Improved Inference of Ecological Interaction Types

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Syed Umair Aziz

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Or

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9
Canada

Abstract

Inference of microbial interaction types allows us to understand the growth and development of microbial life forms found on earth. Numerous methods have been proposed to infer the interaction type(s) of microbes in a microbial communities using a population dynamics model. However, due to dynamic behaviour of microbial communities, these methods can result in erroneous inferences. A method proposed by Xiao et al. in 2017 models the dynamic behaviour of microbial community using sample abundance data overcomes many of these issues, but suffers from a high failure rate of inference, lower confidence on inferred interactions and slower execution speed than the existing algorithms. In this thesis, we propose an improved and more efficient and effective approach to infer the microbial interaction types of larger microbial communities ($N > 10$). Our findings demonstrate that our approach is faster, more fault tolerant, more scalable than the state of the art from 2017, and it has the ability to infer microbial interactions with increased confidence.

Acknowledgements

I am immensely thankful to my esteemed supervisor Dr. Kevin Stanley for his insightful supervision and outstanding support throughout the entire academic program. I could not have imagined addressing such a challenging research problem without his ever-encouraging technical supervision. I am also grateful to our co-researchers Dr. Steven Siciliano and Dr. Steven Mamet for their overwhelming support and encouragement along with P^2IRC for funding this research.

Lastly, I am utterly thankful to my parents, siblings and friends for their everlasting love and support, without which I would not have been able to get through this academic endeavour.

Contents

List of Tables

List of Figures

Glossary

The table below lists all the important terms used in the context of this paper:

Differential Hyperplane (Hyperplane)	A column vector calculated by taking a pairwise difference of active (non-zero) abundance samples.
Differential Hyperplane Matrix (Pair Difference)	A matrix containing all differential hyperplanes with respect to a taxon, represented by M_i.
Perturbation	Randomly changing an index of a proposed sign pattern with the known interaction types.
Sign Pattern	An inference of interaction types for each taxa with respect to a focal taxon. Each index represents an inferred interaction type.
Interaction Type(s)	The three possible types of association among a pair of taxa. It can either be promotion (+1), inhibition (-1) or unknown (0)
N	Number of taxa under consideration
Phi (ϕ)	A quantification of proposed sign pattern by taking a ratio of the total number of intersected hyperplanes with the total number of hyperplanes present in M_i

1 Introduction

A group of microorganisms coexisting in a common space are referred as microbial community [1]. Microbial communities are present in almost all ecosystems including marine, soil, plants and animals. Inference of interaction types between the microbes in a community is a stepping stone towards the understanding of growth and development of their ecological community. Understanding the behaviour of a microbial community such as the ones present in human gut or soil can lead towards building models that can capture the effect of external stimuli applied to the community, which in turn can help in the formulation of life saving drugs or crop enhancing treatments. A comprehensive understanding of intra-community microbial interaction is necessary for achieving this goal. Forecasting the influence of microbes present in a community requires labelling of inter-microbial interaction.

Several approaches have been developed for the inference of microbial interaction. Understanding the dynamic behaviour of microbial communities requires an approach which can evolve with its changing behavior. Microorganisms coexisting in the same ecological space are referred to as a microbial community [1]. In the human gut, they regulate the system of digestion, in soil they impact the growth and development of plants [2], and microbial communities present in water impact marine life [3]. The microbial communities found in any life form have significance influence over the health, growth and behaviour of that organism [4]. Research has been conduced to analyze the host-associated microbial communities present in human body [5].

While predicting and understanding the behaviour of microbial communities is critical for the advancement of our understanding about ecosystems, it is also a complex task, particularly because of the dynamic nature of interactions [6]. Numerous methods have been proposed to infer the structure of a microbial community. Authors of the method proposed in [7] use correlation to infer the microbial interactions for microbes in a microbial community. Correlation-based analysis may suffer from erroneous prediction as the proposed inferences may not be causal, particularly when the community behaviour is highly dynamic [8]. Another approach proposed in the research [9] uses maximum likelihood and Bayesian inference to forecast the dynamics of the microbial community, which tries to address the problem of causality. Several other approaches use an extension of the Generalized Lotka–Volterra model to model the behaviour of microbes [10] [11] [12]. However, previously proposed approaches suffer from one of two shortcomings: 1) the proposed models for the inference of microbial interactions suffer from validity of the inference, meaning that the interaction lacks a causal basis, or 2) the inference methods assume an underlying model of population growth, which may not hold in all circumstances [13]. To overcome these shortcomings, the authors of the research [13] use independent steady-state samples of a microbial community for the inference of microbial interactions

without the assumption of a specific population dynamics model. The method works on the extension of a simple assumption that "*the net ecological impact of species on each other is context-independent, then comparing equilibria (i.e. steady-state samples) consisting of different subsets of species would allow us to infer the interaction types*" [13]. Using this assumption, this method solves a series of differential equations in the form of a Jacobian matrix, which represents the change over time in abundance of each microbial taxon with respect to every other taxa in a given microbial community. The method in [13] incrementally infers the interaction types of each focal taxon by calculating the vector corresponding to that taxon from the Jacobian matrix, whereas the sign of each entry in that matrix represents an interaction type of the focal taxon against every other taxa in the microbial community under consideration. For small N, the method proposed in [13] used a brute-force approach to rigorously check all possible 3^N signs with respect to each focal taxon. For larger N, they proposed a heuristic approach which draws sign patterns with respect to a focal taxon based on the data, effectively reducing the size of search space. The inferences obtained from heuristic method reduce the size of the search space, but may not be able to infer the optimal sign pattern.

The authors of the approach proposed in the research [13] provide a solution to infer the interaction types of a microbial community with limited prior assumptions. The method abstracts the problem into solving a system of partial differential equations in the form of a Jacobian matrix. The sign of each element in the Jacobian matrix can be positive, negative, or zero, representing the dynamics of microbial community interaction. A positive sign represents promotion, a negative sign represents inhibition and zero represents an undetermined relation between interacting microbial taxa. The interaction sign for a taxon with all other taxa is called a sign pattern.

The problem of interaction type inference grows exponentially with increasing numbers of taxa. Each taxon can have positive, negative or zero effect on every other taxon present in the community. The interactions amongst taxa can be represented as edges in an interaction graph where each taxon is represented as a vertex. The interaction graph shows the impact of each taxon on every other taxa present in the community. Capturing the dynamics of microbial community has an exponential growth rate of 3^N per taxon, where N represents number of taxa, imposing a significant computational challenge and requiring computational power beyond availability for a brute-force search of even a moderately sized microbial community. To overcome such computational challenges, authors of [13] proposed a heuristic based approach to effectively reduce the size of search space. However, good heuristic design is a challenging problem in itself. The approach proposed in the research [13] addresses this issue by randomly drawing potential solutions from the data.

The proposed heuristic and brute-force algorithms can be improved in terms of reproducibility, throughput and speed of execution. We systematically analyzed both the heuristic and brute-force methods and addressed the performance based shortcomings using a modular approach. Our primary goal was to improve the throughput of the algorithm by increasing its execution speed, allowing for exhaustive solutions for larger N. The authors of the algorithm proposed in the research [13] devised a solution which is sequential in nature and does not exploit the parallelism present in the problem. We also improved the computation of the quality

score (ϕ) [13], resulting in significantly faster computation of quality metrics for each proposed solution. Our improved implementation allows the execution of a brute-force algorithm on a larger community size. We introduced a block-wise brute-force approximation which can perform a greedy approximation of the brute-force approach on even larger communities ($N \geq 19$) by individually processing blocks of ecologically coherent taxa independently. We also improved the quality of the proposed heuristic by introducing an iterative approach which randomly perturbs a sign pattern for re-evaluation. Because the data under consideration is not noise free as assumed in the research [13], their algorithm may fail to find a valid solution. We introduce a hybrid approach which addresses the frequent failure of their heuristic on noisy data by generating a sign pattern using brute-force until a sufficiently good pattern is found. To address the variance in solution provided by the heuristic from the research [13], we propose a frequency-based voting method which results in significantly lower variance, resulting in more consistent sign patterns over several independent iterations of the algorithm. Each enhancement made to the existing approach proposed by authors in [13] is briefly outlined below.

Accelerated Brute Force Implementation: The enhanced brute-force approach offers two major improvements. It uses a parallel pipeline for the processing of all sign patterns. Our brute-force method does not fail upon the unavailability of a perfect solution; instead it finds the best available solution from the highest obtainable quality measure.

Variable Block Size Brute Force Implementation: As the name suggests, this implementation performs a brute-force solution search on a subset or block of taxa. The number of taxa in each block can be specified by the user. Inference with respect to a taxon is made by accumulating the brute-force inferences provided by each block. As we find the brute-force solution for each block individually, this approach assumes block-wise independence of taxa and provides an incremental and greedy approximation for larger microbial communities.

Hybrid, Robust, Perturbed and Performance Improved Implementation: Due to the exponential growth of the solution space with an increasing number of taxa, for large community sizes ($N \geq 30$), a heuristic method is used to infer a possible solution. To improve the performance of heuristic inference and evaluation, we used a parallel pipeline for the processing of available taxa. To find a better sign pattern close to the one with highest achievable quality, random perturbations on inferred heuristics are performed via random walk. The process of perturbation is repeated until the explored heuristic has a lower quality than its predecessor. This improves the quality of an inferred sign pattern by exploring the local solution space of each proposed sign pattern. The heuristic method may fail to infer a solution if the underlying data contains significant noise. To overcome the frequent failure of sign pattern inference, we introduced a hybrid bypass in the pipeline which is the conjuncture of both the brute-force and heuristic methods. The hybrid method uses the brute-force method to generate a sign pattern for a single taxa when the heuristic module fails to propose a suitable sign pattern. Using this approach, the failure rate of the heuristic module can be significantly reduced for moderately sized ecosystems.

To reduce the variance in inferred interaction types over multiple independent iterations, a frequency based voting method is introduced. This method uses a weighted frequency-based voting system for inferred sign patterns that pass a user specified threshold of quality. The frequency value of the focal taxon for available interaction types (positive, negative or zero) is calculated with respect to all non-focal taxa present in the microbial community, and the most frequently interaction type is considered as the final interaction type. Using this process, we are able to reduce the variance of inference over multiple independent iterations of inference depending upon a user specified threshold.

Algorithm Evaluation: In order to assess the different enhancements, we employed multiple evaluation strategies focusing on individual areas of improvement and compared them against the approach proposed by the authors of [13]. We compared the inference of heuristic sign patterns and their quality against the Hybrid, Robust, and Perturbed approaches. To quantify the improvement in the inference of solution, we compared the algorithm's throughput, failure rate and average sign pattern quality on different samples/taxon ratios on both simulated and field data. To evaluate the brute-force approach, we executed both block-wise and accelerated brute-force implementations on larger community sizes (i.e. N$\geq$11) and compared its failure rate against the brute-force method proposed in the research [13]. We also compared the the quality of sign patterns obtained from the block-wise brute-force method with heuristic method to analyze the improvement in inference quality.

Author Contributions: This research was conduced under the supervision of Kevin Stanley with Steven Siciliano and Steven Mamet as co-authors from the Department of Soil Sciences. Steven Siciliano and Steven Mamet contributed towards gathering of the *Brassica napus* dataset and they provided us with an understanding of microbial interactions and conditioning for algorithm design. Kevin Stanley supervised this entire research by contributing towards the development, design and improvement of the algorithm proposed by the authors of [13]. We are also thankful to P^2IRC for providing us with the research funding and resources to conduct this research.

2 Background

The authors of [13] follow a propose and test methodology to find the interaction types of taxa. A solution can be proposed either via brute-force or heuristic methods. The brute-force method has limited scalability due the exponential growth of search space, meaning that it can only be applied to smaller microbial communities with fewer than 11 taxa for most practical computing systems. On the other hand, the heuristic method is scalable to larger communities as it stochastically proposes sign patterns extrapolated from the underlining data. To quantify the magnitude of change in the abundance of a taxon with respect to every other taxa, a pair wise difference of active abundance samples is taken with respect to each taxa which is known as a differential hyperplane. The quality of each proposed solution is assessed by computing the number of data points that support the proposed sign pattern. The heuristic method uses the underlying abundance data to propose a sign pattern which is then evaluated. It works on the assumption that the possible solution lies within the data. It randomly samples $N-1$ hyperplanes and finds a region of intersection of those hyperplanes which serves as a heuristic. The inference of a solution through the heuristic depends upon the data being noise free which increases the probability that the proposed sign pattern is true, which is rarely the case. Using the underlying data for the inference of solution reduces the search space from 3^N per taxon to a user specified threshold, which accelerates the process of sign pattern inference. The heuristic inference can fail if $N-1$ independent plane do not exist for a given taxon.

2.1 Assumptions

In order to infer the interactions of a microbial community, a number of assumptions were made in the algorithm [13].

1. For both heuristic and brute-force solutions, the underlying abundance data represents a steady state of the sign of Jacobian; that is, the true sign pattern is in the provided data.
2. The steady state abundance data has little to no noise.
3. For the heuristic solution, randomly sampled hyperplanes will infer unique sign patterns.
4. For the heuristic solution, independent inference iterations will be consistent.
5. For the brute-force solution, there will always be a sign pattern amongst 3^N which can intersect all hyperplanes in M_i. Where M_i represents a differential hyperplane matrix.
6. Each taxon will have exactly one solution having $\phi = 1.0$.

2.2 Sign Satisfaction

A pairwise difference of active (non-zero) abundance samples are taken with respect to each taxon to quantify the extent of change caused by a focal taxon. A hyperplane is represented in N-dimensional space and signifies the change in sample abundances with respect to a focal taxon against every other taxa present in the underlying microbial community. With respect to each taxon, multiple hyperplanes are computed depending upon the number of active abundance samples. After the computation of hyperplanes, a sign satisfaction graph with respect to each hyperplane is created in order to assess the quality of an inferred sign pattern. A sign satisfaction graph is a product of element-wise signs of a hyperplane and the inferred sign pattern [13]. The element-wise product of hyperplane and the inferred sign pattern is taken in order to identify if the inferred sign pattern satisfies the changes induced by a focal taxa at a particular time. An inferred sign pattern is considered to have intersected a hyperplane if its sign satisfaction graph for that hyperplane has at least one pair of opposite signs (positive and negative) or it is entirely zero. This measure flows from the assumption that the ecosystem is in a steady state. An entirely positive sign pattern would indicate a taxa growing without bound. An entirely negative sign pattern heralds that the taxa is headed towards extinction.

2.3 Quality Metric

The quality of inferred solution is assessed by computing its intersection with all the available hyperplanes. This quality measure is referred as ϕ and represents the support for the proposed sign pattern. The ϕ measure results in a value ranging between 0 to 1. The value of $\phi = 1$ represents a solution which intersects all the pair difference hyperplanes and is therefore supported by all available data. The ϕ value accurately represents the quality of an inferred sign pattern because it encodes the consistency of each proposed interaction pattern with respect to all evidence.

The ϕ measure accounts for the number of pair difference hyperplanes being intersected by the inferred sign pattern out of the total available hyperplanes, and it shows the consistency of sign pattern with the available changes in abundances with respect to every other taxa in the community. Each sample pair difference is an N dimensional hyperplane as it represents N different interaction types for each taxa with respect to the focal taxon. Each hyperplane is computed by taking a pairwise sample difference and a sign pattern intersecting a hyperplane represents its consistency with the variation in abundance caused by the taxon under consideration.

The brute-force algorithm proposed by the authors of [13] iteratively checks all possible 3^N sign patterns per taxon and finds the sign pattern which intersects all available differential hyperplanes ($\phi = 1$). The sign pattern which intersects all the hyperplanes is considered as the sign pattern with respect to a focal taxon. The step to find the brute-force sign pattern is repeated for each microbial taxon present in the community

sequentially and the sign pattern for each focal taxon is combined to form a matrix encoding the interaction types of microbial community taxa.

The heuristic method proposed in the research [13] has a similar mechanism to the brute-force. Instead of calculating the ϕ quality measure for all 3^N sign patterns per taxon, it proposes a sign pattern extrapolated from the underlying data points to limit the size of search space. The sign pattern with highest value of ϕ is taken as the sign pattern for the focal taxon.

2.4 Brute Force Approach

The brute-force approach checks all $N3^N$ sign patterns for a given set of taxa. This method computes the differential hyperplanes with respect to a focal taxon first, and then iteratively infers and tests all possible 3^N sign patterns per taxon by computing their ϕ score using the pair difference hyperplanes. The sign pattern intersecting all the available hyperplanes is considered as the inference pattern for the focal taxon. The same step is repeated for each taxon to obtain the inference pattern of N taxa. The step-wise description of the brute-force inference method for each taxon as proposed by the authors of [13] is as follows:

Algorithm 1 Brute Force Algorithm

1. Initialize S_i as an empty set.
2. Generate all possible 3^N sign patterns E_i.
3. Compute the differential hyperplane matrix (M_i) from the steady state abundance samples with respect to the ith taxon.
4. For jth column in M_i, check if it intersects any/all sign patterns in E_i. Add all the sign patterns intersecting the jth hyperplane to set S_j.
5. $S_i = S_i \cap S_j$
6. Repeat steps 4 and 5 for each jth hyperplane in M_i
7. Brute force sign pattern for ith taxon is present in S_i

2.5 Step-wise Illustration of the Brute Force Approach

Fig. 2.1 shows the step-wise description of the Brute Force Algorithm. The first step of this algorithm is to compute the pairwise difference matrix, using the active abundance samples of a taxon. Once the differential hyperplane matrix (M_i) has been computed, a sign pattern is generated out of the 3^N available sign patterns per taxon as shown in the step-2. A sign satisfaction graph is constructed with respect to the generated sign pattern and each of the available hyperplane in M_i as shown in the step-3. The quality of the sign pattern is assessed by counting the number of hyperplanes intersected, satisfying the sign satisfaction graph. In the Fig. 2.1, the inferred sign pattern satisfies 2/3 hyperplanes giving it the quality score of 0.66 or 66% as shown in the step-4. Steps 2 to 4 are repeated for all 3^N sign patterns per taxon and the sign pattern satisfying all

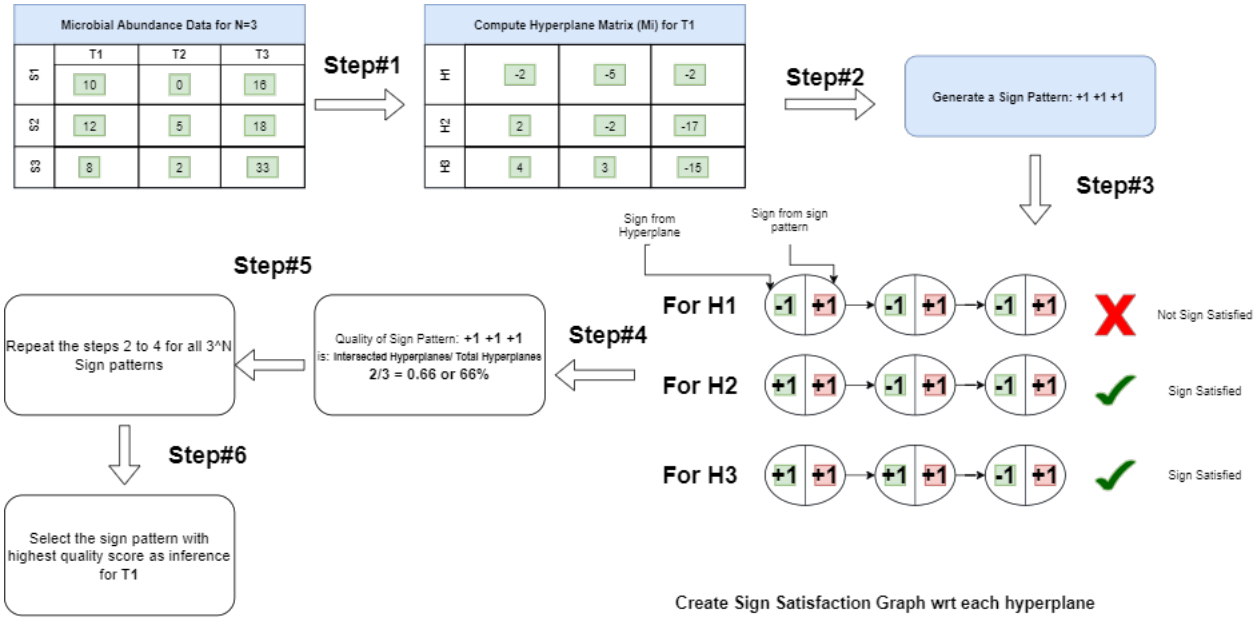


Figure 2.1: Step-wise Illustration of the Brute Force Algorithm

the hyperplanes present in M_i is considered as the final sign pattern for the taxon. All the steps mentioned in Fig. 2.1 are repeated for each taxon present in the microbial community to find sign pattern with respect to each taxon.

2.6 Heuristic Approach

Due to the exponential growth of the solution size, another approach was proposed by the authors of [13] which addresses the problem of sign pattern inference using a heuristic based solution rather than checking all possible sign patterns. This approach employs the data itself to find a viable sign pattern. The heuristic approach checks a user-specified subset of possible sign patterns. It randomly selects $N-1$ hyperplanes from the available differential hyperplanes and finds the intersection of those hyperplanes which then serves as the proposed sign pattern. The step for the inferring and evaluating a solution is repeated as many times as specified by a user defined threshold and the most effective sign pattern (the one with most intersections) is chosen as the final inference for that taxon. This step is repeated for each taxa to get the inference of interaction types (sign pattern). The purpose of using a user specified number of sign pattern(s) is to reduce the $N3^N$ sign pattern search space. The inference of heuristic is based on the assumption that the underlying data is noise free. The step-wise description of the heuristic method as proposed by the authors of [13] is as follows:

Algorithm 2 Heuristic Algorithm

1. Compute the differential hyperplane matrix (M_i) from the steady state abundance samples with respect to ith taxon .
2. Randomly select $N-1$ hyperplanes from M_i.
3. Find a region given by the intersection of $N-1$ hyperplanes and consider it a heuristic H_i.
4. Find the quality measure (ϕ) of heuristic H_i
5. Repeat steps 2 to 4 as many times as specified by the user.
6. The sign pattern for ith taxon is the one with highest value for the ϕ measure.

2.7 Shortcomings in the Existing Approach

The shortcomings in the proposed algorithm that we addressed in our research are highlighted below:

- **Serial Implementation:** The solution as presented is designed to work sequentially, potentially utilizing only a fraction of any available parallelism which limits its throughput as inference with respect to each taxon is independent of the processing of other taxa.
- **Random Draws Lead to Low Consistency:** Multiple iterations of inference of interaction types using the same data and parameters result in different interaction types when using noisy data.
- **Limits on Number of Taxa:** The brute-force solution can not be used for N >10 due to the exponential growth of solution space and inefficiency of implementation.
- **Null Results:** The heuristic approach may not return a solution for a particular taxon due the presence of noise in the data.
- **Brute Force Failure:** The brute-force implementation may fail to infer a solution for a taxon for which a perfect solution ($\phi = 1.0$) does not exist due to noisy data.
- **Heuristic Failure:** The inference of heuristic requires a random selection of $N-1$ hyperplanes, which is not always possible. In order to have $N-1$ unique columns in the differential hyperplane matrix, the count of pair wise combination of active(non-zero) abundance samples for a particular taxon must be greater than the total number of taxa present in the microbial community.
- **Running out of data:** If the number of differential hyperplanes narrowly exceed $N-1$, the heuristic inference may run out of multiple unique combinations of samples, resulting in failure of execution in the subsequent iterations of heuristic inference. This issue is caused by lack of active abundance samples in the data, when "active abundance sample" refers to a sample of abundance having a non zero value for the focal taxon.
- **Unusable Heuristic:** It is possible that all sign patterns inferred by heuristic may result in ϕ with unacceptably small values, indicating only non-viable solutions have been found.

3 Literature Review

The inference of interaction type(s) between the microbes of a microbial community is similar to inferring the missing links in a partially complete social media network. The inference of interaction type involves the prediction of new links (interaction types) using the underlying abundance data; while in the area of social networking the inference of interactions is referred to as link prediction.

The area of network analysis using link prediction has been studied over the past two decades. The task of link prediction or signed linked prediction refers to the inference of an association which is represented by an edge in the graph. An edge signifies trust/distrust or like/dislike between people/items of interest, which are represented as nodes in the network. The association between two nodes can either be signed (positive or negative) or unsigned and the methods used to infer these signed/unsigned associations can be classified into two categories, namely: supervised and unsupervised link prediction [14].

3.1 Supervised Link Prediction

The task of supervised link prediction involves prediction of a positive or negative link based on the features learned from labelled dataset. A labelled dataset is a verified collection of feature values and their respective prediction or responses, the prediction may represent an unsigned or a signed edge depending upon the type of network under consideration [15]. The supervised approach of inference has been widely used in the area of social networking, using labelled data to train a classifier or model to extend the observed behaviour from the observed to the unobserved data points. Knowledge from several social interaction theories have been used to understand the underlying training dataset (a subset of labelled dataset). In 1946, a theory proposed by the authors of [16], which is also known as the balance theory, explains vital aspects influencing the attitude of individuals and their perception about an event. The baseline knowledge provided by balance theory has been used and enhanced in several upcoming studies like the ones mentioned here [17] [18]. Inspired by real-world events and interactions, those studies established the baseline for social interactions which can be incorporated in the model training for improved performance. With the rise of social media, increased attention has been paid to the behaviour of people interacting in an online world. In the context of social media, a signed network consists of nodes representing the people while a positive or negative edge among the nodes represent a possible link (association). To understand the types of possible associations, a trust metric was introduced to categorize an association as positive or negative which may represent trust/similarity or distrust/dissimilarity respectively [19] [20] [21] [22]. The exploration of unsigned networks has become

prevalent in the area of link prediction, where each edge represents a link which does not have a positive or negative type, making it convenient for computational analysis [23] [24]. Unsigned link prediction techniques are used to model the problems where the association is type-less, for example, to model the problem of voting, a voter might vote a candidate as positive or negative but, it is likely that a user might vote neutral [25]. Signed networks are designed to capture the positive and negative associations, where a positive or negative link between the nodes represent the two opposite sides of an association. Mining networks with both positive and negative links is different from mining an unsigned network, as the underlying assumption associated with positive link can not be simply extended to negative links [26] [27]. Moreover, most social media platforms do not provide an explicit way to infer a negative link, making it an area which remained unexplored [28] [29]. The volume of data lends itself to machine learning, data mining and spectral analysis-based techniques for the inference of interaction types (association/link) [30] [31] [32]. Several methods have been proposed to identify noise in labelled data to enhance the accuracy of classifiers by removing the spurious information [33] [34] [35]. The data with reduced noise can then be used for classifier training, increasing its accuracy by selecting only the most relevant chunk of data for training.

Sign prediction has also been used as an approach to infer the type of link between the nodes as shown in this research [14]. Sign prediction refers to the task of sign inference, either positive or negative, on existing unsigned links. The method proposed by the authors of [31] uses the knowledge derived from balance and status theory [32] [36] to infer the sign of a link in a network. Another method proposed by the authors of [37] employs the quantitative measures of social imbalance for sign prediction. Similarly, another research [38] provides an improvement over the PageRank algorithm [39] by integrating the local bias of nodes into the sign prediction of edges. Unlike traditional approaches which use only the topological information of the entire network to infer the links, the approach proposed in the research [40] uses C4.5, a classification algorithm, to train a model using the pattern present in a user's network, then it uses that model to predict possible links making it an effective recommendation system for the user. A probabilistic interpretation based method proposed in research [41] uses user interaction features to infer the new links proved to be of high significance, as it shows that the features provided by node attributes are not as significant as the interaction features [14].

A fairly recent paradigm of link prediction approaches use maximum likelihood to infer the associations between the nodes in a network. Most of these methods assume a network structure and the maximum prior probabilities to obtain the particular parameters required for posterior inference. After the computation of probabilistic parameters, it extends that model to perform the link prediction [42]. A hierarchical graph, which is an extension of maximum likelihood method, can also be used for the task of missing link prediction for the nodes where information is missing [43]. Another study [44], explains how networks exhibit a hierarchical structure where nodes can be subdivided into groups. It uses the information of network subgroups to infer the missing connections with high accuracy in a partly known network.

As the problem of link prediction becomes complex with increasing volume of data, more sophisticated

and complex classifiers were needed to learn and understand the intricate details present in the data. Neural networks have multiple layers of feature extraction and learning, they can learn as much information from the data as possible in the form of network weights, while generalizing it at the same time for unobserved data points. A network for link prediction comprised of three-layer back propagation network, is used to predict links as a multi-class classification problem, in a heterogeneous network [45]. Another approach [46], outperformed its counterparts, which employs complex compositional embedding to handle binary relations. As the heuristic used for the task of link prediction may limit the neural network's ability to converge, the method proposed in [47] provides a new heuristic learning approach for improved performance.

3.2 Unsupervised Link Prediction

As the name suggests, the unsupervised link prediction algorithms infer the links between nodes without using any prior information to train the prediction classifier. Unsupervised link prediction methods use topological information from signed network or node attributes to find a degree of coherence between the network nodes. Unsupervised learning methods also use clustering based on attributes to identify similar subgroups within the solution space, the clustered groups are then used to infer the type for unobserved data points [14].

Most recommendation systems in the online world use an unsupervised link prediction method to make targeted recommendations for new users due to the limited availability of customer centric data. Similarity based measures are widely used in such recommendation systems [42], which work by finding similarities between the different nodes of a network based on a metric that measures the extent of it. Similarity between two nodes can be defined as the overlap between the attributes of nodes [48], the percentage of overlap and its significance is characterized by a similarity metric which shows the relevance of each node in a network. Once the similarity metric is defined, it is used to predict positive/negative or undefined link type. A method [49] uses local optimization partitioning procedure to identify consistent cluster in digraphs with least degree of error. The identified clusters (subgroups) can then be used with a collaborative filtering method as suggested in research [50] to perform the task of link prediction. As the inference of link relies heavily on minimal clustering error, the methods proposed in the research [51] employ the idea of social balance [52] to increase the intra-cluster similarity, while maintaining lower inter-cluster similarity in a signed network. The approach proposed by the authors of [52] uses a local balance index as a measure of clustering quality, which improves clustering quality and link prediction. Another study [53] employs link prediction to enhance the performance of a recommendation system and shows promising results compared to the method using only collaborative filtering.

Propagation based methods are widely used for trust/distrust based sign networks. As the adjacency matrix for trust based sign networks is usually sparse, the idea of propagation based method is to compute a matrix after performing the required propagation operations on the original adjacency matrix and then use it for the link prediction [14]. There are four different propagation operations: *direct propagation, trust coupling,*

co-citation and transpose trust [14]. There are two different propagation techniques which can be used to perform propagation based analysis. *One step distrust propagation* is used to study distrust by propagating multiple steps of trust and then one step of distrust, which shows improved performance over *multi step distrust propagation* where trust and distrust propagate together [14] [36]. The approach proposed by the authors of [54] provides a novel and simpler computation method for the computation of a sub-group while extending the Appleseed nucleus to integrate distrust in the trust metric. Not all interactions exhibit strict binary relation, while in most cases, the association between two nodes can be classified as either positive or negative, in some cases, the presence of association might be vague or non-existent. In such cases, this approach [55] can be used, where it handles the fuzzy association as a multi-class classification problem by using multi-valued representation, making it easier to categorize fuzzy and non-fuzzy associations. A similar problem has been addressed in the research [56] where partial trust/distrust and ignorance information are preserved by deriving the trust from a bilattice.

Various methods have been proposed to efficiently solve the problem of link prediction, by inferring the friend/foe or trust/distrust relationship in a signed network. The study [57] shows the inherit nature of weak structural balance leading towards the global low rank model for the signed network. A weak balance is defined as "*A complete signed network is weakly balanced iff there is no triad in the network that contains two positive edges and one negative edge*" [58][57]. Using this approach [57], the link prediction problem can be solved as a low-rank matrix completion problem. Another approach proposed by the authors of [59] addressed the problem of link labelling by contemplating it as a sign prediction problem for an unsigned link and solving it as a matrix completion problem on a partially observed matrix. To efficiently deal with the larger size of matrices computed during the inference of a link/sign, this method uses a matrix factorization. Another method [60], motivated by singular value thresholding, uses a low-rank tensor based model for representation of adjacency matrix to solve sign inference efficiently.

Correlation refers to a statistical relationship between two random variables which may or may not be causal [61]. The problem of link prediction has also been addressed using statistical correlation by exploiting the similarity between different nodes of signed network. Inference made through correlation could be a result of both direct or indirect affect of one node on another present in a signed graph. This method [62] uses the core properties of dynamic correlation and matrix transformation to discard the indirect or transitive effects of correlation and supports only the direct causal interactions. Other research [63] explores the correlation between the mobility pattern and its impact on the online presence on social media, the correlation between three variable i.e. co-location, social media connectedness and cellular interaction was studied. It was found that co-location (being nearby) strongly correlates with the closeness on social media which shows how real-world correlation based proximity analysis can be used to infer a possible link in the online world.

3.3 Microbiome Graph Building

Microbial communities exhibit a complex and dynamic behaviour. Understanding the interactions amongst the microbes requires the behavioural analysis of associated hosts in a community, which can be effectively studied by network theory as it provides a holistic view of community interactions [64]. Network analysis allows better understanding of the role of microbiota in disease development and growth by using the structural features of graph which remain universal across the complex systems [64]. Inference of links between taxa based on environmental or any other factor is not the only area where the network analysis can be used. Moreover, networks analysis allows for the identification of keystone taxa and community configuration [65]. Different approaches are used to model a microbial community, depending upon approach's efficiency, accuracy and computational complexity [64].

Distance based measures (similarity or dissimilarity) are used to create microbial interaction network [64]. Measures like mutual information or Kullback–Leibler are used to build an interaction network based on the similarity/dissimilarity between taxa [66]. Correlation based networks are also used to create interaction network but it may suffer from causality of inference resulting in spurious inferences [64]. A study in research [67] uses correlation to study the entropy and its relation with functional aspects of microbial communities in human gut [67] [64]. Similarly regression and probabilistic (Bayesian) methods of network analysis are also used to study the even complex poly-microbial or high dimensional interactions, but these measures may suffer from over-fitting, or require an assumption of conditional independence between features, resulting in lack of inference causality [64].

For the better understanding of the relationship between soil microbiome and plant health, studies have been conducted which use random matrix theory to construct an ecological network [68]. To understand the dynamic behaviour of high throughput metagenomics data, a noise tolerant ecological structure identification method named molecular ecological networks (MENs) was developed, based on random matrix theory, to understand microbial communities [69].

Correlation based network association methods have also been used to study microbial communities. As the microbial communities are comprised of large number of bacteria and their associated interactions, studying the impact of such microbial communities become computationally challenging, compelling the usage of microbial correlation networks to compute pairwise associations [70]. Statistical methods such as Kendall Tau, Pearson, and Spearmen have also been used to create the ecological network, but may result in spurious relationship [71].

Research in [72] explains the impact of human gut microbiota in neuropsychiatric diseases and how the gut microbiota influence the neurological system of a human being. Another study shows the impact of exercise and controlled diet on gut microbiota of rugby athletes [73] and how the change in diet affects athlete's performance. The impact of gut microbiota on liver diseases was studied in [74]. Similarly, numerous studies have been conduced to understand the growth and development of plants. A study about the root and

leaf microbiota found that both contain bacteria [75], which helps in understanding the impacts of certain pesticides. The impact of adding silver nano-particles (SNP) in soil microbiota was studied in this research [76] and a negative correlation was found between the soil biomass and application of SNP.

Numerous methods have been proposed to predict and understand the microbial interactions based on either supervised or correlation based approaches. A major drawback of a supervised approach is its dependability on the underlying labelled data which may not accurately represent the evolving behavior of microbial communities and the correlation based methods may suffer from spurious inferences. Therefore, an unsupervised method was needed to overcome both of these shortcomings, which was proposed by the authors of [13].

4 Algorithm Description

As discussed in the section 2.7, the unsupervised approach [13] suffers from a number of shortcomings. In this section, we describe in detail the work we have done to enhance the heuristic and brute-force algorithms proposed in this research [13]. Our proposed improvements focus on increasing the throughput, improving the algorithm execution speed and increasing the accuracy of both heuristic and brute-force methods.

- **Performance:** We parallelized the sequential processing of taxa by exploiting the inference independence of taxa. This implementation uses the potential of available computational power to achieve a decreased execution time for both brute-force and heuristic approaches. We process taxa in parallel by using an available CPU core for each individual taxon. If number of taxa are more than the available CPU cores, then taxa will be processed one after another in the order of completion of execution. Once an individual taxon has been processed, inference results from all taxa are computed, using the same method, which serves as the final inference of the entire microbial community.

- **Block-wise Brute Force:** With increasing number of taxa in the microbial community, the inference of the best sign pattern through exhaustive search of all $N3^N$ sign patterns becomes practically impossible. To address this issue, we introduce a method which performs a block-wise brute-force on blocks of taxa specified by user. This enables us to perform an exhaustive search on individual blocks of taxa, and then fixes the interactions for subsequent computation of remaining blocks.

- **Improved Heuristic:** We improved the inference of heuristic derived sign patterns in two ways: first, we introduce a local sign-space search which randomly perturbs an inferred sign pattern to find a sign pattern with a higher ϕ. Secondly, we introduce a hybrid approach which uses both heuristic and brute-force inference methods. If the heuristic pipeline fails to infer a sign pattern, the brute-force pipeline is used to infer a sign pattern.

- **Variance Reduction:** We observed a high degree of variance in the inference results for independent iterations. To lower the variance, we introduce weighted voting.

4.1 Performance Improved Implementation

To improve the performance of an algorithm, numerous techniques can be employed which can efficiently reduce the run-time of the overall task. However, proper selection of those techniques are important as it is possible that parallelization may end up increasing the run-time, reducing the throughput of the algorithm. For example, generally increasing the computational power would result in better performance. However, it does not always result in a better throughput as an algorithm designed to execute sequentially would process each independent component of program one after another, resulting in computational resources being idle. This in turn would result in limited performance improvement as the independent components of programs are not being executed independently. An algorithm's ability to support parallel execution is limited by the degree of parallelism inherit in the problem, which refers to the number of sub tasks within a larger task that can be executed in parallel. As the degree of parallelism is intrinsic to the algorithm design, sometimes it is necessary to re-design an algorithm for improved performance. Similarly, parallelism is not the only way to improve an algorithm's performance, for example: clever usage of specialized resources like GPUs, can substantially improve execution time. Execution of tasks on specialized hardware like GPUs involve an overhead cost, for example waiting for the resource to be free. Consequently, when putting a task on a specialized hardware, it is important to consider the trade-off between the estimated speed-up and the imposed overhead.

While the previous implementation [13] provides a major contribution in unsupervised inference of interaction types, the proposed algorithm did not discuss parallel implementation. Because computational performance dictates the maximum size of problem one can address using this technique, computational efficiency has a direct impact on the scientific utility of the algorithm. Leveraging the assumption from the research [13] that the inferred sign pattern for each taxon is independent of every other taxon, we employed a parallel processing pipeline to process all available taxa to the limit of computational resources. The parallel implementation approach as shown in Fig. 4.1 exploits the power of available CPU cores for parallel computation of a differential hyperplane matrix (difference of steady state abundance data samples) and differential hyperplane intersection with respect to a proposed sign pattern for each taxon.

The sign satisfaction graph computation and traversal methods proposed in the research [13] relies on a relatively slow sign satisfaction graph. It has been replaced with its equivalent matrix multiplication to further speed up the process. The matrix-based sign satisfaction graph uses the differential hyperplane matrix and proposed heuristic, but instead of creating a graph with nodes and edges, it performs matrix multiplication of sign pattern and differential hyperplane matrix to obtain all possible sign paths, which we define as the sign matrix. The mathematical expression: $B = S \odot H$ shows matrix sign satisfaction computation through the element wise multiplication of S and H, where B represents sign satisfaction matrix, S is the sign pattern and H is the differential hyperplane matrix, while $\odot$ sign represents element-wise matrix multiplication.

$$B = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} \odot \begin{bmatrix} H_{11} & H_{21} & H_{31} \\ H_{12} & H_{22} & H_{32} \\ H_{13} & H_{23} & H_{33} \\ H_{14} & H_{24} & H_{34} \end{bmatrix} = \begin{bmatrix} S_1H_{11} & S_1H_{21} & S_1H_{31} \\ S_2H_{12} & S_2H_{22} & S_2H_{32} \\ S_3H_{13} & S_3H_{23} & S_3H_{33} \\ S_4H_{14} & S_4H_{24} & S_4H_{34} \end{bmatrix} \tag{4.1}$$

Matrix Multiplication of Sign Pattern and Differential Hyperplane Matrix Resulting in Sign Matrix

The left hand side of equation 4.1 represents the inferred sign pattern and differential hyperplane matrix, each entry in the sign pattern matrix represents a proposed interaction type which is yet to be evaluated and each column of the differential hyperplane matrix represents a differential hyperplane calculated through a pairwise difference of sample rows of abundance data with respect to a taxon. Equation 4.1 represents the sign matrix which is computed through the multiplication of a proposed sign pattern and a differential hyperplane matrix. Each column of the resultant matrix of equation 4.1 represents a sign satisfaction path of proposed sign pattern with respect to its respective hyperplane in the differential hyperplane matrix. In the sign matrix, each entry acts as a signed vertex of sign satisfaction graph, while each column represents a possible solution path as proposed in the research [13]. Matrix multiplication effectively reduces the computation time of nodes and edges of sign satisfaction graphs as the creation of a sign satisfaction graph requires O(NM) operations where N represents the number of taxa and M represents the number of hyperplanes, while its equivalent matrix based operation performs the same operation in constant time on appropriate hardware. Furthermore, to check the validity of a path in sign matrix, we simply employ a short circuit evaluation-based technique where the traversal of a path in sign satisfaction graph is terminated as soon as the path has at least one positive and negative node, limiting the path traversal time to O(N) in worst case as maximum number of entries in a column of B is no more than N. This approach reduces the end to end processing time for the computation of a sign pattern's ϕ.

4.2 Perturbed Heuristic Implementation

The heuristic method proposed in by the authors of [13] limits the search space to only those sign patterns present in the data. As each random selection of hyperplane does not always result in a unique heuristic sign pattern and the number of attempts to find a heuristic is also limited by a user specified threshold. This renders us unable to search the entire search space of $N3^N$, and it is possible that a sign pattern with higher ϕ value may exist close in the sign space to a sign pattern generated by the heuristic but that sign pattern would never be evaluated because noise in the data has changed the intersection sign pattern away from ideal. The presence of noise in the data may cause the heuristic pipeline to propose a sign pattern close to, but not actually a sign pattern that would represent a population dynamics of underlying microbial community. The ϕ is computed by checking the intersection of proposed heuristic with each available pair

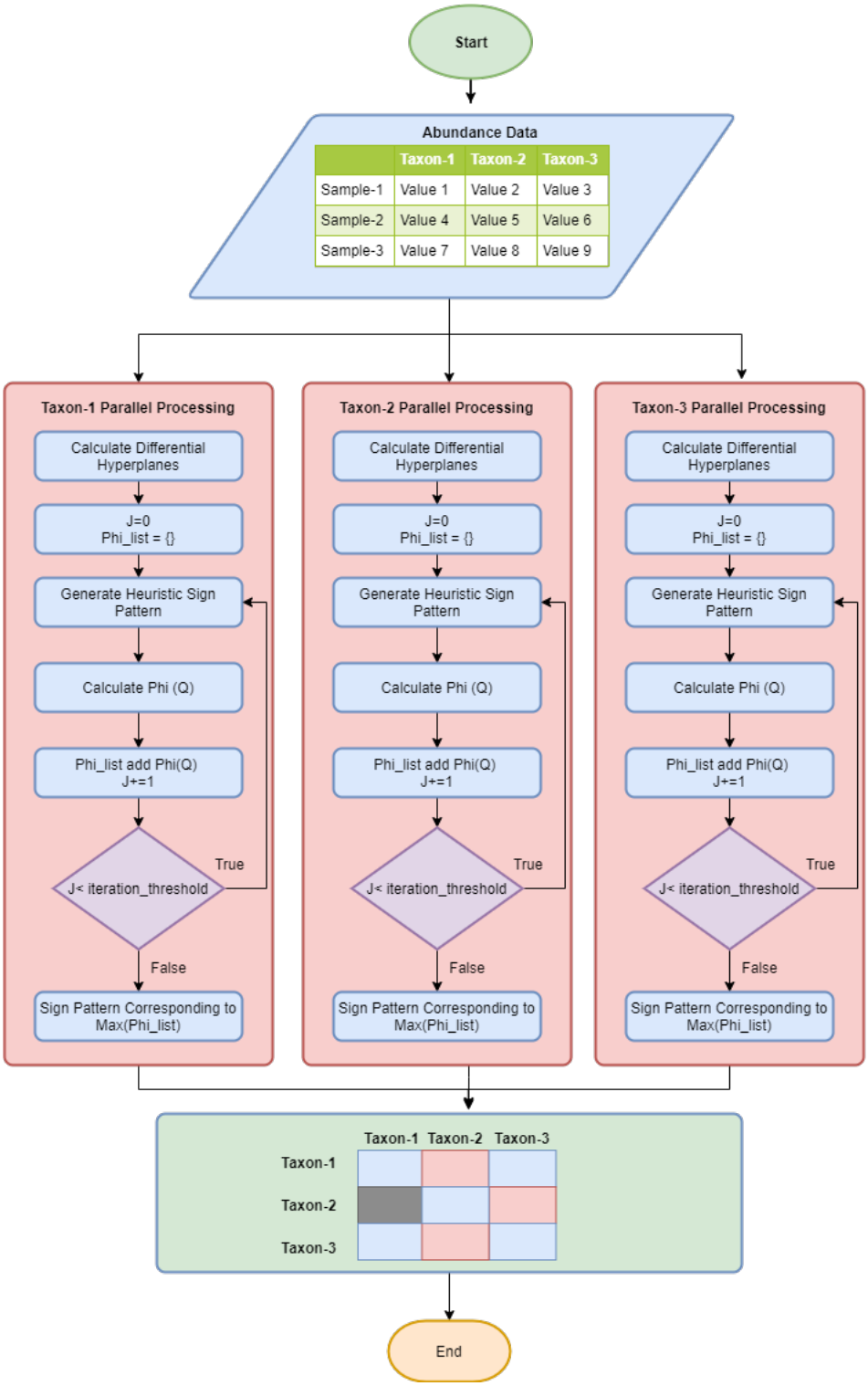


Figure 4.1: Improved Performance Implementation Pipeline

difference hyperplane. A ratio of intersected hyperplanes is taken with respect to the total available pair difference hyperplanes to quantify a heuristic sign pattern.

Our perturbed heuristic method as shown in Fig. 4.2 probes a solution space around a solution proposed by heuristic using a random walk. Once a sign pattern is proposed, ϕ for the un-perturbed sign pattern is computed. The proposed sign pattern is perturbed at a randomly selected index, changing the current sign to one of the other two possibilities. The perturbed sign pattern's ϕ value is calculated. If ϕ is greater than the current sign pattern's ϕ, the perturbed sign pattern becomes the new proposed sign pattern. The sign pattern is randomly perturbed until the ϕ stops improving. To limit circular searches, an entry in the sign pattern is perturbed exactly once. The addition of perturbation step in the existing sign pattern heuristic generation allows us to improve the average match quality by performing a random walk to find a sign pattern having improved ϕ.

4.3 Reduced Variance Implementation

Because the heuristic is stochastic in nature, the interaction pattern proposed by the heuristic approach may not be stable, impairing the algorithm's utility. As randomness plays an integral role in the sign pattern chosen by the heuristic, it also hinders the consistency of interaction types being inferred, potentially leading to variability in inferred interaction types.

We developed a variance-reducing implementation as shown in Fig. 4.3. The objective of this implementation is to reduce the variance in inference of interaction types for the same community while maintaining the solution quality. Our approach assumes that the solution available in the top percentile of ϕ should have high degree of overlap, indicating that there is sufficient data to infer a strong and sufficient solution.

Our method extends the proposed heuristic method in the research [13] by performing a weighted ϕ analysis. All sign patterns with ϕ value lower than a user derived threshold α are discarded, α is derived from a user specified threshold δ as shown in equation 4.2. As per equation 4.2, once a set of sign patterns with ϕ value above the α are selected, all sign patterns are analyzed for their inference type towards every other taxon. For each taxon pair, a weighted frequency for each interaction type (positive, negative and undefined) is calculated. The interaction type with the highest frequency is considered as the most likely interaction type for that taxon pair. For each taxon pair, the step of weighted frequency-based inference is repeated to obtain a weighted sign pattern. The obtained frequency-based sign pattern is checked for sign satisfaction, if the achieved ϕ value for this sign pattern is higher than or equal to α, then this sign pattern is considered the final sign pattern. However, if the ϕ value of the weighted frequency-based sign pattern is lower than the range suggested by α, which is derived from user specified threshold δ, all the sign patterns with a ϕ value ranging within the limits specified by α are matched against the obtained weighted frequency-based sign pattern. The sign pattern amongst the selected sign patterns with $\phi \geq \alpha$ having the least edit distance [77] (minimum number of operations required to transform frequency-based sign pattern

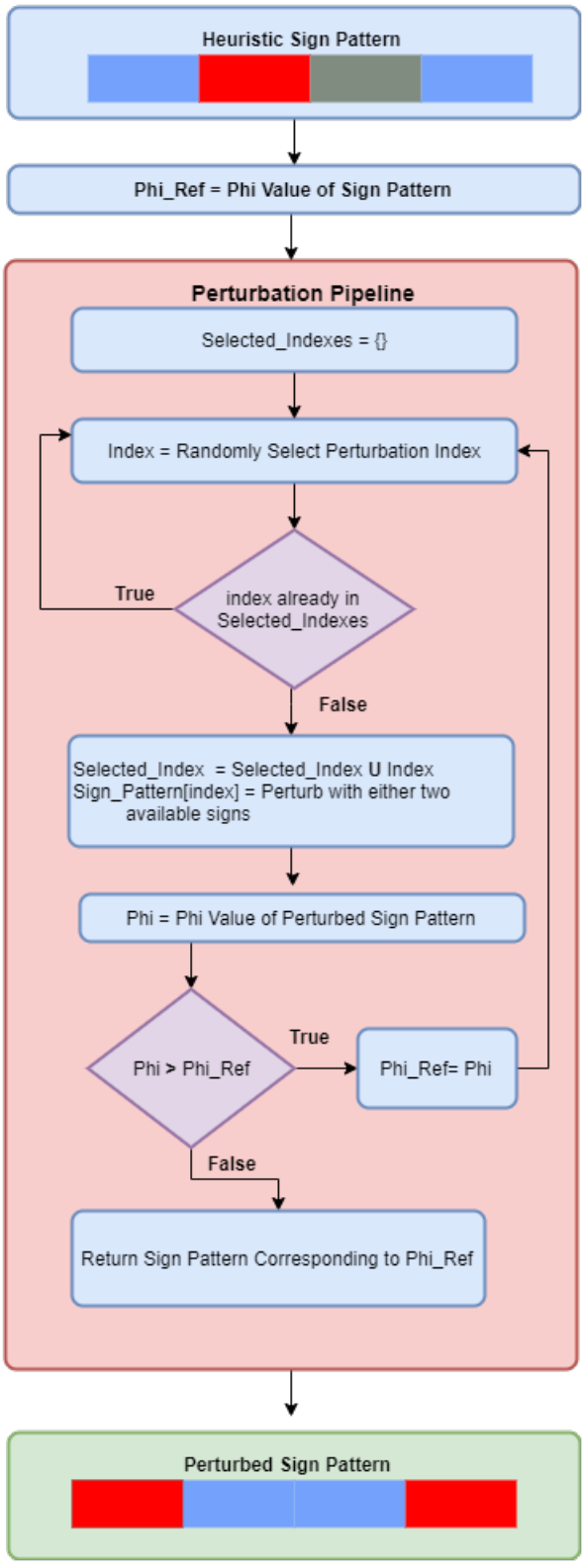


Figure 4.2: Pipeline for Perturbed Sign Pattern Implementation.

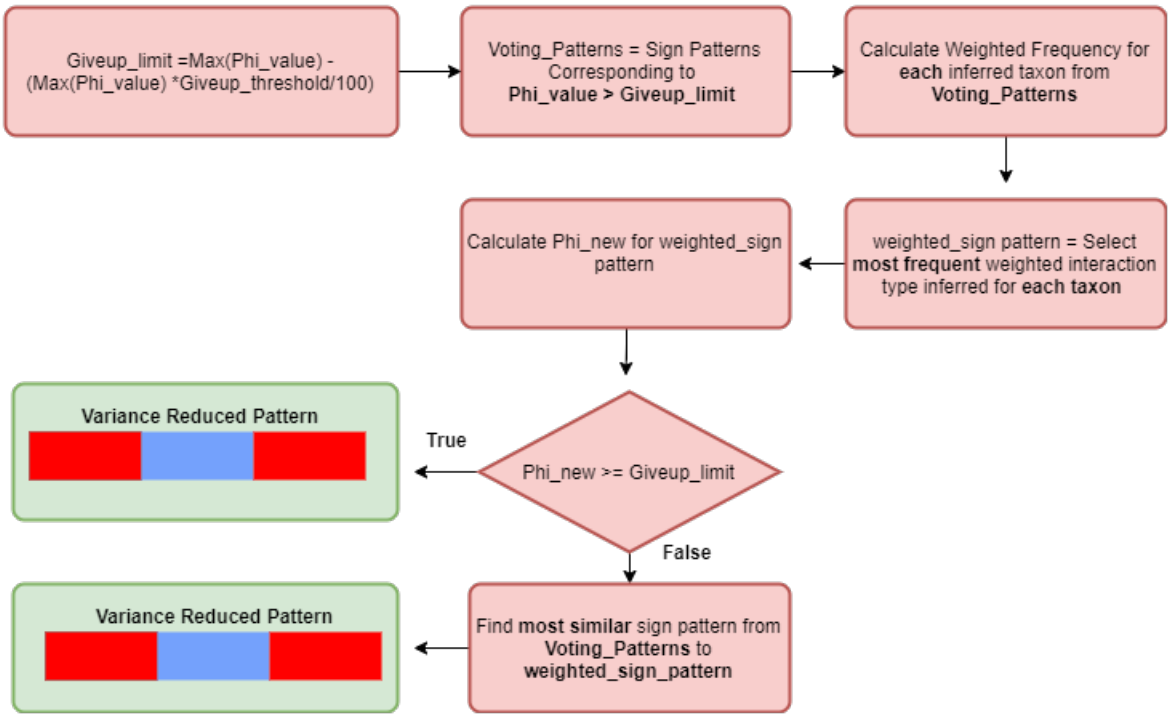


Figure 4.3: Pipeline for Variance Reducing Implementation

into one of the selected sign patterns) from the obtained frequency-based sign pattern is considered as the final inference for taxon under consideration.

$$\alpha = Max(\phi) - Max(\phi) \times \delta/100 \tag{4.2}$$

4.4 Hybrid Implementation

The optimal solution should always be available within the data as long as the data is essentially noise free, which is unfortunately seldom the case. For the *Brassica napus* dataset with 20, 25 and 36 taxa [78][79], the heuristic inference step fails for more than 50% of taxa, resulting in no sign pattern inference. The failure in the inference of heuristic can be caused by insufficient data as the pairwise difference of active samples should always be more than number of taxa in the underlying community. As shown in equation 4.3, the number of pairwise difference hyperplanes computed through active samples should always be greater than N-2, where ξ represents the number of active abundance samples and N represents the number of taxa,

$$\frac{\xi(\xi-1)}{2} \geq N-1 \tag{4.3}$$

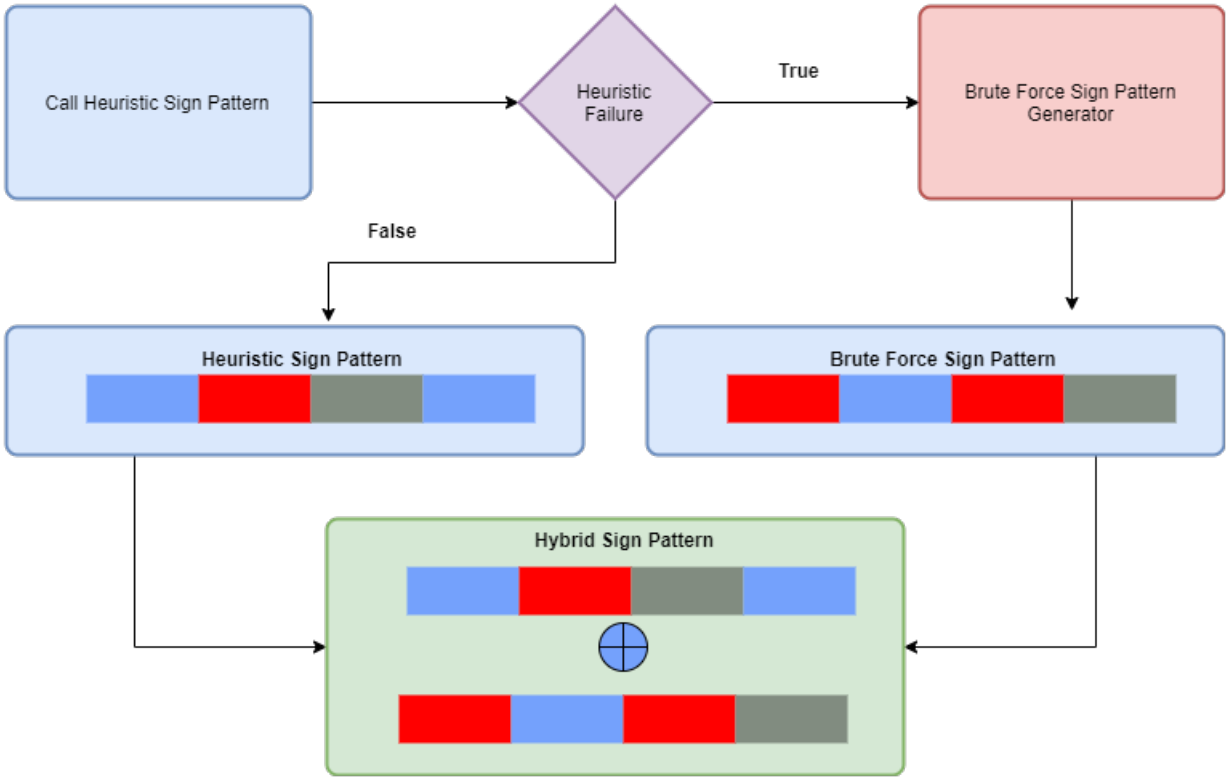


Figure 4.4: Pipeline of the Hybrid Implementation.

Considering the failure rate of the heuristic prediction step, we propose a hybrid approach as shown in Fig. 4.4. The hybrid approach combines both the brute-force and heuristic approaches. When the heuristic step fails to infer a sign pattern, a partial brute-force module is used to obtain a sign pattern inference from 3^N possible combinations of sign patterns for that taxon only. The maximum number of heuristics being inferred by the hybrid module is limited by a user specified threshold. The hybrid module only provides brute-force inferences to taxa for which an inference can not be made by the heuristic module and a sufficient number of non-zero samples exist. It does not search the entire 3^N search space per taxon but rather provides inferences from that search space, derived from the user threshold. Using the hybrid approach, we are able to infer sign pattern(s) for all taxa which failed heuristic inference. Fig. 4.4 shows the workflow of hybrid implementation.

4.5 Accelerated Brute Force Implementation

We introduce a modified brute-force approach as shown in Fig. 4.5, to address the performance issues and stability in the presence of noisy data. The modified algorithm reduces the size of differential hyperplane matrix for faster computation of quality measure (ϕ) with respect to a sign pattern. Additionally, to allow

for solutions when data is noisy, we employ ϕ for sign pattern quantification and iterate over all sign patterns to find the best sign for each taxon. If a taxon has more than one sign pattern with the maximum observed value of ϕ, the first amongst them is selected as the final inference result (final sign pattern).

We also introduce a voting method to settle the tie between sign patterns having the same ϕ score. Our voting method follows the same steps as variance reduction method, which is shown in Fig. 4.3, but with a small change in that the threshold δ is set to 0. This allows for the value of α to be the same as maximum achieved ϕ, allowing only the sign patterns with maximum ϕ to participate in voting, resulting in no loss in the confidence (ϕ) of inferred sign pattern, with increased stability of inference.

In addition to parallel processing of each taxon, the size of each differential abundance matrix is reduced by using a pre-processing step which modifies the differential hyperplane matrix by maintaining only the unique columns (hyperplanes). It discards any repetitive columns and keeps the repetition count of each unique column. This reduction in the size of differential hyperplane matrix requires $O(M^2)$ operations, making it infeasible for the heuristic solution as the heuristic solution checks only a few heuristics and performance gains become insignificant against the $O(M^2)$ time complexity of differential hyperplane matrix size reduction, where M represents the number of hyperplanes in the differential hyperplane matrix. As the brute-force solution iterates through all 3^N possible sign patterns, this reduction in the size of differential hyperplane matrix proves useful.

4.6 Variable Block Size Brute Force (Block-wise) Implementation

The accelerated brute-force approach evaluates all 3^N possible combinations of sign patterns per taxon. However, with the growth of N the accelerated brute-force approach is no longer viable. To circumvent the limitation imposed by the exponential growth of sign space, we introduce a variable block size brute-force method as shown in Fig. 4.7. This implementation uses the assumption of taxon independence as the inference of interaction type with respect to a focal taxon i to j is independent of inference of interaction type for focal taxon i to k. So, instead of inferring a sign pattern for N taxa with respect to a focal taxon, the block-wise method performs brute-force on a block, where the block length of each block is specified by the user, represented by λ. The λ threshold contains a list of block sizes, specified by the user. Each entry in the list represents a block length to be processed, allowing for variable block sizes.

This method incrementally infers the sign pattern of each subsequent block by using the brute-force method. As the subsequent blocks are processed, the inter block sign satisfaction is maintained by evaluating the sign satisfaction on all previously processed blocks, along with the block of signs currently being inferred. As shown in Fig. 4.6, the sign pattern for processed blocks is not recalculated. The already-inferred brute-force sign patterns for previous blocks (fixed blocks) are used as seed, and seed blocks are concatenated with the inferred sign pattern of new block which is yet to be evaluated, resulting in a sign pattern inference for all blocks up till the current processing block, but with fixed inferences for already processed blocks.

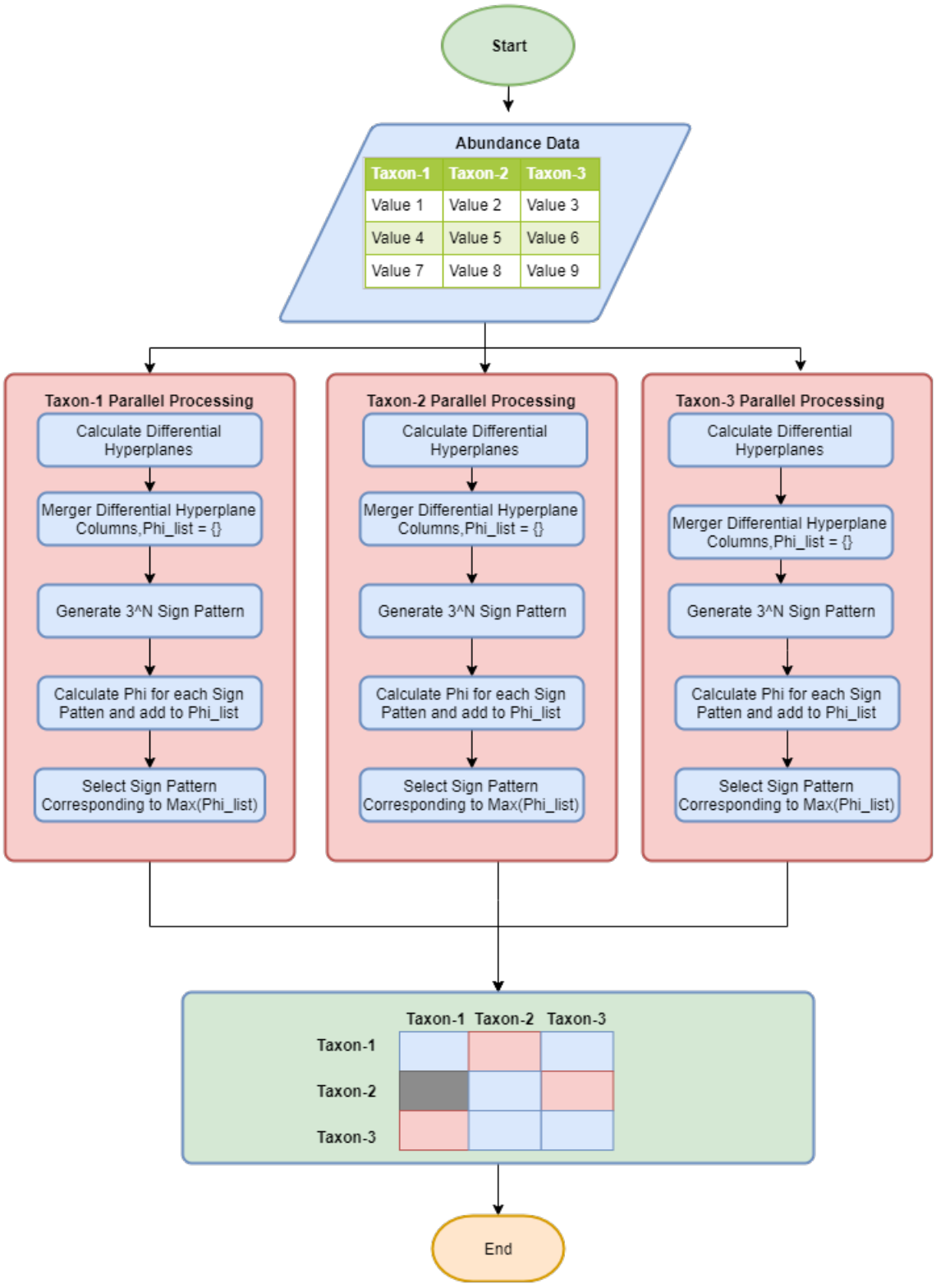


Figure 4.5: Pipeline for Accelerated Brute Force Implementation

This partial processing of a block by using the previously processed block inferences as a seed results in significant reduction of inference complexity as only a single used specified block is being brute-forced at any particular instance. Using the block-wise the sign space complexity is reduced from $N3^N$ to $\frac{N}{\lambda}3^\lambda$ which offers a significant speed-up over the accelerated brute-force method, enabling the processing of $N > 20$, while finding a locally optimal inference of each block of ecologically coherent taxa.

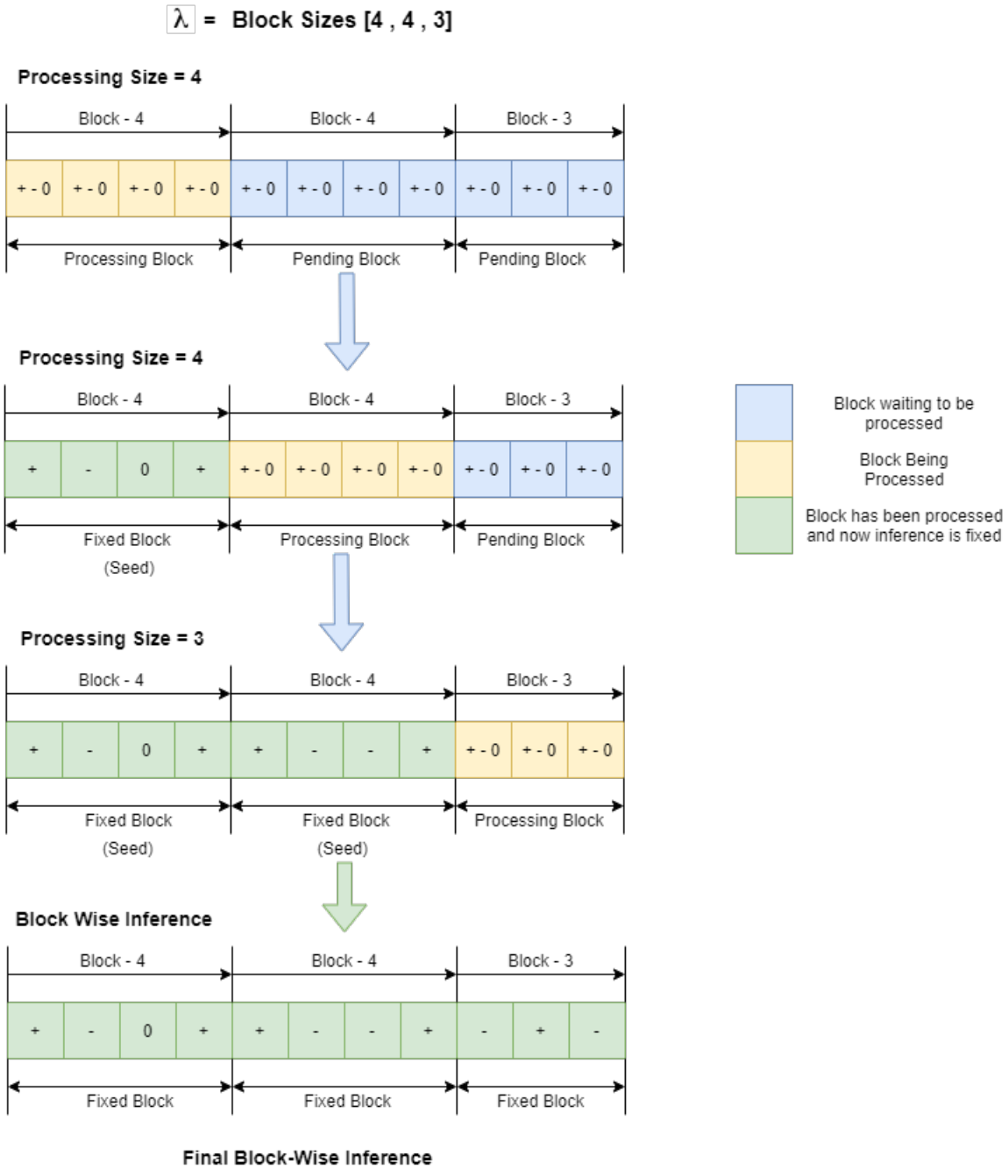


Figure 4.6: Step-wise Illustration of Variable Size Block Processing

Figure 4.7: Pipeline for Variable Block Size Brute Force Implementation

5 Experimental Setup

This section explains our experimental setup which compares each enhanced approach with the default approach [13]. Each subsection explains and compares the achieved improvement with respect to each enhancement made to the default approach.

5.1 Dataset Description

For this study, we used three datasets to showcase the performance and consistency of the proposed algorithm improvements. The datasets are represented as sample-taxon pairs. Each cell contains the abundance of a taxon in that particular sample. Out of the three datasets under consideration, two datasets are real, while one dataset is composed of simulated abundance samples. The *Brassica napus* bacterial community dataset [78] [79] has 3 variants with 20, 25 and 36 amplicon sequence variant (ASV) also referred as taxa. The dataset for the bacterial community of maize roots [80] is composed 7 taxa. While our simulated dataset has three variants of 10, 20 and 30 taxa having 10, 50 and 100 samples per taxon respectively.

5.1.1 Brassica napus Dataset

For sixteen canola genotypes, root and rhizosphere soil samples were collected each week for a period of 10 weeks. Samples were collected from three different sites in Saskatoon, Melfort and Scott in year of 2016 and 2017 in Saskatchewan,Canada [78] [79]. The dataset consists of 13,230 ASVs with and average of 132 ASVs per sample and a total of 1,162,800 reads [78].

The entire dataset was passed through a series of filters before performing the select balance based analysis to identify the smallest group of taxa with highest predictive power. The initial 13,320 ASVs were reduced to 5,754 after noise filtering by selecting the ASVs with more than 2 reads. Similarly, 2073 ASVs are filtered with less than 2 occurrences followed by majority, sample range and predictive analyses to reduce the total number of ASVs to 977. The filtered set of 977 ASVs were passed through the Selbal in R package v 0.1.0 [81] to find the final 81 ASVs which are most predictive of yield performance [78] [79].

Selected ASVs are divided into three sets of 20, 25 and 36 ASVs. The most significant 81 ASVs selected after the selbal filtering were passed through a principle component analysis pipeline weighted by prevalence and then clustered in to three groups. Each group has 20, 25 and 36 ecologically significant ASVs consisting of 357 unique samples. For ease of analytical understanding, the ASV names for both sets were renamed with their respective genus names and the ASVs belonging to the same genus were numbered sequentially.

This data was collected and processed by Dr. Siciliano's group at the University of Saskatchewan. The methods they used are reproduced here for completeness. We are grateful to our co-researchers Steven Siciliano and Steven Mamet from the department of Soil Sciences for the collection and filtering of *Brassica napus* dataset and making it available for our use.

5.1.2 Maize Roots Dataset

This dataset consists of seven bacterial species of synthetic bacterial community of maize roots [80]. The data has been categorized into two sub groups. There are seven groups (steady states) comprised of six species, and one group is comprised of seven species. For analysis, we only use seven groups of six species while leaving the group with seven species as in [13]. Each steady state sample group has six samples making a total of 36 samples per steady state. Steady state samples are renamed for ease of representation as specified in [13]: Enterobacter cloacae (Ecl), Stenotrophomonas maltophilia (Sma), Curtobacterium pusillum (Cpu), Ochrobactrum pituitosum (Opi), Pseudomonas putida (Ppu), Herbaspirillum frisingense (Hfr), Chryseobacterium indologense (Cin) [13] [80].

5.1.3 Simulated Dataset

We are thankful to Juxin Li from University of Saskatchewan Department of Mathematics and Statistics for providing us the source code for the implementation of the Generalized Lotka-Volterra model to generate the simulated data.

The simulated dataset is comprised of 3 different variants of taxa and samples sizes in order to fully gauge the scalability and performance of the improved inference approaches. Our simulated dataset has three variants of 10, 20 and 30 taxa referred as Simulated-10, Simulated-20 and Simulated-30 respectively. Each variant has three sample classifications of 10, 50 and 100 samples per taxon. We have 25 datasets for each individual pair of taxa and sample size.

R package Seqtime [82] and Devtools [83] were used to create the different variants of simulated dataset mimicking the GLV model having the intrinsic growth rate and initial sum of abundances given by the total number of taxa to be generated. All negative abundance values were normalized to zero and all variants of simulated dataset were generated only to test the efficacy of improved algorithm proposed in the research [13].

5.2 Hardware and Software Details

The Python 3.7.4 environment for program execution was setup on both Windows 10 and Ubuntu 18.04.3 environments for each experiment type. The server used for the execution of computational tasks had 1511 GB of memory, Nvidia V100 GPU Computing Accelerator - 16GB as GPU, 64 cores of CPU model: Intel(R) Xenon(R) Gold 6130 CPU with 2.10 GHz clock rate. Furthermore, following versions of Python libraries were

used: Pandas 0.25.1 NumPy 1.16.2, NetworkX 2.3, SciPy.linalg 0.4.9, SciPy 1.2.1, SymPy 1.4, Matplotlib 3.0.3, Seaborn 0.9.0.

5.3 Algorithm Parameters

Any entry in the pairwise difference matrix ranging between $\pm 1e-5$ is considered noise and rounded to zero, represented by threshold $\pm\Omega$. All heuristic based methods were executed for 100 independent iterations for sensitivity analysis, while each independent iterations has a threshold (J) of 100 sign patterns being inferred for each taxon. The accelerated brute-force and the block-wise implementations are executed once per dataset for each classification as it does not use a heuristic inference module, each independent iteration of inference produces the same result.

5.4 Accuracy Evaluation Methods

As the hybrid, perturbed and variance control methods directly impact the interaction types of sign pattern, we performed a comparative analysis of accuracy of default and improved approaches. We used three approaches to compute the accuracy of an inferred sign pattern with respect to the ground truth. The first method is the same as the default method for accuracy, which considers any occurrence of zero in ground truth equivalent to either positive or negative interaction type in the inferred sign pattern, also referred as "Always Right" as proposed in [13]. The second method is where we consider only an exact match of interaction type as correct also referred as "Always Wrong" making a conservative estimation. The third method does not consider zeros during accuracy computation and it is referred as "Without Zero" removing the small or null interactions as ecologically irrelevant.

5.5 Evaluation Methodology of Hybrid Heuristic Implementation

We employ a comparative evaluation approach that compares the heuristic failure rate of the default and the hybrid approaches on both real and simulated datasets. To quantify the impact of either implementations on a dataset, we computed the percentage of failure of heuristic on different datasets. As the purpose of hybrid implementation is to reduce the failure rate of heuristic inference, comparing the failure rates of both default and hybrid approaches provides empirical evidence of the stability of heuristic inferences.

5.6 Evaluation Methodology of Perturbed Heuristic Implementation

The primary purpose of perturbing a heuristic is to find a sign pattern with maximum ϕ in its local search space. As perturbation improves the quality of sign patterns, we perform a comparative analysis of average ϕ to quantify the extent of improvement made through perturbation. The process of perturbation is dependent heuristic. We compared the impact of perturbation on the hybrid heuristic, referred to as hybrid-perturbed, to understand the impact of failure on perturbation. To quantify the significance of the difference between the average ϕ attained through either methods, we also computed the F-score and p-value of the average ϕ distribution, which shows the significance of improvement achieved through the perturbed method. Using these tests, we hypothesize that there would be a higher average ϕ for perturbed sign patterns compared to the default methodology.

5.7 Evaluation Methodology of Variance-Controlled Implementation

In this section we compare the impact of the variance-controlled heuristic approach and the default approach. It was observed during our analysis of each independent execution cycle of algorithm that the variance in inferred sign patterns increases significantly with the increasing number of end to end algorithm cycles. This represents an inconsistency in inferred interaction type over the independent iterations, which makes it difficult to perceive the behaviour of microbial community with certainty. To reduce the variance in inference, we devised an approach to minimize the variance of the inferred sign patterns over the independent iteration of inference, using the user specified threshold δ, which controls the selection of sign pattern based on their ϕ score. We tested our approach on both simulated and real datasets.

As the purpose of variance controlled approach is to infer consistent interaction types over individual iterations, we compared the average variance produced by the default and the variance-controlled approach over a 100 independent iterations of both implementations along with their ϕ. This allows us to quantify the impact of variance control implementation on overall consistency of inference on different datasets and its impact on the quality of inferred sign patterns.

As the variance control directly impacts the interaction types of inferred sign patterns, we also perform a comparative analysis of accuracy of both default and the variance control approaches, using the same three accuracy evaluation techniques used for the perturbed approach.

5.8 Evaluation Methodology of Parallel Execution Implementation

Parallel implementation is used to speed-up the execution by using the available computational power. To evaluate the parallel execution implementation, we perform a comparative execution time analysis of both approaches. For real datasets, we compared the end to end execution time in seconds of both approaches over a 100 independent iterations to avoid the minor differences in execution time duration caused by the availability of hardware resources. For simulated datasets, we also computed the average execution time duration to show the impact of increasing the number of taxa on the execution time of the parallel approach which shows the scalability of parallel implementation. It is to be noted that the parallel implementation does not have any impact on the inference of heuristic or the quality of inferred sign patterns.

5.9 Evaluation Methodology of Accelerated Brute Force Implementation

The brute-force method evaluates all $N3^N$ combinations of heuristics which limits its ability to be executed on larger microbial communities. We performed a comparative analysis of accelerated brute-force approach on the Maize Roots and the Simulated-10 datasets in terms speed. We also perform a comparative analysis of difference between the ϕ of inferred and ground truth sign patterns to quantify the accelerated brute-force's ability to find the most viable solution.

We mapped the distribution of maximum observable sign pattern and used two different methods called voting based or first selection to settle the tie between the sign patterns with maximum observable ϕ. We compared accuracy using the three previously mentioned approaches for both of these sign pattern selection methods to quantify the impact of each methodology on the achieved accuracy.

As we used the modified matrix multiplication approach for the evaluation of quality score for each sign pattern, we also conducted a comparative analysis study to find differences in computed quality scores (ϕ). As per our expectation, the ϕ calculated through the accelerated brute-force method and the graph based sign satisfaction method proposed in the research [13] produced the same quality score for all 3^7 patterns. This supports the equivalency of matrix multiplication method and the default graph search method initially described in [13].

5.10 Evaluation Methodology of Variable Block Size Implementation

The variable block size (block-wise) heuristic algorithm extends the assumption in [13] applied to focal taxa to blocks of taxa. This approach uses accelerated brute-force method to infer sign patterns for blocks of variable sizes. For the block-wise method, we compared the ϕ of the block-wise method with the ϕ score of accelerated brute-force method and we compare the distribution of max scored sign patterns achieved through block-wise and the accelerated brute-force methods on Simulated-10 and Maize Roots datasets. This allow us to quantify the efficacy of block-wise method and compare its performance against the full brute-force method. We also compare the accuracy of block-wise method on all datasets as it is a partial brute-force method and it can be used for larger microbial communities.

6 Experimental Results

6.1 Failure Analysis of Default and Hybrid Approach

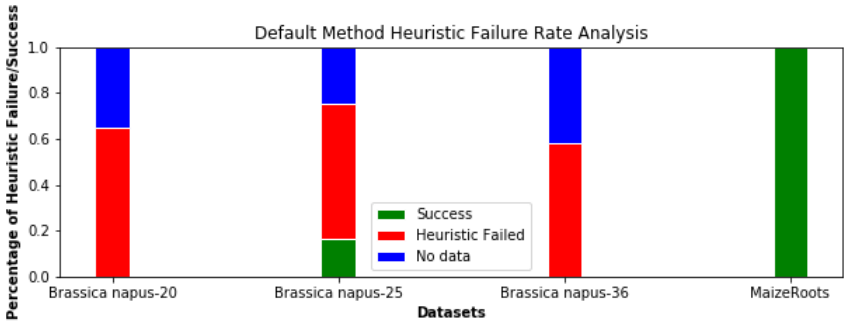
As the hybrid method offers reduced failure rate of heuristic inference. We perform a series of experiments on real and simulated datasets to assess the fault tolerance of the hybrid method as compared to the default method.

6.1.1 Result on Real Datasets

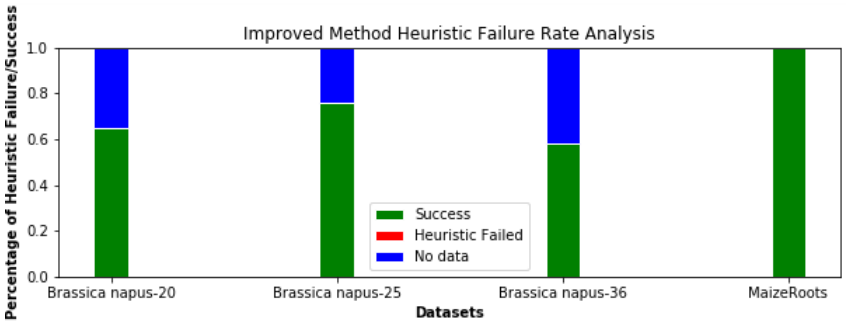
Fig. 6.1 shows the failure rate analysis of enhanced and default approaches on *Brasicca napus* and Maize Root datasets. A mean ϕ score of greater than zero over one hundred independent iterations of heuristic inference shows that a sign pattern was successfully inferred. A heuristic failure can either be caused by too few active abundance samples for the focal taxon, or by the heuristic module failing to infer a solution in the proscribed number of iterations.

It can be seen in Fig. 6.1a that the default heuristic failed to infer a solution for all taxa of the *Brasicca napus*-20 dataset, in which more than 60% of taxa had sufficient active abundance samples. As shown in Fig. 6.1b, the hybrid module was able to infer sign patterns for all taxa having a sufficient number of abundance samples, while failing to infer a sign pattern for taxa with insufficient number of abundance samples. Note that it is mathematically impossible to infer a solution for a taxon with fewer than $N - 1$ active abundance samples using the technique proposed by the authors of [13]. The failure rate of the default approach proposed in [13] for *Brasicca napus*-20 was a 100%, while the failure rate of the hybrid approach was only 35%. As shown in Fig. 6.1a, for the *Brasicca napus*-25 dataset, the default heuristic approach inferred a solution for only 16% of taxa, while as shown in Fig. 6.1b, the hybrid approach was able to infer a sign pattern for 76% taxa. In the same way, the default heuristic module failed to infer any solution for *Brasicca napus*-36 dataset, while the hybrid approach was able to infer a solution for more than 50% of taxa (21 taxa) as shown in 6.1b. It is to be noted that the failure of heuristic method is caused by the data for the mentioned threshold of heuristic inference ($J = 100$). However, it may happen that the heuristic would be able to infer a solution if the threshold J increased.

For the Maize Root dataset, as shown in Fig. 6.1a and b, both the hybrid and default approaches were able to infer a solution for each taxa, as it only has only seven taxa.



(a) Default implementation for *Brassica napus* and Maize Root datasets

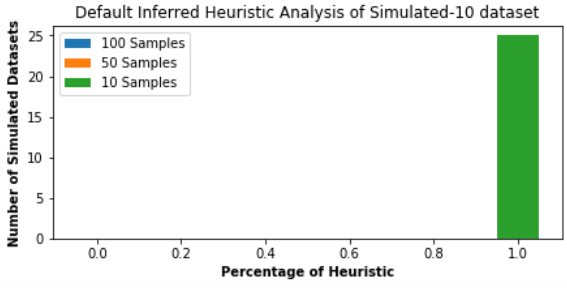


(b) Hybrid implementation for *Brassica napus* and Maize Root datasets

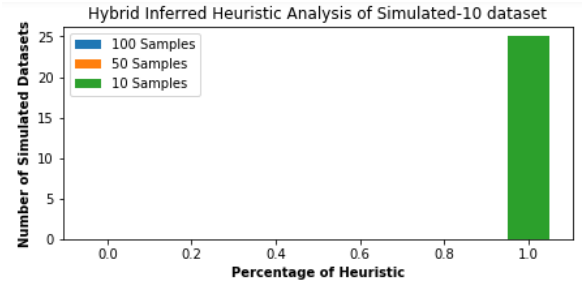
Figure 6.1: Failure Rate Analysis of Default and Hybrid Approaches.

6.1.2 Result on Simulated Datasets

We executed the three Simulated datasets with their 3 sub-classification of samples size 10, 50 and 100 samples per taxon each with 25 datasets to test the efficacy of the hybrid method of simulated dataset. It can be seen in Fig. 6.2 that both the hybrid and the default methods are able to infer a solution for a 100% of taxa for all Simulated-10 datasets.



(a) Distribution of Heuristic Inference for Default Method. Percentage of Heuristic on y-axis represents the number of successfully inferred heuristics out of the total number heuristics.

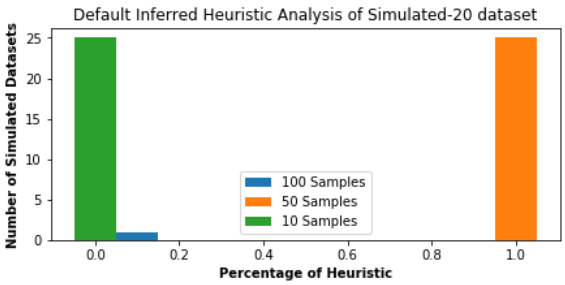


(b) Distribution of Heuristic Inference for Hybrid Method. Percentage of Heuristic on y-axis represents the number of successfully inferred heuristics out of the total number heuristics.

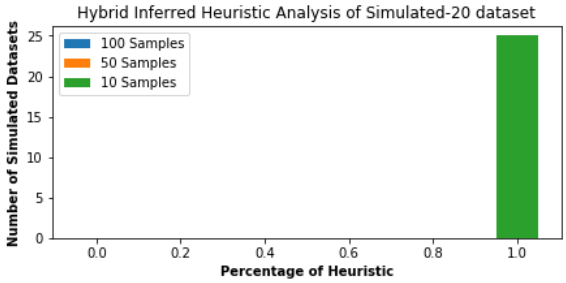
Figure 6.2: Failure Rate Analysis of Default and Hybrid Approaches on Simulated-10 dataset.

For the Simulated-20 dataset as shown in Fig. 6.3, the hybrid method was able to infer a solution for

100% of taxa for each sample type, but the default method failed to infer any solution for the 10-sample class and it also failed to infer a solution for 33% of taxa in datasets having a sufficient number of samples.



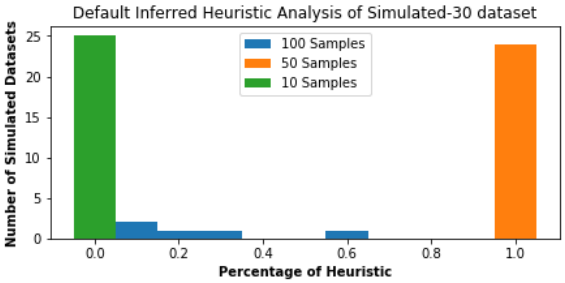
(a) Distribution of Heuristic Inference for Default Method. Percentage of Heuristic on y-axis represents the number of successfully inferred heuristics out of the total number heuristics.



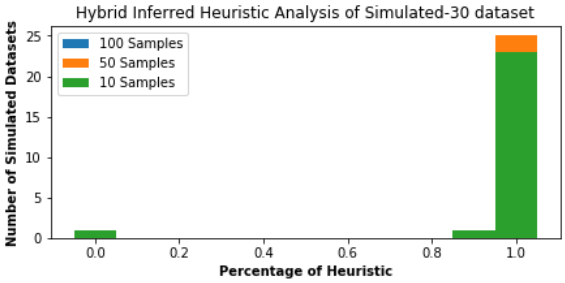
(b) Distribution of Heuristic Inference for Hybrid Method. Percentage of Heuristic on y-axis represents the number of successfully inferred heuristics out of the total number heuristics.

Figure 6.3: Failure Rate Analysis of Default and Hybrid Approaches on Simulated-20 dataset.

For the Simulated-30 dataset as shown in Fig. 6.4, the hybrid method was able to infer a solution for 100% of taxa with 100 and 50 samples per taxon. However, it failed to infer any solution for one dataset with 10 samples per taxa due to an insufficient number of active samples. On the other hand, the default method was unable to infer any solution for all datasets with 10 samples of Simulated-30. The default method was able to infer a solution for 100% of taxa for most datasets with 50 and 100 samples; however, it failed to infer any solution for a dataset with 50 samples.



(a) Distribution of Heuristic Inference for Default Method. Percentage of Heuristic on y-axis represents the number of successfully inferred heuristics out of the total number heuristics.



(b) Distribution of Heuristic Inference for Hybrid Method. Percentage of Heuristic on y-axis represents the number of successfully inferred heuristics out of the total number heuristics.

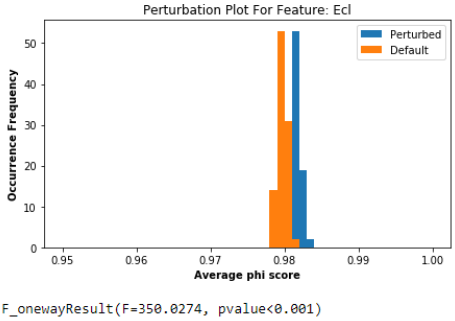
Figure 6.4: Failure Rate Analysis of Default and Hybrid approaches on Simulated-30 dataset

6.2 Average Phi(ϕ) Score Distribution Analysis of Default and Perturbed Heuristic Approaches

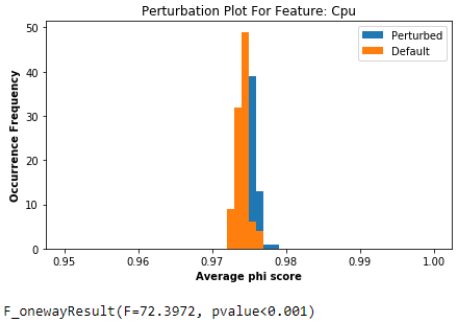
The perturbed method finds a sign pattern with better quality score (ϕ). To assess the extent to improvement, we compared the performance of perturbed approach on both real and simulated datasets.

6.2.1 Result on Real Datasets

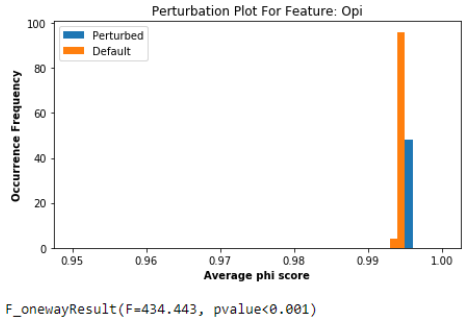
We compared the performance of the perturbed approach against the default approach and by using it in combination with the hybrid approach to quantify the impact of failure on average ϕ score. Fig. 6.5 shows the impact of perturbation on Maize Roots dataset. This dataset is comprised of seven taxa but only three taxa had a less than perfect score (mean $\phi < 1.0$) for either the default or perturbed approach. The degree of improvement achieved by our method is not as significant because the achieved mean ϕ scores of the default approach were already close to perfect. However, the perturbed approach still outperforms the default approach. It should be noted that the failure rate of the Maize Roots dataset was 0.0% for the default approach. It can be seen in Fig. 6.6 that the improvement achieved by the perturbed approach while using the hybrid heuristic inference remains almost the same as it was using the default method. It should be noted that the results are statistically significant but biologically small.



(a) Comparison of Perturbed and Default Approaches for Maize Roots

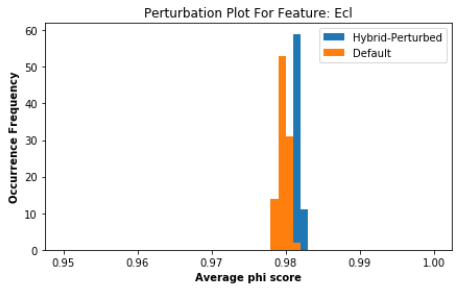


(b) Comparison of Perturbed and Default Approaches for Maize Roots

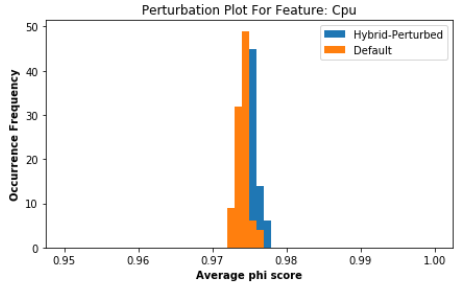


(c) Comparison of perturbed and default approaches for Maize Roots

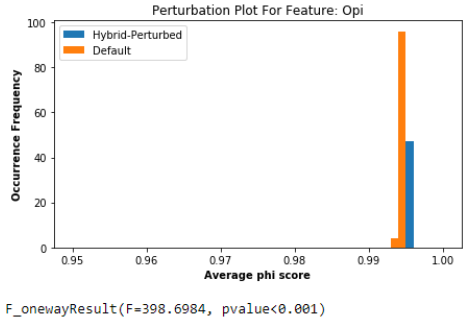
Figure 6.5: Comparison of Perturbed and Default Approaches for Maize Roots



(a) Comparison between hybrid-perturbed and default approaches for Maize Roots



(b) Comparison between hybrid-perturbed and default approaches for Maize Roots

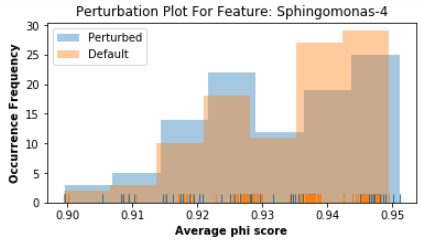


(c) Comparison between hybrid-perturbed and default approaches for Maize Roots

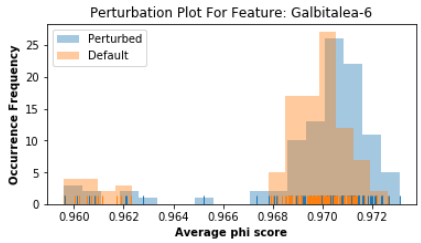
Figure 6.6: Comparison between hybrid-perturbed and default approaches for Maize Roots

For the *Brassica napus* dataset, the failure rate for each taxa was high, resulting in no inference for *Brassica napus*-20 and 36. Only 4 taxa of *Brassica napus*-25 had fewer heuristic failures to have a consistent heuristic inference for all 100 iterations. As shown in Fig. 6.7, the difference between the quality score achieved by the perturbed and default methods remains almost the same, which is also supported by the F-score ($P > 0.05$). The average quality of sign patterns is negatively affected by the high failure rate of default method of heuristic inference as the improvement in quality (ϕ) of sign pattern is still dependent on the inference of heuristic through the default method.

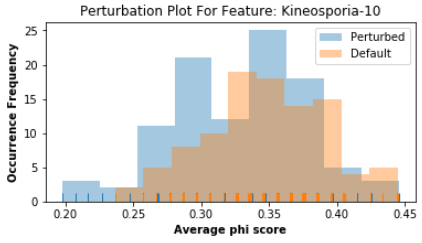
Fig. 6.8 shows the impact of hybrid-perturbed technique on all 4 taxa being inferred by the heuristic algorithm. Each plot shows the distribution of mean ϕ score and F-score, which shows the statistical significance of the difference between the two distributions of mean ϕ scores. For taxa Kineosporia-10 and Flavobacterium-20, it can be seen that their is a significant difference between the ϕ scores of default approach and the perturbed approach. The average of ϕ scores achieved by these features have no overlap with the default approach [13] and the perturbed ϕ score is at least double the quality. For Sphingomonas-4 and Galbitalea-6, the distribution of perturbed and default average ϕ scores show overlap. The ϕ score of perturbed approach tends towards higher mean values, showing that the hybrid-perturbed approach resulted in better ϕ scores as shown in Fig. 6.8. This is also supported by lower p-value ($P < 0.001$), showing that difference between the distributions is significant.



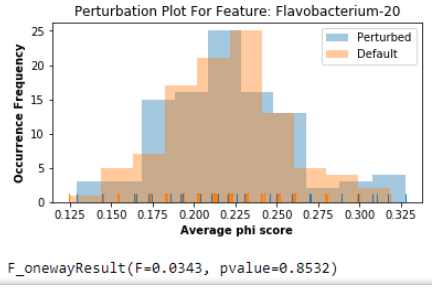
(a) Comparison between perturbed and default approaches *Brassica napus*-25



(b) Comparison between perturbed and default approaches for *Brassica napus*-25

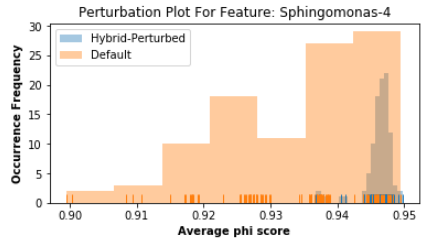


(c) Comparison between perturbed and default approaches for *Brassica napus*-25

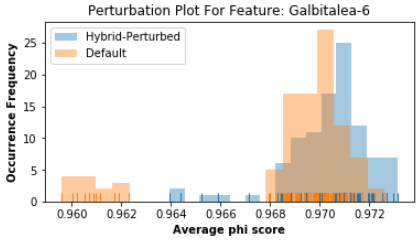


(d) Comparison between perturbed and default approaches for *Brassica napus*-25

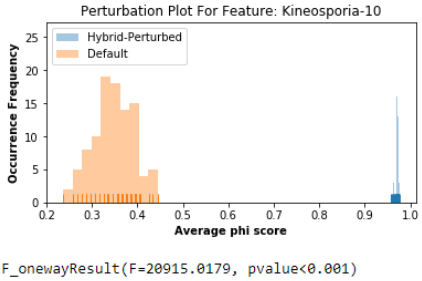
Figure 6.7: Comparison between perturbed and default approaches for *Brassica napus*-25



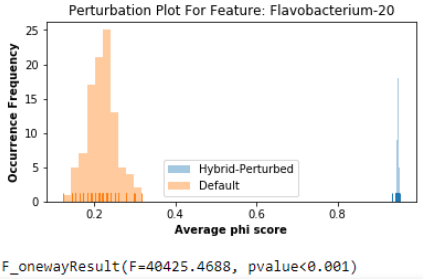
(a) Comparison between hybrid-perturbed and default approaches *Brassica napus*-25



(b) Comparison between hybrid-perturbed and default approaches for *Brassica napus*-25



(c) Comparison between hybrid-perturbed and default approaches for *Brassica napus*-25

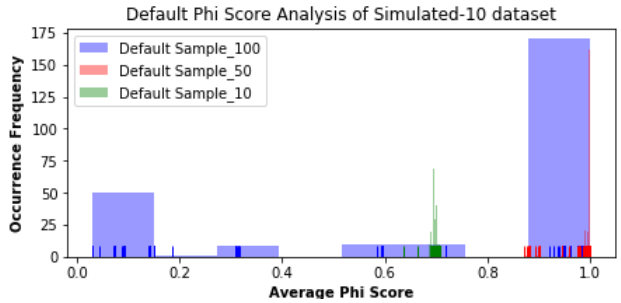


(d) Comparison between hybrid-perturbed and default approaches for *Brassica napus*-25

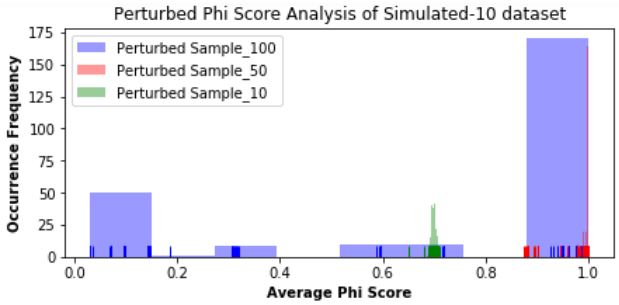
Figure 6.8: Comparison between hybrid-perturbed and default approaches for *Brassica napus*-25

6.2.2 Result on Simulated Datasets

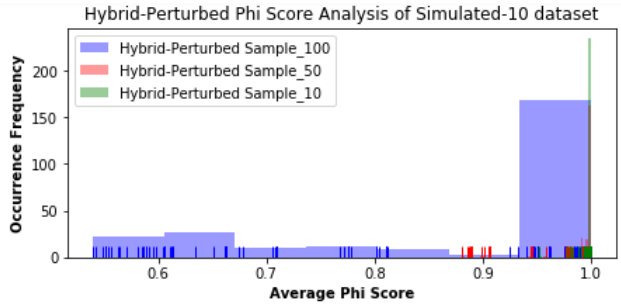
Using the same approach to analyze the Simulated-10, 20 and 30 datasets, we plotted the distribution of average ϕ scores to quantify the improvement achieved by the perturbed method. Fig. 6.9 shows the default, perturbed and the hybrid-perturbed approach results on Simulated-10 dataset. It can be seen 6.9b that the impact of perturbation is not as prominent. However, it becomes prominent in Fig. 6.9c where the hybrid-perturbed approach overcomes the default inference failure, resulting in a higher average ϕ score. The same impact has been shown by Simulated-20 and 30. With increasing number of taxa, the impact of hybrid perturbation becomes more prominent as it can be seen in 6.10c and 6.11c that the average ϕ score is close to 1.0 in majority of cases, which is significantly better than the default approach.



(a) Default Approach ϕ Distribution for Simulated-10 dataset

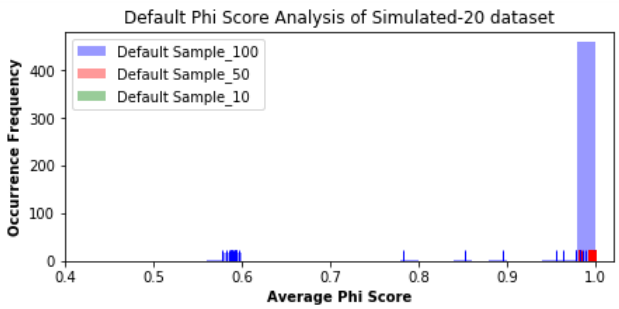


(b) Hybrid Approach ϕ Distribution for Simulated-10 dataset

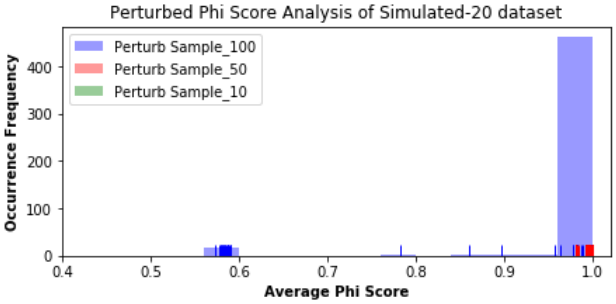


(c) Hybrid-Perturbed Approach ϕ Distribution for Simulated-10 dataset

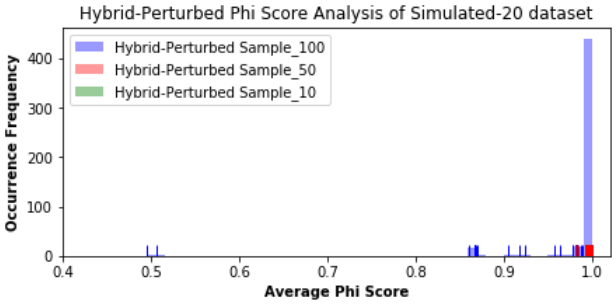
Figure 6.9: Analysis of Perturbation on Simulated-10 dataset



(a) Default Approach ϕ Distribution for Simulated-20 dataset

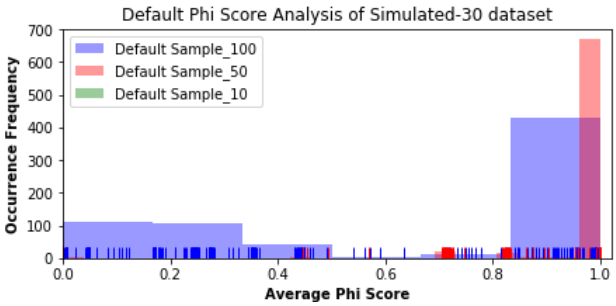


(b) Hybrid Approach ϕ Distribution for Simulated-20 dataset

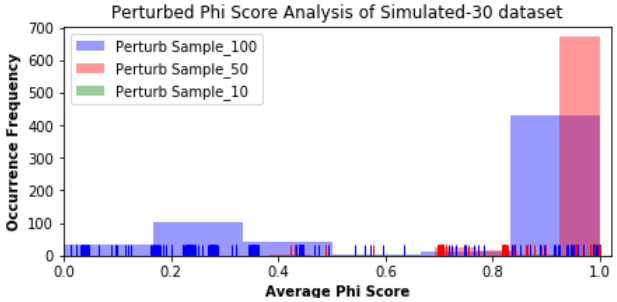


(c) Hybrid-Perturbed Approach ϕ Distribution for Simulated-20 dataset

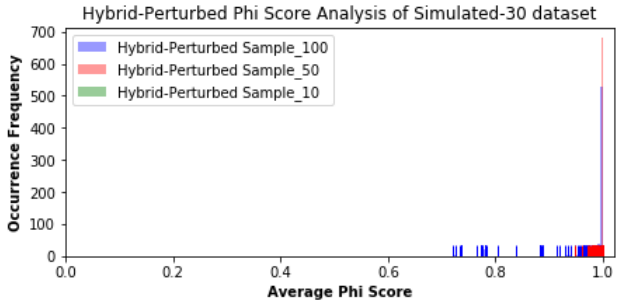
Figure 6.10: Analysis of Perturbation on Simulated-10 dataset



(a) Default Approach ϕ Distribution for Simulated-30 dataset



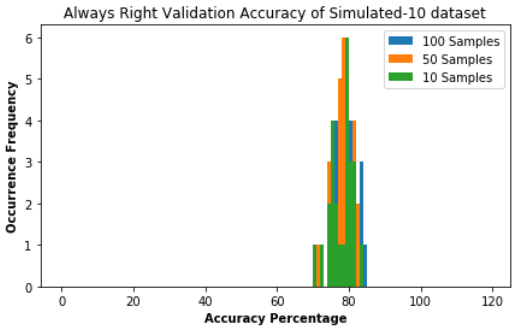
(b) Hybrid Approach ϕ Distribution for Simulated-30 dataset



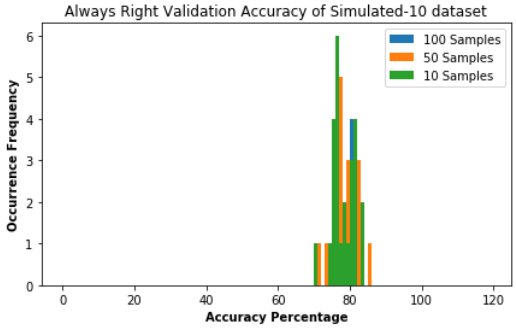
(c) Hybrid-Perturbed Approach ϕ Distribution for Simulated-30 dataset

Figure 6.11: Analysis of Perturbation on Simulated-10 dataset

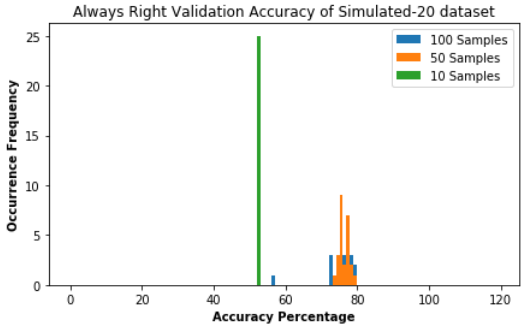
Fig. 6.12, 6.13 and 6.14 show the comparative validation accuracy analysis of default and perturbed approaches. The validation accuracy with respect to the ground truth remains slightly better than or the same as in some cases for the perturbed approach, showing that the higher average ϕ score does not always result in higher validation accuracy. It is to be noted that as the perturbation method relies on the default heuristic, where failure to infer a heuristic will result in higher validation accuracy when using the without zero accuracy computation methods, as an un-inferred sign pattern is represented by a zero pattern with respect to a focal taxon. Therefore, the failure for entire 10-sample class of Simulated-20 and Simulated-30 datasets resulted in a 100% validation accuracy for the without-zero method. The exact match method shows higher validation accuracy for 10-samples classification of Simulated-20 and Simulated-30 datasets as it matches with the zeros in the ground truth which constitutes 50% of the ground truth in that case.



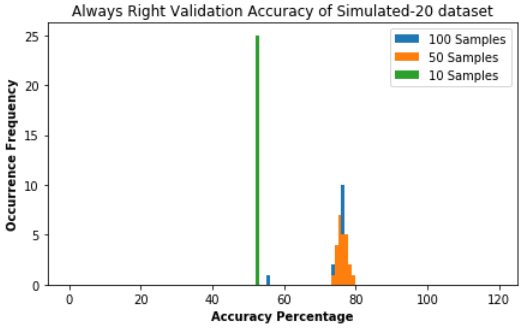
(a) Accuracy Distribution of Default Method on Simulated-10 dataset



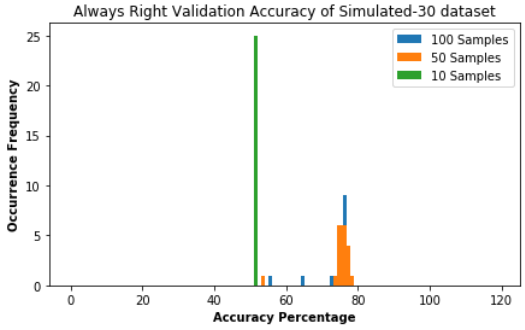
(b) Accuracy Distribution of Perturbed Method on Simulated-10 dataset



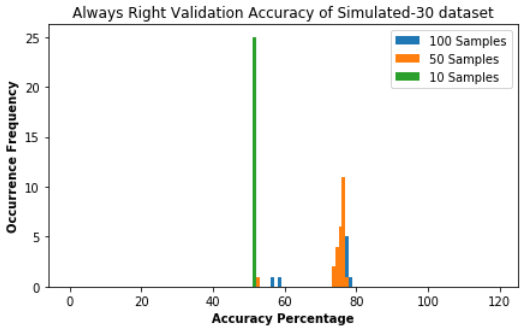
(c) Accuracy Distribution of Default Method on Simulated-20 dataset



(d) Accuracy Distribution of Perturbed Method on Simulated-20 dataset

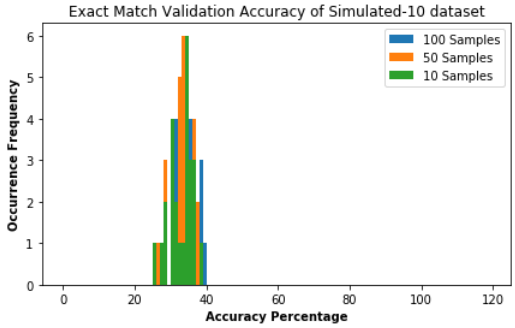


(e) Accuracy Distribution of Default Method on Simulated-30 dataset

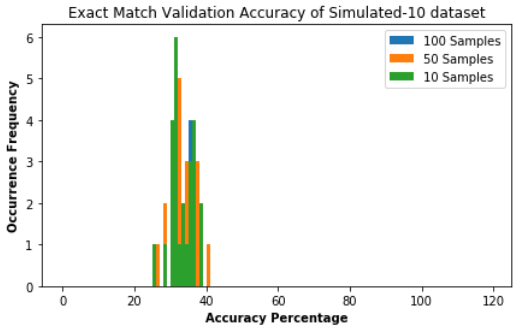


(f) Accuracy Distribution of Perturbed Method on Simulated-30 dataset

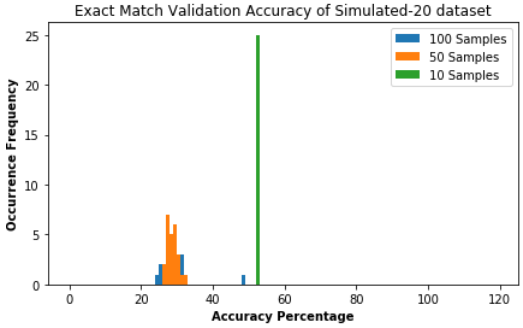
Figure 6.12: Accuracy Analysis Using the default accuracy computation method on Simulated datasets



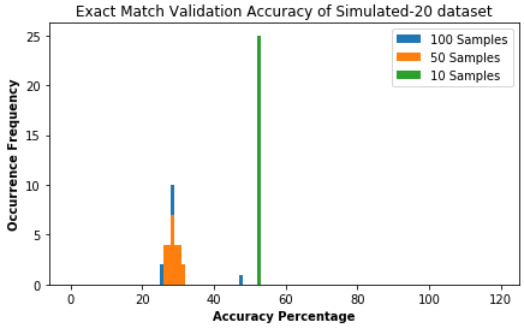
(a) Accuracy Distribution of Default Method on Simulated-10 dataset



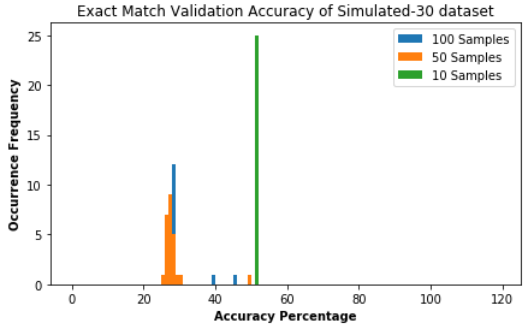
(b) Accuracy Distribution of Perturbed Method on Simulated-10 dataset



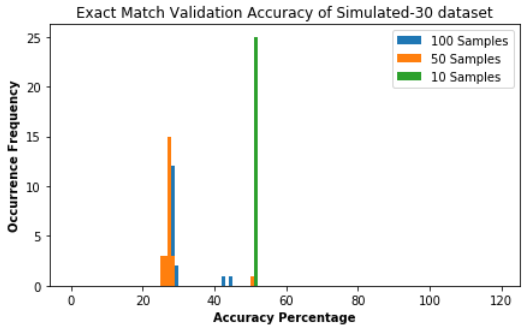
(c) Accuracy Distribution of Default Method on Simulated-20 dataset



(d) Accuracy Distribution of Perturbed Method on Simulated-20 dataset

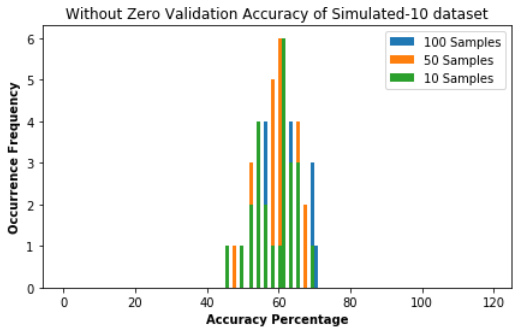


(e) Accuracy Distribution of Default Method on Simulated-30 dataset

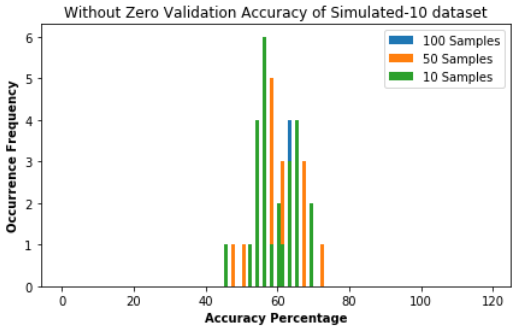


(f) Accuracy Distribution of Perturbed Method on Simulated-30 dataset

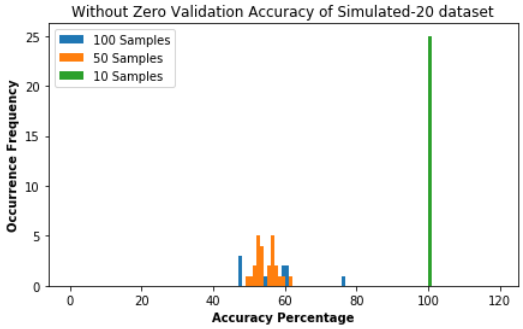
Figure 6.13: Accuracy Analysis Using the exact match accuracy computation method on Simulated datasets



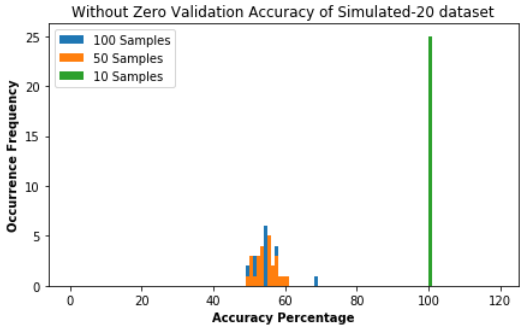
(a) Accuracy Distribution of Default Method on Simulated-10 dataset



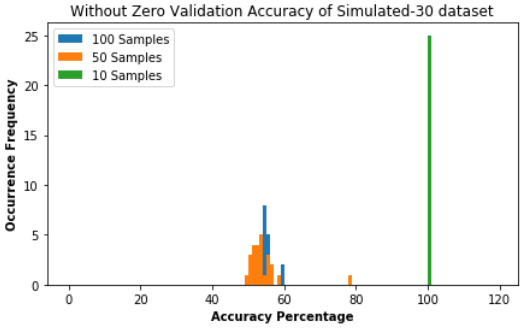
(b) Accuracy Distribution of Perturbed Method on Simulated-10 dataset



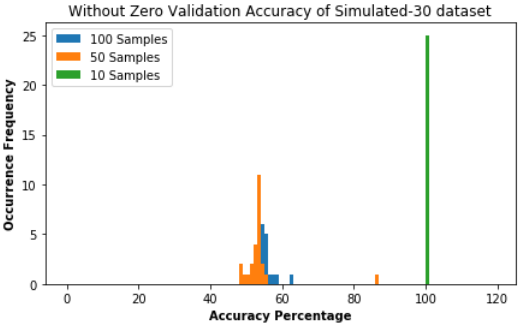
(c) Accuracy Distribution of Default Method on Simulated-20 dataset



(d) Accuracy Distribution of Perturbed Method on Simulated-20 dataset



(e) Accuracy Distribution of Default Method on Simulated-30 dataset



(f) Accuracy Distribution of Perturbed Method on Simulated-30 dataset

Figure 6.14: Accuracy Analysis Using the no zero inclusion accuracy computation method on Simulated datasets

6.3 Variance Analysis of Simple Heuristic and Variance-Controlled Heuristic Approaches

6.3.1 Result on Real Datasets

For *Brassica napus*-20 and *Brassica napus*-36 datasets , there was no inference made by the heuristic module making it impossible to compare the impact of the variance-controlled approach on these datasets. Inference for only 4 features of *Brassica napus*-25 dataset were made by the default heuristic method, while all features of Maize Roots dataset were inferred by the default heuristic. As shown in Fig. 6.15, we compared the mean variances of both datasets. It is evident in Fig. 6.15 that the variance controlled method achieves lower variance than the default method. There was no variance reported for the Maize Roots dataset over a 100 independent iterations of the algorithm, showing that each iteration inferred the exact same inferences for all taxa. For *Brassica napus*-25 dataset, the achieved variance of variance controlled method is slightly lower than the default method as the default heurisitc failure rate is really high for this dataset.

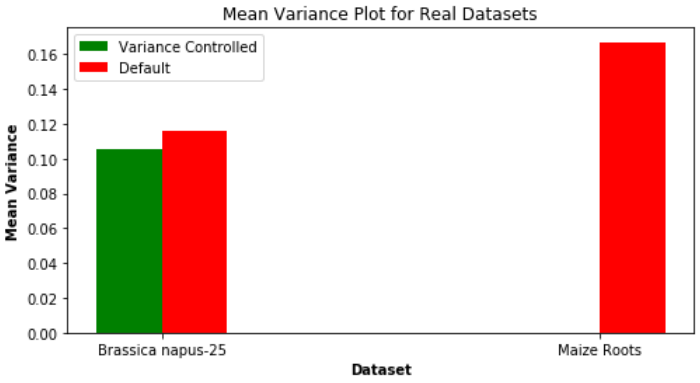
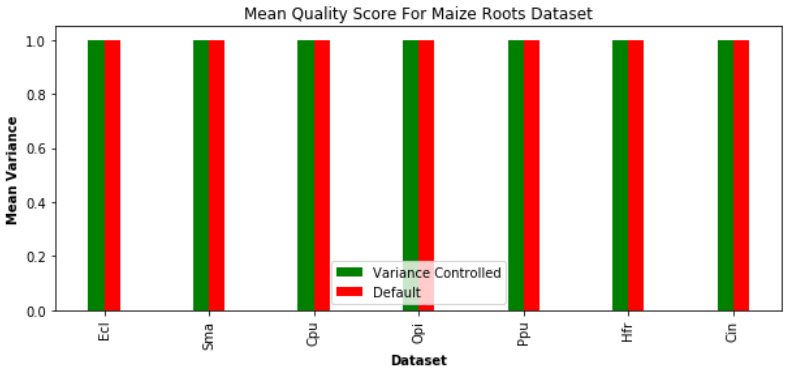
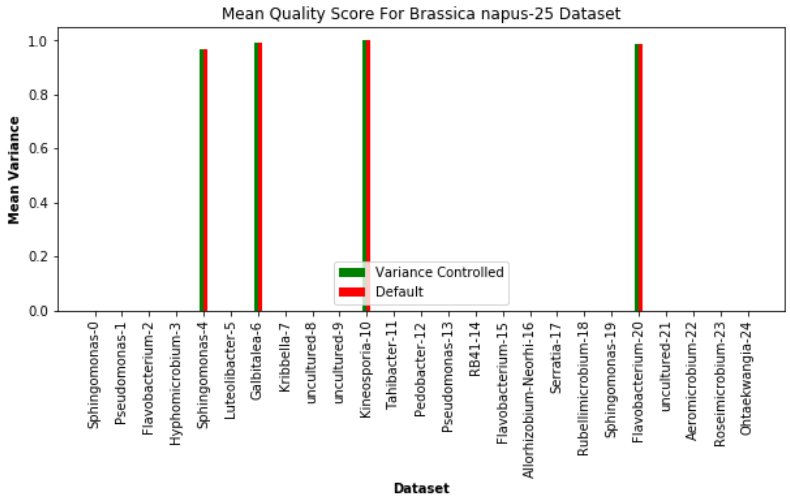


Figure 6.15: Shows the impact of variance control on Real datasets

Fig. 6.16a shows the ϕ score distribution of Maize Roots and four features of *Brassica napus*-25 dataset in Fig. 6.16b. For *Brassica napus*-25 dataset, ϕ score of only those features are compared where the heuristic module did not fail, it should also be noted that the variance control method is dependent on the inference made by either default or hybrid method to perform the variance control. For both datasets, the ϕ score of inferences remain exactly the same, showing that the variance control method is able to achieve lower variance without affecting the quality of proposed inferences.



(a) ϕ for each feature of Maize Root dataset

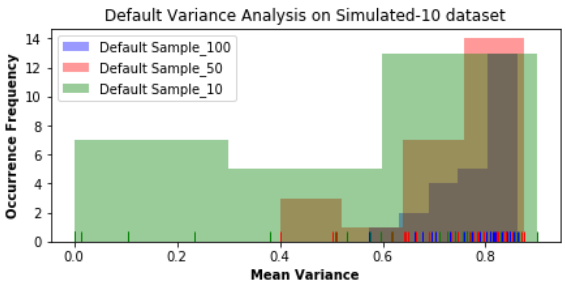


(b) ϕ for each feature of *Brassica napus*-25 dataset

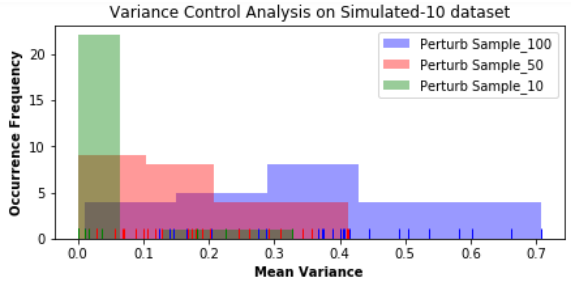
Figure 6.16: Per feature maximum ϕ score of real datasets

6.3.2 Result on Simulated Datasets

We tested the impact of variance control on all simulated datasets. Fig. 6.17, 6.18 and 6.19 show the comparison of variance on Simulated-10, 20 and 30 datasets. It is evident from the figures that the variance control method is more consistent than the default method on all simulated datasets. For the Simulated-10 and Simulated-20 datasets, as shown in Fig. 6.17b, 6.18b respectively, the upper limit of variance from the variance control method is lower than the default method. The distribution of variance is concentrated towards the lower values of variance in the variance control method, as compared to the default method. For Simulated-30 dataset as shown in Fig. 6.19b, even though the maximum variance achieved by both methods are the same, the variance achieved by variance control method is concentrated towards lower variance values.

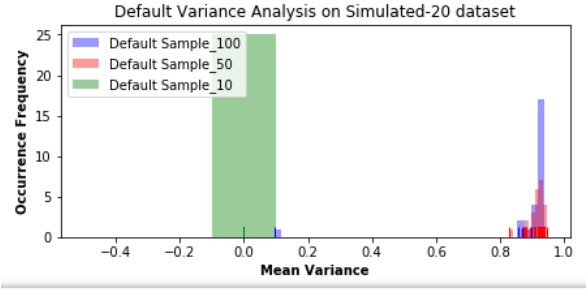


(a) Overall variance for Simulated-10 dataset using the default method

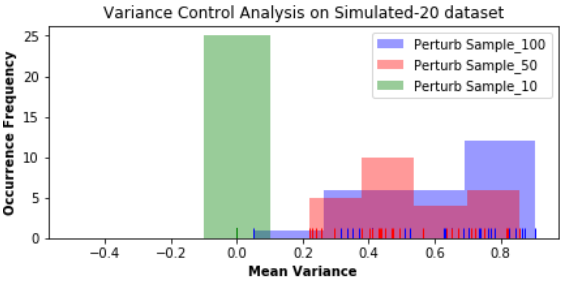


(b) Overall variance for Simulated-10 dataset using variance control

Figure 6.17: Comparative Variance Analysis of Simulated-10 dataset

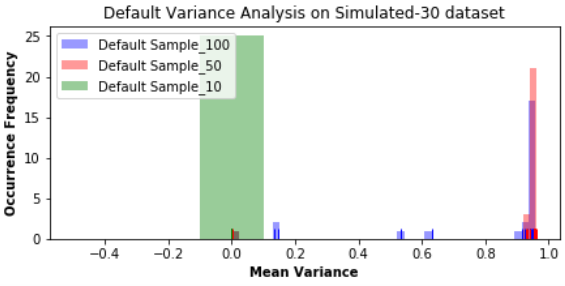


(a) Overall variance for Simulated-20 dataset using the default method

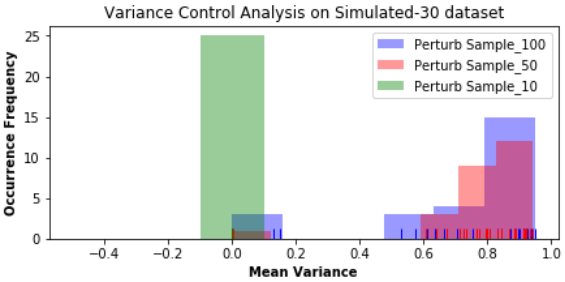


(b) Overall variance for Simulated-20 dataset using variance control

Figure 6.18: Comparative Variance Analysis of Simulated-20 dataset



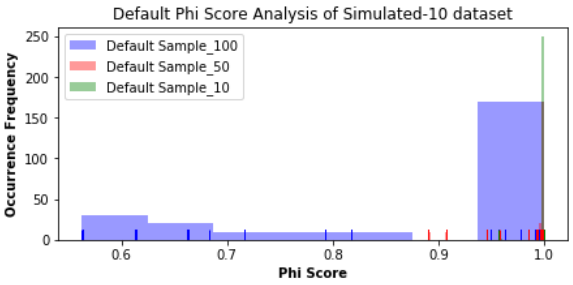
(a) Overall variance for Simulated-30 dataset using the default method



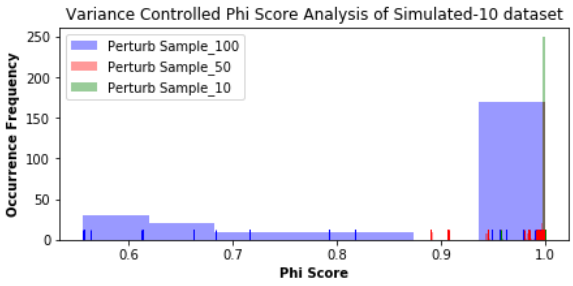
(b) Overall variance for Simulated-30 dataset using variance control

Figure 6.19: Comparative Variance Analysis of Simulated-30 dataset

As shown in Fig. 6.20, 6.21 and 6.22 the ϕ score achieved through the variance control method is almost the same as the default method of heuristic inference, showing that the variance control method was able to achieve lower variance.

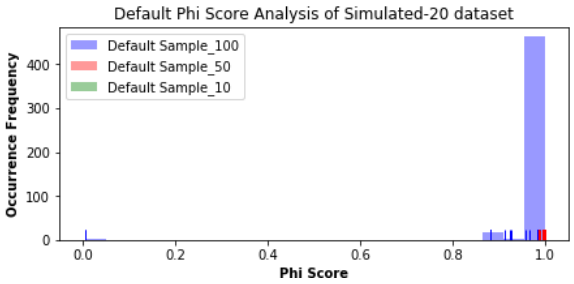


(a) Overall ϕ for Simulated-10 dataset using the default method

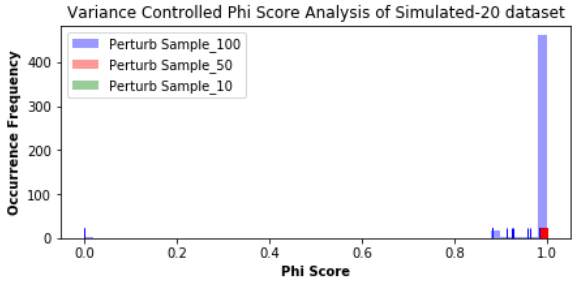


(b) Overall ϕ for Simulated-10 dataset using variance control

Figure 6.20: Comparison of ϕ score between Variance Control and Default methods on Simulated-10 dataset

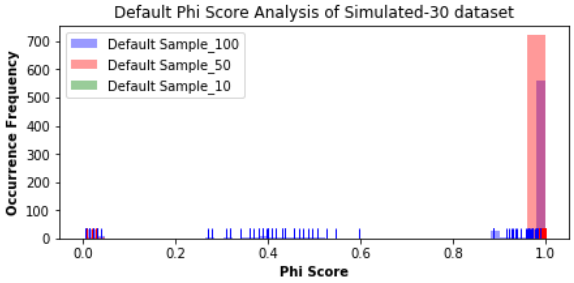


(a) Overall ϕ for Simulated-20 dataset using the default method

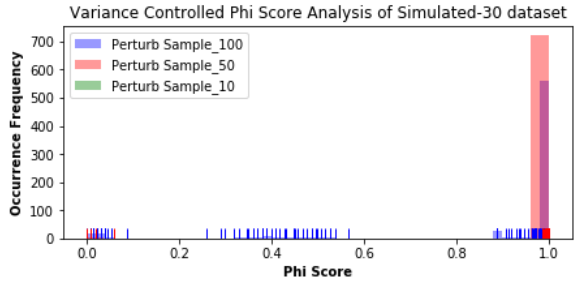


(b) Overall ϕ for Simulated-20 dataset using variance control

Figure 6.21: Comparison of ϕ score between Variance Control and Default methods on Simulated-10 dataset



(a) Overall ϕ for Simulated-30 dataset using the default method

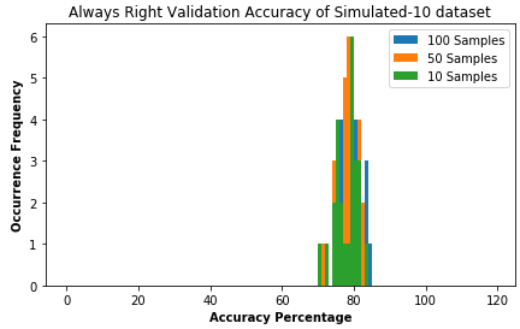


(b) Overall ϕ for Simulated-20 dataset using variance control

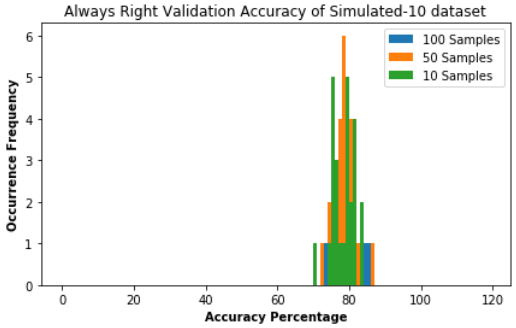
Figure 6.22: Comparison of ϕ score between Variance Control and Default methods on Simulated-30 dataset

Fig. 6.23, 6.24 and 6.25 shows the comparison of validation accuracy of the default and the variance control method. In case of the Simulated-10 dataset, the accuracy achieved by the variance-controlled method is slightly better than the default method which is shown in Fig. 6.23b, 6.24b and 6.25b. For Simulated-20 dataset, the accuracy achieved by the without zero method is slightly better for the default method than the

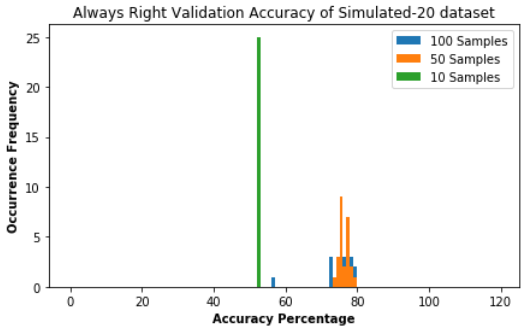
variance control method as shown in 6.25c and d. However, the overall accuracy remains almost the same in other cases for the Simulated-20 and 30 datasets. This shows that the variance control method is able to achieve a lower variance while keeping the quality (ϕ) and the accuracy of inferences remain almost the same as the default method.



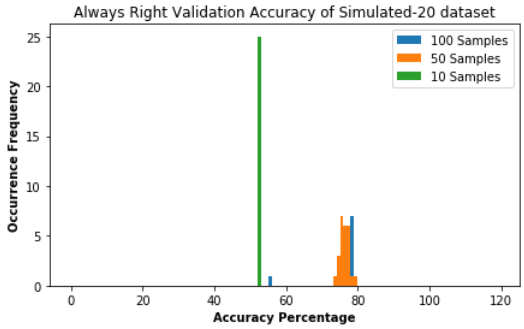
(a) Accuracy Distribution of Default method on Simulated-10 dataset



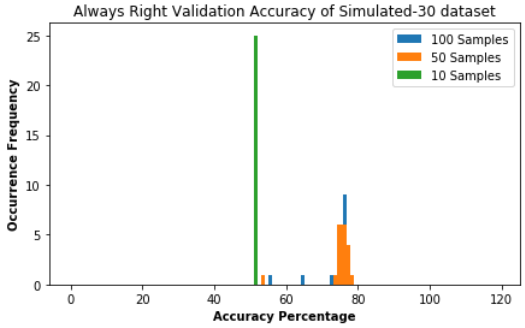
(b) Accuracy Distribution of Variance Control method on Simulated-10 dataset



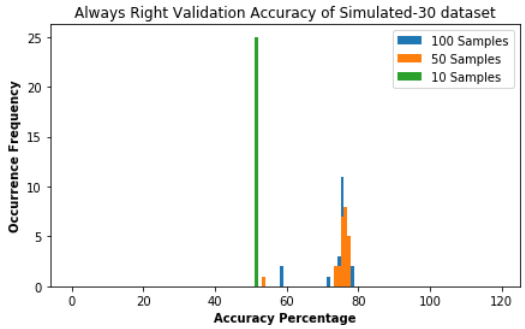
(c) Accuracy Distribution of Default method on Simulated-20 dataset



(d) Accuracy Distribution of Variance Control method on Simulated-20 dataset

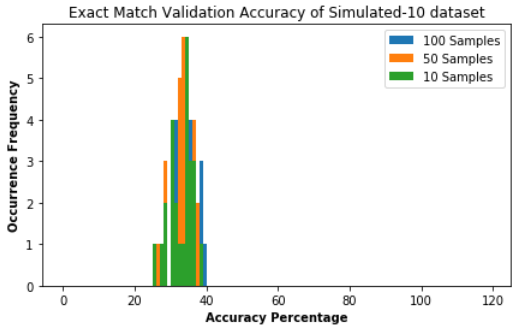


(e) Accuracy Distribution of Default method on Simulated-30 dataset

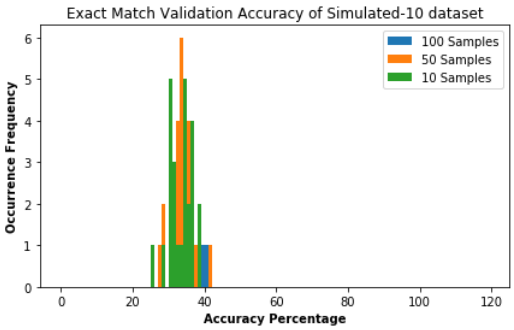


(f) Accuracy Distribution of Variance Control method on Simulated-30 dataset

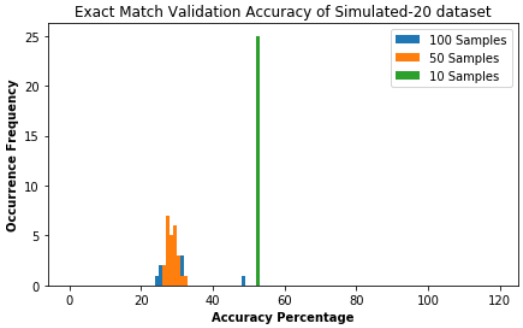
Figure 6.23: Accuracy Analysis using the default accuracy computation method on simulated datasets



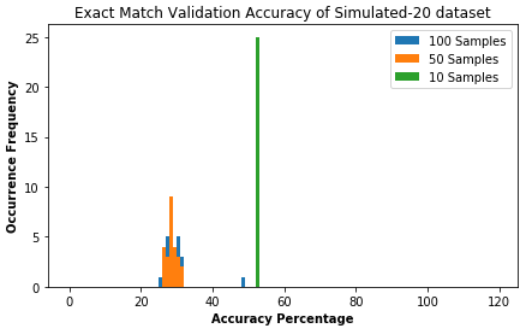
(a) Accuracy Distribution of Default method on Simulated-10 dataset



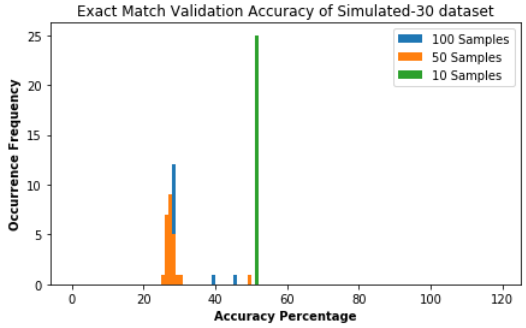
(b) Accuracy Distribution of Variance Control method on Simulated-10 dataset



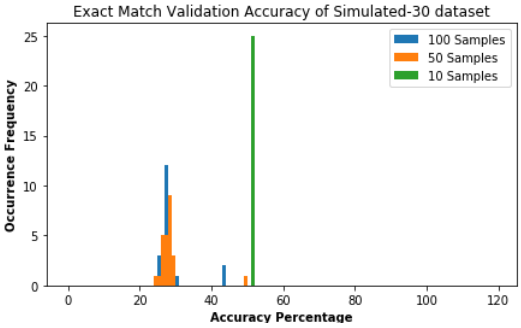
(c) Accuracy Distribution of Default method on Simulated-20 dataset



(d) Accuracy Distribution of Variance Control method on Simulated-20 dataset

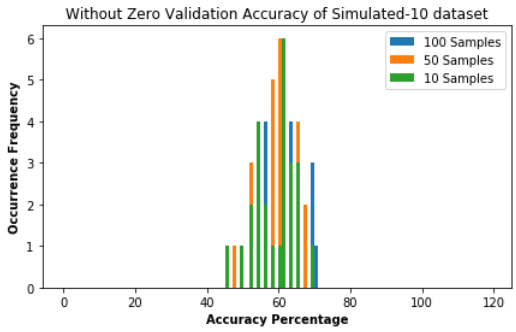


(e) Accuracy Distribution of Default method on Simulated-30 dataset

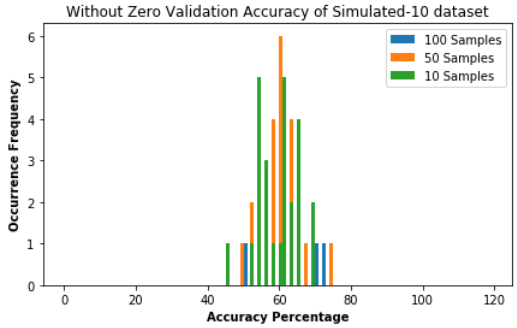


(f) Accuracy Distribution of Variance Control method on Simulated-30 dataset

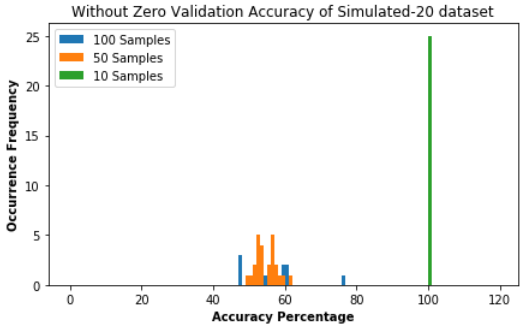
Figure 6.24: Accuracy Analysis using the default accuracy computation method on simulated datasets



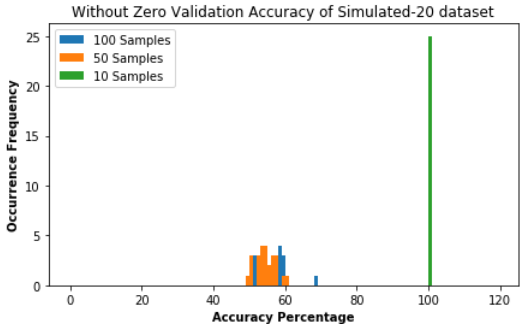
(a) Accuracy Distribution of Default method on Simulated-10 dataset



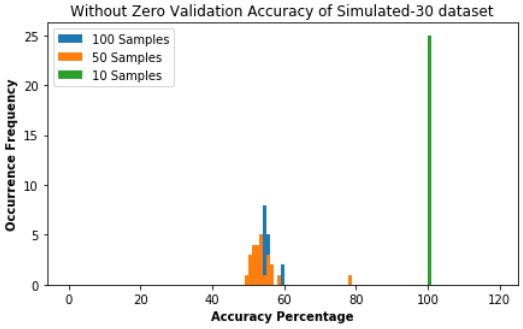
(b) Accuracy Distribution of Variance Control method on Simulated-10 dataset



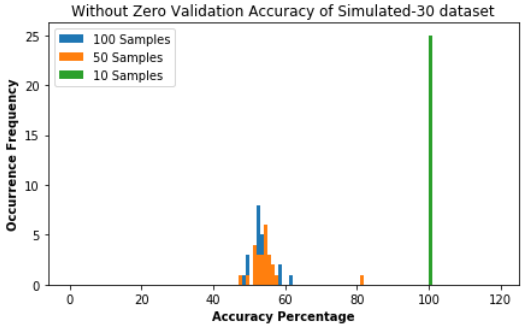
(c) Accuracy Distribution of Default method on Simulated-20 dataset



(d) Accuracy Distribution of Variance Control method on Simulated-20 dataset



(e) Accuracy Distribution of Default method on Simulated-30 dataset



(f) Accuracy Distribution of Variance Control method on Simulated-30 dataset

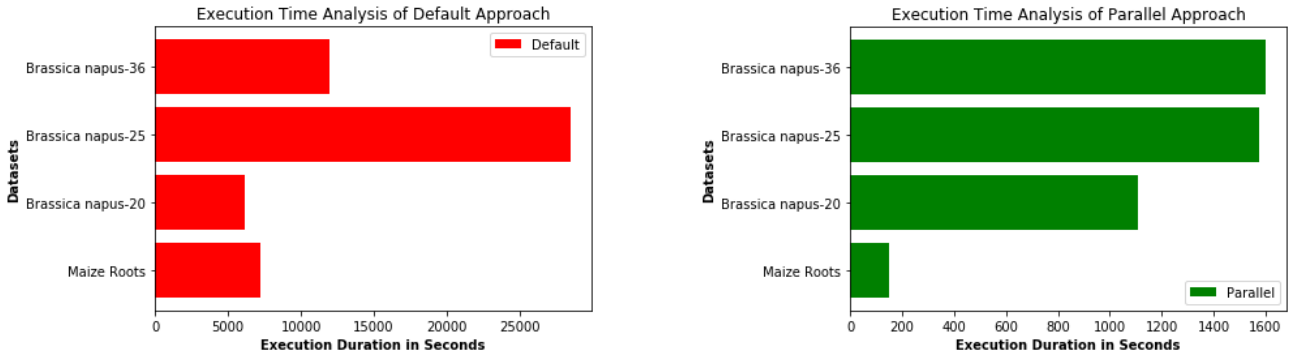
Figure 6.25: Accuracy Analysis using the default accuracy computation method on simulated datasets

6.4 Analysis of Simple and Parallel Algorithm Execution Approaches

6.4.1 Result on Real Datasets

Fig. 6.26 shows the comparison algorithm execution duration on simulated datasets. We executed a 100 independent iterations of both default and improved approaches to avoid minor differences in the execution time of independent iteration caused by the hardware resource allocation. It is evident from Fig. 6.26b that the parallel approach is more than 10 times faster than the default approach in each dataset. The Fig. 6.26a

shows that the default approach takes longer duration to complete the execution of *Brassica napus*-25 dataset than *Brassica napus*-36. This effect is caused by the increased failure rate of *Brassica napus*-36 dataset as it fails to infer a sign pattern for any taxon through the default approach, while the sign patterns for 4 taxa of *Brassica napus*-25 dataset were consistently inferred by the default approach. On the other hand, the execution time duration of parallel approach increases steadily with the increasing number of taxa as it processes each taxon parallelly and the completion of independent iterations is dependent on the complete execution of all individual taxa.



(a) Execution Time of the Default Approach

(b) Execution Time of the Parallel Approach

Figure 6.26: The execution time analysis on real datasets for Default and Parallel approaches

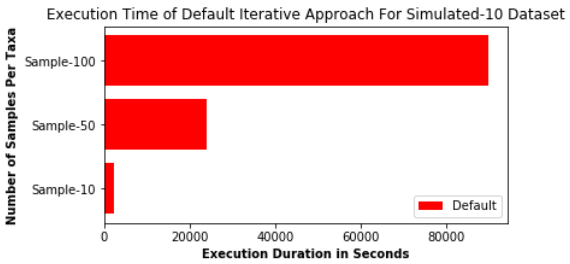
6.4.2 Result on Simulated Datasets

As the default approach takes a long time to complete an end to end execution of one hundred iteration, we only executed the default approach once for each sample classification type of 10, 50 and 100 samples of Simulated-10 dataset and compared the performance of parallel approach on the same dataset. To capture the overall impact of increasing number of taxa, we executed only the parallel approach on all three simulated datasets for all datasets of each sample classification type.

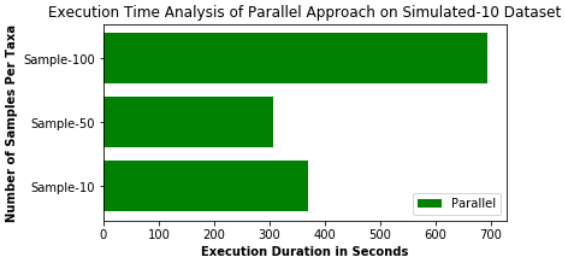
For the comparative analysis on Simulated-10 dataset, we captured the impact of increasing active abundance samples while keeping the same number of taxa for a 100 independent iterations. It can be seen in Fig. 6.27 that the parallel approach outperforms the default approach by a significant margin. As shown in Fig. 6.27a, the execution time duration increases steadily with the increasing number of active abundance samples, showing that the iterative approach takes longer to compute the quality measure (ϕ) because of increased number of differential hyperplanes, which in turned is caused by an increasing number of active abundance samples. Increased duration for the computation of the quality measure causes longer execution times, which slows down the end to end independent iteration completion time due to sequential execution.

On the other hand, as shown in 6.27b, the execution time duration for both 10 and 50 sample size classification of Simulated-10 dataset are somewhat similar, which is a result using matrix multiplication instead of iterative computation of quality measure. The Simulated-10 dataset with 10 samples takes slightly longer than the one with 50 samples, which is a result of parallel thread execution overhead. For small

datasets, with fewer taxa and active abundance samples, using the parallel approach may take slightly longer per taxa than larger datasets, but will remain faster than the default iterative approach.



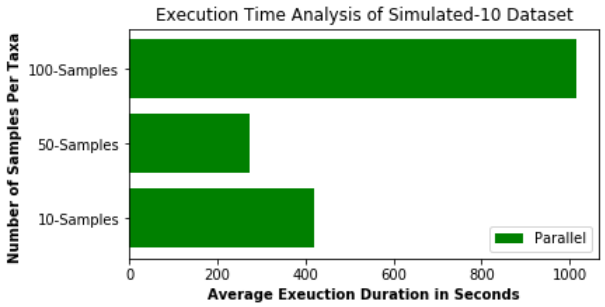
(a) Execution Time of the Default Approach



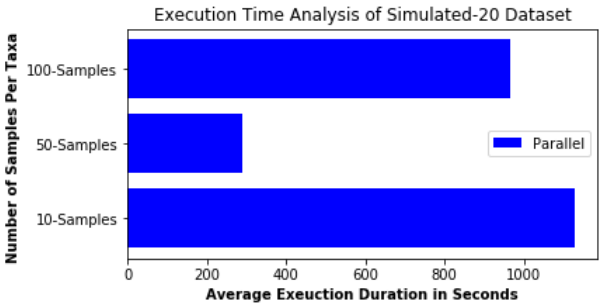
(b) Execution Time of the Parallel Approach

Figure 6.27: The execution time analysis on one instance Simulated-10 dataset for Default and Parallel approaches

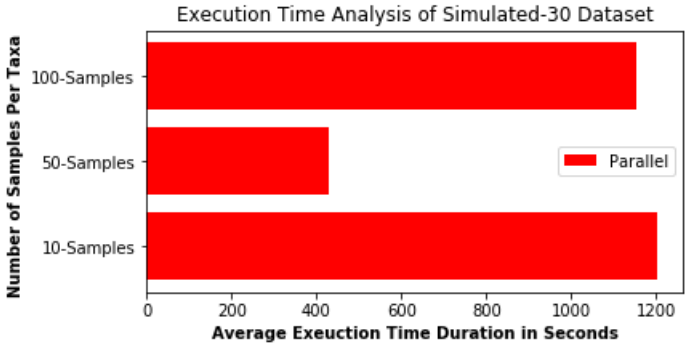
We also computed the average execution time duration as shown in Fig. 6.28. We computed an average execution time over the 25 datasets of each sample classification type for Simulated-10, 20 and 30 datasets. It can be seen that with the increasing number of taxa, the maximum overall execution time remains almost the same, which shows the scalability of the parallel execution approach up to the limit of CPU cores of the system. However, it is also observed that execution time duration of 10 sample class of Simulated-20 and Simulated-30 datasets is more than the 100 sample class, which is caused by the high failure rate and thread overhead caused by the termination of threads. This also shows that for datasets with few active abundance samples, the parallel execution approach may slow, but it will always remain faster than the iterative approach.



(a) Average Execution Time of Parallel Approach on Simulated-10 Dataset



(b) Average Execution Time of Parallel Approach on Simulated-20 Dataset



(c) Average Execution Time of Parallel Approach on Simulated-30 Dataset

Figure 6.28: Average Execution Time of Parallel Approach on Simulated datasets

6.5 Analysis of Default and Accelerated Brute Force Implementations

Brute force is the only method which is guaranteed to find the best solution in the $N3^N$ search space. However, the execution duration of the algorithm is limiting. In this section, we show our comparative analysis of brute-force approach proposed in research [13] and our accelerated brute-force approach, along with the efficacy of the brute-force with respect to the ground truth.

6.5.1 Maize Roots Dataset

Table 6.1 shows the distribution of brute-force algorithm execution time proposed in the research [13] and its comparison with the accelerated brute-force method. We estimated the time duration of the default brute-force method through the execution time of default iterative heuristic method. For the default heuristic method, the time taken for the evaluation of single sign pattern remains the same as the brute-force method, but only a user specified number of sign pattern are tested, instead of $N3^N$ sign patterns. It is evident from the result that the accelerated brute-force method is more than 200 times faster than the former brute-force approach, while achieving the same results. The speed-up shows the increased capacity to perform accelerated brute-force on larger community sizes ($N > 10$). As our brute-force parallel execution method processes each

taxa in parallel, the overall time taken is equivalent to the time taken by the taxon with longest execution duration, as all other taxa complete their execution with-in that duration without hindering the processing of that one taxon due to the availability of more CPU cores than taxa for the Maize Roots dataset.

Table 6.1: Execution time distribution of default vs the accelerated brute-force method for Maize Root dataset

Description	**Mathematical Explanation**
Time taken by 100 iterations of default Brute Force	7193 sec
Time taken for a single iterations out of 100 total iterations by Brute Force	$7193/100 = 71.93$ sec
Average Time for the processing of a total 7 taxa for default Brute Force	$71.93/7 = 10.27$ sec
Average Time for the processing of 100 sign pattern by default Brute Force	$10.27/100 = 0.102$ sec
Estimated Brute Force time to check 3^7 sign pattern per taxon	$0.102 * 3^7 = 224.72$ sec
Estimated Brute Force time to check 3^7 sign pattern for 07 taxa	$224.72 * 7 = 1573.04$ sec
Estimated default Brute Force time for one end to end iteration	$1573.04/60 = 26.217$ **min**
Accelerated Brute Force time for one end to end iteration	5 **sec or** 0.0833 **min**

We conducted a test to quantify the viability of the accelerated brute-force solution with respect to the ground truth. Fig. 6.29 shows the difference between the quality scores (ϕ) of accelerated brute-force method and its corresponding ground truth. The difference between the quality scores is low and it is always with-in the 2.5% margin of error. This shows that the accelerated brute-force method has the ability to accurately infer ground truth sign patterns within an acceptably small error range.

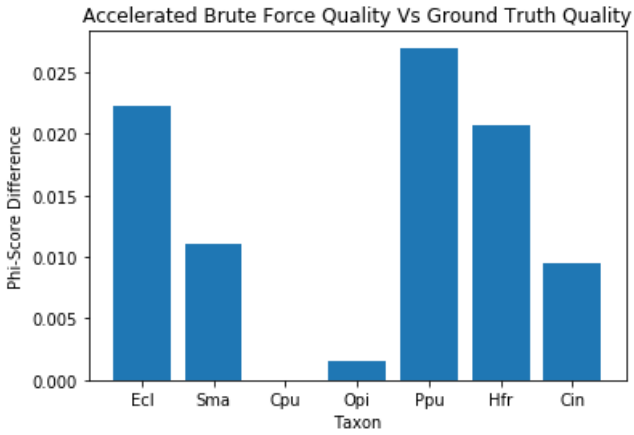


Figure 6.29: Difference between the quality score ϕ of proposed and ground truth sign patterns on Maize Roots dataset

6.5.2 Simulated Dataset

We tested the performance of accelerated brute-force method on the simulated data. Our primary purpose is to compare the speed-up achieved by the accelerated brute-force method against the default iterative approach. Table 6.2 shows the breakdown of comparison of between the default and the accelerated brute-force methods. It is evident that the accelerated brute-force method outperforms the default brute-force method. For the simulated datasets, only the number of active abundance samples increases. The differential hyperplane reduction step provides the accelerated brute-force a significant speed-up on top of the speed-up achieved by parallelism itself.

Similarly, for the simulated datasets we also computed the differences between the computed ϕ score and the ϕ score of ground truth. To quantify the extend of variation, we performed a sensitivity analysis on quality scores of ground truth sign patterns of Simulated-10 with 10, 50 and 100 sample sizes, as it corresponds closely to the real dataset used for accelerated brute-force evaluation. As we have 25 datasets for each classification of sample range, we mapped a histogram of differences of ϕ score between the ground truth and the inferred sign pattern for each classification type. As shown in Fig. 6.30, the most frequent difference is 0.0, for each classification of sample size, which shows that in most cases, there was no difference between the ϕ of accelerated brute-force method and ground truth, demonstrating the accuracy of the accelerated brute-force method. The outliers are mostly caused by the arithmetic underflow and small values of active abundances (less than $1e-05$) which get rounded off to zero.

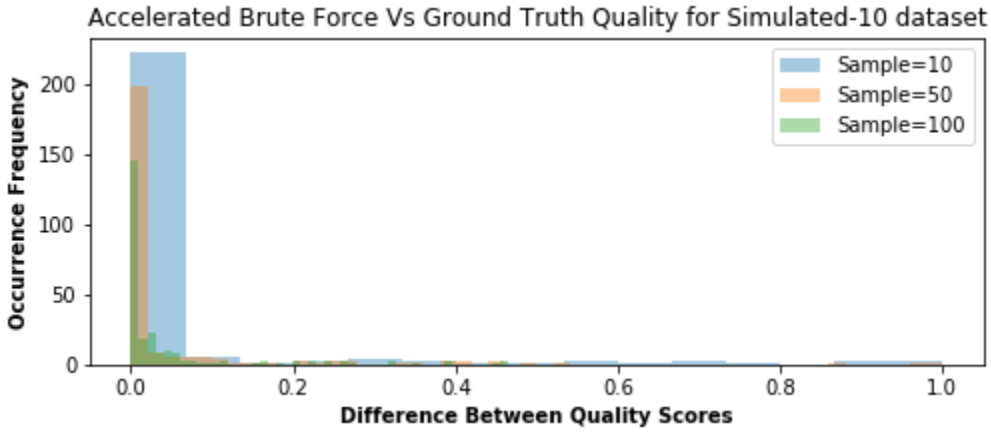


Figure 6.30: Sensitivity Analysis on 25 datasets for each 10, 50 and 100 sample sizes Simulated-10 dataset

Table 6.2: Execution time distribution of default vs the accelerated brute-force algorithm method on Simulated-10 dataset

Description	**10 Samples**	**50 Samples**	**100 Samples**
100 iterations of default Brute Force	2410 sec	24092	89914 sec
for a single iterations out of 100 total iterations by Brute Force	24.1 sec	240.92	899.14
Average Time for the processing of total 10 taxa for default Brute Force	2.41 sec	24.092 sec	89.914 sec
Average Time for the processing of 100 sign pattern by default Brute Force	0.0241 sec	0.2409 sec	0.89914 sec
Estimated Brute Force time to check 3^{10} sign pattern per taxon	1423.08 sec	14226.085 sec	3093.31786 sec
Estimated Brute Force time to check 3^{10} sign pattern for 10 taxa	1423.809 sec	42260.85 sec	530933.1786 sec
Estimated default Brute Force time for one end to end iteration	237.18 **min**	2371.01 **min**	8848.88 **min**
Accelerated Brute Force time for one end to end iteration	10 **sec or** 0.167 **min**	23 **sec or** 0.3834 **min**	40 **sec or** 0.6667 **min**

We also conduced an equivalency test between matrix multiplication based sign satisfaction and graph based sign satisfaction of sign patterns as proposed in the research [13] for all simulated dataset. As per our expectation, the quality score for all 3^{10} sign patterns of each dataset with 10, 50 and 100 samples remained exactly the same as specified in eq 4.1, proving that both methods of sign pattern evaluation are the same.

Fig. 6.31 shows the distribution of maximum ϕ score achieved by the brute-force algorithm while, Fig. 6.32 shows the distributions of number of sign patterns with maximum ϕ score for both first selection and voting based method on Simulated-10 dataset. The maximum quality scores remains 1.0 in most cases. Both methods of pattern selection have a distribution of sign patterns that remains the same as each method uses a different pattern selection technique out of the distribution of max ϕ patterns, but the number of maximum ϕ sign patterns remain the same as if it is not dependent on the method of pattern selection. Fig. 6.31 also shows that there are a large number of patterns with maximum ϕ scores.

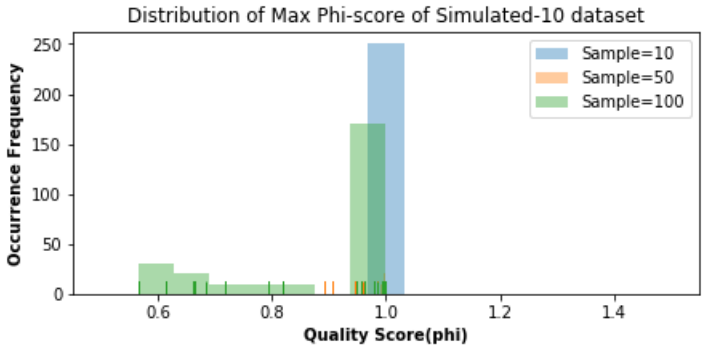
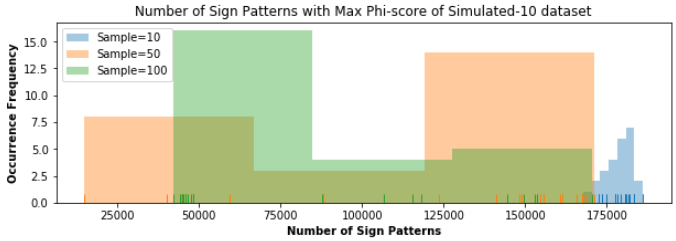
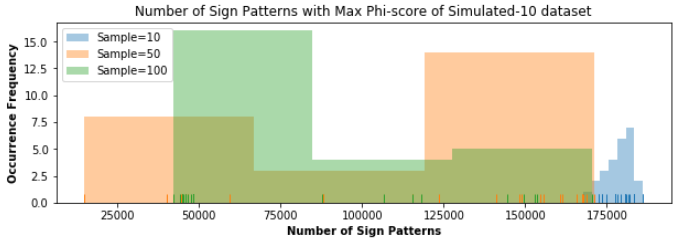


Figure 6.31: Sensitivity Analysis on 25 datasets for each 10, 50 and 100 sample sizes Simulated-10 dataset



(a) Distribution of Number of Sign Patterns with maximum ϕ score for first selection method



(b) Distribution of Number of Sign Patterns with maximum ϕ score for voting based method

Figure 6.32: Distribution of Number of Sign Patterns with maximum ϕ score for both sign pattern selection methods on Simulated-10 dataset

6.6 Validation Analysis of Accelerated Brute Force Approach

We implemented two approaches for the brute-force method, one that selects the first sign pattern (first selection) achieving the maximum ϕ score, while the other performs frequency-based voting of sign patterns with maximum ϕ. We used three approaches to compute the accuracy of an inferred sign pattern with respect to the ground truth.

As each sign pattern with maximum ϕ score is considered a viable solution for a focal taxon, we also mapped a distribution of accuracy scores achieved by the sign patterns having maximum ϕ. The allows us to quantify the maximum achievable accuracy of brute-force method independent of the two methods of sign pattern selection.

6.6.1 Maize Roots dataset

By using the default method of accuracy computation, we achieve a validation accuracy of 59.18% from first selection method, while the voting based method achieves a validation accuracy of 53.02%. For the exact match validation criteria, we achieve a validation accuracy of 57.14% for the first selection method, while we

achieve a validation accuracy of 51.02% for voting based method. Similarly for the method where zeros are not accounted for, the validation accuracy for the first pattern selection method is 58.33%, and the validation accuracy for voting based method is 52.08%. Overall, the validation accuracy achieved through the default accuracy evaluation method using the first pattern selection method is the highest, while the voting based method attains lower validation accuracy than the first pattern selection method, but neither of them are acceptable. This indicates a key shortcoming of accuracy computation method proposed in the research [13] as this method can identify the actual sign pattern, but can be confounded by false positives.

Fig. 6.33 shows the maximum achievable exact match validation accuracy for the Maize Roots dataset. Both voting based and first selection methods performed exactly the same, achieving the lowest validation accuracy of 57% for the Ecl taxon while achieving more than 90% for Cpu taxon. The average maximum accuracy across all taxa remains as high as 81.6% which shows that the brute-force method has the capability to find sign patterns with a validation accuracy of more than 80%. However, it is dependent on the method of selection of sign pattern from the available maximum ϕ sign patterns.

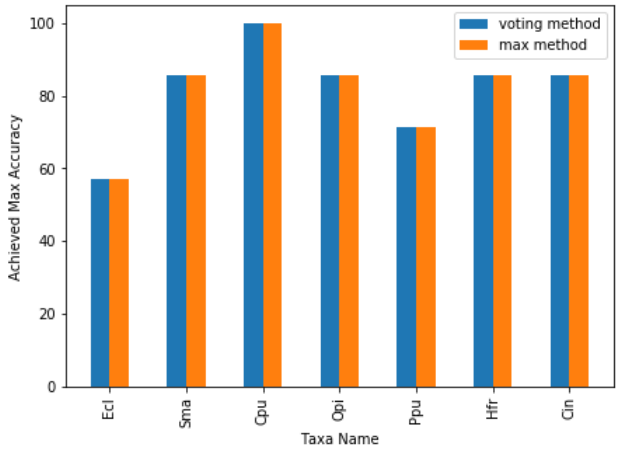
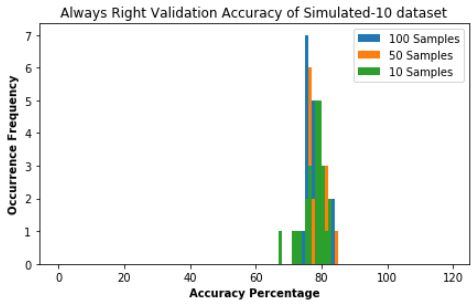


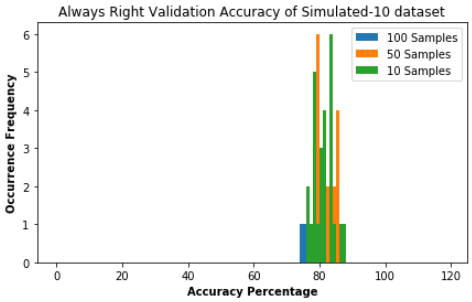
Figure 6.33: Validation accuracy of brute-force method on voting based and first maximum occurrence approaches

6.6.2 Simulated Dataset

Fig. 6.34 shows the distribution of validation accuracy of Simulated-10 dataset on its three classifications of 10, 50 and 100 samples, using first selection and voting based pattern selection methods. It can be seen in Fig. 6.34a that the default method of validation accuracy evaluation achieves the highest validation accuracy of 82.5% with most patterns having an accuracy of around 76%. On the other hand, Fig. 6.34b shows that the voting based validation accuracy method has most patterns with the validation accuracy ranging between 78 to 81%, while the lowest and the highest accuracy of the voting based method are also higher than the first selection method.



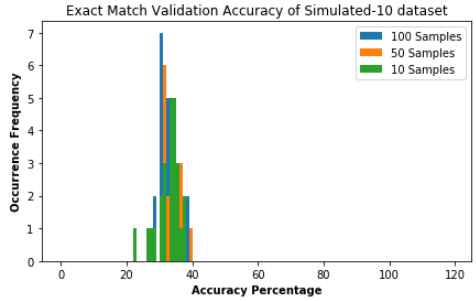
(a) Distribution of Validation Accuracy using the default accuracy computation method and First Selection criteria



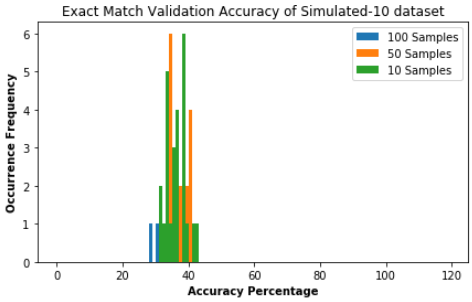
(b) Distribution of Validation Accuracy using the default accuracy computation method and Voting Based Selection criteria

Figure 6.34: Distributions of Validation Accuracy on Simulated Dataset with 10 taxa and 10, 50 and 100 samples

The validation accuracy achieved through the exact match validation computation criteria is lower than both of other methods of accuracy computation. As shown in Fig. 6.35, the validation accuracy achieved through the voting based pattern selection method is better than the first selection method in all aspects but with a small margin.



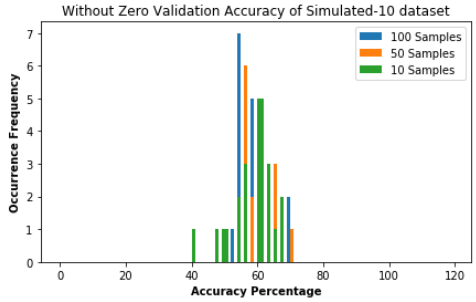
(a) Distribution of Validation Accuracy using the Exact Match and First Selection



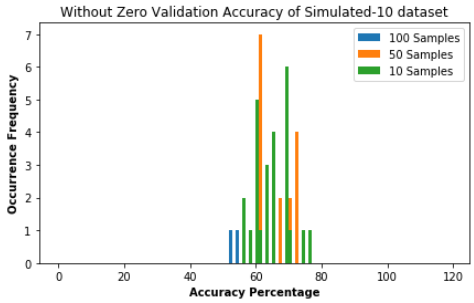
(b) Distribution of Validation Accuracy using the Exact Match and Voting based Selection

Figure 6.35: Distributions of Validation Accuracy on Simulated Dataset with 10 taxa and 10,50 and 100 samples

The achieved accuracy is between the default and the exact match method as shown in Fig. 6.36 for the without zero method. The validation accuracy achieved through the voting based method remains better. This shows that the validation accuracy achieved through the voting based method is slightly better than the first selection method irrespective of the method of accuracy computation, which shows the improved capability of voting based brute-force method in terms of accurate solution inference.



(a) Distribution of Validation Accuracy using the default method of accuracy and First Selection



(b) Distribution of Validation Accuracy using the default method of accuracy and Voting Based Selection

Figure 6.36: Distributions of Validation Accuracy on Simulated Dataset with 10 taxa and 10,50 and 100 samples

6.7 Analysis of Variable Block Size Brute Force Approach

6.7.1 Result on Real Datasets

Fig. 6.37 shows the comparison of the quality measure (ϕ) of sign patterns achieved for each taxon of the Maize Roots dataset. For this implementation, we used two blocks of size three and four for the independent execution of the brute-force method. It is evident in the plot 6.37 that the block-wise brute-force method achieved the same quality sign patterns as the accelerated brute-force approach. The execution time duration of the block-wise method is four seconds which is 20% faster than the accelerated brute-force method, showing the increased efficiency of the block-wise method, while achieving the same quality of sign patterns.

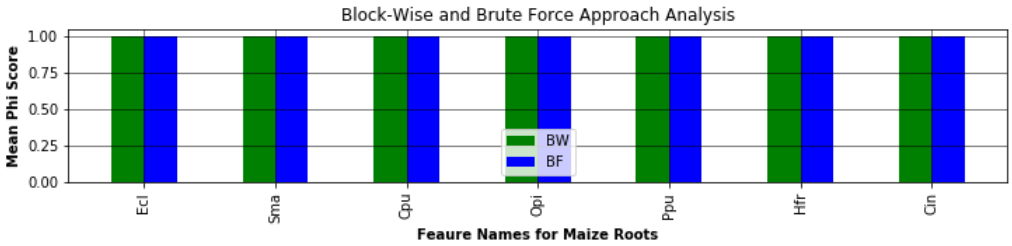
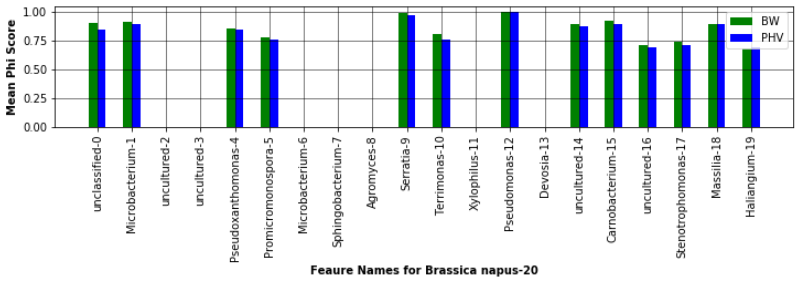


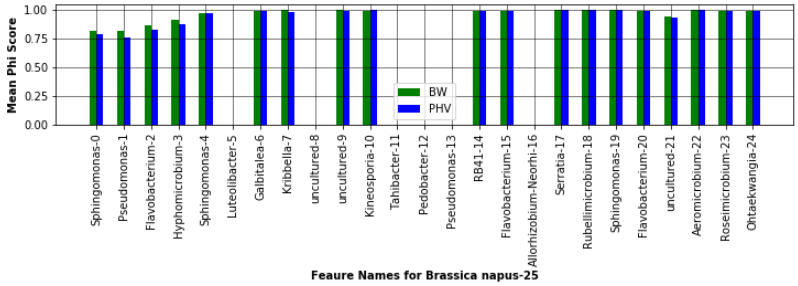
Figure 6.37: shows the results of block-wise implementation for Maize Root dataset

Fig. 6.38 shows the results of the block-wise implementation on *Brasicca napus* dataset. A uniform block size of 10 is used for each dataset while the last block was comprised of 5 and 6 taxa for *Brassica napus*-25 and *Brassica napus*-36 datasets respectively. We compared the quality score of sign patterns obtained from the block-wise approach with the quality score of sign patterns achieved by the perturbed-hybrid-variance controlled approach. Not only does the block-wise method outperform the mean ϕ score of the advanced heuristic method, it also inferred the sign patterns for all taxa except those for which the number of active abundance samples were insufficient. The features with insufficient data are represented by absented bar in

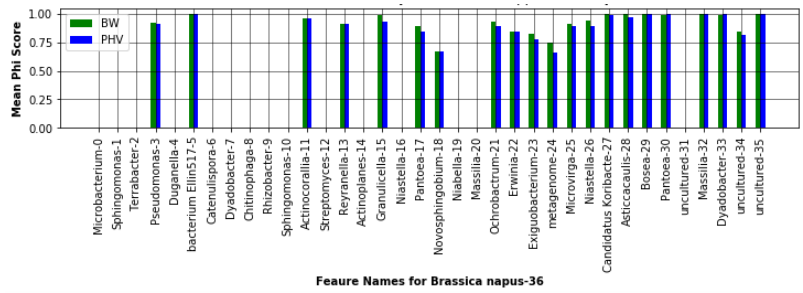
the chart. Due to the large size of *Brassica napus* dataset, we were unable to perform a brute-force analysis. The block-wise method was able to find a partial brute-force solution for all 3 *Brassica napus* datasets within a total time of 30 minutes per dataset, which is a significant speed-up compared to default approach. The execution time of the block-wise implementation remained lower than 15 minutes even for a community with as many as 36 taxa. The estimated accelerated brute-force time for 36 taxa is more than a 1000 hours, which shows that the block-wise approach is highly scalable for larger community sizes ($N > 10$).



(a) Results of block-wise implementation for *Brasicca napus*-20 dataset



(b) Results of block-wise implementation for *Brasicca napus*-25 dataset



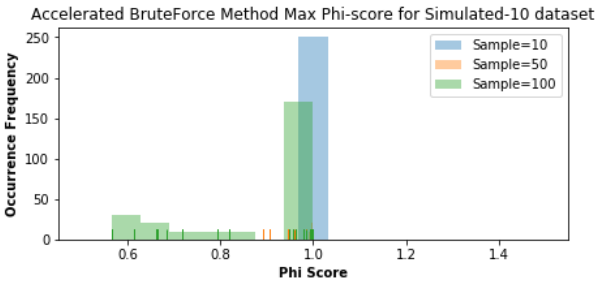
(c) Results of block-wise implementation for *Brasicca napus*-100 dataset

Figure 6.38: BlockWise Implementation result comparison against Perturbed-Hybrid-Variance Controlled Heuristic Method

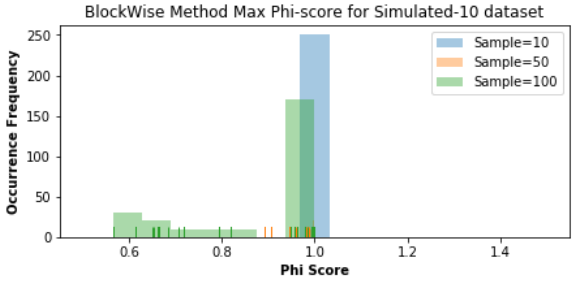
As shown in Fig. 6.38a, only 1 taxa achieved a ϕ score of 1.0 using the blockwise method. However, for *Brassica napus*-25 and 36 shown in 6.38b and c respectively, a total of 6 taxa achieved a $\phi = 1.0$. This shows that the block-wise method is also able to find a perfectly supported solution for larger microbial communities, for which finding a brute-force solution through the method proposed in the research [13] was not possible. However, since the *Brassica napus* dataset does not have a ground truth, we can not determine if these $\phi = 1.0$ sign patterns contain the true solution.

6.7.2 Results on Simulated Datasets

Fig. 6.39 shows the distribution of comparison of quality scores between the block-wise and the accelerated brute-force methods on different sample sizes of Simulated-10 dataset. This distributions shows the results of 25 simulated datasets in each sample size classification. A uniform block size of 5 is used for all execution of block-wise method on Simulated-10 dataset. It can be seen in Fig. 6.39 that the quality of sign patterns obtained through the block-wise remain almost the same for all three classification types with 10, 50 and 100 samples of Simulated-10 dataset. However, it is observed that even though both distributions look almost the same, the quality scores achieved through the accelerated brute-force method are slightly better than the block-wise method. The block-wise method took on an average of 20 seconds to complete the end to end execution of the datasets with 100 samples, which is almost the same as the time taken by the accelerated brute-force method.



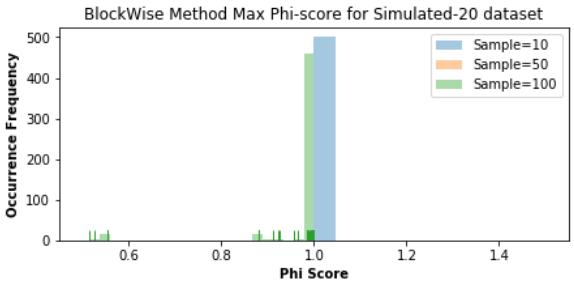
(a) ϕ Distribution of Accelerated Brute Force method for Simulated-10 dataset



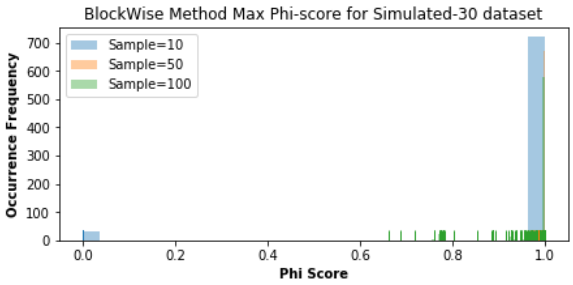
(b) ϕ Distribution of Block-wise Brute Force method for Simulated-10 dataset

Figure 6.39: Comparison of quality score distributions between Block-wise and Accelerated Brute Force Methods over 25 dataset of each classification type of Simulated-10 dataset

Fig. 6.40 shows the distribution of quality scores attained by the block-wise method for Simulated-20 and Simulated-30 datasets for sample sizes of 10, 50 and 100 with each 25 datasets for each sample classification. For both datasets, it can be seen that the $\phi = 1.0$ is the most frequent score, showing that the block-wise method has the ability find the maximum quality score sign pattern even for large communities. The 0.0 score in Fig. 6.40b, shows the failure of the block-wise method caused by an insufficient number of active abundance samples.



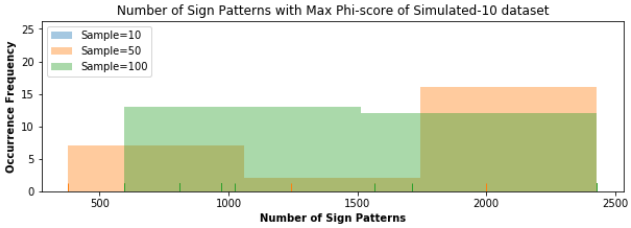
(a) Phi Score Distribution of BlockWise Brute Force method for Simulated-20 dataset



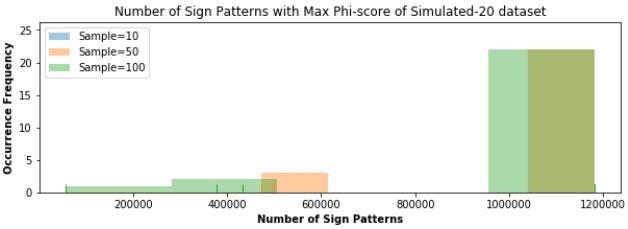
(b) Phi Score Distribution of BlockWise Brute Force method for Simulated-30 dataset

Figure 6.40: Distribution of quality scores of BlockWise brute-force method for Simulated-20 and 30 datasets

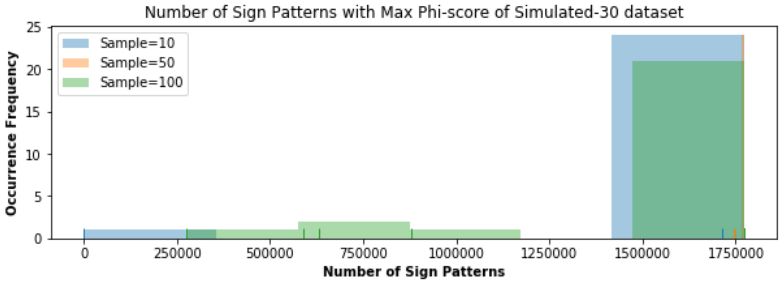
As shown in Fig. 6.41, the block-wise method is able to find a large number of good sign patterns. As for Simulated-10 dataset, a fixed block size of 10 was used, which provides us a total of $3^5 * 10 = 2430$ sign patterns are generated for final selection of selection sign pattern for each classification type of all 25 datasets in Simulated-10. This shows that with increasing number of taxa, the number of sign patterns generated with maximum ϕ score increases. For Simulated-20 and 30, the block size remained 10.



(a) Distribution of Max scored sign pattern for Simulated-10 dataset



(b) Distribution of Max scored sign pattern for Simulated-20 dataset



(c) Distribution of Max scored sign pattern for Simulated-30 dataset

Figure 6.41: Distribution of Max scored sign pattern on all Simulated datasets

6.8 Validation Analysis of Block-wise Brute Force Approach

We employed the same three methods of validation analysis that we used for the accelerated brute-force method and performed the validation analysis of block-wise method on both simulated and real datasets. For the block-wise method, we used the first pattern selection method for the selection of sign pattern with

maximum ϕ score for each taxa. For Simulated-20 and Simulated-30 datasets, we used a fixed block length of 10, while for Simulated-10, we used a block length of 5.

6.8.1 Maize Roots Dataset

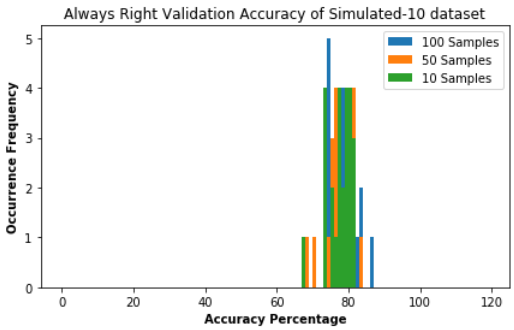
For the Maize Roots dataset, the accuracy achieved through the default method of accuracy evaluation is 63.26%. While with exact match method, we achieved a validation accuracy of 61.22% and similarly for method where zeros are not considered, we achieved a validation accuracy of 62.5%. In all cases, the block-wise method was able to outperform the accelerated brute-force method. It should be noted that the first selection criteria is used for the selection of sign patterns and it is quiet possible that validation accuracy may decrease depending upon the initial seed of sign pattern generation.

6.8.2 Simulated Data

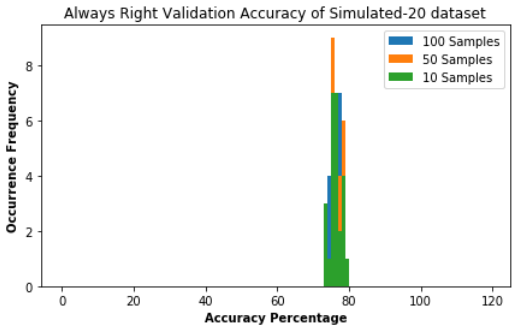
Fig. 6.42 shows the block-wise validation accuracy distribution on all three classifications of Simulated-10,20 and 30 dataset, each with 10,50 and 100 samples. The performance of block-wise method as shown in Fig. 6.42a is almost the same as the the accelerated brute-force method on Simulated-10 dataset, showing that the block-wise method is on par with the brute-force method in terms of accuracy. For Simulated-20 and Simulated-30 datasets, the performance of block-wise method remains consistently good as most of the inferences have a validation accuracy above 75% close to the accuracy considered sufficient in the default approach [13].

For the exact match criteria of accuracy computation, the performance of the block-wise method remains almost the same as the accelerated brute-force method as show in Fig. 6.43a. For Simulated-20 and Simulated-30 dataset as shown in Fig. 6.43b and c, the accuracy of inference decreased significantly as compared to the default accuracy method. Simulated-20 achieves a maximum accuracy of only 32% while Simulated-30 achieves a maximum accuracy of only 30%, without considering the outlier with all zero inferences, which shows that the accuracy of inference decreases with the increasing number of taxa.

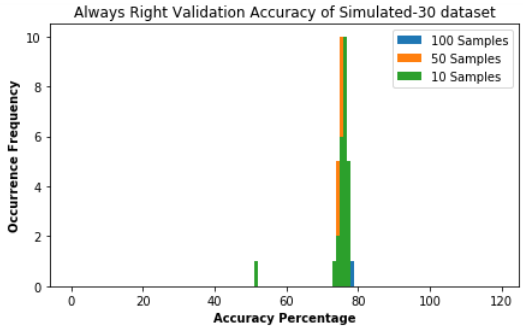
Overall, evaluating the inferences made by the block-wise method through different methods of accuracy computations shows that the block-wise method is highly scalable, and almost as accurate as the accelerated brute-force method. However, it should be noted that the achieved accuracy through the method is not as high as the quality (ϕ) of the sign pattern. This shows that the higher value of ϕ does not always reflect an increased accuracy.



(a) BlockWise Validation Accuracy on Simulated-10

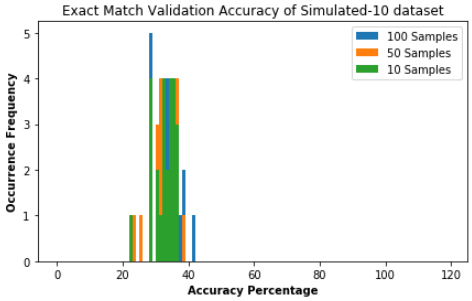


(b) BlockWise Validation Accuracy on Simulated-20

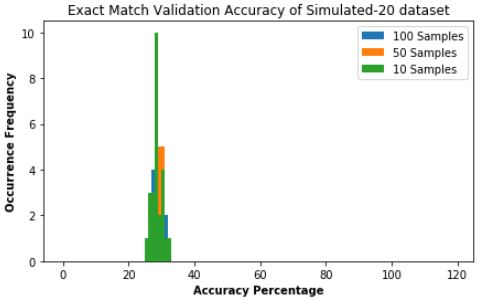


(c) BlockWise Validation Accuracy on Simulated-30

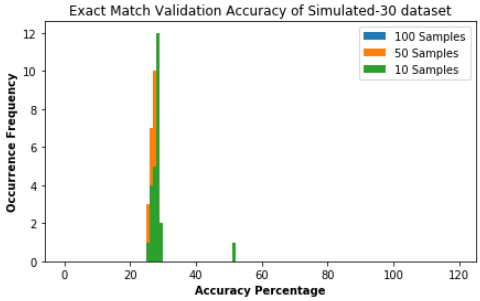
Figure 6.42: BlockWise Validation Accuracy using the default method



(a) BlockWise Validation Accuracy on Simulated-10

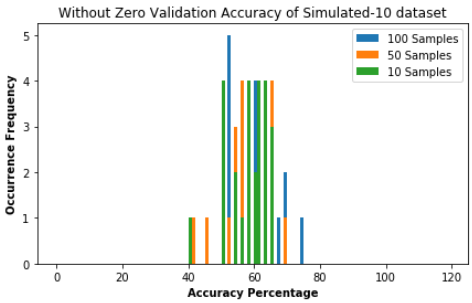


(b) BlockWise Validation Accuracy on Simulated-20

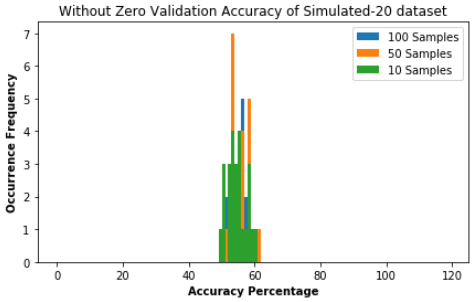


(c) BlockWise Validation Accuracy on Simulated-30

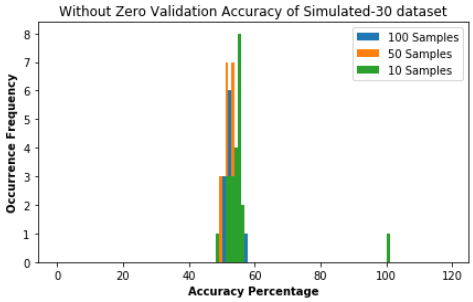
Figure 6.43: BlockWise Validation Accuracy Using exact match method



(a) BlockWise Validation Accuracy on Simulated-10



(b) BlockWise Validation Accuracy on Simulated-20



(c) BlockWise Validation Accuracy on Simulated-30

Figure 6.44: BlockWise Validation Accuracy Using exact match method

7 Discussion

7.1 Result Discussion

Inference of microbial community interaction types without a prior assumption of a population dynamics model is still an open research question. The improvements made on the previously proposed approach proposed by the authors of [13] were targeted towards the speed, failure rate, stability and inference quality. Through this research, we showed how improvements in a particular module of the algorithm such as perturbing or creating a hybrid heuristic can positively impact the quality of inference and reduce the failures caused by noise in the data. The accelerated and block-wise implementation approaches can be used to find complete or partial brute-force solutions of large microbial communities having more than 10 taxa, which was not previously achievable.

The performance of the algorithm is dependent on abundance data being used, where the process of data collection is time consuming and prone to error. Noise in the collected data has the tendency to distort the results, and it may also negatively impact the reproducibility of results even when all the user defined thresholds are kept the same. Moreover, the minimum number of samples required to infer an interaction type with strong statistical support is not always present in the data as most abundance values are zero. The actual cause of recording a zero as an abundance value may differ significantly depending upon the scenario; for example, it can be caused either by instrumental error, human error, or by the actual absence of a taxa in the collected sample. To overcome these issues, we developed a solution which infers consistent interaction types based on a hybrid approach, combining the brute-force and the heuristic approaches.

Each hyperplane in the pair difference matrix is calculated by taking a pair-wise difference between two rows of steady state sample abundance. As we compute the pair-wise difference (H), an entry in the hyperplane may be too small and can be considered as noise. Considering insignificant difference values in H may lead towards spurious inferences as the underlying observed differences may have resulted from a human/instrumental error. To discard all the insignificant differences, we introduced a user specified threshold Ω, as suggested in the research [13], which discards all the hyperplane entries lying within the range of $\pm\Omega$ by rounding them off to zero. The threshold of $\pm\Omega$ for this research is specified in section 5.3. Using Ω, we ensure the usage of only significant differences, resulting in statistically authentic inferences.

As mentioned in the section 2.7, the heuristic inference may fail to return a valid sign pattern. We introduced a zero-sign pattern inference which infers all zero interaction types for each taxon with respect to a focal taxon in case it fails to infer any solution for it. The zero sign pattern (all zero interaction type)

for any row of sign pattern represents either of these reasons of failure: lack of sufficient data for inference, failure in inference of heuristic, or zero differential hyperplane intersections. Usage of a zero sign pattern allows continuous execution of multiple end-to-end cycles of sign pattern inference.

Our research also addressed the issue of inference instability. As shown in section 6.3, the inferences made through the variance control method are more consistent than the default method [13], while it also achieves similar level of quality (ϕ) for sign patterns. However, it can be observed in results of *Brassica napus* dataset shown as in Fig. 6.15 that the variance control does not always work, and in some cases it might result in higher variance as compared to the variance produced by the default approach [13]. This shows that even though the overall consistency of inferences increased through the variance control method, the variance of each individual feature might not always be lower.

To improve the quality of a sign pattern, we introduced a perturbation pipeline that takes a valid sign pattern and finds a sign pattern similar to it but with improved ϕ. The method employed to find an improved quality sign pattern is based on a greedy approach where a sign pattern is randomly perturbed to find a better quality sign pattern. As the method being used is greedy and random, it might not always be able to find a sign pattern in the local search space (a few perturbations away) even if there is one. The degree of ϕ improvement may reduce as ϕ reaches 1.0.

Performance improvement is not the only aspect of algorithmic improvements. To improve the throughput of an algorithm, steps causing frequent failure can be analyzed, and minimizing the failure rate results in a higher throughput. A variety of approaches can be used to reduce the failure rate, depending upon the type of failure. Algorithmic failure analysis involves pin-pointing a range of boundary values and then applying the necessary conditions or alteration to program flow at the onset of an expected failure state. As the proposed algorithm in [13] uses the abundance data for the inference of a heuristic, the presence of noise in the data caused by instrument or human error may hinder the inference of algorithm, resulting in no or partial inferences of sign. Such a failure can be addressed either by reducing noise, or by using a secondary engine for sign inference, which is independent of data.

The block-wise sign pattern finds a heuristic brute-force solution for a block of taxa within the microbial community. As the size of microbial community increases, it becomes harder to find a block-wise sign pattern for a taxon with $\phi = 1.0$. As the block-wise approach is greedy, it can lead to a local minima. The block-wise method represents a compromise between a computationally intensive full solution and the stochastic heuristic.

7.2 Future Work

In this research, we proposed an improved heuristic approach, which is faster and has the ability to infer solutions with more robustness and stability. Our block-wise method is able to perform fast partial brute-force analysis of large microbial communities and is able to find accurate high quality solutions. These

improvements allow us to process large microbial communities in short time duration with increased stability and quality, which was not possible before.

As our research addresses the shortcomings highlighted in section 2.7, the algorithm proposed in the research [13] can be further improved by finding different method to assess the quality of a proposed heuristic. As we discussed earlier, the ϕ measure only checks the ratio of total versus intersected hyperplanes. A possible heuristic could be made by finding a group of brute-force heuristics inferring similar interaction types and having sign patterns with highest ϕ, which can be used as a possible sign pattern. A more holistic approach where a sign pattern is assessed based on its similar predecessors may also reduce the computational time for heuristic assessment. However, in the default approach proposed in the research [13], it was assumed that each sign pattern will have a unique ϕ score. Due to this assumption, no mechanism was introduced for tie breaking, and to address this we used two pattern selection techniques, neither of which performed sufficiently reliable to be a general solution, opening the door to future research.

We also observed that both accelerated and block-wise brute-force methods are able to find a number of sign patterns with maximum achievable ϕ. However, sign patterns with same ϕ score will not have the same accuracy with respect to the ground truth, resulting in erroneous inferences. This raises the need for a method to settle ties between sign patterns having the same ϕ.

We employed a method to discard the exact same hyperplanes, which results in increased speed of ϕ calculation, as it involves checking the intersection of each differential hyperplane with the inferred heuristic sign pattern. Reducing the size of differential hyperplane matrix by discarding the repeated hyperplanes has its associated cost in terms of computational complexity. The method that reduces the differential hyperplane matrix size has a time complexity of $O(M^2)$, where M represents number of hyperplanes, and element-wise comparison of each pair of hyperplanes is considered to take a constant time; as all hyperplane in a pair difference matrix of a particular taxon have the same dimension. Having a quadratic computational time complexity makes the reduction of hyperplanes a time consuming activity. The effect of hyperplane pre-computation is more visible for taxa with more than a 100 active abundance samples. An efficient algorithm to remove the duplicate hyperplanes can significantly improve the performance of the accelerated brute-force method, which can be the focus of future research.

Another potential avenue towards finding the microbial interaction types is exploring semi-supervised learning techniques where the interaction types of a small group of taxa is inferred first through brute-force method and then those interactions are used as labelled data. This labelled data can further be used infer the incremental interaction types of similar taxa which can significantly reduce the inference time. The labelled data can be used as an input to link prediction classifiers which can then be used to infer the causal interaction types and can be validated through the labelled data.

7.3 Summary

Our enhanced approach provides an efficient way to infer microbial interaction types with significantly reduced failure rate and considerably higher inference quality. Our method builds up on the new paradigm introduced by the authors of [13] and it addresses the overall efficiency and effectiveness of microbial interaction inference. This method can be further improved by addressing the remaining shortcomings or by formulating a new approach which uses the essentials of population dynamics model to train and test a semi-supervised link prediction classifier. We proposed a faster, scalable, more stable and fault tolerant method of inference of microbial interaction types. Based on our analysis, we propose that the Accelerated Brute Force method should be used for smaller community sizes ($N \leq 18$). While any combination of the three advanced heuristic methods (Hybrid, Perturbed and Variance-Controlled) can be used for very large communities ($N \geq 50$). For moderately sized communities ($19 \leq N \leq 49$), the block-wise method should be used as it provides a partial brute-force solution which results in better accuracy than the advanced heuristic methods.

References

[1] Microbial communities. `https://www.nature.com/subjects/communities`. Accessed: 2020-03-15.

[2] Janine Bartelt-Ryser, Jasmin Joshi, Bernhard Schmid, Helmut Brandl, and Teri Balser. Soil feedbacks of plant diversity on soil microbial communities and subsequent plant growth. *Perspectives in Plant Ecology, Evolution and Systematics*, 7(1):27–49, 2005.

[3] Stéphane Pesce, Isabelle Batisson, Corinne Bardot, Céline Fajon, Christophe Portelli, Bernard Montuelle, and Jacques Bohatier. Response of spring and summer riverine microbial communities following glyphosate exposure. *Ecotoxicology and environmental safety*, 72(7):1905–1912, 2009.

[4] C. Gaylarde, M. Ribas Silva, and T. Warscheid. Microbial impact on building materials: an overview. *Materials and Structures*, 36(5):342–352, 2003.

[5] Afrah Shafquat, Regina Joice, Sheri L Simmons, and Curtis Huttenhower. Functional and phylogenetic assembly of microbial communities in the human microbiome. *Trends in microbiology*, 22(5):261–266, 2014.

[6] Stefanie Widder, Rosalind J Allen, Thomas Pfeiffer, Thomas P Curtis, Carsten Wiuf, William T Sloan, Otto X Cordero, Sam P Brown, Babak Momeni, Wenying Shou, et al. Challenges in microbial ecology: building predictive understanding of community function and dynamics. *The ISME Journal*, 10(11):2557–2568, 2016.

[7] Jonathan Friedman and Eric J. Alm. Inferring correlation networks from genomic survey data. *PLOS Computational Biology*, 8(9):1–11, 09 2012.

[8] Herbert A Simon. Spurious correlation: A causal interpretation. *Journal of the American statistical Association*, 49(267):467–479, 1954.

[9] Vanni Bucci, Belinda Tzen, Ning Li, Matt Simmons, Takeshi Tanoue, Elijah Bogart, Luxue Deng, Vladimir Yeliseyev, Mary L Delaney, Qing Liu, et al. MDSINE: Microbial dynamical systems inference engine for microbiome time-series analyses. *Genome Biology*, 17(1):121, 2016.

[10] Richard R Stein, Vanni Bucci, Nora C Toussaint, Charlie G Buffie, Gunnar Rätsch, Eric G Pamer, Chris Sander, and Joao B Xavier. Ecological modeling from time-series inference: insight into dynamics and stability of intestinal microbiota. *PLoS computational biology*, 9(12), 2013.

[11] Travis E. Gibson, Amir Bashan, Hong-Tai Cao, Scott T. Weiss, and Yang-Yu Liu. On the origins and control of community types in the human microbiome. *PLOS Computational Biology*, 12(2):1–21, 02 2016.

[12] Steven N Steinway, Matthew B Biggs, Thomas P Loughran Jr, Jason A Papin, and Reka Albert. Inference of network dynamics and metabolic interactions in the gut microbiome. *PLoS computational biology*, 11(6), 2015.

[13] Yandong Xiao, Marco Tulio Angulo, Jonathan Friedman, Matthew K Waldor, Scott T Weiss, and Yang-Yu Liu. Mapping the ecological networks of microbial communities. *Nature Communications*, 8(1):1–12, 2017.

[14] Jiliang Tang, Yi Chang, Charu Aggarwal, and Huan Liu. A survey of signed network mining in social media. *ACM Computing Surveys (CSUR)*, 49(3):1–37, 2016.

[15] Wikipedia labelled dataset. `https://en.wikipedia.org/wiki/Labeled_data`. Accessed: 2020-05-22.

[16] Fritz Heider. Attitudes and cognitive organization. *The Journal of Psychology*, 21(1):107–112, 1946.

[17] Dorwin Cartwright and Frank Harary. Structural balance: a generalization of heider's theory. *Psychological Review*, 63(5):277, 1956.

[18] Michael Moore. An international application of heider's balance theory. *European Journal of Social Psychology*, 8(3):401–405, 1978.

[19] Paolo Massa and Paolo Avesani. Controversial users demand local trust metrics: An experimental study on epinions. com community. In *AAAI*, pages 121–126, 2005.

[20] Giacomo Bachi, Michele Coscia, Anna Monreale, and Fosca Giannotti. Classifying trust/distrust relationships in online social networks. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pages 552–557. IEEE, 2012.

[21] Jian Wu, Francisco Chiclana, and Enrique Herrera-Viedma. Trust based consensus model for social network in an incomplete linguistic information context. *Applied Soft Computing*, 35:827–839, 2015.

[22] Young Ae Kim and Muhammad A Ahmad. Trust, distrust and lack of confidence of users in online social media-sharing communities. *Knowledge-Based Systems*, 37:438–450, 2013.

[23] David Knoke and Song Yang. *Social Network Analysis*, volume 154. SAGE Publications, Incorporated, 2019.

[24] Charu C Aggarwal. An introduction to social network data analytics. In *Social network data analytics*, pages 1–15. Springer, 2011.

[25] Xiaoming Li, Hui Fang, and Jie Zhang. Rethinking the link prediction problem in signed social networks. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[26] Kai-Yang Chiang, Cho-Jui Hsieh, Nagarajan Natarajan, Inderjit S Dhillon, and Ambuj Tewari. Prediction and clustering in signed networks: a local to global perspective. *The Journal of Machine Learning Research*, 15(1):1177–1213, 2014.

[27] Ghazaleh Beigi, Jiliang Tang, and Huan Liu. Signed link analysis in social media networks. In *Proceedings of the 10th International Conference on Web and Social Media, ICWSM 2016*, pages 539–542. AAAI Press, 2016. 10th International Conference on Web and Social Media, ICWSM 2016 ; Conference date: 17-05-2016 Through 20-05-2016.

[28] Jiliang Tang, Shiyu Chang, Charu Aggarwal, and Huan Liu. Negative link prediction in social media. In *Proceedings of the eighth ACM international conference on web search and data mining*, pages 87–96, 2015.

[29] Suhang Wang, Jiliang Tang, Charu Aggarwal, Yi Chang, and Huan Liu. Signed network embedding in social media. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 327–335. SIAM, 2017.

[30] Jérôme Kunegis, Andreas Lommatzsch, and Christian Bauckhage. The slashdot zoo: Mining a social network with negative edges. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, page 741–750, New York, NY, USA, 2009. Association for Computing Machinery.

[31] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 641–650, New York, NY, USA, 2010. Association for Computing Machinery.

[32] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1361–1370, 2010.

[33] Chenhao Tan, Jie Tang, Jimeng Sun, Quan Lin, and Fengjiao Wang. Social action tracking via noise tolerant time-varying factor graphs. In *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1049–1058, 2010.

[34] Braxton E Thomason, Thayne R Coffman, and Sherry E Marcus. Sensitivity of social network analysis metrics to observation noise. In *2004 ieee aerospace conference proceedings (ieee cat. no. 04th8720)*, volume 5, pages 3206–3216. IEEE, 2004.

[35] Roberto CSNP Souza, Denise EF de Brito, Rodrigo L Cardoso, Derick M de Oliveira, Wagner Meira, and Gisele L Pappa. An evolutionary methodology for handling data scarcity and noise in monitoring real events from social media data. In *Ibero-American Conference on Artificial Intelligence*, pages 295–306. Springer, 2014.

[36] R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, page 403–412, New York, NY, USA, 2004. Association for Computing Machinery.

[37] Kai-Yang Chiang, Nagarajan Natarajan, Ambuj Tewari, and Inderjit S. Dhillon. Exploiting longer cycles for link prediction in signed networks. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, page 1157–1162, New York, NY, USA, 2011. Association for Computing Machinery.

[38] Tongda Zhang, Haomiao Jiang, Zhouxiao Bao, and Yingfeng Zhang. Characterization and edge sign prediction in signed networks. *Journal of Industrial and Intelligent Information Vol*, 1(1):19–24, 2013.

[39] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[40] Arti Patidar, Vinti Agarwal, and KK Bharadwaj. Predicting friends and foes in signed networks using inductive inference and social balance theory. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 384–388. IEEE, 2012.

[41] T. DuBois, J. Golbeck, and A. Srinivasan. Predicting trust and distrust in social networks. In *2011 IEEE Third Int'l Conference on Privacy, Security, Risk and Trust (PASSAT) / 2011 IEEE Third Int'l Conference on Social Computing (SocialCom)*, pages 418–424, Los Alamitos, CA, USA, oct 2011. IEEE Computer Society.

[42] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.

[43] Sid Redner. Teasing out the missing links. *Nature*, 453(7191):47–48, 2008.

[44] Aaron Clauset, Cristopher Moore, and Mark EJ Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.

[45] Ji-chao Li, Dan-ling Zhao, Bing-Feng Ge, Ke-Wei Yang, and Ying-Wu Chen. A link prediction method for heterogeneous networks based on bp neural network. *Physica A: Statistical Mechanics and its Applications*, 495:1–17, 2018.

[46] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. International Conference on Machine Learning (ICML), 2016.

[47] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems*, pages 5165–5175, 2018.

[48] Dekang Lin et al. An information-theoretic definition of similarity. In *Icml*, volume 98, pages 296–304, 1998.

[49] Patrick Doreian and Andrej Mrvar. A partitioning approach to structural balance. *Social Networks*, 18(2):149–168, 1996.

[50] Amin Javari and Mahdi Jalili. Cluster-based collaborative filtering for sign prediction in social networks with positive and negative links. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(2):1–19, 2014.

[51] Tibor Antal, Paul L Krapivsky, and Sidney Redner. Social balance on networks: The dynamics of friendship and enmity. *Physica D: Nonlinear Phenomena*, 224(1-2):130–136, 2006.

[52] Deni Khanafiah and Hokky Situngkir. Social balance theory. *arXiv preprint nlin/0405041*, 2004.

[53] Hsinchun Chen, Xin Li, and Zan Huang. Link prediction approach to collaborative filtering. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'05)*, pages 141–142. IEEE, 2005.

[54] Cai-Nicolas Ziegler and Georg Lausen. Propagation models for trust and distrust in social networks. *Information Systems Frontiers*, 7(4-5):337–358, 2005.

[55] Martine De Cock and Paulo Pinheiro Da Silva. A many valued representation and propagation of trust and distrust. In *International Workshop on Fuzzy Logic and Applications*, pages 114–120. Springer, 2005.

[56] Patricia Victor, Chris Cornelis, Martine De Cock, and P Pinheiro da Silva. Towards a provenance-preserving trust model in agent networks. In *WWW2006 Conference Proceedings, Special Interest Tracks, Posters and Workshops*, 2006.

[57] Cho-Jui Hsieh, Kai-Yang Chiang, and Inderjit S. Dhillon. Low rank modeling of signed networks. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, page 507–515, New York, NY, USA, 2012. Association for Computing Machinery.

[58] James A Davis. Clustering and structural balance in graphs. *Human Relations*, 20(2):181–187, 1967.

[59] Priyanka Agrawal, Vikas K. Garg, and Ramasuri Narayanam. Link label prediction in signed social networks. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, page 2591–2597, Beijing, China, 2013. AAAI Press.

[60] Yi Cen, Rentao Gu, and Yuefeng Ji. Sign inference for dynamic signed networks via dictionary learning. *J. Appl. Math.*, 2013, Special Issue:10 pages, 2013.

[61] Wikipedia correlation and dependence. `https://en.wikipedia.org/w/index.php?title=Correlation_and_dependence&oldid=935560828`. Accessed: 2020-02-12.

[62] Baruch Barzel and Albert-László Barabási. Network link prediction by global silencing of indirect correlations. *Nature Biotechnology*, 31(8):720–725, 2013.

[63] Dashun Wang, Dino Pedreschi, Chaoming Song, Fosca Giannotti, and Albert-Laszlo Barabasi. Human mobility, social ties, and link prediction. KDD '11, page 1100–1108, New York, NY, USA, 2011. Association for Computing Machinery.

[64] Mehdi Layeghifard, David M Hwang, and David S Guttman. Disentangling interactions in the microbiome: a network perspective. *Trends in Microbiology*, 25(3):217–228, 2017.

[65] Karoline Faust and Jeroen Raes. Microbial interactions: from networks to models. *Nature Reviews Microbiology*, 10(8):538–550, 2012.

[66] Karoline Faust, J. Fah Sathirapongsasuti, Jacques Izard, Nicola Segata, Dirk Gevers, Jeroen Raes, and Curtis Huttenhower. Microbial co-occurrence relationships in the human microbiome. *PLOS Computational Biology*, 8(7):1–17, 07 2012.

[67] Manimozhiyan Arumugam, Jeroen Raes, Eric Pelletier, Denis Le Paslier, Takuji Yamada, Daniel R Mende, Gabriel R Fernandes, Julien Tap, Thomas Bruls, Jean-Michel Batto, et al. Enterotypes of the human gut microbiome. *Nature*, 473(7346):174–180, 2011.

[68] Hongwu Yang, Juan Li, Yunhua Xiao, Yabing Gu, Hongwei Liu, Yili Liang, Xueduan Liu, Jin Hu, Delong Meng, and Huaqun Yin. An integrated insight into the relationship between soil microbial community and tobacco bacterial wilt disease. *Frontiers in Microbiology*, 8:2179, 2017.

[69] Ye Deng, Yi-Huei Jiang, Yunfeng Yang, Zhili He, Feng Luo, and Jizhong Zhou. Molecular ecological network analyses. *BMC Bioinformatics*, 13(1):113, 2012.

[70] Sophie Weiss, Will Van Treuren, Catherine Lozupone, Karoline Faust, Jonathan Friedman, Ye Deng, Li Charlie Xia, Zhenjiang Zech Xu, Luke Ursell, Eric J Alm, et al. Correlation detection strategies in microbial data sets vary widely in sensitivity and precision. *The ISME Journal*, 10(7):1669–1681, 2016.

[71] WenJun Zhang. Constructing ecological interaction networks by correlation analysis: hints from community sampling. *Network Biology*, 1(2):81, 2011.

[72] Timothy G Dinan and John F Cryan. The impact of gut microbiota on brain and behaviour: implications for psychiatry. *Current Opinion in Clinical Nutrition & Metabolic Care*, 18(6):552–558, 2015.

[73] Siobhan F Clarke, Eileen F Murphy, Orla O'Sullivan, Alice J Lucey, Margaret Humphreys, Aileen Hogan, Paula Hayes, Maeve O'Reilly, Ian B Jeffery, Ruth Wood-Martin, et al. Exercise and associated dietary extremes impact on gut microbial diversity. *Gut*, 63(12):1913–1920, 2014.

[74] D Compare, P Coccoli, A Rocco, OM Nardone, S De Maria, M Cartenì, and G Nardone. Gut–liver axis: the impact of gut microbiota on non alcoholic fatty liver disease. *Nutrition, Metabolism and Cardiovascular Diseases*, 22(6):471–476, 2012.

[75] Davide Bulgarelli, Klaus Schlaeppi, Stijn Spaepen, Emiel Ver Loren Van Themaat, and Paul Schulze-Lefert. Structure and functions of the bacterial microbiota of plants. *Annual Review of Plant Biology*, 64:807–838, 2013.

[76] Mathias Hänsch and Christoph Emmerling. Effects of silver nanoparticles on the microbiota and enzyme activity in soil. *Journal of Plant Nutrition and Soil Science*, 173(4):554–558, 2010.

[77] Wikipedia edit distance. `https://en.wikipedia.org/wiki/Edit_distance`. Accessed: 2020-03-23.

[78] Eric G. Lamb Isobel Parkin Annaliza McGillivray Steven D. Mamet, Bobbi Helgason. Root and exudate selection of yield-beneficial bacteria in brassica napus. *In Review*, In Review(In Review):In Review, In Review.

[79] S. D. Mamet Z. M. Morales S. Williams T. Dowhy Z. Taye E. Lamb M. Links C. Norris S. Shirtliffe M. Arcand S. Vail Helgason Bell, J. K. and S. D. Siciliano. A temporally intensive survey of bacterial communities of brassica napus genotypes grown in three environments. global ecology and biogeography. *Global Ecology and Biogeography*, (In Review), 2019.

[80] Ben Niu, Joseph Nathaniel Paulson, Xiaoqi Zheng, and Roberto Kolter. Simplified and representative bacterial community of maize roots. *Proceedings of the National Academy of Sciences*, 114(12):E2450–E2459, 2017.

[81] Finding highly - associated balances with the response variable. `https://rdrr.io/github/UVic-omics/selbal/`. Accessed: 2020-04-14.

[82] Github hallucigenia-sparsa/seqtime. `https://github.com/hallucigenia-sparsa/seqtime`. Accessed: 2020-04-26.

[83] Github r-lib/devtools. `https://github.com/r-lib/devtools`. Accessed: 2020-04-26.