Towards a Surrogate-assisted PALLAS algorithm for Gene Regulatory Network Inference

Hugo de A. Amorim Neto[1], **Luis Loo**[2], **Marcelo G. P. de Lacerda**[1], **Ulisses Braga Neto**[2], **Fernando Buarque de L. Neto**[1]

[1]Computer Engineering (EComp) – University of Pernambuco (UPE)
50720-001 – Recife – PE – Brazil

[2]Electrical and Computer Engineering – Texas A&M University
77801 – College Station – TX – United States

{haan2, mgpl, fbln}@ecomp.poli.br, {loo, ulisses}@tamu.edu

Abstract. *This paper analyzes the application of surrogate models to improve the efficiency of Gene Regulatory Network (GRN) inference from time-series data. A Radial Basis Function (RBF) surrogate model was integrated with the Penalized mAximum LikeLihood and pArticle Swarms (PALLAS) using a Mixed Fish School Search (MFSS) algorithm to reduce the computational cost associated with evaluating the penalized log-likelihood (PLL) fitness function. Experimental results on the p53-MDM2 negative-feedback loop GRN dataset demonstrate that the surrogate-assisted approach significantly reduced fitness function calls by 50% and 89% while maintaining the quality of the PLL metric, with this showing the potential of surrogate models to accelerate GRN inference.*

1. Introduction

Gene Regulatory Networks (GRN) are essential components of cellular biology, governing the regulation of gene expression and controlling various biological processes. These networks consist of genes, their products (RNA and proteins), and the regulatory relationships among them. Understanding the structure and dynamics of GRNs is vital for deciphering complex biological systems, such as cellular responses to environmental stimuli, developmental processes, and disease mechanisms. Consequently, accurate GRN inference has become a key objective in systems biology and bioinformatics, aiming to reconstruct these networks from high-throughput experimental data, particularly time-series gene expression data. However, the inference of GRNs from such data presents significant challenges due to the high dimensionality, noise, and inherent complexity of the biological systems involved [Marku and Pancaldi 2023].

The Penalized mAximum LikeLihood and pArticle Swarms (PALLAS) algorithm [Tan et al. 2020] is designed for inferring gene regulatory networks from noisy time-series data. It integrates a Penalized Log-Likelihood (PLL) fitness function with the Mixed Fish School Search (MFSS) algorithm, a optimization technique inspired by fish schooling behavior. This approach allows PALLAS to handle both continuous and discrete search spaces, making it well-suited for complex biological systems. The adaptability of PALLAS to scenarios with or without prior knowledge of model parameters further enhances its applicability in diverse biological contexts.

Despite the effectiveness of the PALLAS algorithm in GRN inference, the computational cost associated with repeatedly evaluating the PLL function can be prohibitive, especially when dealing with large datasets or complex network structures. This computational burden arises from the necessity of performing numerous evaluations of the fitness function during the optimization process, each of which is computationally expensive. To address this challenge, researchers have explored the use of surrogate models as a means to reduce the computational costs involved. Surrogate models are simplified representations of complex systems, constructed to approximate the behavior of expensive functions while significantly lowering the computational burden [Ferreira et al. 2019]. These models are particularly useful in scenarios where the original model is too costly to evaluate directly, as they can provide sufficiently accurate predictions at a fraction of the computational cost.

The use of surrogate models in meta-heuristics has gained considerable attention due to their ability to accelerate optimization processes without compromising accuracy. Meta-heuristics, such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Fish School Search (FSS), are optimization techniques that guide the search for optimal solutions in complex spaces by combining exploration and exploitation strategies [Sastry et al. 2005] [Kennedy and Eberhart 1995] [Bastos Filho et al. 2009]. Surrogate models enhance these techniques by replacing costly function evaluations with cheaper approximations, thereby reducing the overall computational time. This approach has been successfully applied across various domains, including the optimization of motor designs [Ibrahim et al. 2022], simulations of human movement [Eskinazi and Fregly 2016], and power integrity analysis in electronic systems [Leal-Romo et al. 2018]. By integrating surrogate models into these optimization frameworks, significant reductions in computational costs have been achieved, while maintaining or even improving the quality of the solutions obtained.

Building on the advantages of surrogate models in various optimization scenarios, their application in GRN inference provides a practical approach for reducing the computational burden associated with optimization-based methods. By integrating surrogate models within algorithms like PALLAS, the frequency of costly PLL function evaluations can be decreased, which in turn allows the algorithm to allocate more resources to exploring promising regions of the search space. This approach is particularly valuable when dealing with large-scale time-series data, where the computational demands are often substantial. Surrogate-assisted optimization not only expedites the inference process but also facilitates the exploration of more complex models that might otherwise be computationally prohibitive.

This paper explores the integration of a Radial Basis Function (RBF) surrogate model within the MFSS algorithm as part of the PALLAS framework for GRN inference. The RBF model is employed to approximate the PLL function, which is a key component of the PALLAS algorithm. By replacing the original PLL function by its RBF approximation during certain iterations of the optimization process, the number of PLL evaluations required is significantly reduced. This surrogate-assisted approach is designed to maintain the accuracy of the inferred networks while achieving substantial computational savings. The effectiveness of this approach is demonstrated through a series of experiments on the p53-MDM2 negative-feedback loop GRN dataset. This dataset, which involves the well-studied interaction between the p53 tumor suppressor protein and its regulator MDM2, serves as a benchmark for evaluating the performance of GRN inference algorithms. The results highlight the potential of surrogate-assisted methods in enhancing the efficiency and scalability of GRN inference, paving the way for more widespread application in systems biology.

The structure of this paper is as follows: Section II presents a brief overview of the employed algorithms, including PALLAS, the PLL function, FSS, MFSS, the surrogate model, and the integration of MFSS with the surrogate model. Section III presents the dataset used, the experimental setup, and the results obtained. Lastly, Section IV presents the conclusion and identifies areas for future work.

2. Methods

The approach proposed in this work combines the PALLAS algorithm with the predictions of surrogate models to reduce the number of calls to the fitness function and with this reduce the overall cost of the process. This sections briefly describes PALLAS, its components (PLL and MFSS), the radial Basis Function interpolation, which is the method used for the surrogate model and the combining strategy used. For full details on the algorithms and equations, please refer to the respective sources cited.

2.1. PALLAS

The PALLAS algorithm [Tan et al. 2020] leverages two key components: a PLL fitness function and the MFSS algorithm. The PLL function is optimized using MFSS, an advanced form of the FSS algorithm [Bastos Filho et al. 2009], which is capable of operating in both continuous and discrete search spaces. PALLAS models regulatory interactions with discrete dimensions and observational noise with continuous ones. It iteratively updates these to maximize the inferred network's likelihood, ensuring robustness to noise in GRN inference. Designed for use with or without prior knowledge of model parameters, PALLAS supports RNA-seq and microarray time-series data, offering flexibility across biological contexts.

2.1.1. Penalized Log-Likelihood

A PLL based on the implementation by [Tan et al. 2020] was used to assess the quality of each individual in the algorithm and to train the surrogate model. Their PLL implementation focus on having the best fit to the data while having a sparse network structure with a small number of edges between genes. The likelihood calculation utilizes a Boolean Kalman Filter [Braga-Neto 2011], implemented through auxiliary particle filtering [Imani and Braga-Neto 2018]. Considering a sample data that consists of n independent time series $Y^j_{1:k} = \{Y^j_1, \ldots, Y^j_k\}$ up to time k, for $j = 1, \ldots, n$. The PLL at time k is defined as

$$\begin{aligned} PLL_k &= \frac{1}{kn} \log p(Y^{(1)}_{1:k}, \ldots, Y^{(n)}_{1:k}) - \eta \sum_{i,j=1}^{2^d} |a_{ij}| \\ &= \frac{1}{kn} \sum_{j=1}^{n} \log p(Y^j_{1:k}) - \eta \sum_{i,j=1}^{2^d} |a_{ij}|, \end{aligned} \tag{1}$$

where $\eta > 0$ is a regularization parameter, which has a default value of $\eta = 0.01$ based on the original implementation. Therefore, the PLL in Eq. 1 is the sum of the average log-likelihood per time series and a negative value times the number of edges in the model. Maximization of Eq. 1 thus encourages the model to both fit the data and be sparse, i.e., contain a small number of edges between genes, consistent with biological understanding.

2.1.2. MFSS

The approach used in this study was adapted from PALLAS [Tan et al. 2020], where the authors introduce the MFSS algorithm, an extension of the FSS. The FSS algorithm was created inspired on the collective behavior of fish schools in the search for food, where each ”fish” gains ”weight” based on its performance, with heavier fish exerting greater influence on the school's movements. The algorithm is structured around four operators, each with different purposes, namely, the individual movement operator performs a greedy local search, the feeding operator adjusts individual weights based on improvements during the individual movement, the collective instinctive movement pushes the school towards promising areas and lastly the collective volitive movement dynamically balances exploration and exploitation based on the algorithm's needs through expansion and contraction of the school radius. The algorithm also reduces the step size at each iteration [Bastos Filho et al. 2009]. The MFSS adaptation gives the algorithm the ability to handle optimization problems involving both large continuous and discrete parameter spaces simultaneously, as it's necessary for the PALLAS algorithm. Algorithm 1 shows the pseudo-code that suits both the FSS and MFSS, It's important to note that all the actions are done for all fish.

Algorithm 1 Pseudo-code of the FSS and MFSS algorithm.

```
initialize randomly all fish
for $iteration = 1, 2, \ldots, N$ do
    Evaluate fitness function
    Execute individual movement
    Evaluate fitness function
    Execute feeding operator
    Execute collective instinctive movement
    Execute collective volitive movement
    Update step size
end for
```

2.2. Surrogate Model

In this study, we apply a surrogate model using Radial Basis Function interpolation to approximate the PLL function from a limited set of data points. RBF is a method for estimating values of unknown locations using a weighted sum of radial basis functions centered at known data points. The weights are determined by solving a system of linear equations, ensuring the interpolant passes through the given data points [Buhmann 2000].

The Python library PySOT [Eriksson et al. 2019] was utilized to implement the surrogate model. After conducting a parametric analysis with Optuna [Akiba et al. 2019], another Python library, RBF interpolation was chosen as the most effective strategy. This analysis involved comparing RBF interpolation against other techniques, including Gaussian Process Regression and Multivariate Polynomial Regression, to determine the optimal approach the problem, and for this specific problem the RBF showed the best results.

2.3. Surrogate-assisted MFSS

The Surrogate-assisted MFSS (SMFSS) is the integration of the surrogate model into the MFSS algorithm. The overall structure of the SMFSS remained the same as the MFSS, however, the main difference was the use of the surrogate model prediction in the place of the fitness function, which is computationally expensive, for a number of iterations. A decision mechanism is used at each iteration to determine whether to use the original PLL fitness function or the prediction from the surrogate model. This decision was made using a parameter $fitnessCheck$, the PLL is computed every $fitnessCheck$ iterations and new points are added to the surrogate model, otherwise the surrogate predicts the fitness value.

The number of points added to the surrogate and their distribution significantly impacted the algorithm's performance. When all points were added during each fitness check, the surrogate prediction eventually became more time-consuming than running the original fitness function, especially towards the end of the algorithm's execution. To address this, we experimented with different strategies for selecting points to add: roulette wheel selection, random selection, and selecting the best individuals. In all cases, we added half of the population size. And a limit to the number of point was also defined, once the limit on the number of points was reached, the surrogate model was reset, incorporating both the current population and the initial population as reference points. This approach ensures a diverse exploration, as the starting population is random, while also promoting exploitation by including all members of the current population. By empiric experiments this limit was sixteen times the number of fish, this value should change according to the computational cost of the fitness function. Our results indicated that adding the better half of the population yielded the best outcomes. Algorithm 2 presents the pseudo-code for the SMFSS.

3. Experimental Methodology And Results

3.1. Dataset

For the experiments, the p53-MDM2 negative-feedback loop GRN dataset was utilized [Batchelor et al. 2009] which is depicted in Figure 1. In this system, p53 functions as a protein that regulates crucial processes such as metabolism, fecundity and also tumor suppressor. MDM2, another protein, that functions as to regulate p53 levels, helps to maintain cellular balance [Vousden and Prives 2009].

The state vector for the system is $\mathbf{X} = (\text{ATM}, \text{p53}, \text{Wip1}, \text{MDM2})$, while dna dsb acts as an external Boolean input signaling DNA damage as p53 is a tumor-suppressing gene that activates DNA repair mechanisms. The gene interaction parameters a_{ij} can be read from Figure 1. For instance, p53 is activated by ATM and inhibited by WIP1 and MDM2, represented as $a_{21} = +1, a_{22} = 0, a_{23} = -1, a_{24} = -1$.

This dataset contains 4 genes, a known network structure, and biases that the PALLAS algorithm can utilize to improve its search process. In this experiment, we assume negative regulation biases of $b_i = -1/2$ for $i = 1, 2, 3, 4$. The transition noise

Algorithm 2 Pseudo-code of the SMFSS.

```
initialize randomly all fish
for $iteration = 1, 2, \ldots, N$ do
    $useSurrogate = iteration \bmod fitnessCheck \neq 0$
    if $useSurrogate$ then
        Predict fitness with surrogate model
    else
        Evaluate fitness function with PLL
        if $pointsAdded < limit$ then
            Add new points to the surrogate model
        end if
    end if
    Execute individual movement
    if $useSurrogate$ then
        Predict fitness with surrogate model
    else
        Evaluate fitness function with PLL
    end if
    Execute feeding operator
    Execute collective instinctive movement
    Execute collective volitive movement
    Update step size
end for
```

parameter p is selected randomly in the interval $[0.01, 0.1]$. The microarray data model has parameters $\mu_i \equiv \mu = 30$, $\delta_i \equiv \delta = 20$, $\sigma_i^2 \equiv \sigma^2 = 49$, for $i = 1, \ldots, 4$.

In the testing phase, the impact of incorporating or omitting prior information on the surrogate model's predictive capabilities was investigated. The dimensionality of the problem increases significantly when the known data is not utilized, expanding from 9 to 24 dimensions. Specifically, the network dimensions increase from 4 to 16, and the bias dimensions grow from 1 to 4, while other dimensions remain unchanged. This expansion escalates the complexity of the problem.

3.2. Experiments

To evaluate the effectiveness of using surrogate models into the GRN inference problem, firstly 30 independent runs of the original PALLAS algorithm were conducted to serve as a reference for comparison, using the dataset both with and without prior information. For these baseline runs, the parameter settings recommended by [Tan et al. 2020] were used, as detailed in Table 1. Subsequently, an optimization procedure was applied to tune the parameters for both the original PALLAS and the SMFSS version using Optuna [Akiba et al. 2019]. With the optimized parameters, 30 runs for each version were executed.

The optimal parameter values for the SMFSS can differ from those of the original PALLAS algorithm. Considering that and accounting for the introduction of a new parameter, $fitnessCheck$, we performed a parameter optimization using Op-

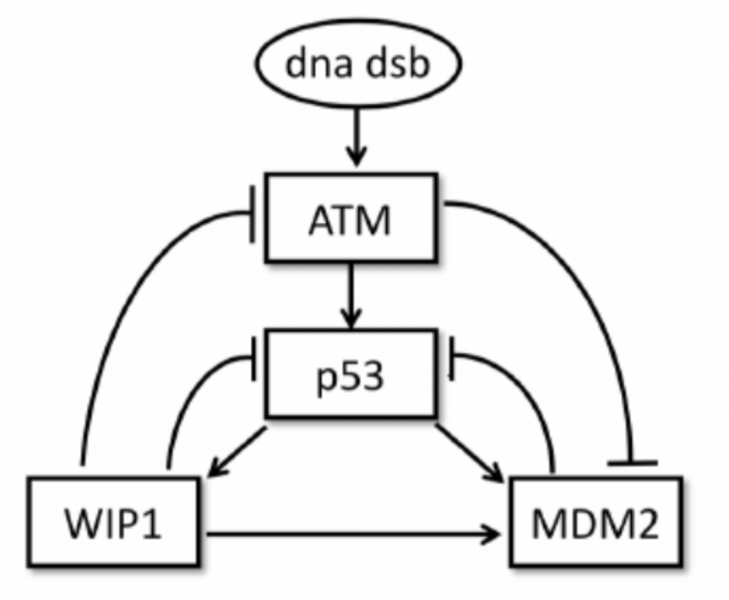

Figure 1. p53-MDM2 negative-feedback loop GRN [Tan et al. 2020].

Table 1. Default Parameters for PALLAS Algorithm

	Value
Number of fish	3 * number of dimension
Iterations	5000
Initial step	0.1 (for all dimension)
Final step	0.00001 (for all dimension)

tuna [Akiba et al. 2019], optimizing parameters for both the original PALLAS and the surrogate-assisted version. This optimization process explored the initial and final steps for each dimension, the number of fish, and the $fitnessCheck$ parameter.

While each algorithm had a different number of fish, we maintained a consistent computational budget by multiplying the Number of fish and Iterations, as defined in Table 1. The optimal values identified by Optuna for the surrogate and PALLAS versions are presented in Table 2 and Table 3, respectively. In the second experiment, the number of iterations was increased to account for the higher computational budget required due to the increase in problem dimensions. This adjustment aligns with the budget definition from the original work [Tan et al. 2020], which scales with the dimensionality of the problem. The parameters net, bias, baseline, delta, and variance represent dimensions of the problem. The $fitnessCheck$ value was different for the execution with and without prior information, being 9 for the case with and 2 for the case without.

3.3. Results

The tests were categorized into two groups: those utilizing prior information and those without. In each group, we evaluated the original PALLAS algorithm with its recommended parameters, the same algorithm with parameters optimized using Optuna, and the SMFSS with parameters also optimized by Optuna. In both scenarios, we observed a performance improvement, in the form of a reduction in the number of fitness function

Table 2. Optuna Parameters for SMFSS

	Value				
Number of fish	206				
Iterations	653				
Fitness Check	9 and 2				
Initial step	Net	Bias	Baseline	Delta	Variance
	0.062	0.033	0.076	0.044	0.025
Final step	0.007	0.006	0.005	0.003	0.004

Table 3. Optuna Parameters for PALLAS version

	Value				
Number of fish	68				
Iterations	1979				
Initial step	Net	Bias	Baseline	Delta	Variance
	0.095	0.061	0.025	0.034	0.019
Final step	0.004	0.008	0.003	0.006	0.006

evaluations required to achieve statistically equivalent results.

For the tests using prior information, the overall final results are shown in table 4 and the convergence curves can be seen in Figure 2. The results of the Mann-Whitney U Test comparing the best results from PALLAS and SMFSS have a U Statistic of 508.00 with a p-value of 0.3953. This indicates that there is not enough evidence to reject the null hypothesis, suggesting that the differences in the distributions of results between the two algorithms are not statistically significant. The box plot of the final results for each algorithm can be seen in Figure 3.

With the $fitnessCheck$ value of 9, the SMFSS evaluated the original fitness function only 72 times out of a total of 653 iterations. This represents 11.02% of the total iterations, resulting in an 88.98% reduction in the number of fitness function calls. This was realized while maintaining results that show no statistically significant difference and, it can be seen in Figure 2, the surrogate version achieves a performance comparable to the other versions while requiring far fewer calls to the fitness functions.

Table 4. Results for execution with prior information

Algorithm Version	PLL Score	Std. Dev.
PALLAS - orig. params	16.78	0.15
PALLAS - optm. params	16.74	0.06
SMFSS - optm. params	16.76	0.11

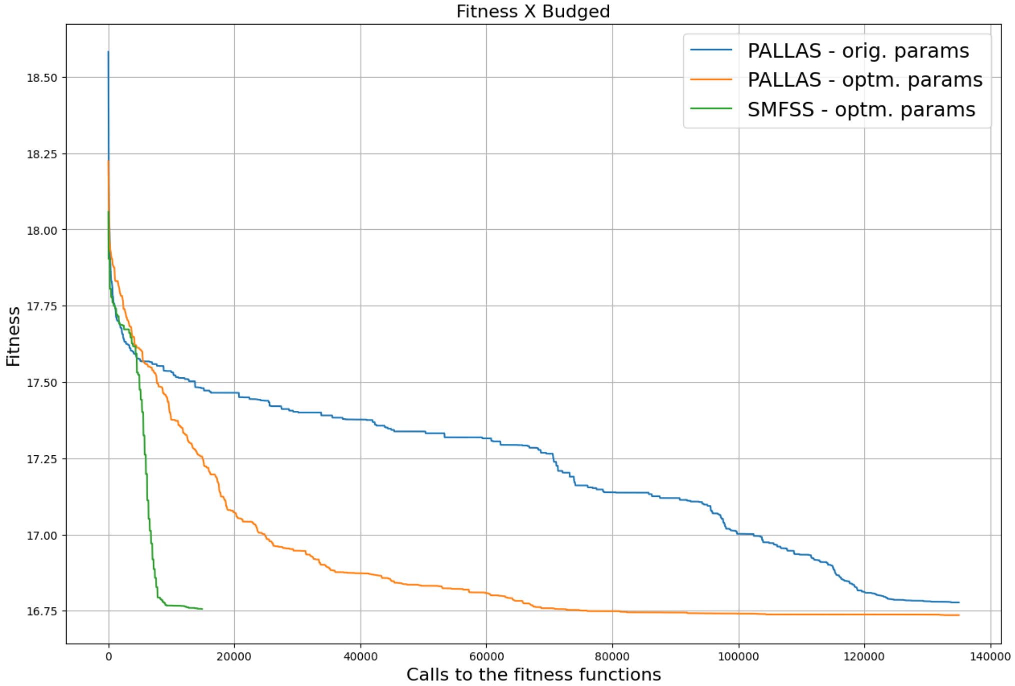

Figure 2. Convergence curve for the algorithms using prior information

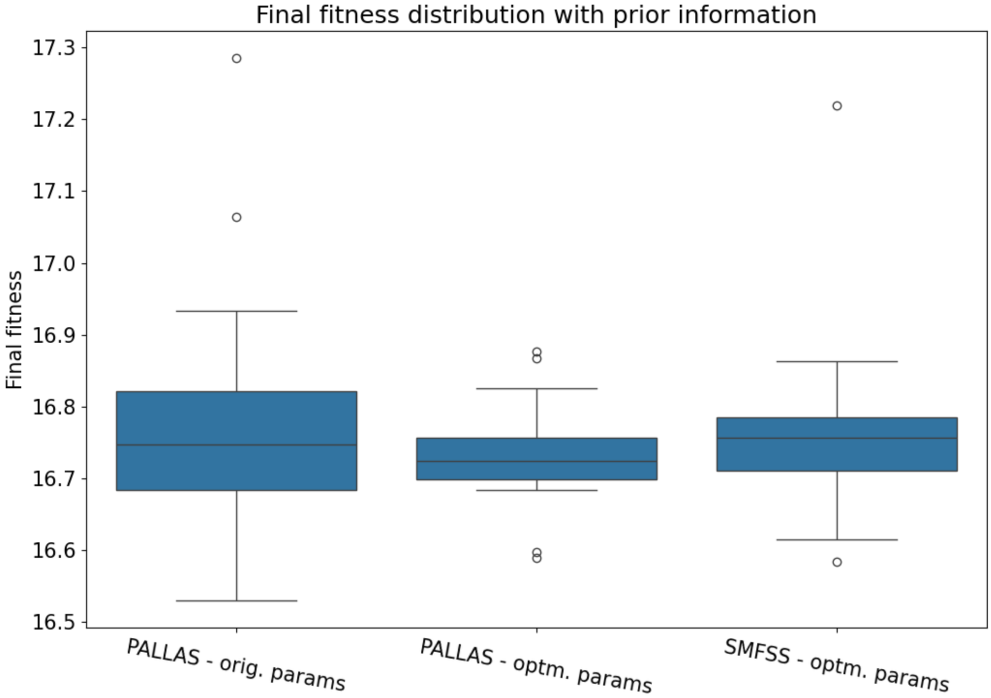

Figure 3. Box plot for the algorithms using prior information

For the experiment without prior data, the Table 5 presents the results, Figure 4 displays the convergence curve and Figure 5 the box plot for the end result. A Mann-Whitney U Test comparing the best PALLAS version, with parameters optimized by

Optuna, and the SMFSS yielded a U Statistic of 478.00 and a p-value of 0.6843. This suggests no statistically significant difference in their result distributions.

Despite having a lower $fitnessCheck$ of 2, this still halves the number of fitness function calls with the surrogate model prediction happening every other iteration. As Figure 4 shows, despite not having a convergence as fast as the PALLAS with Optuna, it still had a relative constant convergence and at the end of the defined budged it achieve results on par with the others versions with half of the number of fitness evaluations.

Table 5. Results for execution without prior information

Algorithm Version	PLL Score	Std. Dev.
PALLAS - original params	17.73	0.18
PALLAS - optm. params	17.72	0.09
SMFSS - optm. params	17.73	0.19

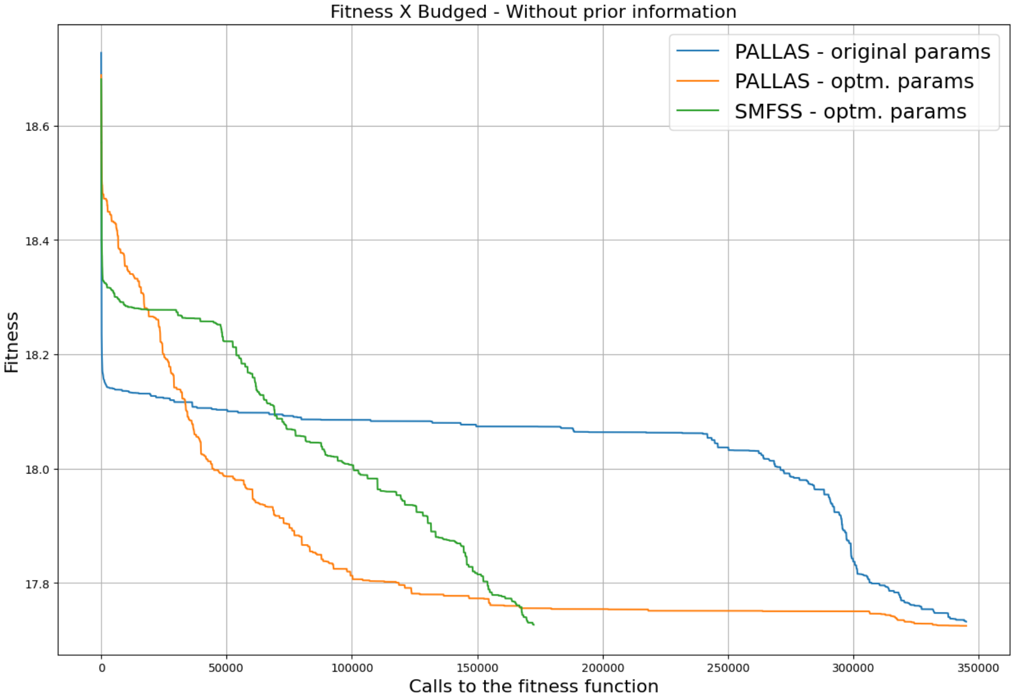

Figure 4. Convergence curve for the algorithm without prior information

4. Conclusion

This study has demonstrated the potential effectiveness of surrogate models, in accelerating GRN inference from time-series data using meta-heuristics and PLL as the metric. By integrating RBF models into the PALLAS-MFSS framework, a reduction was achieved in the number of fitness function calls of 50% and 89%, without compromising the quality of the end results. This efficiency gain happens in two scenarios with different complexities, showing the promise in this approach for making GRN inference more feasible for larger and more complex datasets, and with this helping the understanding of gene regulatory mechanisms.

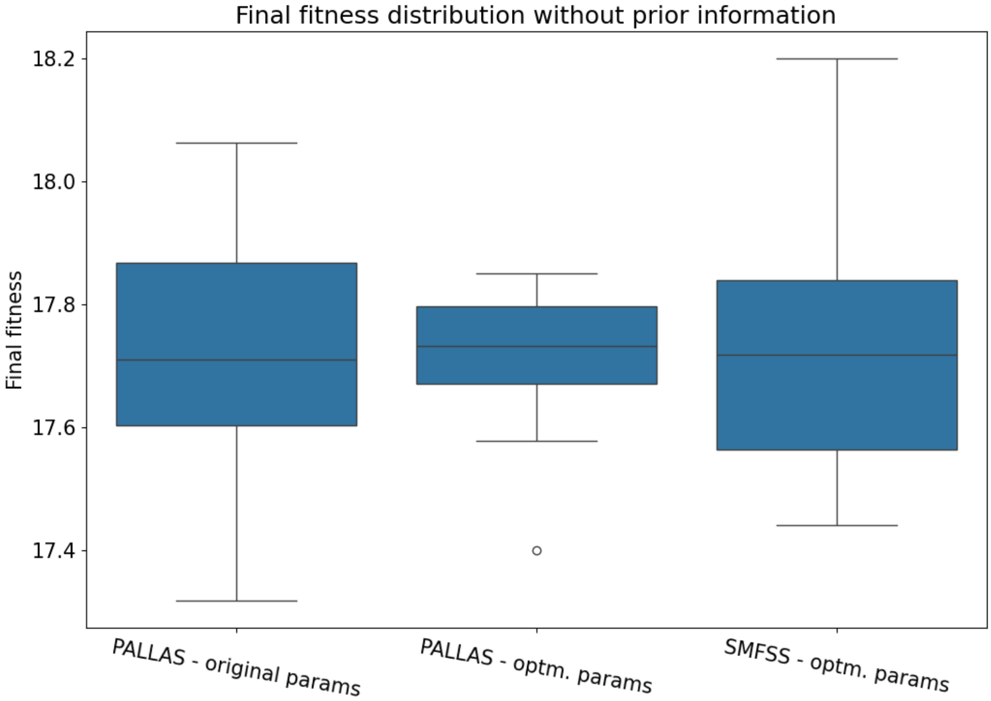

Figure 5. Box plot for the algorithms not using prior information

For future work, there are several possibilities of exploration. Evaluating the approach on diverse datasets, varying in size, number of genes, and network structures, will ensure its generalizability and robustness. Experimenting with alternative surrogate models like neural networks, decision trees, or random forests could further enhance efficiency or accuracy in specific scenarios. Additionally, employing hybrid approaches with multiple surrogate models simultaneously could leverage their strengths for even better performance. By pursuing these research directions, it's possible to improve both the efficiency and effectiveness of GRN inference, making it a more accessible and powerful tool for studying gene regulation.

5. Acknowledgments

We would like to express our gratitude to the authors of the original study [Tan et al. 2020] for making available the source code and dataset utilized in this research. Their availability was necessary for the execution of this work.

Braga-Neto acknowledges the support of the National Science Foundation through NSF awards CCF-2225507 and CCF-1718924.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Bastos Filho, C. J., Neto, F. L., Sousa, M. F. C., Pontes, M. R., and Madeiro, S. S. (2009). On the influence of the swimming operators in the fish school search algorithm. In

2009 IEEE International Conference on Systems, Man and Cybernetics, pages 5012–5017. IEEE.

Batchelor, E., Loewer, A., and Lahav, G. (2009). The ups and downs of p53: understanding protein dynamics in single cells. *Nature Reviews Cancer*, 9(5):371–377.

Braga-Neto, U. (2011). Optimal state estimation for boolean dynamical systems. In *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pages 1050–1054. IEEE.

Buhmann, M. D. (2000). Radial basis functions. *Acta numerica*, 9:1–38.

Eriksson, D., Bindel, D., and Shoemaker, C. A. (2019). pysot and poap: An event-driven asynchronous framework for surrogate optimization. *arXiv preprint arXiv:1908.00420*.

Eskinazi, I. and Fregly, B. J. (2016). An open-source toolbox for surrogate modeling of joint contact mechanics. *IEEE Transactions on Biomedical Engineering*, 63(2):269–277.

Ferreira, J., Pedemonte, M., and Torres, A. I. (2019). A genetic programming approach for construction of surrogate models. In *Computer Aided Chemical Engineering*, volume 47, pages 451–456. Elsevier.

Ibrahim, I., Silva, R., and Lowther, D. A. (2022). Application of surrogate models to the multiphysics sizing of permanent magnet synchronous motors. *IEEE Transactions on Magnetics*, 58(9):1–4.

Imani, M. and Braga-Neto, U. M. (2018). Particle filters for partially-observed boolean dynamical systems. *Automatica*, 87:238–250.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. ieee.

Leal-Romo, F. d. J., Chávez-Hurtado, J. L., and Rayas-Sánchez, J. E. (2018). Selecting surrogate-based modeling techniques for power integrity analysis. In *2018 IEEE MTT-S Latin America Microwave Conference (LAMC 2018)*, pages 1–3.

Marku, M. and Pancaldi, V. (2023). From time-series transcriptomics to gene regulatory networks: A review on inference methods. *PLOS Computational Biology*, 19(8):e1011254.

Sastry, K., Goldberg, D., and Kendall, G. (2005). *Genetic Algorithms*, pages 97–125. Springer US, Boston, MA.

Tan, Y., Neto, F. B. L., and Neto, U. B. (2020). Pallas: Penalized maximum likelihood and particle swarms for inference of gene regulatory networks from time series data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 19(3):1807–1816.

Vousden, K. H. and Prives, C. (2009). Blinded by the light: the growing complexity of p53. *Cell*, 137(3):413–431.