Efficient Selection of Optimal Time Points Over Biological Time-Series Data

1 Methods

1.1 Problem statement

Our goal is to identify a (small) subset of time points that can be used to accurately reconstruct the expression trajectory for *all* genes or other molecules being profiled. We assume that we can efficiently and cheaply obtain a dense sample for the expression of a very small subset of representative genes (here we use nanostring to profile less than 0.5% of all genes) and attempt to use this subset to determine optimal sampling points for the entire set of genes.

Formally, let G be the set of genes we have profiled in our dense sample, $T = \{t_1, t_2, \ldots, t_T\}$ be the set of all sampled time points. We assume that for each time point we have R repeats for all genes. We denote by e^r_{gt} be the expression value for gene $g \in G$ at time $t \in T$ in the r'th repeat for that time point. We define $D_g = \{e^r_{gt}\,, t \in T, r \in R$ as the complete data for gene g over all replicates and time points T.

To constrain the set of points we select we assume that we have a predefined budget k for the maximum number of time points we can sample in the complete experiment (i.e. for profiling all genes, miRNAs, epigenetic marks etc. using high throughput seq experiments). We are interested in selecting k time points from T which, when using only the data collected at these k points, minimizes the prediction error for the expression values of the unused points. To evaluate such a selection, we use the selected values to obtain a smoothing splines [5, 2, 10] function for each gene and compare the predicted values based on the spline to the measured value for the non-selected points to determine the error. In our problem, t_1 and t_T define the first and end points, so they are always selected. The rest of the points are selected to maximize the following objective 1:

Problem 1. *Given D_g for genes $g \in G$, the number of desired time points k identify a subset of $k-2$ time points in $T \setminus \{t_1, t_T\}$ which minimizes the prediction error for the expression values of all genes in the remaining time points.*

1.2 Spline assignments

Before discussing the actual procedure we use to select the set of time points, we discuss the method we use to assign splines based on a selected subset of point k for each gene. There are two issues that needs to be resolved when assigning such smoothing splines: 1. The number of knots (control points) and 2. their spacing. Past approaches for using splines to model time series gene expression data have usually used the same number of control points for all genes regardless of their trajectories [3, 9] and mostly employed uniform knot placements. However, since our method needs to be able to adapt to any size of k as defined above, we select them indirectly through regularization parameter of the fitted smoothing spline where number of knots will be increased until the smoothing condition is satisfied. In contrast to the existing methods, we also select knots when fitting a smoothing spline.

1.3 Iterative process to select points

Because of the highly combinatorial nature of the time points, selection problem we rely on a greedy iterative process to select the optimal points as shown in Algorithm 1.

There are three key steps in this algorithm which we discuss in detail below.

- *Selecting the initial set of points:* When using an iterative algorithm to solve non convex problems with several local minima, a key issue is the appropriate selection of the initial solution set [6, 7]. We have tested a number of methods for performing such initializations. The simplest method we tried is to uniformly select a subset of the points (so if $k = T/4$ we use each 4th point). Another method we tested is to partition the set of all time points T into $k-1$ intervals of almost equal size by dynamic programming. Then, it uses k interval boundaries including t_1 and t_T as initial solution. Finally, we tested a method that relies on the changes between consecutive time points to select the most important ones for our initial set. Specifically, for this we sort all points except t_1 and t_T by average absolute difference with respect to its predecessor time points by computing:

$$m_{t_i} = \frac{\sum_{g \in G} \sum_{d \in D} |e_{gt_{i-1}} - e_{gt_i}|}{2 \sum_{g \in G} |D_g|} \tag{1}$$

  where gt_i is the average or median expression for gene g at time t. We then select the $k-2$ points with maximum m_{t_i} as the initial solution.

- After selecting the initial set, we begin the iterative process of refining the subset of selected points. In this step we repeat the following analysis in each iteration. We exhaustively remove all points from the existing solution (one at a time) and replace it with all points that were not in the selected set (again, one at a time). For each pair of such point, we compute the error resulting from the change (using the splines computed

based on the current set of points evaluated on the left out time points), and determine if the new point reduces the error or not. Formally, let C_n be set of points for iteration n We are interested in finding a point pair $(t_a \in C_n, t_b \in T \setminus C_n)$ which minimizes the following error for the next iteration $C_{n+} = C_n \setminus \{t_a\} \cup \{t_b\}$:

$$error = \frac{\sum_{g \in G} \sum_{d \in D} \sum_{t \in T \setminus C_{n+1}} |\hat{e}_{gt} - e^d_{gt}|}{\sum_{g \in G} \sum_{d \in D} \sum_{t \in T \setminus C_n} |\hat{e}_{gt} - e^d_{gt}|} \tag{2}$$

where $\hat{e}_{gt}$ is our spline based estimate of the expression of gene g at time t. If there are pairs which leads to an error of less than 1 in the above function, we select the best (lowest error) and continue the iterative process. Otherwise we terminate the process and output C_n as the optimal solution. Note that this greedy process is guaranteed to converge to a (local) minima since the number of time points is finite.

Algorithm 1 Iterative k-point selection

```
1:  procedure ITERATIVE–SELECTION
2:      C_0 = select initial k time points by absolute value sorting
3:      e_0 = error of remaining points by fitting splines to C_0
4:      i = 0
5:      do
6:          for each pair (a, b) ∈ (T − C_i) × C_i do
7:              C* = C_i ∪ {a} \ {b}
8:              e* = estimate error by fitting smoothing spline to C*
9:              if e* < e_i then
10:                 C_{i+1} = C*
11:                 e_{i+1} = e*
12:             end if
13:             i = i + 1
14:         end for
15:     while e_{i+1} < e_i
16:     Output C_{i+1} and e_{i+1}
17: end procedure
```

- Third key step of our approach is fitting smoothing spline to gene independently for selected subset of time points. Smoothing splines are capable of modeling arbitrary nonlinear shapes as well they do not have the problems seen in other polynomial fitting methods such as Runge's phenomenon. Smoothing splines perform quite well in preventing overfitting [10]. Let $R = \{(t, y_t), t \in C\}$, and μ be the spline we are interested in fitting, smoothing spline can be found by the following optimization problem which minimizes regularized squared error:

$$min \sum_{(t,y_t) \in R} (y_t - \mu(t))^2 + \lambda \int_0^{T_{max}} (\mu^{''}(x))^2 \tag{3}$$

where λ is the regularization parameter which prevents overfitting. We have estimated regularization parameter by leave one out cross validation in our experiments. λ also affects the number of knots selected.

1.4 Individual vs. Cluster based Evaluation

In section 1.3, we assume that error of each gene has same contribution to the overall error. However, this assumption ignores the fact that expression of genes are correlated with the expression of other genes. To take the correlation between genes into account, we have also performed cluster based evaluation of genes where we analyzed the error by weighting each gene in terms of inverse of the numbers of genes in the cluster it belongs. This scheme ensures that each cluster contributes equally to the resulting error rather than each gene. We find clusters by k-means clustering algorithm over time series-data as well as over a vector of randomly sampled time points on fitted spline [4]. We use Bayesian Information Criterion (BIC) to determine the optimal number of clusters [8].

2 Results

2.1 Datasets and Implementation

We have used mRNA and miRNA expression datasets as well as histone methylation dataset over Mus Musculus in our analysis. mRNA dataset is obtained via NanoString including the expression of 134 selected genes that are effective in lung development. On the other hand, larger miRNA dataset profiles the expression of 599 microRNAs. Both mRNA and miRNA datasets are composed of 42 time points between the first and 28'th days in mouse development. mRNA dataset has between 3 and 5 repeats for each time point whereas miRNA dataset has between 3 and 4 repeats for each time point.

We implemented our method in Python. Our method, analysis code and detailed results are available on `https://github.com/emresefer/geneexpress`.

2.2 Subset of important time points across multiple genes

We identified the subset of important time points across multiple genes over mRNA dataset. Fixing initial (0.5'th day), last (28'th day), and close-mid point (7'th day) in advance, we identified total of 13 points as 0.5, 1.0, 1.5, 2.5, 4, 5, 7, 10, 13.5, 15, 19, 23, 28 out of 42 points we tested for via our method. We find that best informed guess is to initialize using points that are sorted by increasing absolute differences. We also find top 10 solutions lead to errors that are very similar to the ones we used (See Supplementary Figure 1).

We analyze the performance of our method with different initial heuristics. We find significant performance improvement over randomly selected points as in Figure 1 in terms of mean squared error. Sorting initial points by absolute values performs better than other methods in almost all selected number of points. Our method performs as good as noise by increasing number of selected

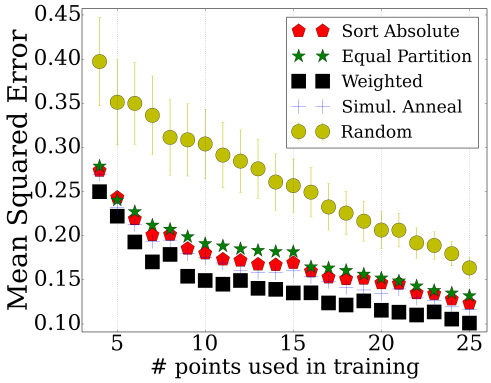
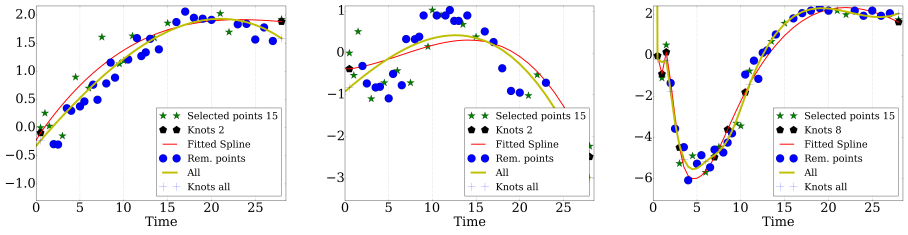


Figure 1: Performance of our method by increasing number of selected points



(a) PDGFRA (b) ELN (c) INMT

Figure 2: Expression profiles over several genes a) PDGFRA, b) ELN, c) INMT

points which is the limiting factor in our method's performance. Overall, we can approximate expression values of each gene quite accurately by using 10 points out of 42 points, almost as good as noise of the data.

Figure 2 shows the predicted expression curves by our method over 13 points (in addition to the initial and last time points) over genes with different expression curves. Selected points can accurately approximate different gene expression profiles.

2.3 Identified time points are predictive for miRNA

We show that the time points identified over mRNA dataset can reconstruct the expression profiles for miRNA dataset quite accurately as in Figure 3. Error decreases by increasing number of selected points similar to mRNA dataset. Identified points perform significantly better than the random set of points of same size. Using the 13 selected mRNA points leads to an average error of 0.3312 whereas optimal points for microRNA lead to relatively similar error (0.3042) indicating that mRNAs can serve as a general proxy for selecting time points.

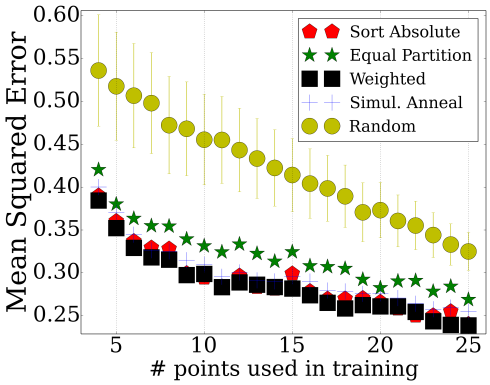


Figure 3: Performance analysis over miRNA dataset

2.4 Detailed Analysis of miRNA Dataset

Detailed analysis of miRNA dataset shows clustered expression profiles of miRNAs as in Figure 4. We identified 8 stable clusters by k-means [6] where number of clusters is selected by Bayesian Information Criteria [8]. We find clusters to change more frequently than mRNA data since miRNA is noisier than mRNA data. Identified clusters are also enriched for several Gene Ontology biological processes [1].

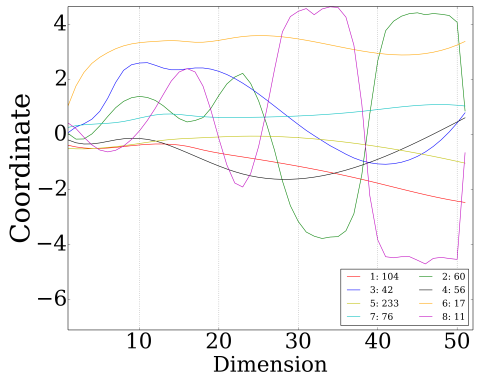
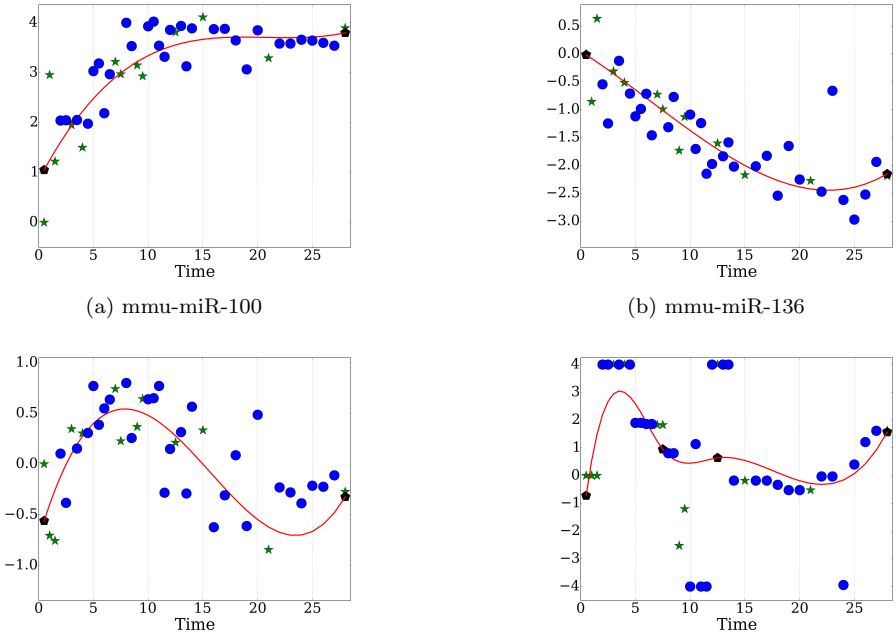


Figure 4: 8 stable clusters

We also predict the expression profiles of several miRNAs by fitting spline over time points identified over mRNA dataset as in Figure 5. Spline-based reconstruction performs better than linear reconstruction similar to mRNA dataset. Accurate prediction of expression profiles of different miRNAs show the importance of our method in reconstructing the expression profiles. Among these miRNAs, mmu-miR-100 targets Fgfr3 and Igf1r, mmu-miR-136 targets Tgfb2. Similarly, mmu-miR-152 targets Meox2, Robo1, Fbn1, Nfya. Lastly,

mmu-miR-219 targets PDGFRA, Eya2, Esr1,2, Efnb2 and Robo1 which are associated with BPD in preterm infants.



(a) mmu-miR-100

(b) mmu-miR-136

(c) mmu-miR-152

(d) mmu-miR-219

Figure 5: Predicted expression profiles of different miRNAs

References

[1] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[2] Ziv Bar-Joseph, Georg K Gerber, David K Gifford, Tommi S Jaakkola, and Itamar Simon. Continuous representations of time-series gene expression data. *Journal of Computational Biology*, 10(3-4):341–356, 2003.

[3] Ziv Bar-Joseph, Anthony Gitter, and Itamar Simon. Studying and modelling dynamic biological processes using time-series gene expression data. *Nature Reviews Genetics*, 13(8):552–564, 2012.

[4] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[5] Carl De Boor. A practical guide to splines. *Mathematics of Computation*, 1978.

[6] John A Hartigan. *Clustering algorithms*. John Wiley & Sons, Inc., 1975.

[7] Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

[8] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[9] Rohit Singh, Nathan Palmer, David Gifford, Bonnie Berger, and Ziv Bar-Joseph. Active learning for sampling in time-series experiments with application to gene expression analysis. In *Proceedings of the 22nd international conference on Machine learning*, pages 832–839. ACM, 2005.

[10] Grace Wahba. *Spline models for observational data*, volume 59. Siam, 1990.