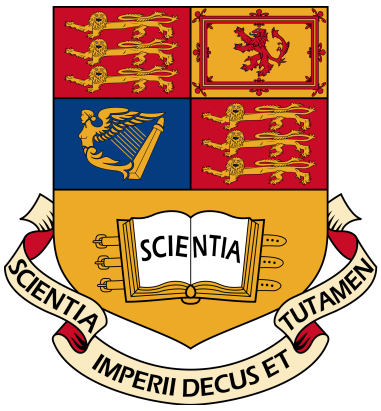
A Clustering-based Short Time Series Analysis Methodology in Omics

Yunsoo Kim, Dr. Tim Ebbels, Slawomir Zukowski

Word Count: 4791

September 5, 2016



IMPERIAL COLLEGE LONDON

COMPUTATIONAL AND SYSTEMS MEDICINE

MSC BIOINFORMATICS AND THEORETICAL SYSTEMS BIOLOGY

Acknowledgments

I would like to acknowledge my supervisor, Doctor Tim Ebbels, for his suggestions and the opportunity to work with him on the project. I would also like to thank Slawomir Zukowsi for his support.

Abbreviations

DB index - Davies Bouldin index

DF - Degrees of freedom

FDR - False Discovery Rate

NMR - nuclear magnetic resonance

PAM - Partitioning Around Medoids

SME - Smoothing-Splines Mixed-Effect

Contents

Abstract

Biological processes are complex temporal events with dynamic behaviors. To understand underlying mechanisms of those processes, time series analysis are vital. However, there is no established methodology for bioinformatics time series mainly because of the characteristics of those time series such as a small number of time points and samples. In this study, a novel method was implemented in Python and extended to include network inference features. It was also tested for sensitivity with 81 scenarios with varying parameters. Mean sensitivity was calculated to be reasonably high (0.8) regardless of the complexity of simulated models, if the models have at least 10 replicates. The method was successfully applied to a real experimental data set from a case-control study of *Schistosoma mansoni* infection. The inferred network highlighted a potential biomarker for the infection, lactate, and portended the methodology's usage as a hypothesis generator.

1 Introduction

Biological processes are temporal events with dynamic and complex behaviors. Thus, for a better understanding of a biological system and its underlying mechanisms, it is very important to study the changes of the system over a period of time. However, in bioinformatics (or omics), high-throughput technologies, such as RNA-Seq, microarray, and nuclear magnetic resonance (NMR) spectroscopy, only provide results from a single time point, so many conventional bioinformatics analyses were limited to a snapshot of the biological system (Voit *et al.,* 2005). To investigate the changes of biological processes over time, results should be collected from mul-

tiple time points. These measurements from multiple time points on an individual biological sample, which is also called as a replicate or an individual such as a patient or a mouse, can be referred as a time series or a time trajectory. These time series analyses can uncover the temporal systematic behavior behind the biological systems. In fact, recent studies on gene expression time series successfully classified genes by time shift patterns, which provided additional information about human brain development (Yuan *et al.,* 2011).

Even though there have been recent successes in omics time series analysis, there is no established (popular) analysis methodology for omics time series, possibly due to the limitations in (or characteristics of) omics time series (Oh *et al.,* 2013). In fact, these limitations make the most of the conventional time series analysis methodologies from other fields such as finance inapplicable. The major limitation is that omics time trajectories are very short (usually less than 20 time points) when compared to those for finance or weather (Gollub *et al.,* 2003). In addition, the number of replicates is often small in omics studies (Bar-Joseph, 2004). In fact, the difficulty of setting up a clinical experiment with many patients and the high cost of other biological samples such as laboratory mice can be the major reasons why the studies have a small number of individuals. Other limitations are highly multivariate data, noise introduced from assay technologies, missing values and non-uniform time sampling from experimental designs (Bar-Joseph, 2004; Voit *et al.,* 2005).

With these limitations in considerations, Ebbels' group in Imperial College has developed a novel method in R to analyze differentially regulated variables between two experimental groups, i.e. controls and cases, for short metabolic time series (unpublished). Although they focused on short metabolic time series data from NMR, their methodology can be applied to short time series from other omics (or short time series in general). This method implemented Smoothing-Splines Mixed-Effect (SME) approach (Berk *et. al.,* 2011). As its name indicates, SME uses a smoothing spline method, which fits a smooth and continuous function to obser-

vations, a fundamental principle of Functional Data Analysis (Ramsay and Silverman, 2005). SME fits a smoothing spline by Expectation Maximization algorithm, which determines the smoothness of the fitted curve by estimating the maximum likelihood and Akaike Information Criterion (Dempster *et al.,* 1977). However, as SME chooses one smoothing factor for each time trajectories, it can often lead to overfitting or underfitting. Thus, to reduce chances of overfitting or underfitting, Ebbels' group developed the method which clusters time trajectories and uses the clusters to determine the smoothness of the spline fit computationally. However, the performance of the methodology was not analyzed with respect to the limitations of short time series or any other limitations (parameters).

Thus, here in this study, this differential time series method was analyzed with various scenarios and parameters to test its performance regarding the limitations of short time series in a rigorous way. The methodology was implemented in Python for its usage as a command line tool as well as a better efficiency. Also, the method was extended to include network inference features for visualization of the structure of time profile similarity/differences between different variables. Two types of networks can be inferred: a correlation network and a differential network. The inferred network can be used to perform further network analysis to get additional information using Cytoscape or any other network analysis tools (Shannon *et al.,* 2003). The method was successfully applied to a real experimental data set as well. The primary research questions are as follows:

1. How does the methodology perform for variations in characteristics (i.e. limitations) of short time series?

2. Does the methodology perform well on a real data set and can the result be used to deduce an underlying mechanism (model)?

3. Can we draw additional information using network inference tools?

2 Methods

The extended methodology can be divided into four main parts: clustering, smoothing spline, permutation test, and network inference. The general overview of the methodology is described in Fig. 1. The methodology accepts a table of time trajectories as an input. The table should have time points as columns, and have each row representing a sample of a variable in an experimental group. If the input table has variables as columns and has time points and samples as rows (a standardized NMR output), then the table is transformed to have the appropriate columns and rows. Annotations for experimental groups should be provided if they are missing from the input table. Missing values in time trajectories are replaced by the data from the previous non-missing time point. It is not recommended to have missing values at the first time point of the input time profile. If the first time point is missing, then the methodology discards that time point for the analysis.

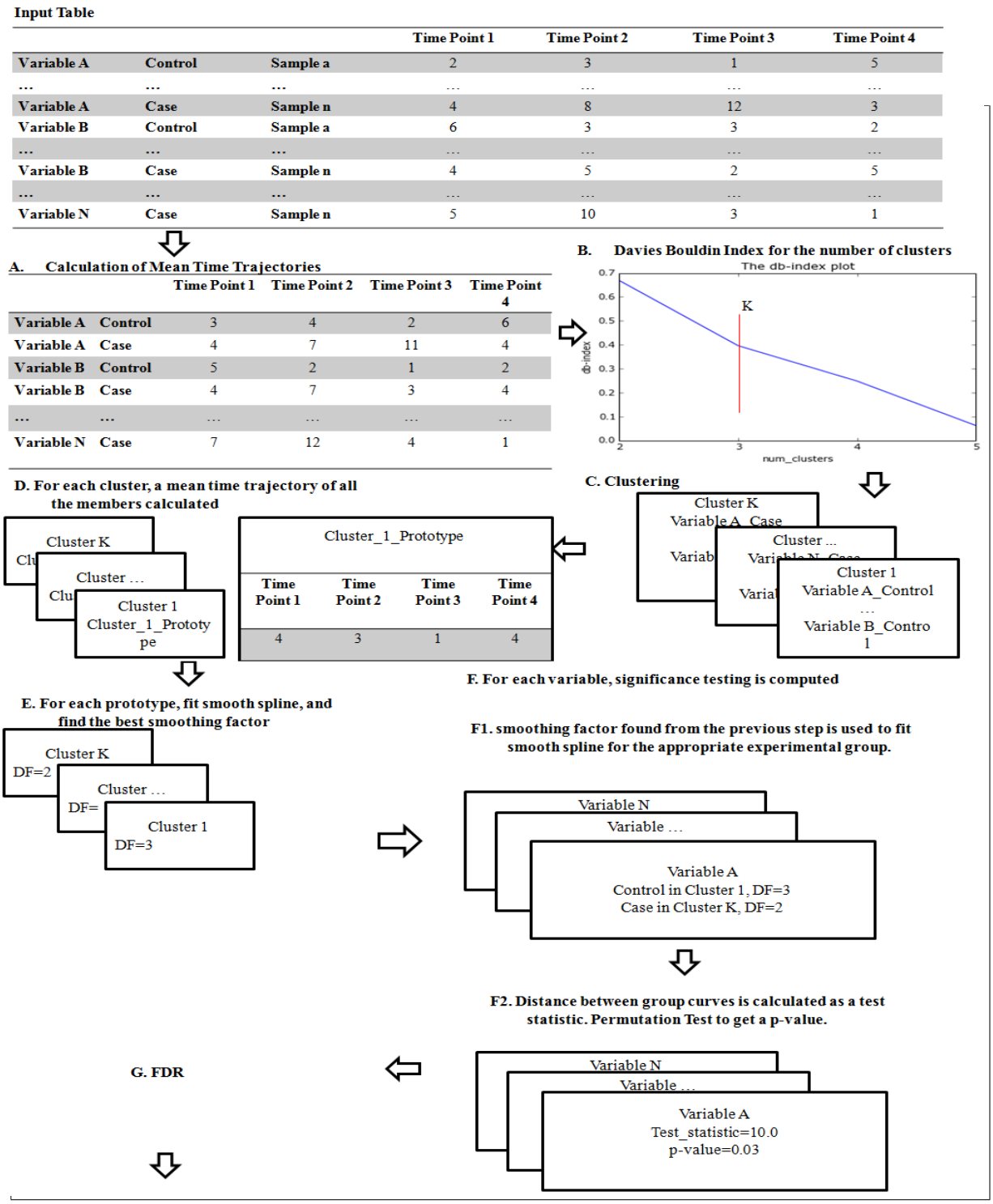


Figure 1: **A workflow of the extended analysis methodology.** This figure provides a general overview of the methodology Input tables should contain time trajectories. Time point measurements can be missing and these missing values are replaced by the previous time point data. A. Mean time trajectory B. Davies Bouldin index to select the appropriate number of the clusters. C. Clustering using K-means or PAM. D. Calculation of mean time trajectory of all members of a cluster. E. Using the cluster prototype (i.e. the mean time trajectory of the cluster), smoothing factor is optimized. F. Significance Testing using the area between fitted curves as Test Statistics and permutation test to calculate raw p-values. G. Benjamini and Hochberg FDR correction of the p-values for multiple testing. Continued on the next page.

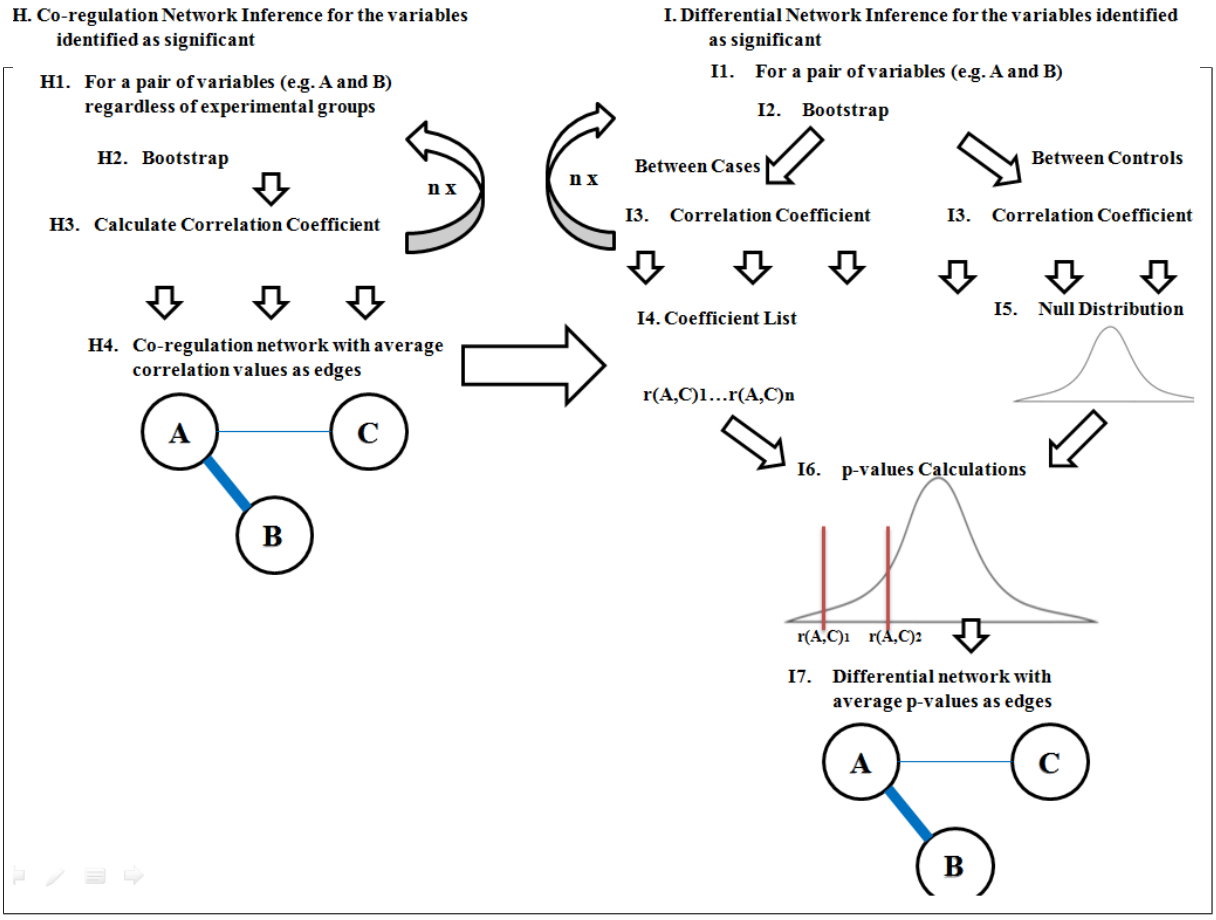


Figure 1. continued. H. Correlation network inference using bootstrapping. It infers a network with an edge representing similarity in time series between a pair of variables. I. Differential network inference using bootstrapping and permutation test. It infers a network with an edge representing how different cases time series to those of controls for a pair of variables.

2.1 Clustering

All the samples in a variable of an experimental group are used to calculate the mean time trajectory. Each mean time trajectory is then scaled by centering (removing the mean) and dividing by the standard deviation, and a distance matrix is calculated for the mean time trajectories. The original methodology used Kendall's tau distance function, but in this extension, the user can choose their preferred distance function, but the default is Kendall's tau distance function (Kendall, 1938). This distance matrix is used to cluster all the mean time trajectories. Partitioning Around Medoids (PAM) or K-means are used to cluster (Theodoridis and Koutroumbas, 2006).

PAM is the default clustering algorithm. It is rank based clustering algorithm and is more robust. K-means is based on Euclidean distances. The methodology provides the user a Davies Bouldin index (DB index) plot, and the user chooses the number of clusters using the plot (Davies and Bouldin, 1979). DB index is a ratio of distances within a cluster to the distances between clusters. Smaller DB index means better clustering. Thus, the number of clusters should be chosen so that it has subsequent numbers of clusters with an increase or very small decrease in DB index values. The user has to consider a good balance between DB index value and the number of clusters in consideration for the decision of the number of clusters.

2.2 Smoothing Spline

The mean of all the profiles in a cluster is calculated as a cluster prototype. Each computed cluster prototype is then used to computationally optimize the smoothing factor for smoothing spline with leave one out cross validation of time points. Here the spline function uses degrees of freedom, which represents the complexity of the model (higher the more complex the model is), as its smoothing factor. The maximum smoothing factor is equal to the number of time points minus one. This maximum limit restricts overfitting. The minimum is 2.0, which goes through all the points (over-fitting). Degrees of freedom (DF) is increased by 0.1 from the minimum to the maximum DF values. The optimization finds the smoothing factor with a local minimum root mean square error (i.e. the difference between the observed value and the predicted value of the spline fit). If there are multiple local minima, then the local minimum at the simpler model (lower DF) is chosen.

2.3 Significance Testing

Then, for each variable, the appropriate smoothing factor determined by the previous step is used to model smooth spline to each experimental group. The area between the fitted curves of the groups is used as test statistics for significance testing. Then, time trajectories are randomly permuted for a number of iterations to generate a null distribution to which the computed test statistics are compared to calculate a p-value. Then, to correct all the p- values

computed for multiple hypothesis testing (family-wise error rate), False discovery rate (FDR) is used (Benjamini and Hochberg, 1995).

2.4 Network Inference

For the variables identified as significant, a correlation network is inferred as well as a differential network. The correlation network shows how similar two significant variables in their time series regardless of experimental groups. For a pair of variables, by bootstrapping (sampling with replacement) n-1 samples, where n is the min-imum number of replicates of the two variables, is done. Then, the bootstrapped samples are used to calculate correlation coefficient by Pearson correlation. This bootstrapping and correlation coefficient calculation are re-peated for a number of times (default: 500). Then, the network is inferred with edges as the average of the correlation coefficients. The correlation network is, therefore, a co-expression/regulation/occurrence network of significantly different variables (i.e. genes, metabolites, or microorganisms) between controls and cases.

The differential network shows how the two variables have significantly different correlation association when comparing cases to controls. It uses a similar approach to the correlation network inference, but for the differential network inference, the two experimental groups are treated as independent of each other, and bootstrapping and correlation coefficient calculation are computed independently. This process is repeated for a number of times, as well. Correlation coefficients calculated from controls make a null distribution to which each correlation co-efficients from cases is compared to calculate raw p-values. FDR is then used to correct the raw p-values. The differential network has edges as p-adjusted (FDR corrected p-values). The differential network highlights sig-nificantly altered relationships between two variables when compared between controls and cases. It shows how different the time profiles are between the two variables in cases when compared to those in controls.

2.5 Implementation

To test the implementation of the methodology in Python, the toy data set was generated with the same parameters that the original method used (Table 1).

The toy data set has three models: linear, parabola, and trigonometry. They are generated with functions listed in Table 1. with parameters values also listed in Table 1. All the models have a Gaussian noise with standard deviation of 15. They also have non-uniform time sampling of 14 time points and have two experimental groups (control and treatment). The time points sampled were (13, 27, 34, 41, 48, 53, 55, 56, 57, 59, 61, 67, and 73 days to mimic a real experimental data set where some time points are closer together than others. Each model has 10 samples (individual time trajectories). Linear model is a null model, where there is no difference between control and cases (treatments). Parabola and trigonometry models have a difference between control and cases (Table 1). The data set was run through the whole methodology except the network inferences because many variables are needed for network inferences.

To analyze the performance of the methodology in a more rigorous way, five models that are frequently observed in biological systems were generated: linear, parabola, exponential decay, logistic (sigmoid), and a Gaussian model to represent a sudden increase and then a decrease in a biological system (Table 2).

Table 1. Toy Data Set. Linear, polynomial, and trigonometry models are used as toy data set. The data are generated using the functions listed below. All the models have two experimental groups (control and case) and have 14 Time Points not uniformly sampled. They also have Gaussian noise with standard deviation of 15. Each model has 10 replicates. Parameters for the functions are also listed. Linear model is a null model (no difference between control and case), and other models have difference between control and case.

Model	Function	Parameters
Linear_control	$a\,x + b$	a=0.5; b=10
Linear_case	$a\,x+b$	a=0.5;b=10
Polynomial_control	$a\,x^2 + b\,x + c$	a=-0.2;b=15;c=50
Polynomial_case	$a\,x^2 + b\,x + c$	a=-0.2;b=10;c=100
Trig_control	$a \times \cos(b\,x) + c$	a=-50;b=0.16;c=100
Trig_case	$a \times \cos(b\,x) + c$	a=50;b=0.16;c=100

Table 2. Performance Analysis Data Set. Linear, polynomial, exponential decay, logistic, and a Gaussian model are used for performance analysis of the methodology. The data are generated using the functions listed below. All the models have two experimental groups (control and case) and have Gaussian noise with standard deviation of 3. The Test column represents the characteristics of time series models that scenarios are analyzing. The List column represents the list of characteristic values that are tested. There is at least one null test for a group (row) of scenarios.

Scenario	Model	Function	Test	List
1-7	Linear	Effect(x)+100	Effect Size	0, 0.005, 0.007, 0.01, 0.015, 0.02, 0.04
8-14	Parabola	$(1/25)(x-50)^2$+effect	Effect Size	0, 0.1, 0.5, 0.75, 1.0, 1.5, 2.5
15-21	Decay	$100^{\text{effect}(x)}$	Effect Size	0, 0.001, 0.0015, 0.002, 0.003, 0.005, 0.01
22-28	Logistic	$\frac{100}{(1+^{\text{ff ct}(x-50))}}$	Effect Size	0, 0.005, 0.007, 0.008, 0.01, 0.02, 0.05
29-35	Gaussian	$100^{\frac{-(x-\text{effect})^2}{2\times 7^2}}$	Effect Size	0, 0.1, 0.2, 0.3, 0.4, 0.5, 1.0
36-41	Linear	Effect(x)+100	Sample Size	5, 7, 10, 12, 15, 20
42-47	Parabola	$(1/25)(x-50)^2$+effect	Sample Size	5, 7, 10, 12, 15, 20
48-53	Decay	$100^{\text{effect}(x)}$	Sample Size	5, 7, 10, 12, 15, 20
54-59	Logistic	$\frac{100}{(1+^{\text{ff ct}(x-50))}}$	Sample Size	5, 7, 10, 12, 15, 20
60-65	Gaussian	$100^{\frac{-(x-\text{effect})^2}{2\times 7^2}}$	Sample Size	5, 7, 10, 12, 15, 20
66-69	Decay	$100^{\text{effect}(x)}$	Number of Time Points	5, 9, 11, 16
70-73	Logistic	$\frac{100}{(1+^{\text{ff ct}(x-50))}}$	Number of Time Points	5, 9, 11, 16
74-77	Gaussian	$100^{\frac{-(x-\text{effect})^2}{2\times 7^2}}$	Number of Time Points	5, 9, 11, 16
78-79	Logistic	$\frac{100}{(1+^{\text{ff ct}(x-50))}}$	Non-uniform Time Points	A constant_interval and An interval proportional to the gradient of the function
80-81	Gaussian	$100^{\frac{-(x-\text{effect})^2}{2\times 7^2}}$	Non-uniform Time Points	A constant_interval and An interval proportional to the gradient of the function

All the models have a Gaussian noise with a standard deviation of 3. The effect of variation of the characteristics is analyzed using many scenarios. The characteristics tested are the limitations of short time series such as sample size, the number of time points, and constant time interval. Then, for all time series profiles, variations in effect sizes, which refer to variations in a parameter of a time series model cases such as slope in linear model cases, and variations in sample sizes in cases were analyzed. In the linear model, the slope was varied. For the parabola, exponential decay, logistic, and Gaussian models, y-intercept (maximum), the rate of decay, the steepness of the curve, and center of the peak (maximum) are varied respectively. For more complex time series models (decay, logistic, and Gaussian models), variations in the number of time points and the effect of non-uniform time point sampling were analyzed (this time both experimental groups are under the same effect). For the default scenario, we have 11 time points uniformly spaced from 0 to 100, and sample size of 20. All the scenarios analyzed in this study are summarized in Table 2. The performance was analyzed by FPR and sensitivity. For all the performance analyses, for each scenario, 50 realizations of positives or negatives were conducted to give a sensitivity value. Then, in total 50 sensitivity values were collected for each scenario. A number of permutation for p-value calculation was 500 iterations. Alpha value for the p-value cutoff was 0.05.

Finally, the real experimental data set from Li *et al.* is used in this study (Li *et al.,* 2011). The study investigated biomarkers for Schistosomiasis using mouse model infected by *Schistosoma mansoni*. Schistosomiasis is a disease found in Asia, South America, and Africa. The original data set had all the metabolites extracted from urine, plasma, and fecal samples of 10 infected mice and 10 control (healthy) mice. The data set used in this study is the selected metabolites from urine samples (Table 3). The selected metabolites are a subset of all the identified metabolites in the original study. The time trajectories have 14 non-uniformly sampled time points. The data was run through the whole methodology with default parameters including the two network inference.

Table 3. Real experimental data set. A list of identified metabolites of urine samples from Li *et al.*

Full name	Short name	Chemical shift
hippurate	Hip	3.97(d);7.84(d);7.55(t);7.64(t)
3-methyl-2-oxovalerate	MOV	2.93(m);1.1(d);1.7(m);1.46(m);0.9(t)
2-oxoadipate	OAP	2.77(t);1.82(m);2.22(t)
2-oxoisocaproate	OIC	2.61(d);2.1(m);
2-oxoisovalerate	OIV	3.02(m);1.13(d)
p-cresol	glucuronide	p-CG
phenylacetylglycine	PAG	7.43(m);7.37(m);3.75(d);3.68(s)
taurine	Tau	3.43(t);3.27(t)
trimethylamine N-oxide	TMA-N	3.28(s)
3-ureidopropionic acid	UPA	2.38(t);3.31(t)
acetate	Ace	1.93(s)
arginine	Arg	3.78(t);1.92(m);1.65(m);3.20(t)
citrate	Cit	2.66(d);2.54(d)
3-carboxy-2-methyl-3-oxopropanamine	CMOPA	2.49(m);1.08(d);3.19(m);3.56(m);3.72(m)
creatine	CRE	3.03(s);3.92(s)
creatinine	CRT	3.03(s);4.05(s)
dimethylamine	DMA	2.72(s)
lactate	Lac	4.11(q);1.32(d)
lysine	Lys	3.78(t);1.92(m);1.47(m);3.03(t);1.72(m)
N-acetylglycoprotein fraction	N-AG	2.06(s)
2-oxoglutarate	OGT	3.01(t);2.45(t)
pyruvate	Pyr	2.36(s)
scyllo-inositol	S-In	3.33(s)
succinate	Suc	2.41(s)

3 Results

3.1 Implementation

The implementation of the methodology was validated with the toy data set generated with the same parameters that were used to verify the original methodology. The implemented method produced a similar distance matrix and determined similar DF factors. Spline fit curves using the determined df factors are presented in Fig. 2. It shows that blue dots are mean time trajectories, and red dots are clustering prototypes. For Fig. 2A., both linear models were identified in the same cluster, and the spline fit works well as the predicted curve is a linear function. For Fig. 2B, both parabola models were identified in the same cluster, and spline fit curve lies really well on the red dots. For panel C and D, each trigonometry models was identified in a cluster of its own, and the spline curves are very different. The result for the significance test agreed with that for the original methodology; polynomial and trigonometry models identified as significant (p adjusted values were less than 0.05). This is expected as linear model time trajectories were sampled with the same parameters, while other models' time profiles were sampled with the different parameters (Table 2).

3.2 Performance Analysis

The performance of the methodology regarding the limitations (characteristics) of short time series was analyzed using the five models represented in Fig. 3 right panels. Variations of effect sizes were tested for all the models, and sensitivity (or FPR for the null model) were calculated. For all the models, an increase in effect size delta increased sensitivity (which is expected as the two groups in a model will be more different with an increase in the effect size delta). Mean of FPR values were around 0.05 as expected. Then, sensitivity did not decrease after it reached 1.0. The overall behavior of the sensitivity plots is similar to that of a logistic plot. In fact, the confidence intervals (the error bars) for the effect size delta around the midpoint of the sigmoid are much larger than those of the ends.

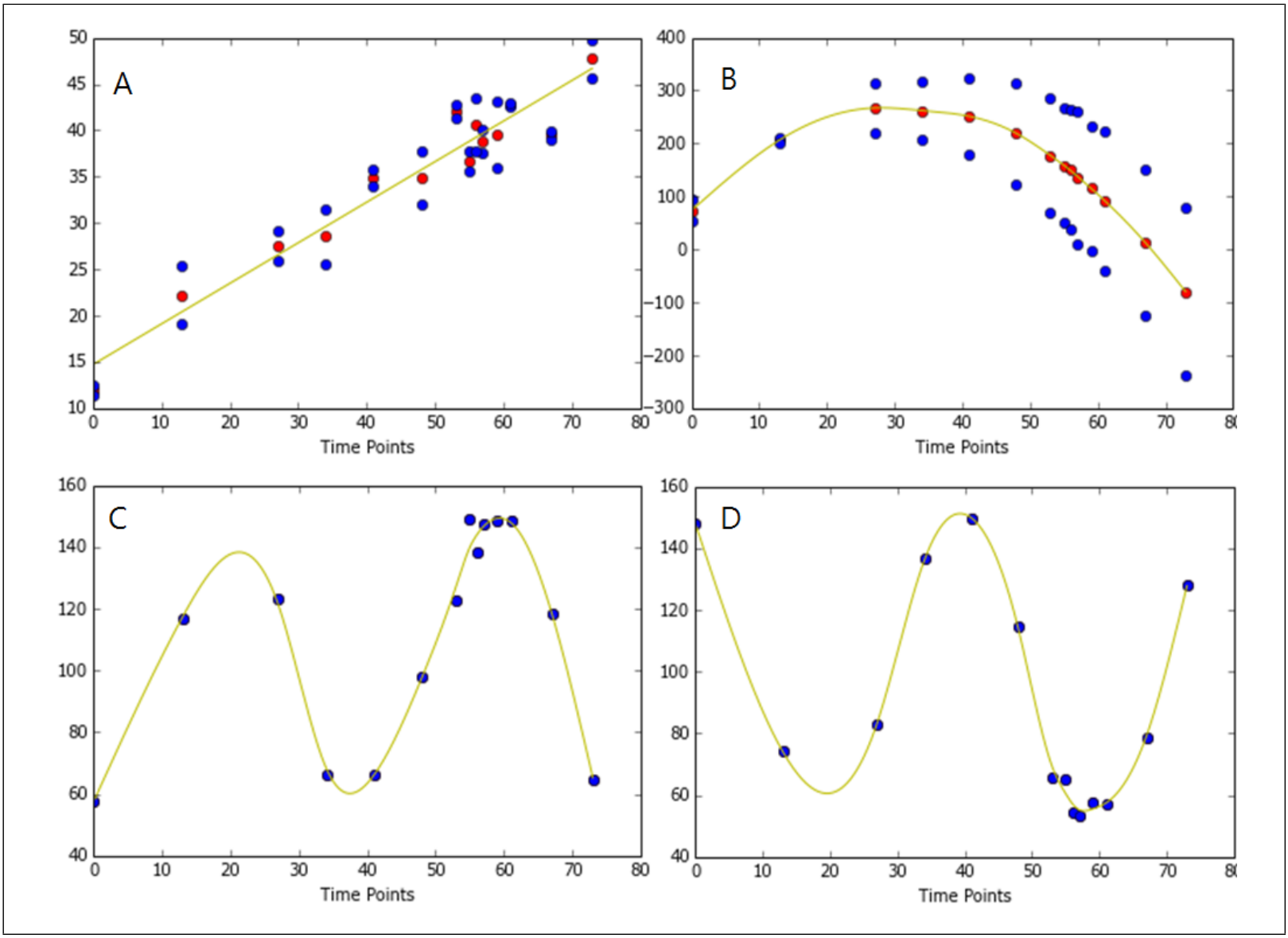


Figure 2: **Spline Fits of the toy data set.** Each panel represents one cluster. A. linear model. B. parabola model. C. trigonometry model. D. trigonometry model. Blue dots represent mean time trajectories. Red dots represent cluster prototypes. If there are no red dots, then the cluster contains only one member (C and D). Yellow lines represent spline fit to cluster prototype. Matplotlib python package was used to plot.

The lowest effect size delta with its 90% confidence interval including 1.0 sensitivity was chosen to generate the right panels. As the right panels of Fig 3. show, 1.0 sensitivity was reached with a detectable difference. A1 has 0.02 as effect size delta, and 1.5, 0.005, 0.02, and 0.5 for B1, C1, D1, and E1 respectively. These effect sizes are used as default values for the further performance analyses of the effect of sample size and number of time points.

Sample sizes of the models were varied to test the effect of smaller samples (Figure 4). All the models de-creased in sensitivity with a decrease in the number of samples in the experiment. Still, the sensitivity did notdecrease to a very low value even with just 5 samples (above 0.5 mean sensitivity). Also, the error bars' widths (i.e. confidence interval) decreased with an increase in sample size.

For more complex models, the effect of the number of time points was investigated. With a decrease in the number of time points, all the models had a decrease in sensitivity. While the decay model had mean of sensitivity values decreased to 0.9, the other models had their mean of sensitivity values decreased to approximately 0.15 with the minimum number of time points tested (5 time points).

For non-uniform time point sampling, negative results were collected. Only logistic and Gaussian models were analyzed because they are the most complex models with their time point (from the number of time point analyses) having a drastic effect on sensitivity values. However, non-uniform time sampling, which was designed to improve the sensitivity value, resulted in lower sensitivity value (Table 4). Negative results for the non-uniform time point sampling. (I am currently running more results to see if the results are actually negative or it was because I chose the wrong time points for the analyses). Both of the models had their sensitivity values decreased by approximately 0.15. Also, the confidence intervals are very similar in all the models.

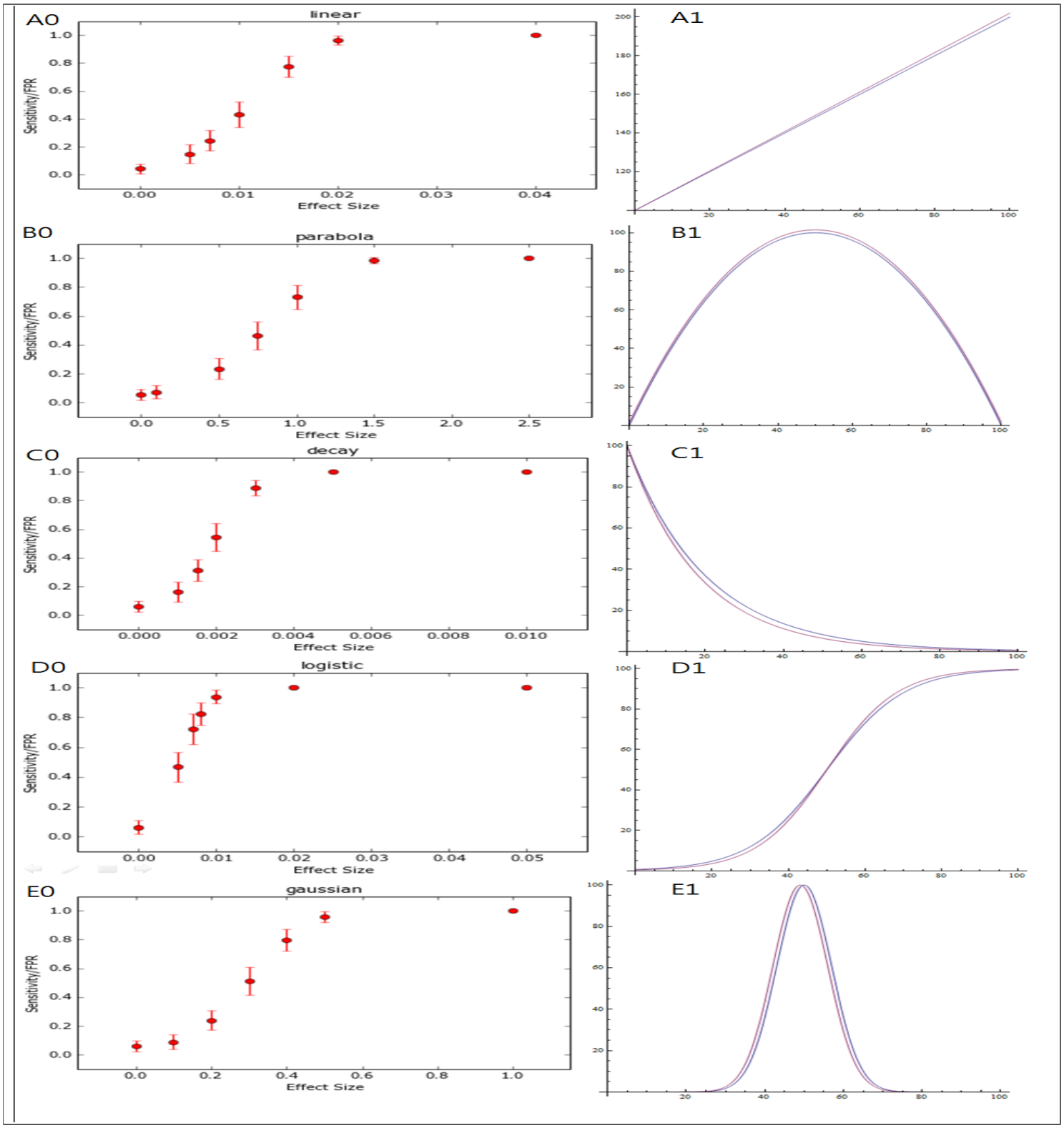


Figure 3: **Analyses on Effect Sizes.** Panels starting with the same letter represent the same model. Panels ending with 0s are effect size delta vs. sensitivity plots. Panels ending with 1s are the visual representation of the appropriate model with the lowest effect size delta with a sensitivity of 1.0 (or near 1.0). A – linear model with its slope as the effect size, B – parabola model with its y-intercept as the effect size, C – decay model with its rate of decay as the effect size, D – logistic model with its steepness as the effect size, E – Gaussian model with its center as the effect size. Error bars on the left panels represent 90% confidence i nterval. The purple lines on the right panels represent cases and the blue lines represent controls. A1 has effect size delta of 0.02, B1 has 1.5, C1 has 0.005, D1 has 0.02, and E1 has 0.5 as their effect size delta. These effect sizes are used for the further performance analyses of the effect of sample size and number of time points. Left panels were plotted using matplotlib, and the rightpanels were plotted using Mathematica.

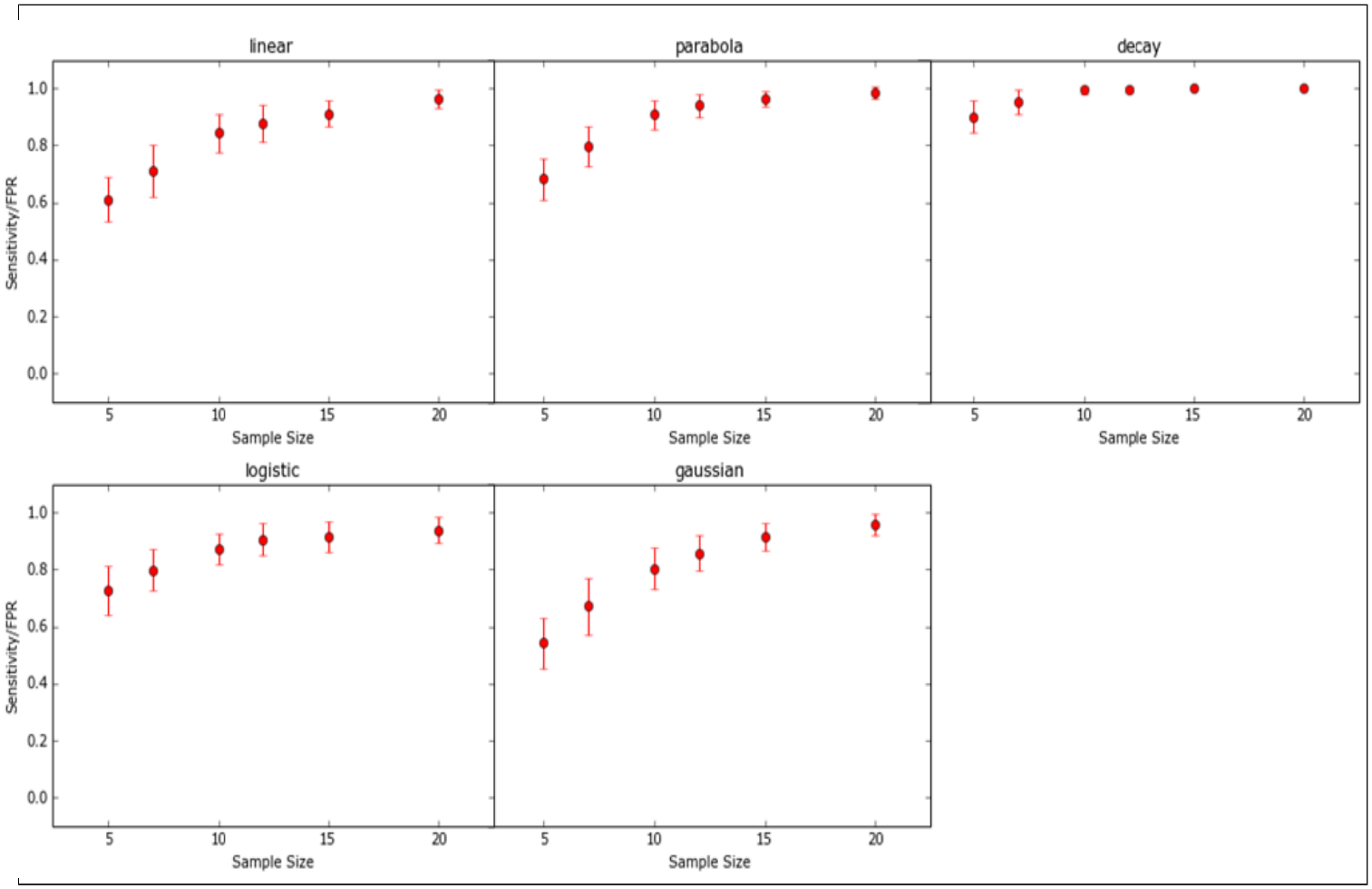


Figure 4: **Analyses on Sample Sizes.** The figure has five subplots and their titles above indicate their models. The plots have sensitivity as y values and sample sizes as x values. Error bars represent 90% confidence interval. Plots were plotted with matplotlib package.

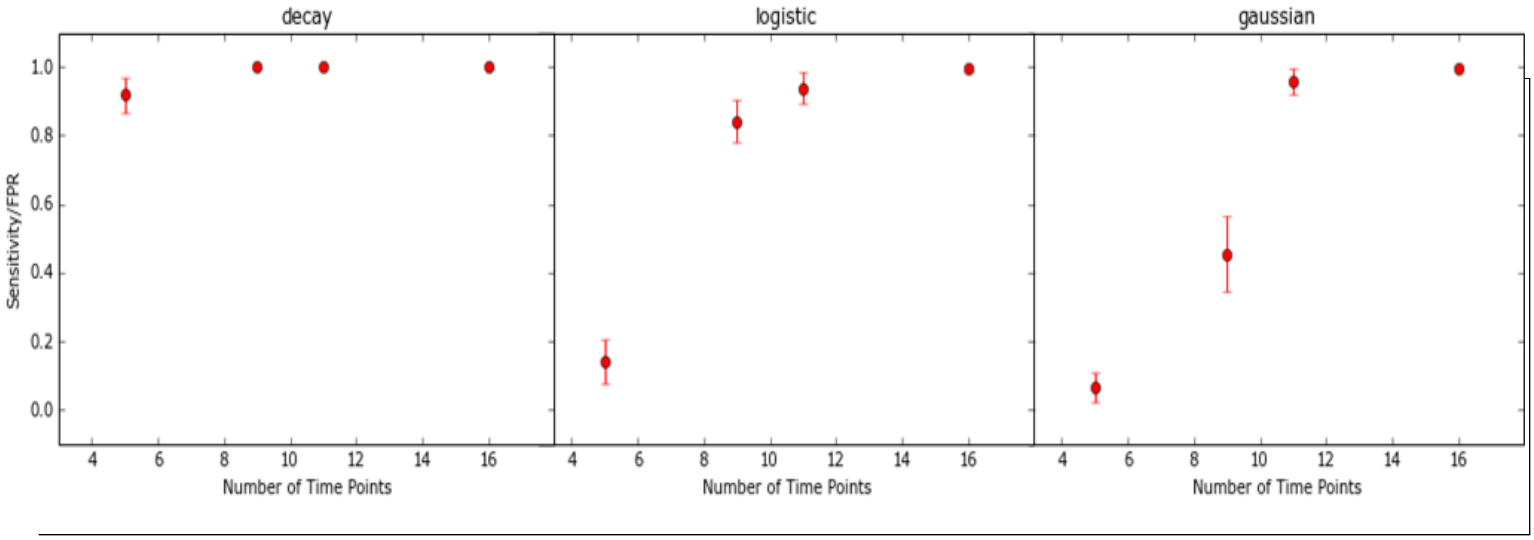


Figure 5: **Analyses on Number of Time Points.** The figure has three subplots and their titles above indicate their models. The plots have sensitivity as y values and sample sizes as x values. Error bars represent 90% confidence interval. Plots were plotted with matplotlib package.

Table 4. Non-uniform Time Sampling. Sensitivity column shows mean of 50 sensitivity values with a 90% confidence interval. The values are rounded to three decimal places.

Model	Sensitivity
Logistic_Uniform	0.466 ± 0.099
Logistic_Non-uniform	0.217 ± 0.075
Gaussian_Uniform	0.511 ± 0.096
Gaussian_Non-uniform	0.362 ± 0.080

3.3 Real Data Analysis

The methodology was applied to a real experimental data set from Li *et al.* The result was comparable to the result acquired from the original methodology (Table 5). All of the significant metabolites from the python package agree with those from the R package. P-adjusted values are only slightly different as well as test statistics.

works were inferred (Fig. 6). The top left panel shows the network of correlation values greater than 0.8, so the edges in the network represent highly positively correlated associations. In other words, they have very similar time trajectories. In fact, the edges are mostly between PAG and itself or p-CG and itself. As highly correlated associations are between the same metabolites, this supports that the network inferred has edges representing sim-ilarity in time trajectories. The reason that the data has several variables for each metabolite is that each metabolite gives rise to several signals in the NMR spectrum.

The differential network provides a different layer of information. It shows how the association between two variables (metabolites) in cases is different from the association between them in controls. In other words, a significant edge shows that time profiles between two variables in cases is much different or less similar when compared to those in controls. For the network on the bottom panel, all the edges are significant, so they represent significantly altered associations between the two significantly altered metabolites when controls and cases are compared. Red colors show negative associations, and green colors represent positive associations. For not significantly altered relationships (the top right panel), it shows that there is no statistically significant evidence to reject that the association between the two variables is altered. In other words, the interactions between the two variables are not significantly altered (at least statistically). Many nodes such as UPA 3.31 and Hip 7.64 are present in both differential networks. Lac 4.11 is notable because it has mostly negative associations in the signif-icant network, while it has mostly positive associations in the non-significant network. As it has edges in red color ranges for the significant network, it shows that time series correlations between lactate and other metabolites are significantly decreased in diseased (case) mice when compared to those in control mice. In other words, the time profiles between lactate and other metabolites in cases are more different (or less similar) than those in controls.

Table 5. Significant Metabolites. Left three columns are the result of the implemented python package and right three columns are the results from the original R package developed by Ebbels' group. Both of the three columns are ordered by the alphabetical order of metabolites. P-adjusted values were rounded to three decimal places. Test statistics were rounded to two decimal places.

metabolites	p-adjusted	Test statistics	metabolites	p-adjusted	Test statistics
Hip_7.55_t	0.004	1096.70	**Hip_7.55_t**	0.011	1097.94
Hip_7.64_t	0.017	474.39	**Hip_7.64_t**	0.011	474.83
Hip_7.84_d	0.010	1061.65	**Hip_7.84_d**	0.011	1062.85
Lac_4.11_q	0.010	1852.45	**Lac_4.11_q**	0.011	1852.11
N-AG_2.06_s	0.017	743.01	**N.AG_2.06_s**	0.011	745.86
OAP_2.22_t	0.028	510.87	**OAP_2.22_t**	0.036	510.87
p.CG_2.3_s	0.000	1102.85	**p.CG_2.3_s**	0.001	1107.19
p.CG_7.06_d	0.000	770.75	**p.CG_7.06_d**	0.003	775.20
p.CG_7.23_d	0.000	1033.47	**p.CG_7.23_d**	0.003	1036.53
PAG_3.68_s	0.004	2783.05	**PAG_3.68_s**	0.001	2819.26
PAG_3.75_d	0.000	4772.53	**PAG_3.75_d**	0.001	4824.12
PAG_7.37_m	0.000	3213.47	**PAG_7.37_m**	0.000	3225.34
PAG_7.43_m	0.000	2054.76	**PAG_7.43_m**	0.000	2062.36
Pyr_2.36_s	0.000	512.66	**Pyr_2.36_s**	0.003	512.11
Tau_3.43_t	0.011	3165.91	**Tau_3.43_t**	0.021	3242.72
UPA_2.38_t	0.000	1726.01	**UPA_2.38_t**	0.004	1726.00
UPA_3.31_t	0.011	1504.53	**UPA_3.31_t**	0.011	1504.96

4 Discussion

In this study, a preexisting method for omics short time series is reviewed, tested, and extended. It was also imple-mented in Python and extended to include network inference features to provide an additional layer of information. The implementation was successful as the results from the implemented methodology confirmed with the results from the original analysis (Fig 2.). The major aim of this research was to investigate the limitations of short time series and the methodology's performance on many different scenarios regarding those limitations. 81 scenarios were analyzed in this study. All the scenarios but 4 of them showed positive (and expected) results. The methodol-ogy can detect differences very well as sensitivity level was very high even with a small change in the effect sizes (Fig. 3). This detection of very small differences between the models can be unrealistic for a real biological data set where the noise level is much higher than that of the one used for the analysis. As the profiles simulated for the study have an amplitude of 100, the signal to noise was about 100/3=33.3 which is very high for biological data. Thus, the detection of a very small difference between the models may be possible because of the low noise level. However, as the methodology was successfully applied to a toy data set and a real biological data set with a higher level of noise, the methodology has a good performance for data sets with noise. Also, sample sizes had the expected effect on sensitivity, but the values did not decrease by a huge amount (Fig. 4.). The biggest change from the highest sensitivity value (1.0) to the lowest (0.55) was 0.45 for the Gaussian model. However, for the other models, the sensitivity value decreased to approximately 0.8. Thus, if the researcher has at least 10 replicates and models with a detectable difference (can be very small as ones in Fig. 3.), the results suggest that regardless of the complexity of the model 0.8 sensitivity can be expected. However, analyses on the number of time points showed a much drastic change in sensitivity values for the logistic and Gaussian models (Fig. 5.). The possible explanation for such a drastic decrease in sensitivity is that these models are more complex as they have regions of stable behaviors (either at 0 or at its plateau) and dynamic behaviors.

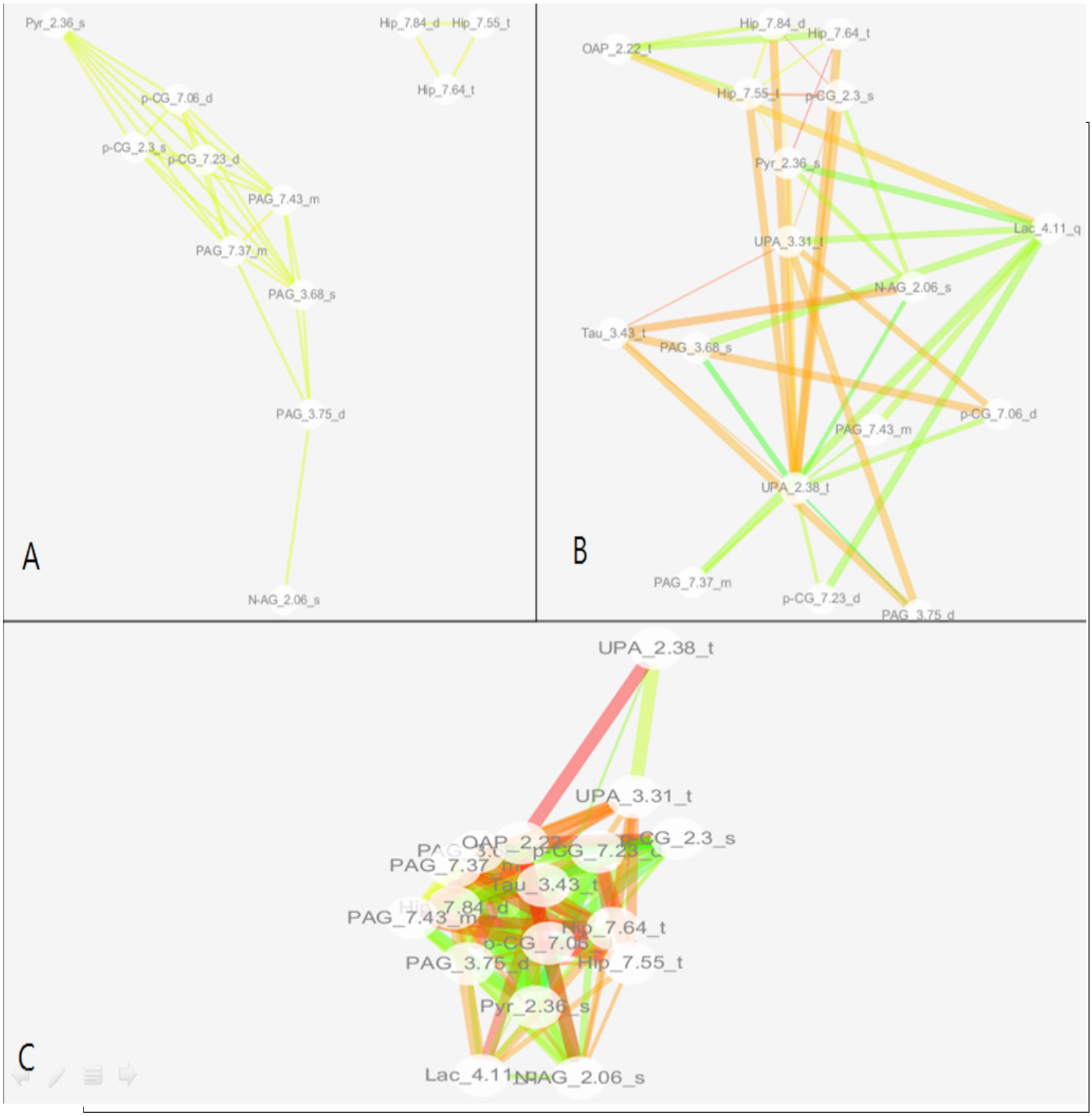


Figure 6: **Inferred Networks.** Top left panel is correlation network. It has edges of only average correlation coefficient greater than 0.8. The top right panel and bottom panel are the differential networks. The bottom is the network with only significant edges (average p-adjusted values less than or equal to 0.05), and the top right one is the network of non-significant edges. The width of the edges for the differential network represents how significant (or not significant the p-value is). Thicker edges represent smaller p-adjusted values (more significant). The nonsignificant network was presented to show that it is not identical to the correlation network. The colors represent if the relationship is negatively associated (red) or positively associated (green). The differential networks are in force-directed layouts.

Thus, if the number of time points is too small, so the measurements are from the stable area of the model, then the difference will not be detected. From the analysis, the underlying mechanisms or models can be deduced with confidence for time series of at least 10 time points. Further investigation is needed on why decay model, although it also has a stable region, did not result in a similar behavior in its sensitivity with respect to a change in the number of time points. Also, further investigation on the non-uniform sampling is necessary as the methodology unexpectedly performed poorly with the manual time sampling (Table 4.). One possible explanation for the unexpected result is that the manual sampling was done incorrectly. Application on real experimental data was also successful as it drew the expected metabolites as the significantly different metabolites. For those significantly different metabolites, networks between them were in-ferred. Correlation network shows a similarity of time trajectories. The same metabolites but with different NMR signals are found to be correlated (have similar time trajectories) by the inference tool (Fig. 6). This is expected as they are the same molecules (with different concentration), so they should be regulated by the same mechanisms.

The differential network is interesting and it provides an additional layer of information. It highlights signifi-cantly altered associations between variables from controls and cases. It can be used as a hypothesis generator as possible changes in interactions of the variables with respect to a condition can be highlighted by the methodology. For instance, Lac 4.11 (lactate) has negative associations with most of its interacting partners for the significantly differential network (Fig. 6). Lactate is known to be involved in energy (glucose) metabolism (Li *et al.,* 2011). Thus, regarding Schistosomiasis, the mice with the disease can have significantly altered glucose metabolism, so its relationship between lactate and other metabolites can be a possible biomarker.

In fact, Li et al. suggested higher lactate levels in mice as the discriminating feature of the infection. Also, a recent study by Howe and colleague suggested that lactate assay can determine the Schistosoma mansoni viability in human (Howe et al., 2015). Although there are many studies highlighting the association between the level of lactate and Schistosomiasis, associations between lactate and its interacting partners regarding the disease have not been thoroughly investigated. Thus, further studies on lactate and its interacting metabolites can discover a new drug target or a new biomarker for Schistosomiasis.

As this methodology has a novel approach, the conclusions made using this method can be quite different from the ones made by using other methodologies. Thus, it is very important to understand the similarities and differences in the approaches to use the appropriate methodology for a specific situation (experiment) and correctly comprehend the results. Smilde *et al.* compared many methods such as for modeling metabolomics time series data (Smilde *et al.,* 2010). The methodologies do not consider sequential ordering of the data, so they treat the data as independent static observations. Also, they used a dataset with the very high number of time points (145), which is unusual for many biological time series data. Smilde *et al.* suggested that the methods that they used are not suitable to model a new biomarker (they are more suitable for modeling pre-selected metabolites). Our method can be used to model a new biomarker using the differential analysis to highlight significantly different variables and network inferences to highlight interactions between variables. Berk *et al.* used smoothing splines to incorporate sequential ordering in the process of analysis. In fact, the analysis uses the fitted curves as the observations (rather than each time point measurements as the observations). However, as the method was at its infancy, it can often overfit a time trajectory with small replicates as it determines each smoothing factor for each time t rajectory. Our method implements smoothing splines as well, but it uses clustering of time series to determine a smoothing factor for the members of the cluster. This clustering approach solved the overfitting issue as it can detect the differences between parabola model of the toy data set (Fig. 2).

The main advantage of the methodology is that it can detect a small difference between models. Also, it can be used on a short time series with a small number of replicates. Although the method's performance is heavily dependent on the number of time points, models with the number of time points greater than 10 should be enough. The novel methodology can be used to model new biomarker and to highlight interesting interactions between variables such as genes or metabolites.

Further investigation on other characteristics of short time series can provide a more rigorous and holistic overview of the performance of the analysis methodology. A number of time points and non-uniform time point sampling can be tested with more scenarios. Implementation of smooth splines in network inference can also be worked in the future.

5 Conclusion

In this study, a novel clustering-based smoothing splines method for omics time series analysis was implemented, extended, and tested. It was extended to included network inference features. It was tested for its sensitivity for 81 scenarios where parameters for time series models were varied. The sensitivity analysis was mainly to investigate the characteristics of short time series which are limitations of omics time series. The result suggested that the methodology can detect a small difference and has a good performance regarding variations in the number of samples. The methodology was successfully applied to a real experimental data set and highlighted a potential biomarker. Thus, the novel method can be used to model unknown biomarkers or to generate hypothesis for biological time series.

6 References

Bar-Joseph,Z. (2004) Analyzing time series gene expression data. *Bioinformatics*. **20**(16), 2493–2503.

Benjamini,Y. and Hochberg,Y. (1995) Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, **57**, 289–300.

Berk,M. *et al.* (2011) A statistical framework for biomarker discovery in metabolomic time course data. *Bioinformatics (Oxford, England)*, **27**, 1979–1985.

Davies,D.L. and Bouldin,D.W. (1979) A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-1**, 224–227.

Dempster,A.P. *et al*. (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B* , **39**, 1–38.

Gollub,J. *et al*. (2003) The Stanford Microarray Database: data access and quality assessment tools. *Nucleic Acids Res.*, **31**, 94–96.

Howe,S. *et al*. (2015) Lactate as a novel quantitative measure of viability in Schistosoma mansoni drug sensitivity assays. *Antimicrob. Agents Chemother.* **59**(2):1193–1199.

Kendall,M. (1938) A New Measure of Rank Correlation. *Biometrika*. **30**, 81–89

Li,J.V. *et al.* (2011) Chemometric analysis of bio fluids from mice experimentally infected with *Schistosoma mansoni*. Parasites & Vectors, 4, 179.

Oh,S. *et al.* (2013). Time series expression analyses using RNA-seq: a statistical approach. *Biomed Res. Int.* **2013**,203681 10.1155/2013/203681

Shannon,P. *et al.* (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, **13**, 2498–2504.

Smilde,A.K. *et al.* (2010) Dynamic metabolomic data analysis: a tutorial review. *Metabolomics*, **6**, 3–17.

Theodoridis,S. and Koutroumbas,K. (2006) Pattern Recognition. *3rd. Academic Press*, San Diego, CA.

Voit,E. *et al*. (2005). Challenges for the identification of biological systems from in vivo time series data. *In Silico Biology* **5**, 83–92

Yuan,Y. *et al.* (2011). Development and application of a modified dynamic time warping algorithm (DTW-S) to analyses of primate brain expression time series. *BMC Bioinformatics* **12**, 347