other prominent algorithms throughout the experiments carried out for this thesis. It is worth mentioning that the performance of PERSONA has been thoroughly analyzed since it was compared with more algorithms using an extensive variety of data sets.

One of the main differences of these methods is that PERSONA population is composed of more flexible members capable of randomly switching between heuristics as part of their individual behavior in contrast to the SUMONA population members that execute a small scale mutation task in each cycle. Besides, PERSONA aligners display an intelligent reaction policy during solution exchange with other aligners. This policy involves filtering the subgraph to receive and deciding on the content of subgraph to send back by querying and evaluating the respective contributions across all alignment objectives. In contrast, SUMONA aligners perform a randomized crossover task for solution exchange and then they finalize each cycle by performing hill climbing for a randomly chosen alignment objective. As a result, PERSONA aligners have more potential to progress at each cycle due to their autonomous behavior. In contrast, SUMONA aligners cause less computational overhead since they are merely performing standard supervision tasks in addition to the core GA operations. Due to the different progression rates, SUMONA aligners evaluate the difference of smaller chunks of alignment compared to PERSONA aligners. As such, SUMONA aligners have more precision in a local region of the search space whereas PERSONA aligners have a broader picture to evaluate their performance and tackle local optima. For this reason, PERSONA is more resistant to convergence due to its global evaluation capabilities and inherent heuristics.

SUMONA and PERSONA have contrasting advantages and disadvantages in terms of setup requirements. One important category of setup requirements is solution quality and objective prioritization. In this category, SUMONA enables to adjust the resolution and objective focus of results by configuring the maximum number of aligners and alignment objectives with application input parameters for run-time. On the other hand, it is required to explicitly assign and configure all the heuristics of each aligner to be employed in the PERSONA population as part of the setup process. For this reason, it is more convenient to configure the expected objectives in SUMONA. Nevertheless, the explicit configuration of PERSONA serves as a means to adjust the priority of alignment objectives in detail. The other important setup requirement category is supervision with input alignments. SUMONA relies on results of external alignment algorithms and uses them in crossover phase of GA search as explained in Section 2.2.4. This kind of setup is a bit handy and it requires experience in using external alignment algorithms. On the other hand, PERSONA is able to create alignments from scratch. Therefore, it doesn't need to configure and classify input alignments. Nevertheless, it may be a good practice to initialize the individual aligners as explained in Section 3.2.4.

SUMONA displays a swap behavior that alters its stationary set of mappings indirectly with respect to the newly proposed non-stationary ones due to possible violations in the one-to-one mapping constraint for both its collaborative and individual tasks. This behavior is based on the UPMX approach explained in Section 2.2.3 and it is computationally efficient although it may cause strong mappings of the stationary part of the solution to be swapped consequently. On the contrary, PERSONA adopts a more conservative approach that preserves the existing mappings when new mapping are proposed by filtering possible one-to-one mapping violations as part of its default behavior. However, such a behavior carries the potential of a premature convergence due to shrinkage in the search space. For this reason, PERSONA ensures diversity and flexibility by periodically removing weaker mappings and providing the possibility of mapping the respective free nodes in different combinations. One final note about the conservative behavior of PERSONA is that it may be disabled by deactivating the filter for one-to-one mapping violations in case of a premature convergence since the algorithm is also able to remove the undesired many-to-many mappings periodically during execution.

One final remark to mention is that SUMONA and PERSONA are both able to process annotational inputs in addition to BS and PPI Network inputs just like some of the existing alignment algorithms. Similar to this extensive input strategy, it is possible to use annotational inputs as the sole Node Similarity input alternative to BS inputs in some other alignment algorithms. A notable pitfall of using annotational inputs in Network Alignment is that it introduces some bias since the annotational inputs are further used in measuring alignment quality despite the fact that they are previously predicted functions by possibly other in-silico methods. As a consequence, the extreme possibilities can be stated as either proceeding a prediction task from a wrong assumption in the worst case or forming a consistent alignment pattern in the best case. For this reason, the alignment quality of both proposed algorithms and their competitors were evaluated by a balanced combination of topological, biological and annotational metrics in order to reduce bias to an acceptable level. In this context, SUMONA was simply compared with its competitors by a Pareto Dominance perspective as well as a separate statistical significance basis for each alignment objective. Conversely, PERSONA and its competitors were effectively compared with each other by means of an aggregated score of overall alignment quality that is extracted from individual topological, biological and annotational scores of each competitor as mentioned in Section 3.3.4.

Appendices

A Implementation Details of SUMONA

A.1 Experiment Parameters

The exact format of the command line arguments used in Supervision Strategy 1,2 and 3 are as follows:

```
./optnetalign −−net1 scere.net −−net2 dmela.net −−bitscores scdm.sim −−annotations1 scere.annos −−annotations2 dmela.annos −−total −−goc −−s3 −−blastsum −−outprefix test1 −−verbose −−popsize 100 −−generations 1000
```

The exact format of the command line arguments used in Supervision Strategy 4 is as follows:

```
./optnetalign −−net1 scere.net −−net2 dmela.net −−bitscores scdm.sim −−annotations1 scere.annos −−annotations2 dmela.annos −−total −−ec −−s3 −−blastsum −−goc −−outprefix test4 −−verbose −−dynparams −−popsize 100 −−generations 1000 −−hillclimbiters 3
```

These command execution formats are compatible with the original OptNetAlign application. The default values for niters, cxrate, cxswappb, mutrate, mutswappb, oneobjrate parameters can also be changed by additional arguments. Please note that the order of optimization objectives have effect on the results. The value of *−−hillclimbiters* argument is multiplied by an embedded coefficient of 500 to calculate the value of niters parameter. Besides, *−−dynparams* argument that dynamically adjusts mutrate, cxrate and oneobjrate to match their success rates can also be used in producing non-dominated alignments.

Source code of SUMONA variants and data conversion tools can be shared upon request.

B Implementation Details of PERSONA

B.1 Explanation for Individual Comparison and Exchange

The messaging context with respect to the alignment state are:

- If the performance of the SubGraph in the sent message is superior to the performance of the receiving aligner in all objectives then all the possible mappings of the alignment within the sent message are added to the receiving aligner's alignment.
- If the performance of the SubGraph in the sent message is inferior to the performance of the receiving aligner in all objectives then the receiving aligner sends its current alignment as a Future message [135] that will transform into a subgraph back to the sender.
- If the performance of the SubGraph in the sent message is inferior to the performance of the receiving aligner in some objectives then the receiving aligner sends the Subgraph of one of its superior objectives by random selection as a Future message back to the sender. The message transforms into a subgraph upon the completion of a concurrent operation and the proper mappings within the message are inserted into the alignment of the initial sender.

The procedure can be summarized in three fundamental actions highlighting the active aligner actor:

1. **Sender:** The initial action of the Sender is about sending Objectives Scores to other Aligners. If the sender pareto dominates the receiver, the sent subgraph is added to the receiver's alignment and 2$^{\text{nd}}$ and 3$^{\text{rd}}$ steps are skipped.
2. **Receiver:** The reaction of the Receiver to the initial action is sending back the subgraph of a randomly selected stronger objective to the previous sender or sending back its whole alignment if the receiver scores pareto dominate the sender scores.
3. **Sender:** The final action of the Sender is adding the received subgraph to its alignment.

B.2 Explanation for Broadcast and Following Leading Aligners

In the global actor environment, the aligner with the globally best score in each objective is tracked by a specific subgraph ID that marks the mappings of that subgraph. These subgraph IDs are periodically broadcasted to all other aligners to make them aware of the current leading performance in that objective and trigger the receivers to add the possible mappings of the corresponding subgraph to their own alignment.

B.3 Objective Specific SubGraphs

Objective specific SubGraphs are used while aligners are exchanging significant sections of their alignments in the scope of the “Individual Comparison and Exchange” and “Broadcast and Follow Leading Aligners” tasks. In the scope of the “Individual Comparison and Exchange” task,the inferior objectives of the sent message are listed according to the performance of the receiving aligner and one of them is randomly chosen to send back its respective subgraph. The significant subgraph from the chosen objective is extracted as follows:

- GOC Specific SubGraph: Node Pairs having k common GO terms or more are extracted.
- BS Specific SubGraph: Node Pairs having a BS similarity above a certain threshold or more are extracted.
- EC Specific SubGraph: Edge Pairs are extracted in which a pair of node pairs are connected to each other in both organisms. (A node pair is composed of a single alignment mapping that has a pair of nodes belonging to the compared organisms)
- ICS [87] Specific SubGraphs: SubGraphs with aligned node pairs of top centrality scores are extracted when ICS option is picked. The respective centrality measure for this task is randomly picked among the system wide centrality metrics of “betweenness”, “harmonic”, “closeness” and “pagerank” parameters.
- S^3 Specific SubGraphs: Random Selection from specific aligned subgraphs of “Non-Induced Mapping Subgraph” and “top scoring connectivity degree subgraph”. Non-Induced Mapping Subgraph is built by removing aligned edges without counterparts in the opposite organism from the subgraph in order to prevent the S^3 score penalty that they cause.

B.4 Possible Heuristics of an Aligner

We developed several alignment heuristics and gathered them as a suite to be used by members of the PERSONA population in this study. These heuristics are used in progressing an individual alignment and they identify new valid node mappings based on the initial state of a particular alignment as well as their runtime parameters. The runtime parameters are recommended to be altered in each execution cycle in order to cover more search space during progression of a particular solution. The most common practices of assigning runtime parameters are random selection from a valid range and gradual relaxation. Each aligner in the population may adopt a behavior that gathers a selection of various heuristics complementing each other in a characteristic way. Users of PERSONA are strongly encouraged to change the behavior of each aligner by changing its assigned heuristics and their range of runtime parameters in order to achieve a new prioritization of objectives. While doing so, the meta-heuristic collaboration actions and individual heuristics of each aligner should be kept in a balanced state that enables efficient collaboration without convergence. One key point to keep this balance is to configure the execution periods of collaboration actions and individual heuristics without overlap and dominance against each other in the scheduler of PERSONA. (The library implemented within PERSONA may also be used to develop other aligners or other alignment optimizers in additon to configuration.)

Our Aligner Heuristics Suite may be classified into three main and one support group. The first one of these, the seed-and-extend heuristics group includes several centrality heuristics to initiate a seed. A centrality heuristic mainly relies on one of the betweenness centrality, closeness centrality, harmonic centrality, pagerank [138] and connectivity degrees approaches and it may further have particular biological and annotational similarity thresholds to identify central nodes with a secondary dimension. The central nodes identified with one of these heuristics may be extended with an edge-forming heuristic. On the other hand, the second group includes some cluster mapping heuristics that mainly rely on clusters of "Louvain Modularity" or "Label Propagation" [138] as well as secondary topological similarity and node similarity constraints. The cluster membership of each node is computed and stored in the graph database with respect to its interactions and the overall topology. Cluster Mapping heuristics assume that the node pairs from a significant cluster pair are both topologically and individually similar. For this reason, they give priority to mapping members of such significant clusters. As the third group of our heuristics suite, we present a set of other heuristics that prioritize node similarity objectives over topological ones. These heuristics are a combination of various queries searching for biological and annotational similarity and they still try to improve topological similarity objectives given that they satisfy certain constraints of node similarity. Finally, we present some supplementary heuristics for randomization, cleaning and progression that aim to complement the three main heuristic groups as part of a complete aligner behavior. The heuristic groups and their individual members are listed below in detail:

a) Seed & Extend:

Heuristic for Mapping Central Edges: This heuristic intends to map node pairs that reside on edge pairs provided that they have high centrality scores.

1. First, both node pairs on an edge pair is mapped if their dupleConnectivity, tripleConnectivity and quadrupleConnectivity values are above certain thresholds (ordered by the decreasing sum of common annotations in both node pairs).
2. Then, a single node pair on an edge pair where the other node pair is already aligned is mapped if the respective dupleConnectivity, tripleConnectivity and quadrupleConnectivity values are above certain thresholds (ordered by decreasing common annotations in the unaligned node pair.

Heuristic for Aligning Central Node Pairs with Connectivity Degrees: This heuristic is used as the seed operation and the following operations are chained to its completion for better propagation. In this operation, a node pair with biological similarity is mapped with respect to the number of common annotations, their dupleConnectivity sums, tripleConnectivity sums and quadrupleConnectivity sums ordered by decreasing quadrupleConnectivity sums followed by decreasing tripleConnectivity sums followed by decreasing dupleConnectivity sums of both nodes.

Heuristic for Aligning Central Node Pairs with Connectivity Degrees From Top: This heuristic is the same with “Heuristic for Aligning Central Node Pairs with Connectivity Degrees” except that it orders the results with respect to descending connectivity degree scores from top limited by the number of records to be returned and the percentile of the minimum result. The powerMode parameter can be: 2,3 or 4.

Heuristic for Aligning Central Node Pairs with Alternative Centrality Algorithms: This is heuristic can be used as a seed and other operations can be chained to its completion for better propagation (algorithm: betweenness, harmonic, pagerank and closeness). In this operation, a node pair with biological similarity is mapped with respect to the number of common annotations and each node's centrality threshold ordered by decreasing centrality sums of both nodes.

Heuristic for Aligning Central Node Pairs with Alternative Centrality Algorithms From Top: This heuristic is the same with "Heuristic for Aligning Central Node Pairs with Power Connectivity" except that it orders the results with respect to descending score of the respective Centrality algorithm from top limited by the number of records to be returned and the percentile of the minimum result. The algorithm parameter can be: betweenness, harmonic, pagerank and closeness.

Heuristic for Mapping Connected Edges: This heuristic aims to map cliques, semi cliques or small interacting patterns with each other. It has demonstrated poor performance in our experiments.

b) Cluster Mapping:

Heuristic for Aligning a Node Pair in two Given Clusters: A node pair of given clustertypes and given cluster IDs with biological similarity is mapped limited by the minimum common annotations that they share (clusterType: labelpropagation, louvain).

Heuristic for Aligning Edges in two Given Clusters: This heuristic can be used in two modes one of which aims to map an edge pair completely and the other aims to complement an existing node pair to an edge pair (clusterType: labelpropagation, louvain).

1. In the edge pair mode, an edge pair of given clustertypes is mapped and given cluster IDs having similarity limited by the minimum common annotations that each nodes pair shares.
2. In the complementing mode, a node pair of given clustertypes and given cluster IDs with biological similarity on an edge pair is mapped where the other node pair is already aligned limited by the minimum common annotations that the nodes of the pair shares.

c) Node Similarity Prioritization:

Heuristic For Prioritizing Biological Similarity (BS) Followed By Annotational Similarity (GOC): This heuristic has a gradual constraint relaxation approach. It initially tries to improve EC as a secondary objective as well as BS and GOC. After then it maps both GOC and BS improving node pairs. Finally, it maps all possible BS improving node pairs.

1. First, an edge pair with biological similarity on both node pairs is mapped if their number of common annotations is above a certain threshold, (ordered by decreasing biological similarity sum)
2. Then, one node pair of an edge pair with biological similarity on that single node pair is mapped while the other edge is already aligned if their number of common annotations is above a certain threshold, (ordered by decreasing biological similarity followed by decreasing common annotations)
3. Finally, a node pair with biological similarity is mapped if their number of common annotations is above a certain threshold (ordered by decreasing biological similarity followed by decreasing common annotations)

Heuristic For Prioritizing Annotational Similarity Followed By Biological Similarity: This heuristic applies a similar gradual constraint relaxation approach with the previous heuristic. The only difference is it prioritizes GOC before BS

1. First, an edge pair with biological similarity on both node pairs is mapped if their number of common annotations is above a certain threshold, (ordered by decreasing common annotations sum)
2. Then, a node pair with biological similarity is mapped if their number of common annotations is above a certain threshold (ordered by decreasing common annotations)
3. Finally, one node pair of an edge pair is mapped while the other edge is already aligned if their number of common annotations is above a certain threshold, (ordered by decreasing common annotation sum)

Heuristic for Listing the Top Scoring Biological Similarity Values: A predefined number of Node pairs with top scoring biological similarities are mapped to each other.

Heuristic for Listing the Top Scoring Annotational Similarity Values: A predefined number of Node pairs with top scoring common node pair annotations are mapped to each other.

Heuristic for Improving Annotational and Biological Similarity along with Edge Consistency: Edges will be mapped to each other in the two steps below provided that they meet user defined thresholds for biological and annotational similarity.

1. If the limiting annotation and biological similarity values are not zero, an Edge Pair with positive biological similarity is mapped on both node pairs limited by their number of common annotations and biological similarity on each node pair (ordered by decreasing biological similarity)
2. Otherwise, one node pair of an edge pair with biological similarity on that single node pair is mapped while the other edge is already aligned limited by their number of common annotations and biological similarity (ordered by decreasing biological similarity)

Heuristic for Improving Annotational Similarity (GOC) as well as DupleConnectivity Connectivity: This heuristic initially tries to improve bitscore similarity as a secondary objective as well as its targeted objectives. After this step, it directly improves annotational similarity and DupleConnectivity Connectivity without any other constraint. Mapping node pairs with high dupleConnectivity connectivity improves the chances of propagating the mapping locally with the neighboring edges and provides a potential for improving the topological similarity objectives EC and S^3 by consecutively executing complementary heuristics (i.e. Heuristic for Forming Edge Pair Mappings from Existing Node Pair Mappings)

1. If the given similarity is greater than zero, one node pair with biological similarity is mapped limited by the number of common annotations, similarity value and the dupleConnectivity values for both nodes (ordered by decreasing common annotations followed by decreasing biological similarity)
2. Otherwise, one node pair is mapped limited by the number of common annotations and the dupleConnectivity values for both nodes (ordered by decreasing common annotations followed by decreasing dupleConnectivity value)

d) Supplementary Heuristics:

Heuristic for Aligning Edges: This heuristic should be executed in the final stages of an alignment in order to reduce the number of unaligned edges.

Heuristic for Forming Edge Pair Mappings from Existing Node Pair Mappings: This heuristic complements many other heuristics by forming an edge from existing mappings.

1. (Given that the biological similarity is greater than zero), one node pair of an edge pair with biological similarity is mapped where the other node pairs are aligned if the number of common annotations and biological similarity of the subject node pairs are above certain threshold values.

2. (Otherwise), one node pair of an edge pair is mapped where the other node pairs are aligned if the number of common annotations of the subject node pair is above a certain threshold.

Heuristic for Adding a Given Number of Random Mappings: This heuristic maps a random node pair limited by the given pair count. It is very useful for random search. It can be used in the post processing phase to finalize the one-to-one alignment.

Heuristic for Removing Low Scoring Mappings: This heuristic removes node pair mappings scoring below the threshold values of common annotations, biological similarity and according to edge presence. It can be decided to keep the edges regardless of the other threshold values or remove them if they score low in the other threshold values. There is also another version of the same heuristic that chooses the low scoring mappings to be removed randomly when the number of mappings to be removed are limited by the respective parameter.

Heuristic for Removing Mappings Without Edges: This heuristic removes node pairs that do not constitute a conserved edge for increasing search space in order to achieve better topological scores. There is also another version of the same heuristic that chooses the low scoring mappings to be removed randomly when the number of mappings to be removed are limited by the respective parameter.

Heuristic for Removing Inductive Mappings: This heuristic removes the less contributing side of inductive mappings without violating the threshold values in the parameters in order to eliminate them from the inductive mapping definition.

Heuristic for Removing Many-To-Many Mappings: PERSONA is a one-to-one alignment optimizer since it focuses on the Global Network Alignment problem by mapping each node of the smaller network into a single node of the larger network. The application periodically removes unexpected one-to-many or many-to-many mappings that occur during some complicated queries or parallel executions that violate atomicity in graph database transactions. The main reason of applying this cleaning process is providing a feasible solution to the initial one-to-one problem definition and it doesn't imply superiority of any particular mapping approach. This heuristic is used to remove those unexpected mappings. It has two versions one of which removes the latterly added mappings and the other one removing the weaker mapping among the one-to-many alternatives. Removing the weaker mappings is computationally more demanding, so the one removing the latterly added mappings is used throughout the system.

It is finally worth noting that, the new sets of mappings performed by either the search tools or the above listed heuristics are added to an alignment unless they do not violate the one-to-one mapping constraint of that aligner. If a subset of the mappings violate the constraint, then the violating subset is removed from the set and the rest is added to the alignment.

B.5 Architecture of Critical Modules

The most critical component of the overall architecture is the concurrency infrastructure responsible for retrieving information from the single source graph database in a concurrent fashion. The frameworks for achieving this task are Akka Concurrency Framework [135] that provides a high level abstraction for multi-threading and Neo4j graph database technology [137]. The specific entities of the concurrency infrastructure are described below:

a) Future Messages: Sending a message and receiving a reply is essentially a two-step process. Future data structures are used to adapt this mechanism to a concurrent environment. A future is a data structure that is used along with actors when it is desired to avoid blocking calls and retrieve the result of certain concurrent operations in the meantime. The concurrent operation is invoked by the actor and its result can either be accessed synchronously or asynchronously [135].

b) Typed Actors: Typed Actors do not directly include a behavior like standard actors and instead they implement the active object pattern with a combination of object oriented definitions and asynchronous actors that process method calls. The Active Object Pattern only decouples method execution from method invocation into different threads of execution in order to introduce concurrency and fault tolerance. Nevertheless, different behaviors are embedded to Typed Actors in PERSONA by instructing their reaction when they receive different types of messages [135].

c) TransactionTemplate: Since transactions can collide with other transactions in a parallel environment, TransactionTemplates are used for requesting and detecting the operating resources of the graph database. Transaction templates check whether the database is occupied or not periodically and request their operation to be executed when they detect it is available. Nevertheless, some transactions may still collide when they detect the availability of the database at the same time and such cases are handled in exception handling mechanisms by triggering the same transactions with 0.5 probability so that one of the requesters may give up the resource eventually.

d) Router: In Akka, a router directs the incoming messages from the source to the outbound destination actors with a specific order in order to manage the concurrent execution of multiple actors. It is possible to embed a routing approach according to the messaging characteristics in an Akka Router. All routing approaches have their own characteristic mechanism of picking the next routee among the round robin, random, smallest mailbox, broadcast, scatter gather first completed and tail chopping mechanisms for sending the incoming messages [135].

In PERSONA, routing is used to pick the next recipient Aligner of the active sender Aligner while performing the periodically scheduled collaboration operations. The specific scheduling mechanism of the router depends on the Tail Chopping Algorithm that will first send an availability confirmation message to a randomly picked recipient and then after a small delay to a second randomly picked recipient and so on. The router with this approach waits for the first reply and forwards the first reply back to the original sender informing the availability of that recipient and discarding other replies. The goal of this scheduling mechanism is to decrease latency by performing redundant queries to multiple recipients, assuming that a randomly selected actor may be faster to respond than the others.

e) Scheduler: In Akka, it is possible to send scheduled messages to actors and execute scheduled tasks whenever the need for performing tasks in the future arises. There is a unique instance of a scheduler in every Actor System that is internally used internally scheduling tasks to happen at specific points in time. Akka Schedulers are able to schedule up to millions of triggers and it is intended to be used in various use-cases including actor receive timeouts, future timeouts, circuit breakers and other time dependent events [135]. In PERSONA, periodical operations performed by the Actor System are scheduled by this entity. These periodical operations are used to simulate real time behaviors of aligner actors via a turn based fashion. The period of individual alignment behaviors, broadcast messages and inter-alignment messages are subject to change by the domain expert for further testing expectations.

f) Aligners: Each aligner in the system is implemented with Typed Actors decoupled with their interfaces. The essential structure of Aligners are implemented in their onReceive methods in order to establish their reaction policy upon receiving messages. This method differentiates the type of the received message and applies a different procedure for different message types. It directly adds the possible mappings whenever it receives a subgraph from another aligner and similarly it adds the marked entities associated with a subgraph whenever it receives a broadcast message from leading aligners. On the other hand, the objective scores comparison operations between the sender and receiver are also implemented within this method, regarding the comparison results of the sender and the receiver as states and switching to different policies with respect to these comparison results. Each aligner in the Actor System has a characteristic behavior implemented with a combination of alignment heuristics that may all execute with different probabilities in each system cycle. The behavior of aligners are subject to change according to the desired purpose in terms of the applied heuristics and their probability of occurrence in each cycle. In principle, the domain expert is encouraged to establish different behaviors out of the provided alignment heuristics in order to ensure diversity and collaboration among aligners.

B.6 Experiment Parameters & Remarks

- HubAlign is analyzed to be more efficient with parameter λ ranging from 0.1 to 0.2 that improves AFS over EC and parameter α ranging from 0.7 to 1.0 improving EC over AFS in its own publication. Therefore it is performed with those corresponding parameter values.

- One implementation detail to consider for NETAL is that it should take the smaller network as the first argument and the bigger network as the second argument. On the other hand the only significant parameter of the algorithm is found to be the α parameter since the other arguments had no visible effect on the results, therefore the algorithm has been executed by adjusting the α parameter with a step size of 0.1 between 0 and 1 boundary values.

- There is no clear recommendation of parameter values in SPINAL. For this reason, its alpha value has been incremented by 0.1 from 0 to 1 for invoking both modes SPINAL. The average of results (apart from the discarded outliers) has been compared with other competitors.

- In order to map a larger chunk of node pairs in each cycle without compromising topological quality, the finalMappingFactor parameter should at least be around a quarter of the size of the smaller network to rapidly achieve satisfactory sequence similarity results.

- OptNetAlign has been invoked with the following command for all S3, GOC and BS (blastsum) objectives:

```
time ./optnetalign −−net1 mm.interaction −−net2 dm.interaction −−total −−s3 −−bitscores
mmdm.sim −−blastsum −−annotations1 mm.annos −−annotations2 dm.annos −−goc
−−cxrate 0.05 −−cxswappb 0.75 −−mutrate 0.05 −−mutswappb 0.0001 −−oneobjrate
0.75 −−dynparams −−popsize 200 −−generations 250000000 −−hillclimbiters 10000
−−timelimit 180 −−outprefix mm-dm3 −−finalstats » mm-dm3.finalstats
```

- PINALOG has been invoked with the following command for all topological, biological and annotational objectives:

  ```
  /usr/bin/time -o dmmm_mentha_time.txt ./pinalog1.0 dm.interaction mm.interaction
  dmmm.sim mentha.gaf
  ```

- SANA was performed with its commit (Mar. 13, 2019) for unnormalized sim files and it has been invoked with the following command with balanced coefficients across all topological, biological and annotational objectives:

  ```
  ./sana -g1 DMelanogaster -g2 SCerevisiae -s3 0.2 -go_k 0.4 -esim 1 0.4 -simFile 1
  DMelanogaster_SCerevisiae_blastws3ev10.out -simFormat 1 1
  ```

- MAGNA++ has been invoked with the following command for all topological and biological and objectives:

  ```
  /usr/bin/time -o time_mmdm.txt
  ./MAGNA++_CLI_Linux64_01_30_2015/magnapp_cli_linux64 -G mm.gw -H dm.gw
  -m S3 -o mmdm.result -d mmdm.mat -p 15000 -t 4 -e 0.5 -n 2000 -a 0.5
  ```

- PISwap has been invoked with its execution script based on the following line for topological and biological objectives:

  $(S, M) = psb09.processOnce(G, G2, GS, M0, 0.6, 200)$

- PROPER has been invoked with the following command having the biological input threshold 150 in all experiments except for DM-MM Mentha comparison where PROPER scored better with the threshold values 10 and 150 together. The average of results has been taken into account in this particular case.

  ```
  ./exe/proper
  mentha2021-swissprot-uniprotgoa/dm.interaction
  mentha2021-swissprot-uniprotgoa/sc.interaction
  mentha2021-swissprot-uniprotgoa/dmsc.sim 1 150 proper_mentha_dmsc_10.aln
  ```

- GEDEVO has been invoked from its particular plugin in Cytoscape GUI. Annotational and Biological Similarity files has been given as optional inputs during experiments.
- PERSONA has been invoked with "gopp" and "epp" flags that improve annotational search and random search during post processing. Plus, In order to map a larger chunk of node pairs in each cycle without compromising topological quality, the finalMappingFactor parameter of PERSONA should at least be around a quarter of the size of the smaller network to rapidly achieve satisfactory sequence similarity results. It should also be noted that, PERSONA enables changing objective priority through aligner behavior modification.
- The Biological Sequence Similarity scores were extracted from FASTA files with parameters of *-word_size* of 3 and $-evalue$ of 10 by BLASTP in BioGRID 2017 and Mentha 2021 data sets.
- Akka 2.5.X series do not have any issues with artifactID value of akka-actor-2.11. Typed Actors are available until Akka version 2.5.8
- The concept of "connectivity degrees" are mentioned as "powers" in the source code of PERSONA.

Table B.1 is a summary table showing the possible input files and strengths of each aligner in objectives:

Appendix Table B.1: Possible Input Types and Strengths of Each Aligner. The "Optimized Objectives" column display the objectives that the respective Aligner is strong against.

Alignment Algorithm	Input Types	Optimized Objectives
HubAlign	Network, Sequence Similarity	EC, BS, LCCS
OptNetAlign	Network, Sequence Similarity, Annotations	EC, S3, GOC, GOE, BS
NETAL	Network, Sequence Similarity	EC, S3, BS
PINALOG	Network, Sequence Similarity, Annotations	EC, GOC, BS, LCCS, Resnik
PROPER	Network, Sequence Similarity	EC, BS, LCCS
PERSONA	Network, Sequence Similarity, Annotations	GOE, GOC, BS, LCCS, EC
SPINAL	Network, Sequence Similarity	EC, BS, LCCS,GOC
GEDEVO	Network, Sequence Similarity, Annotation Similarity	EC, LCCS
MAGNA++	Network, Sequence Similarity	EC, S3
SANA	Network, Sequence Similarity, Annotation Similarity	S3, EC
PISwap	Network, Sequence Similarity	EC, BS, Resnik

Bibliography

[1] D. Lee, O. Redfern, and C. Orengo. Predicting protein function from sequence and structure. *Nature Reviews Molecular Cell Biology*, 8(12):995–1005, Number: 12 Publisher: Nature Publishing Group, December 2007.

[2] W. M. Fitch. Homology: a personal view on some of the problems. *Trends in Genetics*, 16(5):227–231, May 2000.

[3] P. Mier, M. A. Andrade-Navarro, and A. J. Pérez-Pulido. orthoFind Facilitates the Discovery of Homologous and Orthologous Proteins. *PLOS ONE*, 10(12):e0143906, Publisher: Public Library of Science, December 2015.

[4] V. Memišević, T. Milenković, and N. Pržulj. Complementarity of network and sequence information in homologous proteins. *Journal of Integrative Bioinformatics*, 7(3):275–289, Publisher: De Gruyter Section: Journal of Integrative Bioinformatics, December 2010.

[5] O. Kuchaiev and N. Przulj. Global Network Alignment. *Nature Precedings*, June 2010.

[6] O. Kuchaiev and N. Pržulj. Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, 27(10):1390–1396, Number: 10, May 2011.

[7] G. A. Pavlopoulos, M. Secrier, C. N. Moschopoulos, T. G. Soldatos, S. Kossida, J. Aerts, R. Schneider, and P. G. Bagos. Using graph theory to analyze biological networks. *BioData Mining*, 4(1):10, April 2011.

[8] M. Koyutürk. Algorithmic and analytical methods in network biology. *WIREs Systems Biology and Medicine*, 2(3):277–292, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wsbm.61, 2010.

[9] H. G. Vikis and K.-L. Guan. Glutathione-S-Transferase-Fusion Based Assays for Studying Protein-Protein Interactions. In H. Fu, editor, *Protein-Protein Interactions: Methods and Applications*, Methods in Molecular Biology, pages 175–186. Humana Press, Totowa, NJ, 2004. ISBN 978-1-59259-762-8. doi: 10.1385/1-59259-762-9:175. URL *https://doi.org/10.1385/1-59259-762-9:175*.

[10] O. Puig, F. Caspary, G. Rigaut, B. Rutz, E. Bouveret, E. Bragado-Nilsson, M. Wilm, and B. Séraphin. The Tandem Affinity Purification (TAP) Method: A General Procedure of Protein Complex Purification. *Methods*, 24(3):218–229, July 2001.

[11] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences*, 98(8):4569–4574, Publisher: National Academy of Sciences Section: Biological Sciences, April 2001.

[12] A.-C. Gavin, M. Bösche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J. M. Rick, et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868):141–147, Number: 6868 Publisher: Nature Publishing Group, January 2002.

[13] D. Stoll, M. F. Templin, J. Bachmann, and T. O. Joos. Protein microarrays: applications and future challenges. *Current Opinion in Drug Discovery & Development*, 8 (2):239–252, March 2005.

[14] W. G. Willats. Phage display: practicalities and prospects. *Plant Molecular Biology*, 50(6):837–854, December 2002.

[15] P. E. Hodges, A. H. Z. McKee, B. P. Davis, W. E. Payne, and J. I. Garrels. The Yeast Proteome Database (YPD): a model for the organization and presentation of genome-wide functional data. *Nucleic Acids Research*, 27(1):69–73, January 1999.

[16] T. S. Keshava Prasad, R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, et al. Human Protein Reference Database—2009 update. *Nucleic Acids Research*, 37(Database issue):D767–D772, January 2009.

[17] K. Han, B. Park, H. Kim, J. Hong, and J. Park. HPID: The Human Protein Interaction Database. *Bioinformatics*, 20(15):2466–2470, October 2004.

[18] J. Y. Chen, S. Mamidipalli, and T. Huan. HAPPI: an online database of comprehensive human annotated and predicted protein interactions. *BMC Genomics*, 10(1):S16, July 2009.

[19] J. Yu, S. Pacifico, G. Liu, and R. L. Finley. DroID: the Drosophila Interactions Database, a comprehensive resource for annotated gene and protein interactions. *BMC Genomics*, 9(1):461, October 2008.

[20] H. W. Mewes, C. Amid, R. Arnold, D. Frishman, U. Güldener, G. Mannhaupt, M. Münsterkötter, P. Pagel, et al. MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Research*, 32(suppl_1):D41–D44, January 2004.

[21] A. Zanzoni, L. Montecchi-Palazzi, M. Quondam, G. Ausiello, M. Helmer-Citterich, and G. Cesareni. MINT: a Molecular INTeraction database. *FEBS Letters*, 513 (1):135–140, _eprint: https://febs.onlinelibrary.wiley.com/doi/pdf/10.1016/S0014-5793%2801%2903293-8, 2002.

[22] S. Kerrien, Y. Alam-Faruque, B. Aranda, I. Bancarz, A. Bridge, C. Derow, E. Dimmer, M. Feuermann, et al. IntAct—open source resource for molecular interaction data. *Nucleic Acids Research*, 35(suppl_1):D561–D565, January 2007.

[23] I. Xenarios, D. W. Rice, L. Salwinski, M. K. Baron, E. M. Marcotte, and D. Eisenberg. DIP: the Database of Interacting Proteins. *Nucleic Acids Research*, 28(1):289–291, January 2000.

[24] G. D. Bader, I. Donaldson, C. Wolting, B. F. F. Ouellette, T. Pawson, and C. W. V. Hogue. BIND—The Biomolecular Interaction Network Database. *Nucleic Acids Research*, 29(1):242–245, January 2001.

[25] C. Stark. BioGRID: a general repository for interaction datasets. *Nucleic Acids Research*, 34(90001):D535–D539, Number: 90001, January 2006.

[26] E. R. Jefferson, T. P. Walsh, T. J. Roberts, and G. J. Barton. SNAPPI-DB: a database and API of Structures, iNterfaces and Alignments for Protein–Protein Interactions. *Nucleic Acids Research*, 35(suppl_1):D580–D589, January 2007.

[27] R. Mosca, A. Céol, A. Stein, R. Olivella, and P. Aloy. 3did: a catalog of domain-based interactions of known three-dimensional structure. *Nucleic Acids Research*, 42(D1): D374–D379, January 2014.

[28] M. Kuhn, C. von Mering, M. Campillos, L. J. Jensen, and P. Bork. STITCH: interaction networks of chemicals and proteins. *Nucleic Acids Research*, 36(suppl_1):D684–D688, January 2008.

[29] D. Szklarczyk, A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, P. Minguez, T. Doerks, M. Stark, et al. The STRING database in 2011: functional interaction

networks of proteins, globally integrated and scored. *Nucleic Acids Research*, 39 (suppl_1):D561–D568, January 2011.

[30] S. Orchard, M. Ammari, B. Aranda, L. Breuza, L. Briganti, F. Broackes-Carter, N. H. Campbell, G. Chavali, et al. The MIntAct project—IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Research*, 42(D1): D358–D363, Number: D1, January 2014.

[31] D. Alonso-López, F. J. Campos-Laborie, M. A. Gutiérrez, L. Lambourne, M. A. Calderwood, M. Vidal, and J. De Las Rivas. APID database: redefining protein–protein interaction experimental evidences and binary interactomes. *Database*, 2019(baz005), January 2019.

[32] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research*, 13(11):2498–2504, Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab, November 2003.

[33] J. Ji, A. Zhang, C. Liu, X. Quan, and Z. Liu. Survey: Functional Module Detection from Protein-Protein Interaction Networks. *IEEE Transactions on Knowledge and Data Engineering*, 26(2):261–277, Conference Name: IEEE Transactions on Knowledge and Data Engineering, February 2014.

[34] M. W. Gonzalez and M. G. Kann. Chapter 4: Protein Interactions and Disease. *PLOS Computational Biology*, 8(12):e1002819, Publisher: Public Library of Science, December 2012.

[35] E. Sprinzak and H. Margalit. Correlated sequence-signatures as markers of protein-protein interaction. *Journal of Molecular Biology*, 311(4):681–692, August 2001.

[36] M. Deng, S. Mehta, F. Sun, and T. Chen. Inferring domain-domain interactions from protein-protein interactions. *Genome Research*, 12(10):1540–1548, October 2002.

[37] T. M. W. Nye, C. Berzuini, W. R. Gilks, M. M. Babu, and S. A. Teichmann. Statistical analysis of domains in interacting protein pairs. *Bioinformatics*, 21(7):993–1001, Publisher: Oxford Academic, April 2005.

[38] H. B. Fraser, A. E. Hirsh, D. P. Wall, and M. B. Eisen. Coevolution of gene expression among interacting proteins. *Proceedings of the National Academy of Sciences of the United States of America*, 101(24):9033–9038, June 2004.

[39] S. P. Kanaan, C. Huang, S. Wuchty, D. Z. Chen, and J. A. Izaguirre. Inferring protein-protein interactions from multiple protein domain combinations. *Methods in Molecular Biology (Clifton, N.J.)*, 541:43–59, 2009.

[40] K. S. Guimarães and T. M. Przytycka. Interrogating domain-domain interactions with parsimony based approaches. *BMC Bioinformatics*, 9(1):171, March 2008.

[41] J. Gertz, G. Elfond, A. Shustrova, M. Weisinger, M. Pellegrini, S. Cokus, and B. Rothschild. Inferring protein interactions from phylogenetic distance matrices. *Bioinformatics*, 19(16):2039–2045, November 2003.

[42] C.-S. Goh, A. A. Bogan, M. Joachimiak, D. Walther, and F. E. Cohen. Co-evolution of proteins with their interaction partners 1 1Edited by B. Honig. *Journal of Molecular Biology*, 299(2):283–293, June 2000.

[43] T. Dandekar, B. Snel, M. Huynen, and P. Bork. Conservation of gene order: a fingerprint of proteins that physically interact. *Trends in Biochemical Sciences*, 23(9): 324–328, September 1998.

[44] E. M. Marcotte, M. Pellegrini, H. L. Ng, D. W. Rice, T. O. Yeates, and D. Eisenberg. Detecting protein function and protein-protein interactions from genome sequences. *Science (New York, N.Y.)*, 285(5428):751–753, July 1999.

[45] A. J. Enright, I. Iliopoulos, N. C. Kyrpides, and C. A. Ouzounis. Protein interaction maps for complete genomes based on gene fusion events. *Nature*, 402(6757):86–90, Number: 6757 Publisher: Nature Publishing Group, November 1999.

[46] R. Mosca, T. Pons, A. Céol, A. Valencia, and P. Aloy. Towards a detailed atlas of protein–protein interactions. *Current Opinion in Structural Biology*, 23(6):929–940, December 2013.

[47] E. Demir, M. P. Cary, S. Paley, K. Fukuda, C. Lemer, I. Vastrik, G. Wu, P. D'Eustachio, et al. The BioPAX community standard for pathway data sharing. *Nature biotechnology*, 28(9):935–942, Publisher: Nature Publishing Group, September 2010.

[48] H. Hermjakob, L. Montecchi-Palazzi, G. Bader, J. Wojcik, L. Salwinski, A. Ceol, S. Moore, S. Orchard, et al. The HUPO PSI's molecular interaction format–a community standard for the representation of protein interaction data. *Nature Biotechnology*, 22(2):177–183, February 2004.

[49] S. Kerrien, S. Orchard, L. Montecchi-Palazzi, B. Aranda, A. F. Quinn, N. Vinod, G. D. Bader, I. Xenarios, et al. Broadening the horizon–level 2.5 of the HUPO-PSI format for molecular interactions. *BMC biology*, 5:44, October 2007.

[50] S. Bandyopadhyay, R. Sharan, and T. Ideker. Systematic identification of functional orthologs based on protein network comparison. *Genome Research*, 16(3):428–435, March 2006.

[51] S. E. Brenner. Errors in genome annotation. *Trends in Genetics*, 15(4):132–133, Publisher: Elsevier, April 1999.

[52] M. G. Reese, G. Hartzell, N. L. Harris, U. Ohler, J. F. Abril, and S. E. Lewis. Genome Annotation Assessment in Drosophila melanogaster. *Genome Research*, 10(4):483–501, Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab, April 2000.

[53] K. Sjölander. Phylogenomic inference of protein molecular function: advances and challenges. *Bioinformatics*, 20(2):170–179, January 2004.

[54] The Gene Ontology Consortium. The Gene Ontology project in 2008. *Nucleic Acids Research*, 36(suppl_1):D440–D444, January 2008.

[55] The Gene Ontology Consortium. The Gene Ontology Resource: 20 years and still GOing strong. *Nucleic Acids Research*, 47(D1):D330–D338, January 2019.

[56] A. Bairoch. The ENZYME database in 2000. *Nucleic Acids Research*, 28(1):304–305, January 2000.

[57] A. Bairoch and R. Apweiler. The SWISS-PROT Protein Sequence Data Bank and Its New Supplement TREMBL. *Nucleic Acids Research*, 24(1):21–25, January 1996.

[58] A. Ruepp, A. Zollner, D. Maier, K. Albermann, J. Hani, M. Mokrejs, I. Tetko, U. Güldener, et al. The FunCat, a functional annotation scheme for systematic

classification of proteins from whole genomes. *Nucleic Acids Research*, 32(18): 5539–5545, September 2004.

[59] M. Kanehisa and S. Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1):27–30, January 2000.

[60] R. Caspi, T. Altman, R. Billington, K. Dreher, H. Foerster, C. A. Fulcher, T. A. Holland, I. M. Keseler, et al. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. *Nucleic Acids Research*, 42(D1):D459–D471, January 2014.

[61] G. Wu and R. Haw. Functional Interaction Network Construction and Analysis for Disease Discovery. *Methods in Molecular Biology (Clifton, N.J.)*, 1558:235–253, 2017.

[62] M. P. Samanta and S. Liang. Predicting protein functions from redundancies in large-scale protein interaction networks. *Proceedings of the National Academy of Sciences of the United States of America*, 100(22):12579–12583, October 2003.

[63] M. Tantardini, F. Ieva, L. Tajoli, and C. Piccardi. Comparing methods for comparing networks. *Scientific Reports*, 9(1):17557, Number: 1 Publisher: Nature Publishing Group, November 2019.

[64] R. Shen and C. Guda. Applied Graph-Mining Algorithms to Study Biomolecular Interaction Networks, https://www.hindawi.com/journals/bmri/2014/439476/. ISSN: 2314-6133 Pages: e439476 Publisher: Hindawi Volume: 2014, April 2014.

[65] J. Fan, A. Cannistra, I. Fried, T. Lim, T. Schaffner, M. Crovella, B. Hescott, and M. D. M. Leiserson. Functional protein representations from biological networks enable diverse cross-species inference. *Nucleic Acids Research*, 47(9):e51–e51, May 2019.

[66] M. Cao, H. Zhang, J. Park, N. M. Daniels, M. E. Crovella, L. J. Cowen, and B. Hescott. Going the Distance for Protein Function Prediction: A New Distance Metric for Protein Interaction Networks. *PLOS ONE*, 8(10):e76339, Publisher: Public Library of Science, October 2013.

[67] G. R. G. Lanckriet, M. Deng, N. Cristianini, M. I. Jordan, and W. S. Noble. Kernel-based data fusion and its application to protein function prediction in yeast. In

Biocomputing 2004, pages 300–311. WORLD SCIENTIFIC, December 2003. ISBN 978-981-238-598-7. doi: 10.1142/9789812704856_0029. URL *https://www.worldscientific.com/doi/abs/10.1142/9789812704856_0029*.

[68] C. S. Leslie, E. Eskin, A. Cohen, J. Weston, and W. S. Noble. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467–476, March 2004.

[69] Y. Pan, J. Wang, and M. Li. *Algorithmic and artificial intelligence methods for protein bioinformatics*. Wiley, Hoboken, New Jersey, 2014. ISBN 978-1-118-56781-4 978-1-118-34578-8. OCLC: 915592946.

[70] C.-S. Liao, K. Lu, M. Baym, R. Singh, and B. Berger. IsoRankN: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25(12):i253–i258, Number: 12, June 2009.

[71] R. Singh, J. Xu, and B. Berger. Pairwise Global Alignment of Protein Interaction Networks by Matching Neighborhood Topology. In T. Speed and H. Huang, editors, *Research in Computational Molecular Biology*, Lecture Notes in Computer Science, pages 16–31, Berlin, Heidelberg, 2007. Springer. ISBN 978-3-540-71681-5. doi: 10.1007/978-3-540-71681-5_2.

[72] C.-Y. Ma and C.-S. Liao. A review of protein–protein interaction network alignment: From pathway comparison to global alignment. *Computational and Structural Biotechnology Journal*, 18:2647–2656, January 2020.

[73] B. P. Kelley, B. Yuan, F. Lewitter, R. Sharan, B. R. Stockwell, and T. Ideker. PathBLAST: a tool for alignment of protein interaction networks. *Nucleic Acids Research*, 32(Web Server issue):W83–W88, July 2004.

[74] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, October 1990.

[75] M. Kalaev, M. Smoot, T. Ideker, and R. Sharan. NetworkBLAST: comparative analysis of protein networks. *Bioinformatics*, 24(4):594–596, February 2008.

[76] M. Koyutürk, Y. Kim, U. Topkara, S. Subramaniam, W. Szpankowski, and A. Grama. Pairwise Alignment of Protein Interaction Networks. *Journal of Computational Biology*, 13(2):182–199, Publisher: Mary Ann Liebert, Inc., publishers, March 2006.

[77] J. Flannick, A. Novak, B. S. Srinivasan, H. H. McAdams, and S. Batzoglou. Græmlin: General and robust alignment of multiple large interaction networks. *Genome Research*, 16(9):1169–1181, September 2006.

[78] B.-S. Seah, S. S. Bhowmick, and C. F. Dewey, Jr. DualAligner : a dual alignment-based strategy to align protein interaction networks. *Bioinformatics*, 30(18):2619–2626, September 2014.

[79] A. Alcalá, R. Alberich, M. Llabrés, F. Rosselló, and G. Valiente. AligNet: alignment of protein-protein interaction networks. *BMC Bioinformatics*, 21(6):265, November 2020.

[80] P. H. Guzzi and T. Milenković. Survey of local and global biological network alignment: the need to reconcile the two sides of the same coin. *Briefings in Bioinformatics*, 19(3):472–481, May 2018.

[81] A. Elmsallati, C. Clark, and J. Kalita. Global Alignment of Protein-Protein Interaction Networks: A Survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 13(4):689–705, July 2016.

[82] R. Singh, J. Xu, and B. Berger. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences of the United States of America*, 105(35):12763–12768, September 2008.

[83] O. Kuchaiev, T. Milenković, V. Memišević, W. Hayes, and N. Pržulj. Topological network alignment uncovers biological function and phylogeny. *Journal of The Royal Society Interface*, 7(50):1341–1354, Number: 50, September 2010.

[84] T. Milenković, W. L. Ng, W. Hayes, and N. Pržulj. Optimal Network Alignment with Graphlet Degree Vectors. *Cancer Informatics*, 9:121–137, June 2010.

[85] V. Memišević and N. Pržulj. C-GRAAL: Common-neighbors-based global GRAph ALignment of biological networks. *Integrative Biology*, 4(7):734–743, Publisher: The Royal Society of Chemistry, June 2012.

[86] N. Malod-Dognin and N. Pržulj. L-GRAAL: Lagrangian graphlet-based network aligner. *Bioinformatics*, 31(13):2182–2189, July 2015.

[87] R. Patro and C. Kingsford. Global network alignment using multiscale spectral signatures. *Bioinformatics*, 28(23):3105–3114, December 2012.

[88] M. Sarich, N. D. Conrad, S. Bruckner, T. O. F. Conrad, and C. Schütte. Modularity revisited: A novel dynamics-based concept for decomposing complex networks. *Journal of Computational Dynamics*, 1(1):191, Company: Journal of Computational Dynamics Distributor: Journal of Computational Dynamics Institution: Journal of Computational Dynamics Label: Journal of Computational Dynamics Publisher: American Institute of Mathematical Sciences, 2014.

[89] A. E. Aladağ and C. Erten. SPINAL: scalable protein interaction network alignment. *Bioinformatics*, 29(7):917–924, Number: 7, April 2013.

[90] S. Hashemifar and J. Xu. HubAlign: an accurate and efficient method for global alignment of protein–protein interaction networks. *Bioinformatics*, 30(17):i438–i444, Number: 17, September 2014.

[91] L. Chindelevitch, C.-Y. Ma, C.-S. Liao, and B. Berger. Optimizing a global alignment of protein interaction networks. *Bioinformatics*, 29(21):2765–2773, November 2013.

[92] C. Clark and J. Kalita. A multiobjective memetic algorithm for PPI network alignment. *Bioinformatics*, 31(12):1988–1998, Number: 12, June 2015.

[93] J. Flannick, A. Novak, C. B. Do, B. S. Srinivasan, and S. Batzoglou. Automatic Parameter Learning for Multiple Network Alignment. In M. Vingron and L. Wong, editors, *Research in Computational Molecular Biology*, Lecture Notes in Computer Science, pages 214–231, Berlin, Heidelberg, 2008. Springer. ISBN 978-3-540-78839-3. doi: 10.1007/978-3-540-78839-3_19.

[94] S. M. E. Sahraeian and B.-J. Yoon. SMETANA: Accurate and Scalable Algorithm for Probabilistic Alignment of Large-Scale Biological Networks. *PLOS ONE*, 8(7): e67995, Publisher: Public Library of Science, July 2013.

[95] J. Hu, B. Kehr, and K. Reinert. NetCoffee: a fast and accurate global alignment approach to identify functionally conserved proteins in multiple networks. *Bioinformatics*, 30(4):540–548, February 2014.

[96] F. Alkan and C. Erten. BEAMS: backbone extraction and merge strategy for the global many-to-many alignment of multiple PPI networks. *Bioinformatics*, 30(4): 531–539, February 2014.

[97] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, September 2003.

[98] X. Gandibleux and M. Ehrgott. 1984-2004 – 20 Years of Multiobjective Metaheuristics. But What About the Solution of Combinatorial Problems with Multiple Objectives? In C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, editors, *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, pages 33–46, Berlin, Heidelberg, 2005. Springer. ISBN 978-3-540-31880-4. doi: 10.1007/978-3-540-31880-4_3.

[99] N. Mamano and W. B. Hayes. SANA: simulated annealing far outperforms many other search algorithms for biological network alignment. *Bioinformatics*, 33(14): 2156–2164, July 2017.

[100] V. Saraph and T. Milenković. MAGNA: Maximizing Accuracy in Global Network Alignment. *Bioinformatics*, 30(20):2931–2940, Number: 20, October 2014.

[101] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli. Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing*, 11(6):4135–4151, September 2011.

[102] V.-D. Cung, S. L. Martins, C. C. Ribeiro, and C. Roucairol. Strategies for the Parallel Implementation of Metaheuristics. In C. C. Ribeiro and P. Hansen, editors, *Essays and Surveys in Metaheuristics*, Operations Research/Computer Science Interfaces Series, pages 263–308. Springer US, Boston, MA, 2002. ISBN 978-1-4615-1507-4. doi: 10.1007/978-1-4615-1507-4_13. URL *https://doi.org/10.1007/978-1-4615-1507-4_13*.

[103] E.-G. Talbi, M. Basseur, A. J. Nebro, and E. Alba. Multi-objective optimization using metaheuristics: non-standard algorithms. *International Transactions in Operational Research*, 19(1-2):283–305, January 2012.

[104] E. Alba and M. Tomassini. Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 6(5):443–462, October 2002.

[105] M. Idzik, A. Byrski, W. Turek, and M. Kisiel-Dorohinicki. Asynchronous Actor-Based Approach to Multiobjective Hierarchical Strategy. In V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, and J. Teixeira,

editors, *Computational Science – ICCS 2020*, Lecture Notes in Computer Science, pages 172–185, Cham, 2020. Springer International Publishing. ISBN 978-3-030-50420-5. doi: 10.1007/978-3-030-50420-5_13.

[106] M. Starzec, G. Starzec, A. Byrski, and W. Turek. Distributed ant colony optimization based on actor model. *Parallel Computing*, 90:102573, December 2019.

[107] N. V. Blamah, A. A. Oluyinka, G. Wajiga, and Y. B. Baha. MAPSOFT: A Multi-Agent based Particle Swarm Optimization Framework for Travelling Salesman Problem. *Journal of Intelligent Systems*, 30(1):413–428, Publisher: De Gruyter Section: Journal of Intelligent Systems, January 2021.

[108] D. Krzywicki, W. Turek, A. Byrski, and M. Kisiel-Dorohinicki. Massively concurrent agent-based evolutionary computing. *Journal of Computational Science*, 11:153–162, November 2015.

[109] M. Burgin. Swarm Superintelligence and Actor Systems. *International Journal of Swarm Intelligence and Evolutionary Computation*, 06(03), 2017.

[110] M. Cannataro and P. H. Guzzi. *Data management of protein interaction networks*. Wiley, Oxford, 2011. ISBN 978-0-470-77040-5. OCLC: 751147580.

[111] R. Raveaux, J.-C. Burie, and J.-M. Ogier. A graph matching method and a graph matching distance based on subgraph assignments. *Pattern Recognition Letters*, 31 (5):394–406, April 2010.

[112] L. A. Zager and G. C. Verghese. Graph similarity scoring and matching. *Applied Mathematics Letters*, 21(1):86–94, January 2008.

[113] Y. Tian, R. C. McEachin, C. Santos, D. J. States, and J. M. Patel. SAGA: a subgraph matching tool for biological graphs. *Bioinformatics*, 23(2):232–239, January 2007.

[114] C. Clark and J. Kalita. A comparison of algorithms for the pairwise alignment of biological networks. *Bioinformatics*, 30(16):2351–2359, Number: 16, August 2014.

[115] X. Cao, W. Zhang, and Y. Yu. A Bootstrapping Framework With Interactive Information Modeling for Network Alignment. *IEEE Access*, 6:13685–13696, 2018.

[116] B. Neyshabur, A. Khadem, S. Hashemifar, and S. S. Arab. NETAL: a new graph-based method for global alignment of protein–protein interaction networks. *Bioinformatics*, 29(13):1654–1662, Number: 13, July 2013.

[117] D. Park, R. Singh, M. Baym, C.-S. Liao, and B. Berger. IsoBase: a database of functionally related proteins across PPI networks. *Nucleic Acids Research*, 39(Database): D295–D300, Number: Database, January 2011.

[118] W. K. Kim and E. M. Marcotte. Age-Dependent Evolution of the Yeast Protein Interaction Network Suggests a Limited Role of Gene Duplication and Divergence. *PLOS Computational Biology*, 4(11):e1000232, Publisher: Public Library of Science, November 2008.

[119] S. M. E. Sahraeian and B.-J. Yoon. A Network Synthesis Model for Generating Protein Interaction Network Families. *PLOS ONE*, 7(8):e41474, Publisher: Public Library of Science, August 2012.

[120] Y. Zhu, Y. Li, J. Liu, L. Qin, and J. X. Yu. Discovering large conserved functional components in global network alignment by graph matching. *BMC Genomics*, 19(S7): 670, Number: S7, September 2018.

[121] L. Meng, A. Striegel, and T. Milenković. Local versus global biological network alignment. *Bioinformatics*, 32(20):3155–3164, Number: 20, October 2016.

[122] F. E. Faisal, L. Meng, J. Crawford, and T. Milenković. The post-genomic era of biological network alignment. *EURASIP Journal on Bioinformatics and Systems Biology*, 2015(1):3, Number: 1, December 2015.

[123] S. Maskey and Y.-R. Cho. Survey of biological network alignment: cross-species analysis of conserved systems. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2090–2096, San Diego, CA, USA, November 2019. IEEE. ISBN 978-1-72811-867-3. doi: 10.1109/BIBM47256.2019.8983132. URL *https://ieeexplore.ieee.org/document/8983132/*.

[124] J. Crawford, Y. Sun, and T. Milenković. Fair evaluation of global network aligners. *Algorithms for Molecular Biology*, 10(1):19, Number: 1, December 2015.

[125] Y. Sun, J. Crawford, J. Tang, and T. Milenković. Simultaneous Optimization of both Node and Edge Conservation in Network Alignment via WAVE. In M. Pop and H. Touzet, editors, *Algorithms in Bioinformatics*, Lecture Notes in Computer Science, pages 16–39, Berlin, Heidelberg, 2015. Springer. ISBN 978-3-662-48221-6. doi: 10.1007/978-3-662-48221-6_2.

[126] P. H. Guzzi, M. Mina, C. Guerra, and M. Cannataro. Semantic similarity analysis of protein data: assessment with biological features and issues. *Briefings in Bioinformatics*, 13(5):569–585, September 2012.

[127] C. Pesquita. Semantic Similarity in the Gene Ontology. In C. Dessimoz and N. Škunca, editors, *The Gene Ontology Handbook*, Methods in Molecular Biology, pages 161–173. Springer, New York, NY, 2017. ISBN 978-1-4939-3743-1. doi: 10.1007/978-1-4939-3743-1_12. URL *https://doi.org/10.1007/978-1-4939-3743-1_12*.

[128] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA, August 1995. Morgan Kaufmann Publishers Inc. ISBN 978-1-55860-363-9.

[129] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, September 1997.

[130] C. Camacho, G. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos, K. Bealer, and T. L. Madden. BLAST+: architecture and applications. *BMC Bioinformatics*, 10(1):421, December 2009.

[131] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, Perth, WA, Australia, November 1995. IEEE. ISBN 978-0-7803-2768-9. doi: 10.1109/ICNN.1995.488968.

[132] J. S. Arora. Multiobjective Optimum Design Concepts and Methods. In *Introduction to Optimum Design*, pages 543–563. Elsevier, 2004. ISBN 978-0-12-064155-0. doi: 10.1016/B978-012064155-0/50017-3. URL *https://linkinghub.elsevier.com/retrieve/pii/B9780120641550500173*.

[133] M. Mahmoodabadi, A. A. Safaie, A. Bagheri, and N. Nariman-zadeh. A novel combination of Particle Swarm Optimization and Genetic Algorithm for Pareto optimal design of a five-degree of freedom vehicle vibration model. *Applied Soft Computing*, 13(5):2577–2591, Number: 5, May 2013.

[134] Z. Fan, T. Wang, Z. Cheng, G. Li, and F. Gu. An Improved Multiobjective Particle Swarm Optimization Algorithm Using Minimum Distance of Point to Line. *Shock and Vibration*, 2017:1–16, 2017.

[135] M. K. Gupta. *Akka essentials: a practical, step-by-step guide to learn and build Akka's actor-based, distributed, concurrent, and scalable Java applications*. Community experience distilled. Packt Publ, Birmingham, 2012. ISBN 978-1-84951-828-4. OCLC: 820375063.

[136] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Rydberg, P. Selmer, and A. Taylor. Cypher: An Evolving Query Language for Property Graphs. In *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, pages 1433–1445, New York, NY, USA, May 2018. Association for Computing Machinery. ISBN 978-1-4503-4703-7. doi: 10.1145/3183713.3190657. URL *https://doi.org/10.1145/3183713.3190657*.

[137] I. Robinson, J. Webber, and E. Eifrem. *Graph databases*. O'Reilly, Beijing, second edition edition, 2015. ISBN 978-1-4919-3089-2. OCLC: ocn911172345.

[138] M. Needham and A. E. Hodler. *Graph algorithms: practical examples in Apache Spark and Neo4j*. O'Reilly Media, Sebastopol, California, first edition edition, 2019. ISBN 978-1-4920-4768-1. OCLC: on1066191517.

[139] A. de Bernardi Schneider, C. T. Ford, R. Hostager, J. Williams, M. Cioce, U. V. Çatalyürek, J. O. Wertheim, and D. Janies. StrainHub: a phylogenetic tool to construct pathogen transmission networks. *Bioinformatics*, 36(3):945–947, Number: 3, February 2020.

[140] L. Igual and S. Seguí. *Introduction to Data Science: A Python Approach to Concepts, Techniques and Applications*. Undergraduate Topics in Computer Science. Springer International Publishing : Imprint: Springer, Cham, 1st ed. 2017 edition, 2017. ISBN 978-3-319-50017-1. doi: 10.1007/978-3-319-50017-1.

[141] E. G. Tuncay. Graph Based Methods To Retrieve And Predict Epidemiological Statistics. Research Report, Robert Koch Institute, Berlin, September 2020.

[142] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, March 1955.

[143] E. Kazemi, H. Hassani, M. Grossglauser, and H. Pezeshgi Modarres. PROPER: global protein interaction network alignment through percolation matching. *BMC Bioinformatics*, 17(1):527, Number: 1, December 2016.

[144] V. Vijayan, V. Saraph, and T. Milenković. MAGNA++: Maximizing Accuracy in Global Network Alignment via both node and edge conservation. *Bioinformatics*, 31 (14):2409–2411, Number: 14, July 2015.

[145] H. T. T. Phan and M. J. E. Sternberg. PINALOG: a novel approach to align protein interaction networks—implications for complex detection and function prediction. *Bioinformatics*, 28(9):1239–1245, May 2012.

[146] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, Number: 7043 Publisher: Nature Publishing Group, June 2005.

[147] R. Ibragimov, M. Malek, J. Guo, and J. Baumbach. GEDEVO: An Evolutionary Graph Edit Distance Algorithm for Biological Network Alignment. In T. Beißbarth, M. Kollmar, A. Leha, B. Morgenstern, A.-K. Schultz, S. Waack, and E. Wingender, editors, *GCB 2013*, volume 34 of *OpenAccess Series in Informatics (OASIcs)*, pages 68–79, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-939897-59-0. doi: 10.4230/OASIcs.GCB.2013.68. URL *http://drops.dagstuhl.de/opus/volltexte/2013/4229*. ISSN: 2190-6807 Artwork Size: 12 pages Medium: application/pdf Publisher: Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany.

[148] A. Elmsallati, A. Msalati, and J. Kalita. Index-Based Network Aligner of Protein-Protein Interaction Networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(1):330–336, Conference Name: IEEE/ACM Transactions on Computational Biology and Bioinformatics, January 2018.

[149] A. Elmsallati, S. Roy, and J. K. Kalita. Exploring Symmetric Substructures in Protein Interaction Networks for Pairwise Alignment. In I. Rojas and F. Ortuño, editors, *Bioinformatics and Biomedical Engineering*, Lecture Notes in Computer Science, pages 173–184, Cham, 2017. Springer International Publishing. ISBN 978-3-319-56154-7. doi: 10.1007/978-3-319-56154-7_17.

[150] The UniProt Consortium. UniProt: the universal protein knowledgebase. *Nucleic Acids Research*, 45(D1):D158–D169, January 2017.

[151] R. P. Huntley, T. Sawford, P. Mutowo-Meullenet, A. Shypitsyna, C. Bonilla, M. J. Martin, and C. O'Donovan. The GOA database: Gene Ontology annotation updates for 2015. *Nucleic Acids Research*, 43(D1):D1057–D1063, Number: D1, January 2015.

[152] A. Calderone, L. Castagnoli, and G. Cesareni. mentha: a resource for browsing integrated protein-interaction networks. *Nature Methods*, 10(8):690–691, August 2013.

[153] B. Aranda, H. Blankenburg, S. Kerrien, F. S. L. Brinkman, A. Ceol, E. Chautard, J. M. Dana, J. De Las Rivas, et al. PSICQUIC and PSISCORE: accessing and scoring molecular interactions. *Nature Methods*, 8(7):528–529, Bandiera_abtest: a Cg_type: Nature Research Journals Number: 7 Primary_atype: Correspondence Publisher: Nature Publishing Group Subject_term: Bioinformatics;Proteomics Subject_term_id: bioinformatics;proteomics, July 2011.

[154] N. del Toro, M. Dumousseau, S. Orchard, R. C. Jimenez, E. Galeota, G. Launay, J. Goll, K. Breuer, et al. A new reference implementation of the PSICQUIC web service. *Nucleic Acids Research*, 41(Web Server issue):W601–W606, July 2013.

[155] S. Orchard, S. Kerrien, S. Abbani, B. Aranda, J. Bhate, S. Bidwell, A. Bridge, L. Briganti, et al. Protein interaction data curation: the International Molecular Exchange (IMEx) consortium. *Nature Methods*, 9(4):345–350, April 2012.

[156] E. Triantaphyllou. *Multi-criteria Decision Making Methods: A Comparative Study*, volume 44 of *Applied Optimization*. Springer US, Boston, MA, 2000. ISBN 978-1-4419-4838-0 978-1-4757-3157-6. doi: 10.1007/978-1-4757-3157-6. URL *http://link.springer.com/10.1007/978-1-4757-3157-6*. Series Editors: _:n291.

[157] L. Bennett, B. Melchers, and B. Proppe. Curta: A General-purpose High-Performance Computer at ZEDAT, Freie Universität Berlin. page 5 S., Artwork Size: 5 S. Publisher: Freie Universität Berlin, 2020.

[158] V. Vijayan, D. Critchlow, and T. Milenković. Alignment of dynamic networks. *Bioinformatics*, 33(14):i180–i189, July 2017.

[159] V. Vijayan and T. Milenković. Aligning dynamic networks with DynaWAVE. *Bioinformatics*, 34(10):1795–1798, Number: 10, May 2018.

[160] N. Malod-Dognin, K. Ban, and N. Pržulj. Unified Alignment of Protein-Protein Interaction Networks. *Scientific Reports*, 7(1):953, Number: 1 Publisher: Nature Publishing Group, April 2017.

[161] Neo4j, Inc. Graph management - Neo4j Graph Data Science, https://neo4j.com/docs/graph-data-science/2.0/management-ops/. May 2022.

[162] R. T. Fielding. *Architectural styles and the design of network-based software architectures*. phd, University of California, Irvine, 2000. AAI9980887 ISBN-10: 0599871180.

[163] R. Richards. Representational State Transfer (REST). In R. Richards, editor, *Pro PHP XML and Web Services*, pages 633–672. Apress, Berkeley, CA, 2006. ISBN 978-1-4302-0139-7. doi: 10.1007/978-1-4302-0139-7_17. URL *https://doi.org/10.1007/978-1-4302-0139-7_17*.

[164] J. Xin, C. Afrasiabi, S. Lelong, J. Adesara, G. Tsueng, A. I. Su, and C. Wu. Cross-linking BioThings APIs through JSON-LD to facilitate knowledge exploration. *BMC Bioinformatics*, 19(1):30, December 2018.

[165] M. Zaslavskiy, F. Bach, and J.-P. Vert. Global alignment of protein–protein interaction networks by graph matching methods. *Bioinformatics*, 25(12):i259–1267, June 2009.

[166] R. Sharan, S. Suthram, R. M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R. M. Karp, and T. Ideker. Conserved patterns of protein interaction in multiple species. *Proceedings of the National Academy of Sciences*, 102(6):1974–1979, Publisher: National Academy of Sciences Section: Biological Sciences, February 2005.

[167] B. P. Kelley, R. Sharan, R. M. Karp, T. Sittler, D. E. Root, B. R. Stockwell, and T. Ideker. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proceedings of the National Academy of Sciences*, 100(20):11394–11399, Publisher: Proceedings of the National Academy of Sciences, September 2003.

[168] L. A. Mirny and M. S. Gelfand. Using orthologous and paralogous proteins to identify specificity determining residues. *Genome Biology*, 3(3):preprint0002.1, February 2002.

[169] T. G. Stützle. *Local search algorithms for combinatorial problems: analysis, improvements, and new applications*. Number Bd. 220 in Dissertationen zur künstlichen Intelligenz. Infix, Sankt Augustin, 1999. ISBN 978-3-89601-220-3.

[170] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming. An overview of population-based algorithms for multi-objective optimisation. *International Journal of Systems Science*, 46(9):1572–1599, July 2015.

Zusammenfassung

Die ‘Globale Netzwerkausrichtung’ in Protein-Protein-Interaktionsnetzwerken ist ein NP-vollständiges Problem aufgrund der widersprüchlichen Natur der biologischen und topologischen Ausrichtungsziele. Es wurden bereits mehrere Aligner entwickelt, die sich auf verschiedene Prioritäten und Ziele des Problems konzentrieren. Keine dieser Alignment-Heuristiken liefert jedoch exakte Lösungen, obwohl sie die Problemziele bis zu einem gewissen Grad erreichen. Aus diesem Grund ist die Forschungsfrage, wie man stärkere Aspekte von unterschiedlichen Network Alignment Heuristiken vereinen kann, sehr vielversprechend. In dieser Arbeit wird das Ziel verfolgt, die Methoden zum Durchsuchen des Suchraumes dieses Problems zu verbessern, indem die gleichzeitige Verwendung mehrerer Heuristiken verwaltet wird, und zu diesem Zweck werden zwei neuartige populationsbasierte meta-heuristische Methoden vorgeschlagen.

Bei der ersten dieser Methoden (SUMONA) handelt es sich um einen überwachten genetischen Algorithmus, der eine Erweiterung des rechenintensiven memetischen Multi-Zielsetzung algorithmus OptNetAlign darstellt. Diese Methode zielt darauf ab, den Ausrichtungsprozess zu beschleunigen und zu leiten, indem der Crossing-Over-Mechanismus des genetischen Algorithmus mit Eingaben von anderen Alignern/Heuristiken modifiziert wird. Der Algorithmus basiert auf einer generischen Prozedur, die mehrere Alignments mit wechselnden Heuristiken und Eingabeparametern generiert, die generierten Alignments klassifiziert, einen randomisierten Alignment-Auswahlmechanismus aus den klassifizierten Alignments für das Cross-over etabliert und schließlich globale und lokale Suchparameter anpasst. Mit dieser Methode ist es möglich, eine bessere Laufzeitleistung zu erzielen, bestimmte Ziele gegenüber anderen zu priorisieren und auch die sekundären Ziele zu optimieren.

Die zweite Methode (PERSONA) ist ein von einem Partikelschwarm inspirierter kollaborativer Ansatz, der mehrere Aligner so orchestriert, dass sie ihre Teillösungen kontinuierlich austauschen, während sie Fortschritte machen. Diese Aligner bilden gemeinsam einen Partikelschwarm, der in einer reaktiven Akteur-Umgebung nach multiobjectiven Lösungen für das Alignment-Problem sucht. Innerhalb des Schwarms senden die führenden oder prominenten Akteure den stärkeren Teil ihrer Lösung als Teilgraphen an andere Akteure und erhalten nach Auswertung dieser Teillösungen die stärkeren Teilgraphen der Gegenpartei zurück. Die individuellen Alignment-Heuristiken wurden ebenfalls im Rahmen derselben Forschung entwickelt und auf der Grundlage von Alternativen wie Seed-and-Extend-Ansätzen mit verschiedenen Zentralitäts- und Sequenz-Seeds, Cluster-Mapping-Ansatz und Knotenähnlichkeitspriorisierung implementiert. Sowohl die populationsbasierten meta-heuristischen Aufgaben als auch die individuellen heuristischen Aufgaben wurden in einer nicht-deterministischen Weise implementiert, um die Flexibilität zu verbessern und zu verhindern, dass man in lokal optimalen Lösungen gefangen ist. Die mit dieser Methode erzielten Ergebnisse sind sowohl für topologische als auch für Knotenähnlichkeitsziele bemerkenswert optimiert und ausgewogen.

Selbstständigkeitserklärung

Name: TUNCAY
Vorname: Erhun Giray
Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Dissertation selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht. Diese Dissertation wurde in gleicher oder ähnlicher Form noch in keinem früheren Promotionsverfahren eingereicht.

Mit einer Prüfung meiner Arbeit durch ein Plagiatsprüfungsprogramm erkläre ich mich einverstanden.

Datum: 25. Mai 2022 Unterschrift: ______________________