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 spline [7, 3, 16] 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 [4, 15] 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 cubic smoothing spline where number of knots will be increased until the smoothing condition is satisfied [16]. Regularization parameter is estimated by leave-one-out cross-validation (LOOCV).

1.3 *TempSelect*: 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 [9, 12]. 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 4'th point). Another method we tested is to partition the set of all time points T into $k-1$ intervals of almost equal size. This method determines these boundares by estimating the cumulative number of points until each time point and selecting time points with cumulative values $\frac{T}{k-1}, 2\frac{T}{k-1}, \ldots, (k-2)\frac{T}{k-1}$ respectively. 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, we sort all points except t_1 and t_T by average absolute difference with respect to its predecessor time points by computing:

$$m_{t_i} = \frac{\sum_{g \in G} |Md(e_{gt_{i-1}}) - Md(e_{gt_i})| + |Md(e_{gt_{i+1}}) - Md(e_{gt_i})|}{2|G|} \quad (1)$$

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

- *Iterative improvement step:* 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 $T^- = T \setminus \{t_1, t_T\}$ and 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 ratio for the next iteration $C_{n+} = C_n \setminus \{t_a\} \cup \{t_b\}$:

$$error\ ratio = \frac{error(C_{n+})}{error(C_n)} = \frac{\sum_{g \in G} \sum_{r \in R} \sum_{t \in T \setminus C_{n+}} (\hat{e}_{gt}^{C_{n+}} - e_{gt}^{r})^2}{\sum_{g \in G} \sum_{r \in R} \sum_{t \in T \setminus C_n} (\hat{e}_{gt}^{C_n} - e_{gt}^{r})^2} \quad (2)$$

where $\hat{e}_{gt}^{C_n}$ is our spline based estimate of the expression of gene g at time t by fitting smoothing spline over points C_n. If there are pairs which leads to an error ratio of less than 1 in the above function, we select the best (lowest error), assign it to C_{n+1} 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 *TempSelect*: Iterative k-point selection

```
1:  procedure ITERATIVE–TEMPORAL–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 (t_a, t_b) ∈ (T^- \ C_i) × C_i do
7:              C* = C_i ∪ {t_a} \ {t_b}
8:              e* = estimate error by fitting smoothing spline to C* where
                     regularization parameter is estimated by LOOCV
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 and e_i
17: end procedure
```

- *Fitting smoothing spline:* Third key step of our approach is fitting smoothing spline to every gene independently for selected subset of time points. Smoothing splines are capable of modeling arbitrary nonlinear shapes as well as 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 [16]. Let $I_g = \{(t, Md(e_{gt})),\ t \in C\}$, and μ be the spline we are interested in fitting, smoothing spline can be found by the following optimization problem which minimizes penalized least-squares error:

$$\min \sum_{(t,y_t)\in I_g} (y_t - \mu(t))^2 + \lambda \int_{t_1}^{t_T} \mu^{''}(x)^2 dx \tag{3}$$

where λ is the regularization parameter which prevents overfitting. We have estimated regularization parameter by leave-one-out cross-validation (LOOCV) 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 profiles of genes are correlated with the expression of other genes. To take the correlation between gene profiles 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 by treating each gene as a point in R^T space as well as over a vector of randomly sampled T time points on fitted spline [6]. We use Bayesian Information Criterion (BIC) to determine the optimal number of clusters [14].

1.5 More Complex Iterative Improvement Procedures

We also propose the following more complex iterative improvement procedures for *TempSelect*:

- We add and remove b time points in each iteration instead of a single point. This increases the complexity of each iteration from $O(GT^2Q)$ to $O(GT^{2b}Q)$ where Q is the complexity of fitting a smoothing spline.
- We run simulated annealing to escape from local minima [10]. In this case, we do not try all pairs of points and select the one with minimum error in each iteration, but instead move to a neighbouring solution with probability 1 if its error e^r is lower than error of current solution e^i whereas we move to a solution with probability $e^{-T(e^r - e^i)}$ if $e^r \geq e^i$. Here, T is the temperature that increases by increasing number of iterations and the probability of moving to a solution with larger error decreases over time.

Even though both approaches should escape from local minima theoretically better than the greedy approach we described above, they do not perform better in practical instances.

2 Results

2.1 Datasets and Implementation

We developed a method *TempSelect* to select a subset of k time points from an initial larger set of n points such that the selected subset provides an accurate, yet compact, representation of the temporal trajectory. The method utilizes splines to represent temporal profiles and implements a cross validation strategy to evaluate potential sets of points. Following initialization which is based on the expression values, we employ a greedy search procedure that adds and removes points until a local minima is reached. The resulting set is then used for the larger genomic and epigenetic experiments. To test this method and to demonstrate its ability to reduce time, costs and samples while still providing accurate description of the temporal profiles, we focused on experiments related to lung development in mice. We implemented *TempSelect* in Python. Its implementation, code, detailed results, and datasets are available on `https://github.com/emresefer/geneexpress`.

We have used mRNA and miRNA expression datasets as well as temporal histone methylation dataset over Mus Musculus in our analysis. mRNA dataset is obtained by profiling the expression of 134 selected genes that are determined to be effective in lung development via NanoString array at a high sampling rate. On the other hand, larger miRNA dataset profiles the expression of 599 microRNAs. Both mRNA and miRNA datasets includes samples at 40 time points between the half and 28'th days in mouse development. mRNA dataset has between 2 and 4 repeats for each time point whereas miRNA dataset has between 3 and 4 repeats for each time point. We normalized mRNA dataset by quantile normalization followed by $\log 2$ transformation whereas miRNA values are normalized by variance mean normalization [**?**]. Methylation data has 3 repeats for time points 0.5, 1.5, 2.5, 5, 10, 15, 19, 26 for 266 loci belonging to 13 genes. Among these genes all of them except Zfp536 also exist in mRNA dataset. Supplementary Table 1 summarizes the number of loci for each gene in methylation dataset. We used shifted percentage of methylation at each time point as our dataset which is obtained by subtracting the median percentage of methylation at initial time point (baseline) from all data points for each gene.

2.2 *TempSelect* identifies subset of important time points across multiple genes

While our method can be used to select any number of time points, to demonstrate its utility we have tested it in the following setting. First we fixed a set of points in advance (first (0.5'th day) and last (28'th day), which are required for any setting and day 7 which was previously determined to be of importance to lung development, see Supporting Results for other settings). In addition, we have asked *TempSelect* to further select 10 more points (for a total of 13). For this setting, the method selected the following points: 0.5, 1.0, 1.5, 2.5, 4, 5, 7, 10, 13.5, 15, 19, 23, 28 out of 40 points. While we do not know the ground truth,

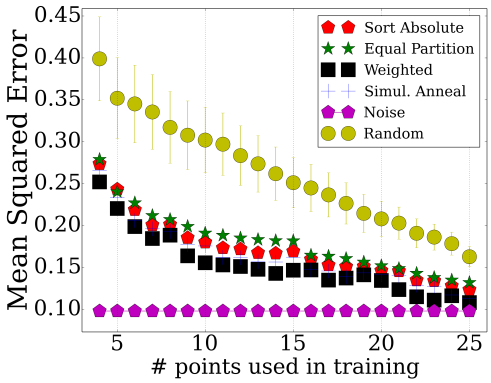
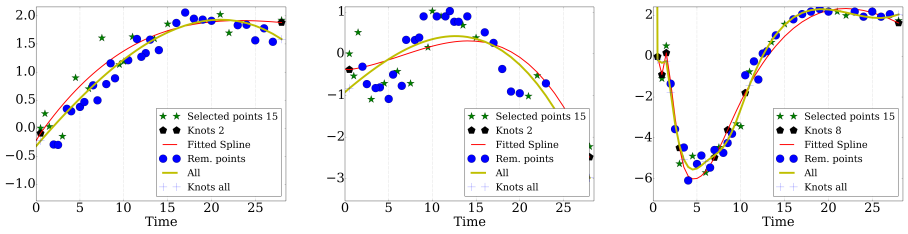


Figure 1: Performance of *TempSelect* by increasing number of selected points

the larger focus on the earlier time points determined by the method (with 7 of the 13 points for the first 7 days) makes sense in this context as several aspects of lung differentiation are determined in this early phase [8]. The other 3 weeks were more or less uniformly sampled by our method. This highlights the usefulness of an unbiased approach to sampling time points rather than just uniformly sampling through the time window.

We tested the performance of our method by using it to select subsets of size 3 to 25 time points. To determine the accuracy of the reconstructed profiles using the selected points, we computed the average mean squared error for points that were not used by the method (Methods). The results are presented in Figure 1 which also plots a comparison between the performance of our method and two baseline approaches: a random selection of points and uniform sampling which is often used in such experiments [4]. We have also compared the performance of the different strategies to initialize the set of points (sort by absolute differences, equal partition) and to perform the search (simulated annealing, weighting genes by cluster size). Finally, the figure includes a comparison between the performance of each of these strategies and the noise in the data (computed based on the repeat information) which is the theoretical limit for the performance of any profile reconstruction method. As can be seen in the Figure, we find significant performance improvement over randomly selected points in terms of mean squared error. Sorting initial points by absolute values further improves the performance highlighting the importance of initialization when searching large combinatorial spaces. As the number of points used by the method increases, it leads to results that are very close to the error represented by noise in the data (0.098) even when only using half the points. Relative order of the methods do not change if we also use additional anchor points E16.5 and E18.5 in our analysis (Supplementary Figure 7).

Figure 2 presents the reconstructed and measured expression values for a few genes using 13 time points (in addition to the initial and last time points). Figure plots median value for each time point as well as selected knots for



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

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

spline reconstruction. Note that even though each of these genes had a different trajectory and different inflection points, the selected set of points enables our method to fit all of these pretty well without overfitting.

2.3 Identified time points over mRNA are predictive of miRNA profiles

To test the usefulness of our method for predicting the correct sampling rates for other genomic datasets, we next profiled mouse miRNAs for the same developmental process. miRNAs are important for lung development as several miRNAs are differentially expressed during lung development [17] and some miRNA families are involved in suppression of lung tumors [11]. Unlike the mRNA dataset, which utilized prior knowledge to profile less than 1% of all genes, the miRNA dataset profiled 599 miRNAs, more than 50% of known mouse miRNAs. Thus, such data represents an unbiased sample and can provide information on whether using one type of genomic data can be helpful for determining rates for other types.

To test the usefulness of our approach, we used the miRNA expression values for the time points determined by the mRNA analysis to reconstruct the complete trajectories for each miRNA. The results are presented in Figure 3. In addition to the comparison included in the mRNA figure, the miRNA figure includes the optimal results for using miRNA data (as opposed to mRNA data) to select the points. As can be seen, the points selected by the mRNA analysis leads to reconstruction that is much better than when using random points ($p < 0.01$) highlighting the relationship between the two datasets and the ability to use one to determine points for the other. Further, performance using the mRNA set is very similar to the performance using the miRNA data itself. For example, when using the 13 selected mRNA points the average mean squared error is 0.3312 whereas when using the optimal points based on the miRNA data itself the error is 0.3042. This serves as a strong indication that mRNAs can serve as a general proxy for selecting time points.

Figure 4 presents the reconstructed and measured expression values for a few miRNAs using time points identified over mRNA dataset. Spline-based recon-

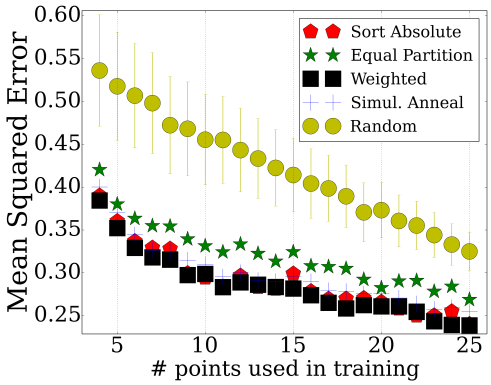


Figure 3: Performance analysis over miRNA dataset

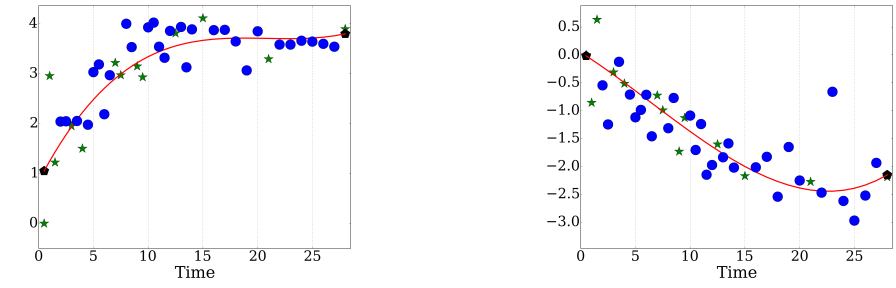
struction performs better than linear reconstruction similar to mRNA dataset. Accurate prediction of different miRNA profiles show the importance of identified points over mRNA dataset. Among these miRNAs, mmu-miR-100 targets Fgfr3 and Igf1r, mmu-miR-136 targets Tgfb2. Similarly, mmu-miR-152 targets Meox2, Robo1, Fbn1, Nfya. Lastly, mmu-miR-219 targets PDGFRA, Eya2, Esr1, Esr2, Efnb2 and Robo1 some of which are associated with BPD in preterm infants [13].

2.4 miRNA Clusters Are Enriched For Several Biological Processes

Detailed analysis of miRNA dataset shows clustered expression profiles of miRNAs as in Figure 5. We identified 8 stable miRNA clusters by k-means algorithm [9] where number of clusters is selected by Bayesian Information Criteria [14]. We find clusters to change more frequently than mRNA data as miRNA is noisier than mRNA data (See Supplementary Figure 8 for mRNA clusters). After mapping each miRNA to corresponding genes by TargetScan [1], we run gene-enrichment analysis by FuncAssociate [5]. We find clusters to be enriched for several Gene Ontology biological processes [2]. For instance, cluster 4 is enriched for single-organism cellular process, positive regulation of biological process, regulation of metabolic process, etc (See Supplementary Results for details).

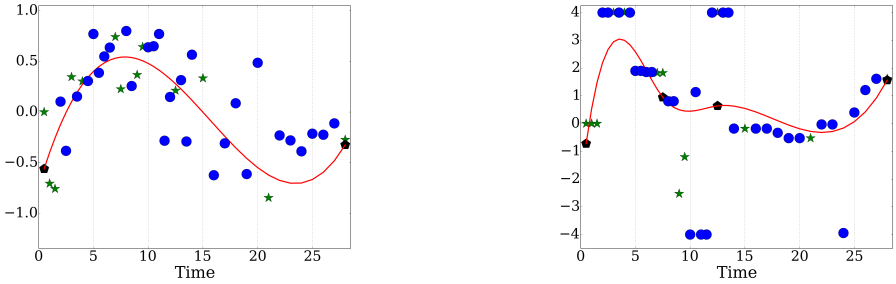
2.5 Comparison of Temporal Methylation Data With Expression Data

TempSelect selected 0.5, 5, 15, 26 out of 8 time points over temporal methylation data when we consider each gene locus independently. We identified high similarity between methylation and gene expression datasets showing the possibility of using the identified subset of time points over gene expression also



(a) mmu-miR-100

(b) mmu-miR-136



(c) mmu-miR-152

(d) mmu-miR-219

Figure 4: Predicted expression profiles of miRNAs a) mmu-miR-100, b) mmu-miR-136, c) mmu-miR-152, d) mmu-miR-219.

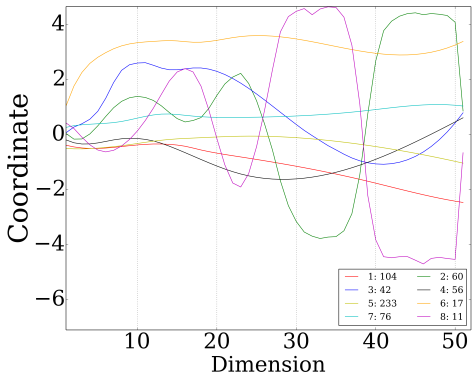
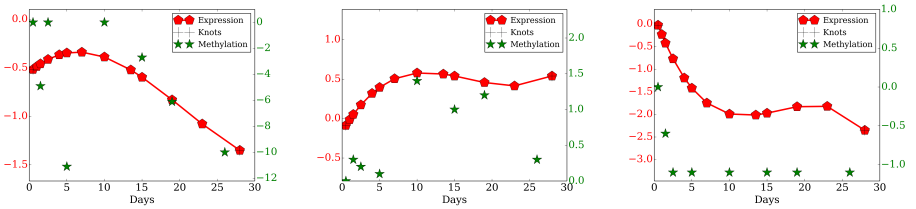


Figure 5: 8 stable clusters



(a) Dnmt3a (b) Src (c) Lox

Figure 6: Comparison of gene expression and methylation data for genes a) Dnmt3a, b) Src, c) Lox.

in methylation experiments. We plot both datasets jointly as in Figure 6 for Dnmt3a, Src, and Lox genes. Among the multiple methylation locus belonging to a gene, we select the most similar one in terms of pearson correlation (See Supplementary Table for detailed correlation analysis).

3 Conclusion

We develop *TempSelect* to efficiently identify subset of important time points over densely sampled gene expression profiles. We show that these points can be used as candidates for high-throughput profiling experiments as well as other temporal experiments such as methylation. Additionally, identified points can serve as a proven benchmark to reduce the experimental cost.

References

[1] Vikram Agarwal, George W Bell, Jin-Wu Nam, and David P Bartel. Predicting effective microrna target sites in mammalian mrnas. *eLife*, 4:e05005, 2015.

[2] 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.

[3] 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.

[4] 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.

[5] Gabriel F Berriz, Oliver D King, Barbara Bryant, Chris Sander, and Frederick P Roth. Characterizing gene sets with funcassociate. *Bioinformatics*, 19(18):2502–2504, 2003.

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

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

[8] Martin Guilliams, Ismé De Kleer, Sandrine Henri, Sijranke Post, Leen Vanhoutte, Sofie De Prijck, Kim Deswarte, Bernard Malissen, Hamida Hammad, and Bart N Lambrecht. Alveolar macrophages develop from fetal monocytes that differentiate into long-lived cells in the first week of life via gm-csf. *The Journal of experimental medicine*, 210(10):1977–1992, 2013.

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

[10] Scott Kirkpatrick, C Daniel Gelatt, Mario P Vecchi, et al. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[11] Madhu S Kumar, Stefan J Erkeland, Ryan E Pester, Cindy Y Chen, Margaret S Ebert, Phillip A Sharp, and Tyler Jacks. Suppression of non-small cell lung tumor development by the let-7 microrna family. *Proceedings of the National Academy of Sciences*, 105(10):3903–3908, 2008.

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

[13] Antonia P Popova, J Kelley Bentley, Tracy X Cui, Michelle N Richardson, Marisa J Linn, Jing Lei, Qiang Chen, Adam M Goldsmith, Gloria S Pryhuber, and Marc B Hershenson. Reduced platelet-derived growth factor receptor expression is a primary feature of human bronchopulmonary dysplasia. *American Journal of Physiology-Lung Cellular and Molecular Physiology*, 307(3):L231–L239, 2014.

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

[15] 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.

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

[17] Andrew E Williams, Sterghios A Moschos, Mark M Perry, Peter J Barnes, and Mark A Lindsay. Maternally imprinted micrornas are differentially expressed during mouse and human lung development. *Developmental Dynamics*, 236(2):572–580, 2007.

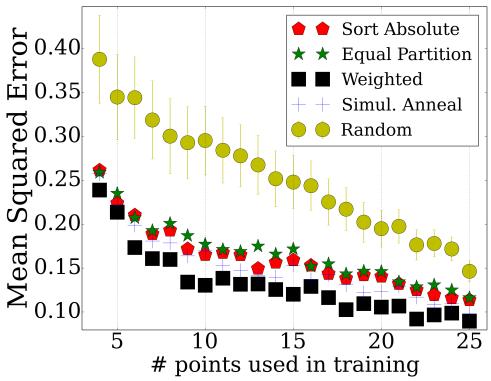
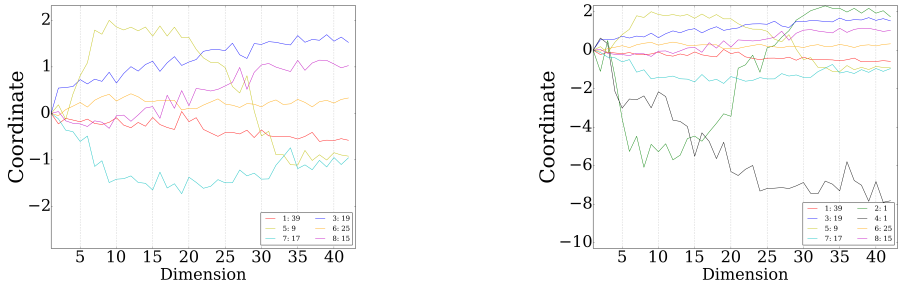


Figure 7: Performance of *TempSelect* by increasing number of selected points



(a) 6 stable clusters

(b) Centroids

Figure 8

A Supplementary Information

Gene	Number of loci		Gene	Number of loci
Cdh11	14		Zfp536	16
Src	11		Igfbp3	34
Sox9	16		Wif1	21
Dnmt3a	41		Vegfa	20
Eln	20		Tnc	4
Foxf2	41		Lox	17
Akt1	11			

Table 1: Summary of methylation dataset