Stony Brook University

A Mutliscale, Geometric Algorithm for Non-parametric Data Exploration with an Application to Genomic Data

A Dissertation Presented

by

Joseph McQuown

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Applied Mathematics and Statistics

(Statistics)

Stony Brook University

December 2007

Stony Brook University

The Graduate School

Joseph McQuown

We, the dissertation committee for the above candidate for the Doctor of Philosophy degree, hereby recommend acceptance of this dissertation.

James Glimm - Dissertation Advisor
Chair and Distinguished Professor, Department of Applied Mathematics and Statistics

Stephen Finch - Chairperson of Defense
Professor, Department of Applied Mathematics and Statistics

Wei Zhu
Professor, Department of Applied Mathematics and Statistics

Bhubaneswar Mishra - Outside Member
Professor, Department of Computer Science and Mathematics, Courant Institute, New York University

This dissertation is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

A Mutliscale, Geometric Algorithm for Non-parametric Data Exploration with an Application to Genomic Data

by

Joseph McQuown

Doctor of Philosophy

in

Applied Mathematics and Statistics

(Statistics)

Stony Brook University

2007

This thesis presents an efficient and adaptive multi-scale algorithm for analyzing measurement data, composed of two categories: a regular set of measurements that can be described by means of a dominant geometry, and a set of "outliers", i.e.,

measurements that deviate from this underlying geometry. The algorithm uses a stopping-time construction in order to identify local regions of different sizes and shapes where the data is concentrated around local lines (or d-planes) and excluding local percentages of putative outliers that reside outside such regions. Thus it is able to construct efficiently a description of the dominant "geometry" in terms of a curve (or d-dimensional graph). Using the local geometric properties, it then detects the outliers. Our approach need not make any assumption about the distributional properties of the noise and it exhibits robustness against noise and outliers. Furthermore, the speed of our algorithm is linear in the size of the data and it can handle high-dimensional data without a blow-up of computational expense. Genomic expression data is an application that can be assayed quite well within this framework. This paper explores experimental results of such phenomena and describes some of the mathematical underpinnings of this algorithm and its various properties.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisor Professor James Glimm for his sage advice and utterly profound wisdom. I would also like to thank Professor Wei Zhu for encouraging me to finish this thesis and reminding me that it was something I could do. I am greatly indebted to her. I also want to thank Professor Stephen Finch for his patience, interest and encouragement. I am humbled by the attention, advice, and financial support given to me by Professor Bud Mishra while working in the New York University Computational Biology laboratory. He is a true inquirer, if ever there was, having an indefatigable appetite for all things new or unknown. Lastly, I am greatly indebted to Professor Gilad Lerman for generously teaching me this fruitful area of research and encouraging me to complete this dissertation. It was a wonderful privilege to work with Professor Lerman while at New York University. I have learned a great deal from his steady, deliberate and exacting approach to research and try daily to duplicate it in life as well. I would like to acknowledge that much of this work was a joint effort with Professor Lerman during his post-doctoralship and my fellowship at the New York University Computational Biology Laboratory.

Joseph McQuown

Chapter 1

Introduction

Complex geometries characterize many interesting phenomena that we encounter in disparate domains: cloud chamber trajectories in physics, optical intensities of spots on a microarray in biology, features of corpora of texts in computational linguistics, etc. Each single measurement in any of these domains either delineates the underlying dominant geometry a little bit more clearly, or points to novel unexplained dynamics because of its deviation from this geometry. The interplay, between the dominant geometry and the deviations from it points to a better understanding of these phenomena. However, measurements are noisy and often distorted in different manner at different scales, and pose many challenges, when one attempts to both elicit the dominant geometry and simultaneously identify the "outliers" from a massive collection of measurements.

As a concrete example, consider microarray analysis in biology, a high-throughput method to measure abundance of multiple species of target DNA by simultaneous hybridization to an array of DNA probes. These applications use comparative methods: In a "two-color" scheme, simultaneous array hybridization detects "normal" target DNA at one fluorescent wavelength and an abnormal or experimental target DNA at

another. Those target DNAs that have truly differential behavior from one experiment to the other are called "enriched" or "expressed" (outliers) and are the objects sought after in these high-throughput experiments. Enriched/expressed targets are found in two steps: first, the measurements are transformed through a "normalization" step in order to assign identical statistical properties to targets unaffected by changes in conditions from one experiment to the other, next the normalized data is further analyzed through well-chosen statistical criteria for "identification" of enriched targets that are truly different in two experiments.

Normalization and identification steps are intertwined. If outliers are used in fitting the model, the model can become corrupted. It is thus necessary to avoid fitting these unusual measurements. While such a desideratum may appear circular, we will see that this circularity can be avoided by local stopping rules, and a strong assumption of the existence of an intrinsic geometric structure–two ideas exploited by our methods.

More specifically, we assume a data set with a substantial part concentrated around a low-dimensional set which is a d-dimensional graph of a function (we refer to it as the "stable" part) and another set of "outliers", which are not engendered by the fundamental model of the data but are influenced by something else. We have devised a justifiably fast algorithm for constructing from this data both the underlying geometry (the d-dimensional graph) as well as identifying outliers with few false positives. In the situations we study, the noise of either the "stable" part or of the "outliers" may depend on the range of the data, in a non-uniform way.

Our algorithm makes no assumptions about the distributions of noise or outliers, except on the "small deviations" of the "stable set" when assigning p-values. Returning to our earlier example of microarray-based biological data analysis, we notice that it is possible to model the bulk of the data as lying along a curve, and the outliers as deviating significantly from the local mean (a point on the curve) by

a significant distance, when suitably scaled by the local standard deviation. Note that when some parts of the data are sparse, a naive algorithm for outlier detection based only on local density, will work poorly. It may classify part of the main curve as outliers, since the low density in that region can introduce a large variability in the estimation of the mean. Similarly, errors in variance estimation can result in the enriched targets to be identified as part of the model, especially if they happen to appear as highly concentrated (e.g. in case of ChIP-on-chip data). The latter case of ChIP-on-chip data is an example where outliers are not necessarily few, but simply follow an unexpected geometry, significantly deviating from the dominant geometry modeled. Our algorithm is well-suited to such cases.

The algorithm also handles high-dimensional data, though it is not discussed in this thesis. In [46], we exemplify such data by analyzing patches (e.g. 3×3 pixel-arrays) obtained from images of unknown and wide-ranges of noise. By viewing the image data in the transformed space of patches, we can visualize the detected outliers as edges in the original image. Unlike many common image-processing techniques, it does not rely on finding the geometry through local gradients, a method highly sensitive to noise.

Technically, the algorithm uses a divide-and-conquer strategy in this space of an arbitrary dimension, by starting with an interval of interest and a region that is cylindrical over the interval. It then partitions the points in that region to putative outliers and regular sets using a constant proportion hyper-strip. As it evaluates this division in terms of the ratio of outliers to regular points, the variance of the regular points, the size of the set of points, it decides if it should divide them further in to sub-regions and recur, or simply stop at that level for that region. The exact choice is determined via several auxiliary numerical parameters.

The theoretical roots of our algorithm are found in the mathematics of harmonic analysis and singular integrals. There, the key ingredient is a multi-scale

stopping-time construction, which has appeared in various forms such as the Calderon-Zygmund cube decomposition [16]. Other more recent extension of these ideas in an intrinsic geometric setting appears in the work of Jones [40], David and Semmes [20; 21], Lerman [48; 45] and Jones and Lerman [**?**] (the latter describes a multi-scale representation of special measures or data sets with geometric features). Notably, [48] makes use of a "localized" metric with regard to points near a manifold in order to effectively bin the data by this metric.

A plethora of algorithms to recover or "learn" the intrinsic geometry of nonlinear data manifolds have been introduced of late. Such works include [65; 74; 24; 42], and [3]. Algorithms such as [65; 24; 3] are based on local preservation of distances between points on the manifold and the same points on a low-dimensional representation. Others such as [74] preserve global, geodesic distances. Whatever the case may be, these algorithms all use kernels (a semi-definite, symmetric matrix) to induce a local neighborhood structure on the data and map the points on the manifold to lower dimensional subspaces. See [35] for a brief review of kernel methods for dimensionality reduction and [69] for an encyclopedic guide, replete with accompanying Matlab® code. More generally, Sapiro and Mémoli [66] describe a way to make geometric comparisons of manifolds given by point clouds directly, without surface reconstruction.

In contrast to these algorithms, our algorithm, and the contribution of this thesis, is explicitly designed to handle noise and also determine when data do not strictly reside on a manifold, and can be labeled as "outliers" with respect to such a manifold. Work in a similar vein by Arias-Castro et al. [14; 15] use dyadic cubes and local estimates to "detect" geometric structures in noisy data. Arias-Castro's ideas of "good continuation" have an analogy to our multiscale curve. Central to the assumptions in [14] and [15], however is that noise is engendered uniformly. Our algorithm makes no such supposition. Indeed, central to the tenant of our method-

ology is that noise may not be uniform. Both methods establish neighborhoods by recursively-constructed regions within the data, as opposed to the ones based on a weighted Euclidean threshold or k-nearest-neighbors, say. Our algorithm controls the neighborhoods with prescriptions on the side lengths of the cubes, and focuses on isolation of outliers without seeking an embedding of any kind. The algorithm is also fast. I.e. it has an optimal time-complexity, which is linear with respect to the size of the data and involves a small constant (determined by the complexity of the stopping rules). However, unlike all of the competing methods, discussed above, it only assumes simple geometric structures in the form of graphs of functions. Most importantly, unlike methods involving a kernel, the computation-time does not increase with the number of observations.

1.1 Background

In the last two decades, multiscale analysis (in particular wavelet analysis) has become fundamental in the areas of signal analysis and image segmentation. The incorporation of geometric (and computational) analysis into the multiscale methodology has resulted in significant improvements of various algorithms (e.g., the introduction of curvelets and bandlets in image analysis). Further methods of both geometric and multiscale nature are expected to be crucial for the analysis of more general data sets. Indeed, many important and non-trivial data sets contain parts which are concentrated around low-dimensional manifolds embedded in higher dimensional spaces such as microarray data.

There are several difficulties in developing techniques and theoretical methods in order to study data sets having intrinsic low-dimensional structures. First, there is a variety of possible low dimensional structure embedded in high dimensions that has to be addressed theoretically and algorithmically. Second, it is hard to handle

noise with geometric techniques, because they often introduce high dimensional local structures which may vary at different locations. It is clear from these obstacles that a general solution cannot be trivial and that it also requires multiscale techniques in order to deal with the various structures observed at different scales.

Peter W. Jones [40] and Gilad Lerman [45] developed a theoretical framework to study the d-dimensional structures of sets (or measures) in $\mathbb{R}^D$ by approximating them at different scales and locations by d-planes. The corresponding techniques are parallel to multiscale methods for analyzing functions, but are adapted to sets or measures supported on lower dimensional subsets (e.g. manifolds) of higher dimensional ambient spaces.

In this report, special attention is given to data procured via gene-chip technologies such as microarrays. Such data has become of vital interest to biologists and applications involving them have become ubiquitous in statistics, applied mathematics and computer science not to mention the new fields of bioinformatics and computational biology. Microarrays and gene-chips provide a way of characterizing transcriptional properties of thousands of genes and studying their interactions simultaneously under many different experimental conditions. To that end, many different methods of knowledge discovery have been applied and adapted to microarray data in order to group genes and arrays with similar expression patterns. We formalize and propose solutions for the main obstacles that must be addressed and overcome when analyzing such data. In this case, not only can the dimension be imposing (one could be analyzing tens or hundreds of experiments) but the noise evoked from such experiments precludes applying standard statistical techniques. In so-called ChIP-on-chip data, the problems become more unwieldy, which will be explained later. The problems in general microarray data involve the existence of both statistical noise, systematic variation, and in ChIp-to-chip data, assymetry. Even in the case of $d = 2$, control vs. experimental conditions, the problem is far from trivial. The noise en-

gendered by various sources, to be discussed later, is hard to deconvolve; and in the absence of detailed knowledge to balance such problems, it is difficult to properly distinguish specific groupings of genes and there corresponding role, based on expression intensity data.

It can now be explained why this problem gives rise to high-dimensional data sets exhibiting low-dimensional intrinsic geometric structures with noise. Assume that the microarray data is of $d = D$ dimension. Then the identification problem can be formulated in a mathematical language as follows: Given a set of points in $\mathbb{R}^D$ concentrated around a curve, find the curve (e.g., Lipschitz graph or chord-arc curve representing the locally-dependent mean) and strip (e.g., Lipschitz graph representing locally-dependent covariance) so that we obtain a footing for distinguishing *deviating* points (differentially expressed genes or *enriched* in the case of ChIP-to-chip data) and the main bulk of points (non-differentially expressed genes or *non-enriched*).

Indeed, low-dimensional data sets are usually more complicated than the curve-like set in the example above. In many cases, such a point cloud may contain different geometric structures, which correspond to different dimensions. In addition, each d-dimensional part of a set can have non-smooth structures that involve corners and intersections and surfaces that have sharp edges. Lerman [48] defines an algorithm constructed to define clusters not by correlation or distance but by the possibly lower dimensional subsets in which the points reside. Lerman bins data by their *Jones number* and makes divisions based on their histogram profiles. When the d-dimensional data is combined with noise, a parsimonious description of such geometrical aspects of the data becomes a formidable problem. Another difficulty studying low-dimensional subsets is that they can be sampled in a non-uniform manner. Some regions of the data may contain a dense set of points (in the non analytical sense) and other regions may be relatively sparse.

Another prominent difficulty in pursuing our goal is known as the "curse of

dimensionality"; a phrase originally coined by Bellman [1]. The problems include, roughly:

1) Norms in $\mathbb{R}^d$ are not equivalent for large d so that the same function, and its approximation will have different levels of smoothness, necessitating different criteria of smoothness, for different norms

2) Approximating a function to within ϵ requires $O(\epsilon^{-d})$ evaluations in d dimensions

3) Integration and therefore numerical density estimation, with equidistant grids, requires $O(\epsilon^{-d})$ evaluations on each grid to maintain a level ϵ of accuracy in d dimensions

Thus, precaution is taken by restricting ourselves to estimating low, D-dimensional sets to mitigate the exponential drift in computational expense. We want our constants to depend exponentially only on the intrinsic dimension of the set and weakly on the its ambient dimension.

We would like to describe a low-dimensional data set by a collection of d-dimensional parts with some noise. Furthermore, we would like to associate each d-dimensional part with a d-manifold having a sufficiently small d-volume (d-dimensional volume). This requires a precise definition and discussion of d-manifolds with finite d-volume. When $d = 1$, we define a rectifiable curve to be a compact and connected set. It is known that these curves can be parameterized by a Lipschitz function defined on a closed interval (see e.g. [21]).

When $d \geq 2$ there is a problem defining the right class of d-dimensional manifolds with finite d-dimensional measure.The difficulty is illustrated by the following fact: any finite set in $\mathbb{R}^D$, where $D \geq 3$, can be embedded in a d-manifold ($d \geq 2$)

with arbitrarily small d-volume. Such a d-manifold can be taken to be a thin tube around a curve of finite length which contains the given set. This manifold, however, does not correctly reflect the underlying data set. The correct approach is to demand that the approximating manifolds satisfy a "flatness" condition at all scales. Such a condition might limit the class of investigated sets. Because of this difficulty, our attention is focused on the case $d = 1$ and generalize later.

The existing techniques for approximating general low-dimensional data sets with additive noise by low-dimensional surfaces are not fully satisfactory. This approach is known as manifold learning (see e.g. [5]). In this setting, one presumes that the data set lies on a smooth low-dimensional submanifold of $\mathbb{R}^D$. A simple version of this problem, which has been studied extensively, is curve or surface reconstruction in $\mathbb{R}^2$ or $\mathbb{R}^3$ respectively. There are two main disadvantages of the current manifold learning algorithms, however. First, they mainly work for data sets which lie on a smooth d-manifold and cannot tolerate mixed dimensions and certain singularities like sharp edges. Second, the current algorithms usually require very special sampling laws. In particular, they are extremely sensitive to noise.

1.2 Rectifiability, Lerman & Jones work

The motivation for approximating general low-dimensional data sets by low-dimensional surfaces is based on work developed by Peter Jones [40] in order to solve the so-called Jones's version of the Traveling Salesman Problem (JTSP). The JTSP is composed of two parts. The first, whether a given set is contained in a rectifiable curve; and the second applies to sets, which are known to be contained in a rectifiable curve. It entails the construction of a curve that contains the set, whose length is comparable to the shortest one among all such curves. In contrast, the original Traveling Salesman Problem looks for the minimal length closed curve which contains all points of

a given finite set (see e.g. [43]). In order to solve the JTSP, Jones [40] introduced the β_∞ numbers, which record normalized L_∞ approximation errors of a set by lines at different scales and different locations. The collection of Jones's β_∞ numbers for all dyadic cubes in $\mathbb{R}^n$ is sufficient to determine the solution of both parts of the JTSP. In this context, Jones's numbers are used in a synthesis problem. Moreover, Jones's numbers can be used in order to characterize smoothness properties of a rectifiable curve of almost minimal length containing the set in question. One useful tool for such an analysis problem is the Jones function, which is formed by adding the squares of Jones's β_∞ numbers at different scales and the same location (see e.g. [7]). Jones's function is used to bound from above the length of the shortest curve containing the investigated set, or any subset of it and can be used in certain cases to characterize the existence of a tangent at a point of the set (see e.g. [7]).

We first review the notions of rectifiability and quantitative rectifiability and then explain our work in terms of these notions. Rectifiability expresses in a general (weak) sense "manifold-like" structures of measures. A measure μ on $\mathbb{R}^n$ is called d-rectifiable if there exists a countable union of d-dimensional Lipschitz graphs $\{G_i\}_{i\geq 1}$ on $\mathbb{R}^n$ such that $\mu(\mathbb{R}^n \setminus \overline{\cup_{i\geq 1} G_i}) = 0$. The weakest definitions which is sometimes referred to as d-subrectifiability is used here. Note that this definition does not take into account any quantitative estimates like the Lipschitz constants of the graphs. The latter estimates are the scope of quantitative rectifiability.

Lerman [45] extended the theory to a wide class of d-rectifiable measures using an L^2 variant of Jones' β numbers (see [40],[20]). These numbers measure the scaled "deviations" of the given measure from approximating d-planes at different scales and locations. The dimension d is usually fixed in advance. Scales and locations are obtained by restricting the measure into cubes in an extended dyadic grid. More precisely, a large cube Q_0 which corresponds to the largest scale of interest is fixed (for the simplicity of the discussion we may assume that Q_0 contains the support of

μ). Then, a dyadic grid $\mathcal{D} \equiv \mathcal{D}_{Q_0}$ which include all dyadic subcubes of Q_0 is formed. In order to avoid some edge effects, we use an extended dyadic grid $\widetilde{\mathcal{D}} \equiv \widetilde{\mathcal{D}}_{Q_0}$ which includes several shifts of the grid $\mathcal{D}$. If Q is a cube in $\widetilde{\mathcal{D}}$, then the L^2 Jones number of Q, $\beta_2(Q)$ is defined by the formula:

$$\beta_2(Q) \equiv \beta_2^{\mu,d}(Q) = \begin{cases} \min\limits_{d-\text{planes } P_d} \left(\dfrac{1}{\mu(Q)} \int\limits_Q \left(\dfrac{\text{dist}(z, P_d)}{l(Q)} \right)^2 \mathrm{d}\mu(z) \right)^{\frac{1}{2}}, & \text{if } \mu(Q) > 0\,, \\ 0, & \text{otherwise}\,. \end{cases} \tag{1.1}$$

The d-dimensional Jones function J^d of the given measure is formed by adding the squares of the d-dimensional L^2 Jones numbers at different scales and the same location. That is,

$$J^d(x) = \sum_{Q \in \widetilde{\mathcal{D}}} \beta_2^2(Q) \cdot \chi_Q(x)$$

where χ_Q is the indicator function of a point $x \in Q$. J^d plays the same role in geometric measure theory as the square function in harmonic analysis (see e.g. [7] where this idea first appeared with an L^∞ version of the Jones function). In this thesis, 1.1 is adapted to a discrete version and the minimization is over the best lines L_Q instead of d-planes. Since we are interested in finitely many points, we are not compelled to use the measure μ. The point is the same, however. We are summing normed deviations scaled by $l(Q)$. Also of interest is what will be defined f_Q, the ratio of points inside and outside a certain region (whose projection is Q) and its sum across the dyadic grid $\mathcal{D}(Q_0)$.

The theory of quantitative rectifiability examines a data set at all locations and scales, as one does in multiscale analysis. The difference resides in application; multiscale analysis is usually applied to functions, whereas Jones's numbers and the theory he developed in general is used to analyze sets. The same way wavelet coefficients are used to characterize smoothness properties of functions, Jones'sβ_∞ numbers are used to characterize smoothness properties of sets. The problem of reconstructing a

function from its wavelet coefficients is replaced by the somewhat similar problem of constructing the curve which contains the underlying set by using Jones's numbers. Moreover, the algorithm in the paper is not unlike those used for multiscale image processing (see Donoho, Candes, Arias-Castro, Huo, etc).

Problems exist in developing a numerical algorithm based on the existing theory of quantitative rectifiability. Namely, the construction of the curve or surface which appears in the theory attempts to fit all points and this becomes impractical for large sets of data. It is also undesirable for noisy data; one does not want to fit noise. For similar reasons, it is clear that approximation by lines or d-planes that minimize the L_∞ distance is undesirable. This obstacle is overcome by fitting a short curve only to a certain portion of the data set and using L_2 variants of Jones' β_∞ numbers and square functions. Following Jones, a square function is formed by adding the squares of β numbers from different scales and the same location. This function and the β numbers can be computed for any dimension $d << n$.

After describing a small amount of the theoretical foundation and formally developing the algorithm for a $d = 1$ dimensional curve, the multiscale algorithm is applied to ChIP (Chromatin Immunoprecipitation)-on-chip data in collaboration with biologists from the NYU Cancer Institute. In another paper [46], we generalize the algorithm for high-dimensional pixel data so that we may exemplify the virtues of the algorithm in high dimension by isolating structures poorly approximated by a line or curve (and thus, outliers) and demonstrating that such outliers are actually the edges in the pictures.

1.3 Collaboration and prior publications

The work for this thesis was developed in conjunction with publication of [47] and [46]

Chapter 2

The Basic Algorithm

2.1 Basic Notation and Definitions

Throughout the paper we assume a fixed data set E of N points in $\mathbb{R}^D$. We fix an intrinsic dimension d in order to approximate the set (or a substantial part of it) by a d-dimensional graph. In the rest of the paper we assume that $d = 1$; that is, a substantial part of the data is well approximated by a one-dimensional graph. We fix a line L that approximates the set E globally (see Subsection 2.3.2 for details). We also assume that the data has been transformed orthogonally so that L coincides with the first coordinate axis (the orthogonal transformation is described in Subsection 2.3.2). We denote by Q_0^* the closed interval of minimal length containing the projection of E onto L.

If L' is a line in $\mathbb{R}^D$, that is not orthogonal to the x-axis (the line L), then we denote by $\underline{L'}(x)$ the function from $\mathbb{R}$ to $\mathbb{R}^D$ such that $L' = \{(x, \underline{L'}(x)) : \ x \in \mathbb{R}\}$. L' is the regression line (or $L'(x) = (L_1(x), ..., L_{D-1}(x))$ a linear regression function) for a region K $\subset \mathbb{R}^D$ or a data set $E' \subset E$, if the linear real-valued functions $L_i(x), \ i = 1, ..., D-1$, minimize the least square error.

If K is a subset of $\mathbb{R}^D$, we denote by $|K| \equiv |K \cap E|$ the number of points of E in K. If Q is an interval, we denote its length by $\ell(Q)$ and by χ_Q the indicator function of Q ($\chi_Q(x) = 1$ if $x \in Q$ and $\chi_Q(x) = 0$ otherwise).

We consistently use the notation $\underline{u}$, $(x_1, \underline{y})$ or $(x, \underline{y})$ for a point in $\mathbb{R}^D$; That is, $\underline{y} = (x_2, \ldots, x_D)$ and $\underline{u} = (x_1, \underline{y})$. We denote by $\|\underline{u}\|_2$ the Euclidean norm of $\underline{u}$. We denote by $\text{dist}(\underline{u}_1, \underline{u}_2)$ the Euclidean distance between $\underline{u}_1$ and $\underline{u}_2$ (equivalently $\|\underline{u}_1 - \underline{u}_2\|_2$).

The properties of the function $\underline{f}$ defining the underlying one-dimensional graph depend on the data set. The fastest version of the algorithm produces an approximation which is a Lipschitz function. However, we also explain how to adapt the algorithm for underlying functions with higher degrees of smoothness (more derivatives). Formally we define a Lipschitz function as follows: The Lipschitz norm of a function $\underline{f}$ from $\mathbb{R}$ to $\mathbb{R}^D$ is a global bound on the "approximate derivative" of a function and is defined as the smallest number L^* such that

$$\max_{x_1, x_2 \in \mathbb{R}} \|\underline{f}(x_2) - \underline{f}(x_1)\|_2 \leq L^* |x_2 - x_1|.$$

L^* is often denoted $\|\underline{f}\|_{\text{Lip}}$. A function $\underline{f}$ from $\mathbb{R}$ to $\mathbb{R}^D$ is Lipschitz if and only if $\|\underline{f}\|_{\text{Lip}} < \infty$. The Lipschitz graph associated with $\underline{f}$ has the form: $\Gamma = \{(x, \underline{y}) |\ x \in [a, b] \text{ and } \underline{y} = \underline{f}(x)\}$. From a practitioner point of view, a Lipschitz function has a uniformly bounded derivative, except for "few" points where the function is continuous but has no derivative.

2.2 Structure of the input data

We analyze here data sets represented as a cloud of points distributed around a curve with additional sets of "outliers" separated from the main curve. Natural and important instances of this model have been described in the introduction.

More precisely, we presuppose that each point of the data set E was sampled from a distribution F which is a mixture of two other distributions F_{in} and F_{out}. The mixture parameters of F may depend on x. That is,

$$F(x, \underline{y}) = P_{\text{in}}(x) \cdot F_{\text{in}}(x, \underline{y}) + P_{\text{out}}(x) \cdot F_{\text{out}}(x, \underline{y}).$$

We denote $\varepsilon_1 := \max P_{\text{out}}(x)$ and assume that $\varepsilon_1 \ll 1$. Practically, we assume that the projection of F on the real line is supported or concentrated around a finite interval.

We denote the regression function (of $\underline{y}$ on x) associated with F_{in} by $\underline{C}$, that is, $\underline{C}(x_0) = \mathrm{E}(F_{\text{in}}(x, y)|\ x = x_0)$. We assume further that $\underline{C}$ is a Lipschitz function. We may also allow higher smoothness of $\underline{C}$ We denote the conditional standard deviation of the $\underline{y}$ variable given x by S, that is, $S(x) = \sqrt{\mathrm{Var}(F_{\text{in}}(x', y)|\ x' = x)}$, and assume that $S(x) < \infty$ for all $x \in \mathbb{R}$.

The distribution F_{out} represent "outliers"; points deviating from the "stable" model (with underlying distribution F_{in}). The notion of "outliers" may depend on the application we consider. Indeed, for ChIP-on-chip experimental, "outliers" may include significant subpopulations of points (not necessarily negligible in quantity), that deviate from the main curve of normalization. F_{out} could be a complicated mixture of some subpopulations whose local means are separated from the local mean of F_{in}, where the size of separation depends on the local variances of both F_{in} and F_{out}. One important requirement of both the underlying distribution of "outliers" and our suggested algorithm is robustness. That is, small changes in ε_1, result in small changes in the identification of points sampled from F_{out} by the algorithm. We cannot verify it for a given model, nonetheless we discuss some numerical estimates of a similar criterion for special choices of F_{in} and F_{out} in Subsection 3.1.

The assumptions above restrict the distribution of any single sampled point. The algorithm takes into account joint properties of the distribution of points only

when estimating local statistical properties of the variables in local regions. That is, the algorithm fixes a cylindrical local region $\widetilde{Q}$ in $\mathbb{R}^D$ and estimates local statistical properties of $\underline{y}_i|x_i$, where $(x_i, \underline{y}_i) \in \widetilde{Q}$. The assumption is that those statistical properties represent the properties of the underlying distribution. Clearly, independence of the variables $\underline{y}|x$ is sufficient to guarantee such a property, however weaker assumptions allowing the weak law of large numbers in a more general setting, will also hold.

2.3 Idea of the Algorithm

The algorithm builds on the following intuition: it is possible to approximate the data set by lines and planes with cylindrical regions or sheaths around them at various locations and scales. These lines are combined together in order to estimate the underlying Lipschitz graph (the graph of the function $\underline{C}$). The cylindrical regions exclude "initial outliers". Inside them the algorithm estimates the local standard deviation S and then use this estimate together with the one for $\underline{C}$ in order to reassess "outliers" of the whole data set. A sketch of the algorithm appears in **Alogrithm 1**.

Step 1 of the algorithm as well as Step 4 are explained in Subsection 2.3.2. The first step varies in different cases which we list in Subsection 2.3.1. Steps 2 and 3 are explained in Subsection 2.3.3. The reassessment and ranking of outliers is described in Subsection 2.4.2. The steps Output 1 and 2 are described in Subsection 2.4, whereas Subsection 2.4.1 describes a smoothed version of these output functions. The steps Output 3 and 4 are described in Subsection 2.4.2. Finally in Subsection 2.5 we describe how to choose the main parameter of the algorithm: α_0.

Algorithm 2.1: MSC scheme

Input: Data E and parameters
Output: Curve $\widetilde{\underline{C}}$, Standard deviation function $\widetilde{S}$, Ranking R, A set of outliers
Initialization:
Approximate E by a line L and apply a rigid transformation to E, so that $L = x$-axis
Set Q_0 := interval containing the projection of E on L
Set: $l = 0$, $Good\_int = \{Q_0\}$, $Stop\_int = \emptyset$, $N_{\text{stop}} = 0$.
while $l \leq l_0$, $N_{\text{stop}} < N$ **do**
Step 1 Form cylindrical regions around all intervals in $Good\_int$
Step 2 Compute basic quantities in each cylindrical region (ang(L_Q, L), F_Q, σ_Q, β_Q)
Step 3 Identify new stopping intervals in $Good\_int$ satisfying the "stopping time criteria"
$Stop\_int$:= $Stop\_int$ $\cup$ new stopping intervals
N_{stop} := number of points in all intervals in $Stop\_int$
$Good\_int = Good\_int \setminus Stop\_int$
Step 4 $Good\_int$:= set of dyadic subintervals of intervals in $Good\_int$
l:= l+1
end while
Output 1 Record local standard deviations for stopping time cylindrical regions (computed before) to get $\widetilde{S}$
Output 2 Record local line approximations for stopping time cylindrical regions (computed before) to get $\widetilde{\underline{C}}$
Output 3 Obtain ranking R by using $\widetilde{S}$ and $\widetilde{\underline{C}}$
Output 4 Identify outliers according to R

2.3.1 Different cases

We perform the algorithm according to several cases based on certain knowledge of the data

1. The L-case (linear case).

   Here we assume that the function $\underline{C}$ is linear. In practice, we classify this case when the data is concentrated around a line.

2. The G-case (general case).

   Here $\underline{C}$ can be any Lipschitz graph. In practice, the data seems to concentrate around a curve (Lipschitz graph) with some local nonzero curvature (that is, it is not a line).

3. The S-case (symmetric case).

The distribution $F_{\text{in}}(x, \underline{y} |\ x = x_0)$ is approximately radial around the point $(x_0, \underline{C}(x_0))$. That is, $F_{\text{in}}(x_0, \underline{y})$ is well approximated by a distribution depending only on $\|\underline{y} - \underline{C}(x_0)\|_2$ (the local skewness is close to zero). In practice, we classify this case when a big portion of the points (e.g. at least 85%) seem to be symmetric around the main normalizing curve.

4. The A-case (asymmetric case).
   In this case the assumptions of the S-case are not valid. That is, asymmetric distribution of points around the curve is noted. Currently we have only considered this case when $D = 2$.

What's more, we may have the G or L-cases together with the A or S-cases giving us LS ,LA, GS and GA.

2.3.2 Definitions of multiscale grids, regions and lines

The algorithm starts by finding an approximating line L for the whole data set along which dyadic intervals are then formed. It first finds the top principal axis of the data set (right singular vector with a corresponding maximal singular value); it then shifts and rotate the data so that the x_1-axis coincides with that principal axis; it then linearly regresses each one of the coordinates $x_2, \ldots, x_D$ of points in E onto the coordinate x_1 and obtain a linear function $\underline{L}$ from $\mathbb{R}$ to $\mathbb{R}^{D-1}$, $\underline{L}$ ($\underline{L}(x_1) = (L_2(x_1), \ldots, L_D(x_1))$, representing the line L; lastly, it transforms the data in the following way so that L coincides with the x_1-axis: $E = f(E)$, where $f(x_1, \underline{y}) = (x_1, \underline{y} - \underline{L}(x_1))$.

A natural grid of dyadic intervals can be defined along L by fixing an interval $Q_0 := [a_0, b_0)$ of almost minimal length containing the projection of E onto L. In Subsection 2.4.1 we use several intervals of this form and partition Q_0 repeatedly

(until level l_0) into smaller intervals. For any given interval Q, we define its dyadic subintervals, denoted by Q_L and Q_R (left and right), as the two half-closed, half-open intervals of equal lengths whose union is Q. We partition repeatedly Q_0 to dyadic subintervals, but not more than l_0 times. We denote the set of all such intervals by $\mathcal{D}(Q_0)$. It contains half-closed half-open subintervals of Q_0, where for each $j = 0, \ldots, l_0$, there are exactly 2^j intervals of side length $2^{-j} \cdot l(Q_0)$. If $Q \in \mathcal{D}(Q_0) \setminus \{Q_0\}$, then denote by P_Q the dyadic parent of Q according to the grid $\mathcal{D}(Q_0)$ and similarly define $P_{Q_0} := Q_0$.

For each $Q \in \mathcal{D}(Q_0)$ we associate an infinite strip, $\widehat{Q}$, a cylindrical region, $\widetilde{Q}$, and another region for testing outliers inside $\widetilde{Q}$ denoted by $T(\widetilde{Q})$. We also form an approximating line L_Q for the points in $\widetilde{Q}$. The infinite strip $\widehat{Q}$ in $\mathbb{R}^D$ has the form:

$$\widehat{Q} = Q \times \mathbb{R}^{D-1}.$$

The definitions of the other regions and lines depend on the different cases as follows:

1) *LS*-case:

The corresponding regions and line for an interval $Q \in \mathcal{D}(Q_0)$ are exemplified in Figure 2.1(a) and defined as follows:

$$\widetilde{Q} = \begin{cases} \{(\underline{y}, x) \in \widehat{Q} : \|\underline{y} - \underline{L}_Q(x)\|_2 \leq c_0 \cdot \ell(Q)\}, & \text{if } \ Q \subseteq Q_0; \\ \widehat{Q}_0, & \text{if } \ Q = Q_0 \ , \end{cases} \tag{2.1}$$

$$T(\widetilde{Q}) = \{(\underline{y}, x) \in \widetilde{Q} : \|\underline{y} - \underline{L}_Q(x)\|_2 > \frac{c_0 \cdot \ell(Q)}{2}\},$$

$$L_Q \equiv L.$$

2) *GS*-case:

The corresponding regions and lines are defined recursively from top to bottom

levels. They are exemplified in Figure 2.7.4. If $Q = Q_0$, then we define $\widetilde{Q} = \widehat{Q}_0$, $L_Q = L$ and $L_{Q_L} \equiv L_{Q_R} := L$, where Q_L and Q_R are the left and right dyadic subintervals of Q. If $Q \subsetneq Q_0$, then L_Q has been defined in the previous level, and

$$\widetilde{Q} = \left\{(x, \underline{y}) \in \widehat{Q} : \|\underline{y} - \underline{L}_Q(x)\|_2 \leq c_0 \cdot \ell(Q)\right\}. \tag{2.2}$$

The lines L_{Q_L} and L_{Q_R} are set as the regression lines for $\widehat{Q}_L \cap \widetilde{Q}$ (left part of $\widetilde{Q}$) and $\widehat{Q}_R \cap \widetilde{Q}$ (right part of $\widetilde{Q}$) respectively. Finally, define

$$T(\widetilde{Q}) = \left\{(x, \underline{y}) \in \widetilde{Q} \cap \widehat{Q}_L : \ \|\underline{y} - \underline{L}_{Q_L}(x)\|_2 > \frac{c_0 \cdot \ell(Q)}{2}\right\}$$
$$\bigcup \left\{(x, \underline{y}) \in \widetilde{Q} \cap \widehat{Q}_R : \|\underline{y} - \underline{L}_{Q_R}(x)\|_2 > \frac{c_0 \cdot \ell(Q)}{2}\right\}.$$

3) *A*-case:

The A-case is motivated mainly by our experience analyzing ChIP-on-chip data furnished by biologists at the NYU Cancer Institute and NYU Medical Center and published in [8]. In this case the data is two-dimensional and there will only be one-sided significant "outliers". We thus describe an algorithm for the general asymmetric case for two-dimensional data (that is $D = 2$). The linear asymmetric case is a special case.

The idea is to form asymmetric local regions. We thus choose two parametric constants c_+ and c_- (instead of c_0) to define such regions (see Figure 2.1(b)). It is possible to have either $c_+ = \infty$ or $c_- = -\infty$ (but not both). In the case of ChIP-on-chip data, we choose $c_+ = \infty$ (i.e. we don't exclude any points from the estimation that are above L_Q).

If $Q = Q_0$, then we define $\widetilde{Q} = \widehat{Q}_0$, $L_Q = L$ and $L_{Q_L} \equiv L_{Q_L} := L$, where Q_L and Q_R are the left and right dyadic subintervals of Q. If $Q \subsetneq Q_0$, then L_Q has

been defined in the previous level, and

$$\widetilde{Q} = \left\{(x,y) \in \widehat{Q} : |y - \underline{L}_Q(x)| \leq c_0 \cdot \ell(Q)\right\}.$$

The lines L_{Q_L} and L_{Q_R} are the regression lines for $\widehat{Q}_L \cap \widetilde{Q}$ (left part of $\widetilde{Q}$) and $\widehat{Q}_R \cap \widetilde{Q}$ (right part of $\widetilde{Q}$) respectively. Finally, define

$$\begin{aligned} T(\widetilde{Q}) =& \widetilde{Q} \setminus \left(\left\{(x,y) \in \widetilde{Q} \cap \widehat{Q}_L : \frac{c_-}{2} \cdot \ell(Q) \leq y - \underline{L}_{Q_L}(x) \leq \frac{c_+}{2} \cdot \ell(Q)\right\} \right. \\ & \left. \bigcup \left\{(x,y) \in \widetilde{Q} \cap \widehat{Q}_R : \frac{c_-}{2} \cdot \ell(Q) \leq y - \underline{L}_{Q_R}(x) \leq \frac{c_+}{2} \cdot \ell(Q)\right\}\right). \end{aligned}$$

The above regions and lines are exemplified in Figure 2.1(b) and Figure 2.1(d).

In most of the paper we assume the GS-case and examine the A-case for ChIP-on-chip data appear in Chapter 4.

2.3.3 Local computation and the stopping time criteria

The algorithm computes at each visited interval Q (with corresponding regions $\widehat{Q}$, $\widetilde{Q}$, $T(\widetilde{Q})$) the following quantities: L_Q, f_Q, F_Q, σ_Q and β_Q. We have explained above how it finds the local regression lines L_Q. The fraction f_Q is the ratio of "putative local outliers" to the total number of points projected on Q. That is,

$$f_Q = \frac{|T(\widetilde{Q})|}{|\widehat{Q}|}.$$

The quantity F_Q adds up such fractions of all parenting intervals (including current interval), that is,

$$F_Q = \sum_{\substack{Q' \in \mathcal{D}(Q_0) \\ Q' \supseteq Q}} f_{Q'}.$$

It can be thought of as assesing the "fraction" of outliers in a conic region above the interval 2.2, which is the union of regions of the form $T(\widetilde{Q})$ for all parenting intervals and current interval. The algorithm computes F_Q with a top-bottom procedure: First,

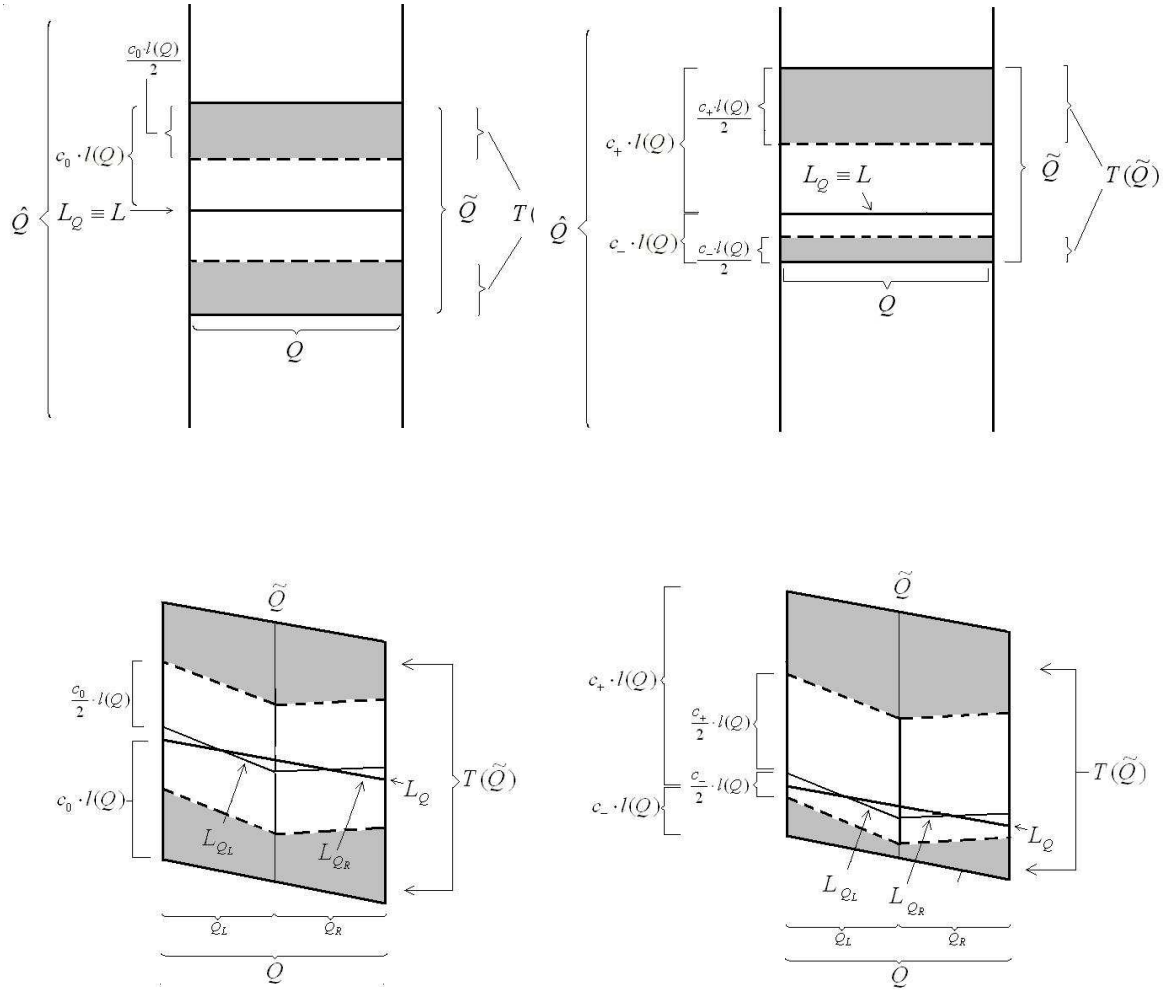
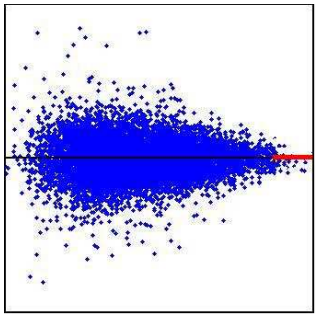
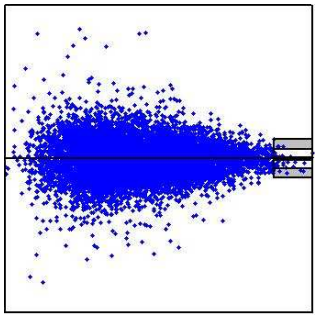


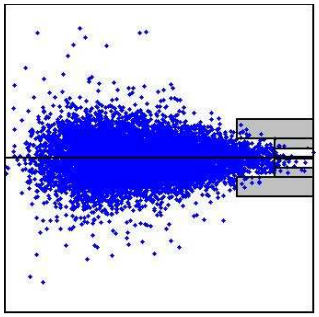
Figure 2.1: Upper-left figure represents the LS-case, upper-right figure represents the LA-case, bottom left figure is the Symmetric, GS-case and bottom right figure is the Asymmetric, GS-case



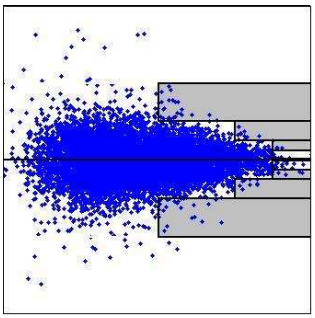
(a) lowest level (4th level) stopping time interval (red)



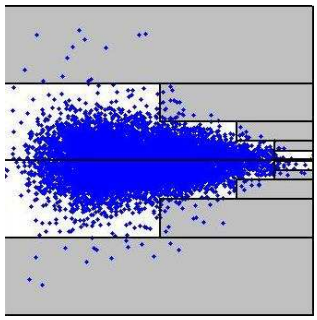
(b) the region $T(\widetilde{Q})$ and $\widetilde{Q}$ for the stopping time interval



(c) 3rd level parent of the stopping interval

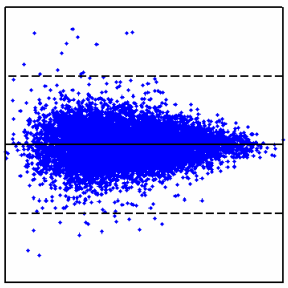


(d) 2nd level parent of the stopping time interval

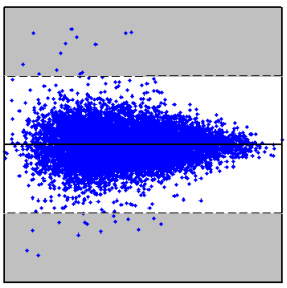


(e) 1st level parent of stopping time interval

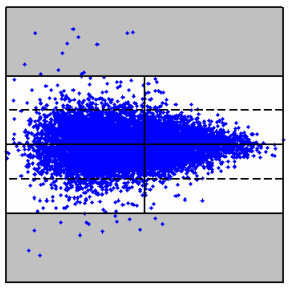
Figure 2.2: Conical regions of a stopping time interval $\mathcal{Q} \in \mathcal{D}(Q_0)$



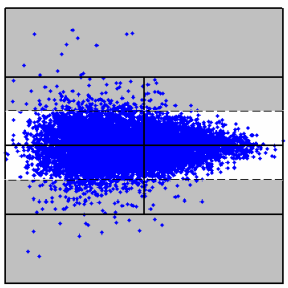
(a) Data around the line (interval Q_0) and the regions corresponding to Q_0: $\widetilde{Q}_0$, the whole rectangle, and $T(\widetilde{Q}_0)$, union of two vertical rectangles outside of dashed lines



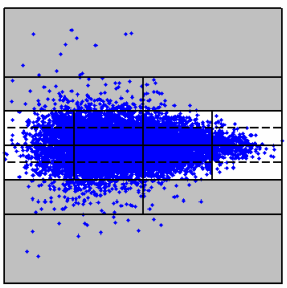
(b) Points in gray area ($T(\widetilde{Q}_0)$) are detected as "outliers" and excluded



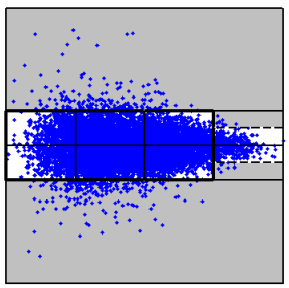
(c) 2nd level of algorithm, Q_0 is partitioned into two intervals Q_1 and Q_2, their rectangles are formed together with the regions $T(\widetilde{Q}_1)$ and $T(\widetilde{Q}_2)$ outside dashed lines



(d) Again, points in $T(\widetilde{Q}_1)$ and $T(\widetilde{Q}_2)$ (new gray area) are detected as "outliers" and excluded

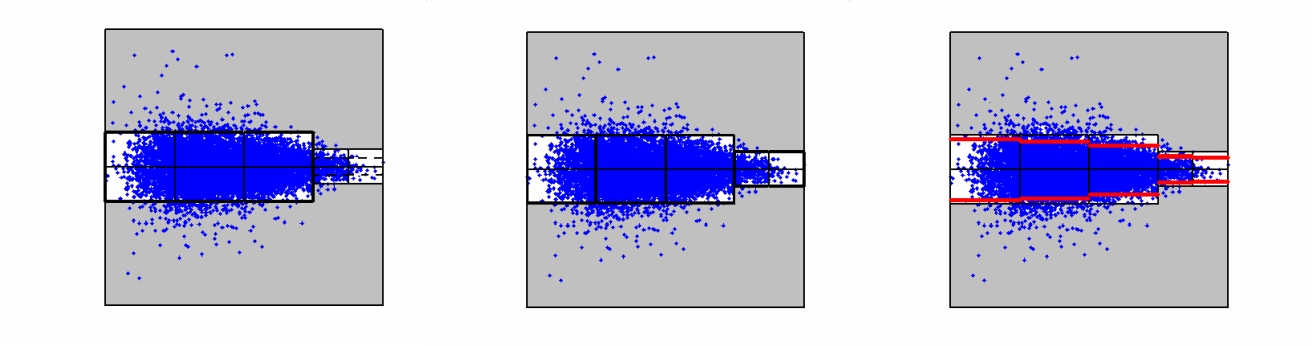


(e) 3rd level of algorithm



(f) Stop at rectangles "without significant outliers" (3 rectangles denoted by bold lines) while exclude "outliers" detected at rightmost rectangle

Figure 2.3: Pictorial representation of how the algorithm works when applied to an artificial and simple set (LS-case). The plots continue are are concluded in Figure 2.4



(a) 4th level of the algorithm, only the two rightmost rectangles may be partitioned further

(b) Picture of final so-called "stopping-time" rectangles denoted by bold lines (regions $\widetilde{Q}$ associated with "stopping-time" intervals Q)

(c) Red lines indicate the strip $\widetilde{S}$, which estimates local standard deviation in each "stopping time" rectangle

Figure 2.4: Continuation of Figure 2.3. The so-called "strip" is indicated by red lines in the last plot

it initializes $F_Q \equiv 0$ for all $Q \in \mathcal{D}(Q_0)$. Then, it applies the reduction formula (from coarse levels to fine levels):

$$F_Q = F_{P_Q} + f_Q.$$

The quantity σ_Q estimates the local conditional $(y|x)$ "standard deviation" of the residuals in $\widetilde{Q}$ (assuming the local mean of $y|x$ is zero, else use $y - \widehat{y}(x)$) and β_Q is its ratio to the length of Q, that is

$$\sigma_Q = \left(\frac{1}{|\widetilde{Q}|} \sum_{(x,y) \in \widetilde{Q} \cap E} |y|^2 \right)^{\frac{1}{2}} \quad \text{and} \quad \beta_Q = \frac{\sigma_Q}{\ell(Q)}.$$

While proceeding from top to bottom levels, the algorithm stops at an interval $Q' \in \mathcal{D}(Q_0)$ (together with all of its descendants in $\mathcal{D}(Q_0)$) if one of the following conditions is satisfied:

1. $F_{Q'} > \alpha_0$. (2.3)

2. $|\widetilde{Q}'| < n_0$. (2.4)

3. $\beta_{\widetilde{Q}'} > \delta_0$ (optional). (2.5)

4. angle and lift stopping rule with parameter θ_0 (optional). (2.6)

The first stopping time condition is the crucial one for controlling the total number of outliers (see Subsection 2.7.1). The second one is necessary in order to have valid local estimates in each interval. The third one allows us to control the rate of change of the output function $\widetilde{S}$, it may be ignored by setting $\delta_0 = c_0$. The last stopping time condition allows us to control the Lipschitz constant of the averaged version of $\underline{\widetilde{C}}$. It can be ignored by setting $\theta_0 = \frac{\pi}{2}$.

We next explain the last stopping time condition which allows us to control the Lipschitz constant of the averaged version of $\underline{\widetilde{C}}$ This criterion is empty at level 0. Assume that the stopping time conditions have been applied until level $j, j < l_0$ and that the union of the intervals where the algorithm stopped at (stopping-time intervals) is not Q_0. We fix an interval $Q = [a_Q, b_Q)$ in $D_j(Q_0)$ which is not contained in a stopping time interval. We first define Q^+ and Q^- and then use them to define precisely the angle and lift stopping-rule. If the interval $[b_Q, b_Q + \ell(Q))$ is not contained in a stopping-time interval, then we define $Q^+ = [b_Q, b_Q + \ell(Q))$. Otherwise Q^+ is the stopping-time interval which contains $[b_Q, b_Q+\ell(Q))$. Similarly, if the interval $[a_Q-\ell(Q), a_Q)$ is not contained in a stopping-time interval, then we define $Q^- = [a_Q - \ell(Q), a_Q)$. Otherwise Q^- is the stopping-time interval which contains $[a_Q - \ell(Q), a_Q)$. Then, Q^- satisfies the angle and lift stopping-time rule with the parameter θ_0 if it satisfies at least one of the following equations:

$$\tan(angle(L_Q, L)) = \theta_0 \tag{2.7}$$

$$\|\underline{L}_Q(b_Q) - \underline{L}_{Q^+}(a_{Q^+})\|_\infty > \theta_0 \cdot \ell(Q) \tag{2.8}$$

$$\|\underline{L}_Q(a_Q) - \underline{L}_{Q^-}(b_{Q^-})\|_\infty > \theta_0 \cdot \ell(Q) \tag{2.9}$$

2.4 The output functions

We next describe several functions which are the output of the algorithm. We first define them as piecewise constant (or piecewise linear) functions on Q_0. In practice, they are computed by the algorithm only for points in E (therefore those functions are restricted to the projection of E onto L). In Chapter 2.4.1 we explain how to smooth those output functions.

The first function, $\widetilde{\underline{C}}$, estimates the underlying curve. The next function, $\widetilde{S}$, estimates standard deviations in local regions around $\widetilde{\underline{C}}$. The function $\widehat{S}$ corrects $\widetilde{S}$ so that it has a valid meaning for points outside the local regions (in practice, both functions are comparable in scale, with a scale-constant close to 1).

In the following definitions we use the following sets:

We denote

$$\mathcal{Q} = \{Q \in \mathcal{D}(Q_0) : \ Q \text{ is a stopping time interval}\},$$
$$\mathcal{B} = \{Q \in \mathcal{Q} : \ |\widetilde{Q}| < n_0, \ \beta_Q > \delta_0 \ \text{ and } \ \text{ang}(L_Q, L) > \theta_0\}$$

The function $\widetilde{\underline{C}}$ is defined by the formula:

$$\widetilde{\underline{C}}(x) = \sum_{Q \in \mathcal{Q} \setminus \mathcal{B}} L_Q(x) \cdot \chi_Q(x) + \sum_{Q \in \mathcal{B}} L_{P_Q}(x) \cdot \chi_Q(x). \tag{2.10}$$

That is, $\widetilde{\underline{C}}$ is the fitted least-squares line for $Q \in \mathcal{Q} \setminus \mathcal{B}$ and the least-squares fit of the parent of $Q \in \mathcal{B}$, that is any Q satisfying any of the stopping criteria of equations (2.4), (2.5) or (2.6).

The algorithm computes the function $\widetilde{S}$ as follows:

$$\widetilde{S}(x) = \sum_{Q \in \mathcal{Q} \setminus \mathcal{B}} \sigma_{\widetilde{Q}} \cdot \chi_Q(x) + \sum_{Q \in \mathcal{B}} \sigma_{\widetilde{P}_Q} \cdot \chi_Q(x).$$

Note that this function estimates locally the square root of the second moments of the distances of a subset of E (excluding "initial outliers") to the curve $\underline{\widetilde{C}}$. We refer to it as an estimate for the "local standard deviations".

The problem is that the function $\widetilde{S}$ does not estimate the "standard deviation" of F_{in}, but the "standard deviation" of its restriction to $\cup_{Q\in\mathcal{Q}}\widetilde{Q}$. We suggest a fix for this problem by assuming that the data in the regions $\{\widetilde{Q}\}_{Q\in\mathcal{Q}}$ can be well approximated by a restriction of a normal distribution. Note that we do not assume anything about the tail of the distribution and that this is the only place where we introduce parametric assumptions. The function $\widehat{S}$ lends itself to a clear statistical interpretation—it has a more global meaning and it can directly assign p-values (see Subsection 2.4.2). That said, in practice $\widetilde{S}$ and $\widehat{S}$ are very similar and $\widetilde{S}$ can be used if it is important to avoid using parametric assumptions altogether. It is also possible to compute p-values nonparametrically.

In order to simplify the definition of $\widehat{S}$, we assume here that $D = 2$ and discuss the general case (either symmetric or asymmetric). We later explain what we do in the symmetric case when $D > 2$. For any interval Q we denote $a_Q = c_- \cdot \ell(Q)$ and $b_Q = c_+ \cdot \ell(Q)$. Note that in the GS-case $b_Q = -a_Q = c_0 \cdot \ell(Q)$. Many of our data sets of current interest (e.g., ChIP-on-chip data, see [47]) fall in the category of the A -case, where $b_Q = \infty$.

We first define $\widehat{\sigma}_Q$ as the solution to the equation:

$$\sigma_Q^2 = -\frac{\widehat{\sigma}_Q}{\sqrt{2\pi}} \cdot \left(b_Q \cdot e^{-\frac{b_Q^2}{2\cdot\widehat{\sigma}_Q^2}} - a_Q \cdot e^{-\frac{a_Q^2}{2\cdot\widehat{\sigma}_Q^2}} \right) + \frac{\widehat{\sigma}_Q^2}{2} \cdot \left(\mathrm{erf}\left(\frac{b}{\sqrt{2}\cdot\widehat{\sigma}_Q} \right) - \mathrm{erf}\left(\frac{a_Q}{\sqrt{2}\cdot\widehat{\sigma}_Q} \right) \right) \quad (2.11)$$

and then obtain $\widehat{S}$ as follows:

$$\widehat{S}(x) = \sum_{Q\in\mathcal{Q}\setminus\mathcal{B}} \widehat{\sigma}_Q \cdot \chi_Q(x) + \sum_{Q\in\mathcal{B}} \widehat{\sigma}_{P_Q} \cdot \chi_Q(x).$$

Remark 2.4.1. *In order to solve equation (2.11), we have used the Matlab subprogram FZERO with the initial value* $(\widehat{\sigma}_Q)_0 = \sigma_Q$.

Remark 2.4.2. *Formula (2.11) is based on the following proposition:*
If $a < 0 < b$, $X \sim N(0, \widehat{\sigma}^2)$ *is a normal random variable with density function* $f = \frac{1}{\sqrt{2\pi\widehat{\sigma}^2}} \cdot e^{-\frac{x^2}{2\widehat{\sigma}^2}}$ *and if* $\sigma_{a,b}{}^2 := \int_a^b x^2 f(x)$, *then*

$$\sigma_{a,b}{}^2 = \frac{-\widehat{\sigma}}{\sqrt{2\pi}} \cdot \left(b \cdot e^{-\frac{b^2}{2\cdot\widehat{\sigma}^2}} - a \cdot e^{-\frac{a^2}{2\cdot\widehat{\sigma}^2}} \right) + \frac{\widehat{\sigma}^2}{2} \cdot \left(\operatorname{erf}\left(\frac{b}{\sqrt{2}\cdot\widehat{\sigma}} \right) - \operatorname{erf}\left(\frac{a}{\sqrt{2}\cdot\widehat{\sigma}} \right) \right).$$

This proposition follows from two simple observations. First, note that

$$\sigma_{a,b}{}^2 = \int_a^b x^2 f(x)\,\mathrm{d}x = \frac{-\widehat{\sigma}^2}{\sqrt{2\pi\widehat{\sigma}^2}} \int_a^b x \cdot \frac{d}{\mathrm{d}x} e^{-\frac{x^2}{2\widehat{\sigma}^2}}\,\mathrm{d}x = \frac{-\widehat{\sigma}}{\sqrt{2\pi}} \left[x e^{-\frac{x^2}{2\widehat{\sigma}^2}} \right]_a^b + \widehat{\sigma}^2 \int_a^b f(x)\,\mathrm{d}x. \qquad (2.12)$$

Second, recall that $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\,\mathrm{d}t$ *and note that* $\operatorname{erf}(\frac{x}{\sqrt{2}\cdot\widehat{\sigma}}) = \frac{1}{\sqrt{2\pi\widehat{\sigma}^2}} \int_0^x e^{-\frac{t^2}{2\widehat{\sigma}^2}}\,\mathrm{d}t$.

Remark 2.4.3. *It is possible to have* $a_Q = -\infty$ *or* $b_Q = \infty$, *but not both. Assume e.g.* $b_Q = \infty$ *(our choice with ChIP-on-chip data); In this case equation (2.11) obtains the form*

$$\sigma_Q^2 = \frac{\widehat{\sigma}_Q}{\sqrt{2\pi}} \cdot a_Q \cdot e^{\frac{-a_Q^2}{2}\widehat{\sigma}_Q^2} + \frac{\widehat{\sigma}_Q^2}{2} \left(1 - \operatorname{erf}\left(\frac{a_Q}{\sqrt{2}\cdot\widehat{\sigma}_Q} \right) \right).$$

Remark 2.4.4. *If* $D > 2$ *then we assume that the covariance matrix of the normal distribution is a scalar matrix with the scalar* σ_Q. *We also replace the interval* $[a, b]$ *(or* $[-a, a]$*) by a* $D-1$ *dimensional ball with center at* 0 *and radius* a. *The above lower dimensional techniques then extend as follows:* $\widehat{\sigma}_Q$ *is the solution of equation (2.11), where* $b_Q = -a_Q = c_0 \cdot \ell(Q)$.

2.4.1 Smoothing the output functions

It is possible to derive a smooth version of the above functions as follows: First, generate many instances of piecewise constant functions according to different grids. Then average these piecewise constant functions over all those instances (see Figure 2.4). This is similar to the notion of smoothed histograms via shifts in [70].

The Lipschitz condition is very natural because the averaging of step functions is Lipschitz. By using kernel methods and avoiding averaging, it is possible to obtain higher degrees of smoothness, if necessary. However, the averaging idea described below results in the fastest implementation of the algorithm and also in the most efficient storage.

The details of the averaging are as follows. Set $Q_0^* := [a_0^*, b_0^*]$ the shortest closed interval containing the projection of E onto L. Define the following n_{sh} intervals:

$$Q_i = [a_0^*, 2 \cdot b_0^* - a_0^*) - \frac{i \cdot \ell(Q_0^*)}{n_{sh}},\ i = 1, \ldots, n_{sh}.$$

That is, we uniformly shift the interval $Q_1 \equiv [a_0^*, 2 \cdot b_0^* - a_0^*)$ to the left.

Denote the strips formed with respect to the intervals Q_i, $i = 1, \ldots, n_{sh}$ by $\widetilde{\underline{C}}_i$, $\widetilde{S}_i$ and $\widehat{S}_i$.

The "averaged" strips are obtained as follows

$$\widetilde{\underline{C}} = \frac{1}{n_{sh}} \cdot \sum_{i=1}^{n_{sh}} \widetilde{\underline{C}}_i\,, \quad \widetilde{S} = \frac{1}{n_{sh}} \cdot \sum_{i=1}^{n_{sh}} \widetilde{S}_i \quad \text{and} \quad \widehat{S} = \frac{1}{n_{sh}} \cdot \sum_{i=1}^{n_{sh}} \widehat{S}_i\,.$$

2.4.2 Ranking and identification of outliers

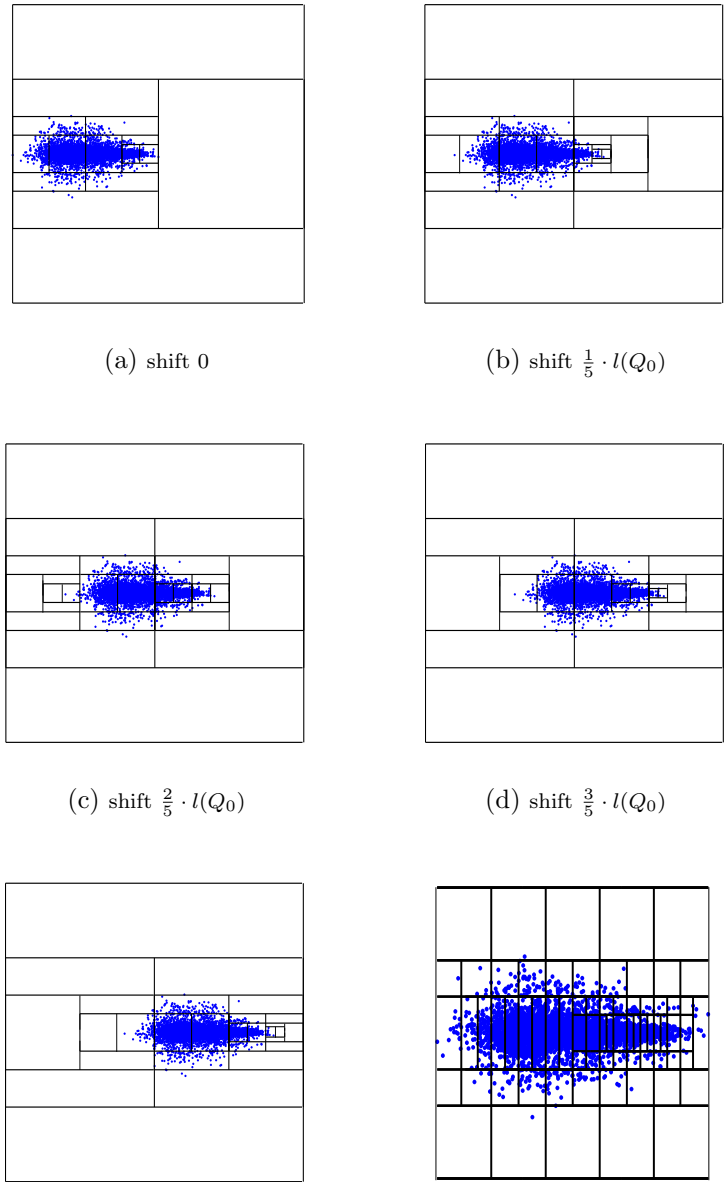
We assign $\widetilde{R}$ and $\widehat{R}$ to any point $(x, \underline{y}) \in E$ by:

$$\widetilde{R}(x, \underline{y}) = \frac{\|\underline{y} - \widetilde{\underline{C}}(x)\|_2}{\widetilde{S}(x)} \quad \text{and} \quad \widehat{R}(x, \underline{y}) = \frac{\|\underline{y} - \widetilde{\underline{C}}(x)\|_2}{\widehat{S}(x)}\,.$$

We mainly use the ranking $\widehat{R}$, although in practice, $\widetilde{R} \cong \widehat{R}$.

We may fix a threshold level λ and identify a corresponding set of outliers containing all points with $\widehat{R}$ greater than λ. In order to avoid arbitrariness of λ and assess the significance of outliers, we use $\widehat{R}$ to assign p-values as follows:

$$p\text{-val}(x, \underline{y}) = \frac{2}{\sqrt{2\pi}} \int\limits_{\widehat{R}(x,\underline{y})}^{\infty} e^{-\frac{t^2}{2}}\, \mathrm{d}t = \left(1 - \text{ erf}\left(\frac{\widehat{R}(x, \underline{y})}{\sqrt{2}}\right)\right) \tag{2.13}$$



(a) shift 0

(b) shift $\frac{1}{5} \cdot l(Q_0)$

(c) shift $\frac{2}{5} \cdot l(Q_0)$

(d) shift $\frac{3}{5} \cdot l(Q_0)$

(e) shift $\frac{4}{5} \cdot l(Q_0)$

(f) combination of all shifts on original grid

Figure 2.5: Subplots a-f are the shifted grid progressions indicating $n_{sh} = 5$ iterations, the partitions for the last shift (not shown) will be precisely the same as the zero*th* shift so it is not included (note the "exhausted" intervals are different for each shift so that the corresponding "strips" fitted over each exhausted cube will therefore also be different) - the lowest plot is the combination of all grids picture above it

If $D = 2$ and if we assume the A-case, with $b_Q = \infty$ for all $Q \in \mathcal{D}(Q_0)$, and define

$$\widehat{R}(x, \underline{y}) = \max\left(\frac{-(y - \widetilde{C}(x))}{\widehat{S}(x)}, 0\right) \tag{2.14}$$

then using 2.13 and 2.14 we get

$$p\text{-val}(x, \underline{y}) = \frac{1}{\sqrt{2\pi}} \int_{\widehat{R}(x,\underline{y})}^{\infty} e^{-\frac{t^2}{2}}\, \mathrm{d}t = \frac{1}{2}\left(1 - \text{ erf}\left(\frac{\widehat{R}(x, \underline{y})}{\sqrt{2}}\right)\right). \tag{2.15}$$

It is possible to use resampling techniques in order to assign p-values without parametric assumptions as done in [80] by taking percentiles of a bootstrapped $\widehat{R}^*_B(x, \underline{y})$, where B is the number of bootstraps. This, of course, adds an immense computational cost.

2.5 Determination of α_0

Until now we have not elucidated a way to approximate α_0 which, indeed, is the linchpin of the MSC algorithm. In many cases (e.g. expression profiling cDNA arrays) we have an explicit range or a specific value of expected percentage of outliers which we may use for α_0. In other cases (e.g. DNA arrays of ChIP-on-chip experimental data), we have no prior knowledge on the percentages of outliers that may assist in determining an appropriate value for α_0 or small range of values. In order to determine such a value, we apply the algorithm with different values of α_0. For each fixed α_0, we select few p-values and record the numbers $N_{out}(\alpha_0, p)$ of outliers that the algorithm detects for any fixed α_0 and p-values. We then look for the value α_0^* where highest jump has been observed in the number of outliers across various p-values. In order to justify the method, we assume a data set generated according to our model. For simplicity, assume further that the data set is planar ($D = 2$), generated according

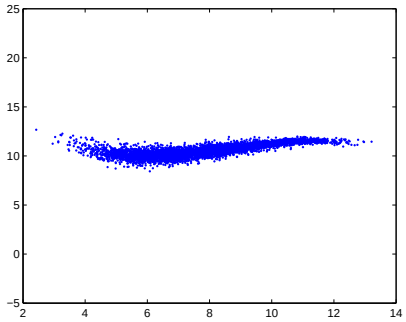
to the GS-model (general symmetric) with $P_{out}(x) \equiv P_{out}$. We want to that our proposed choice for α_0 will be a good estimator for P_{out}, the fraction of the outlier component. In order to justify our method, we need to understand for any given value of α, $(0 < \alpha_0 < 0.5)$ the dependence of N_{out}, the number of outliers detected by the algorithm, on the threshold p-value and the sensitivity of this relation on the parameter α_0. Equivalently, we need to understand for any given value of α_0, the dependence of N_{out} on the constant $B \equiv B(\alpha_0)$ such that outliers are identified outside the strips $\underline{\widetilde{C}}(x) + B \cdot \widehat{S}(x)$ and $\underline{\widetilde{C}}(x) - B \cdot \widehat{S}(x)$ (note that B determines the p-values, given $\underline{\widetilde{C}}(x)$ and $\widehat{S}(x)$). We start by considering such dependence (and its sensitivity to changes of α_0) in the very special case where there is no outlier component in our model ($P_{out} = 0$). In this case, it is clear that the curves describing the dependence of N_{out} on B(or the corresponding p-values) vary continuously with the parameter α_0. In Figure 2.5(a) we describe a sample from this model, whereas in Figure 2.5(b) we show that our method does not detect any jump, i.e. no layer of outliers. Assume next that ($P_{out} > 0$) and the following very special case: The underlying curve lies on the x-axis (i.e. the conditional mean of the stable distribution is zero) and the conditional variance of the stable distribution is constant (i.e. homoscedastic random variable). Moreover we assume that the outlier distribution is homoscedastic, bimodal and symmetric around 0 with constant means $\pm\mu$ and constant conditional variances σ around each mode, such that $\sigma \ll \mu$. Note that if $\alpha_0 > P_{out} + \epsilon$, $\epsilon > 0$ is a sufficiently small constant), then we expect the algorithm to peel out most of outliers and thus use mainly points sampled from the stable distribution to estimate the stable standard deviation. Therefore, in a similar way to the case above ($P_{out} = 0$), we expect continuous variation in the profile curves. Similarly, if $\alpha_0 < P_{out} - \epsilon$, the underlying distribution is trimodal and small variation in α_0 result in small variation of the sampled points and thus small variation in the profile curves. However, when $\alpha_0 \sim P_{out} - \epsilon$, we encounter a transition from an underlying unimodal distribution to

trimodal distribution. Therefore for a fixed B (sufficiently large) we notice a jump in the number of outliers detected by the algorithm when transitioning from $P_{out} - \epsilon$ to $P_{out} + \epsilon$.

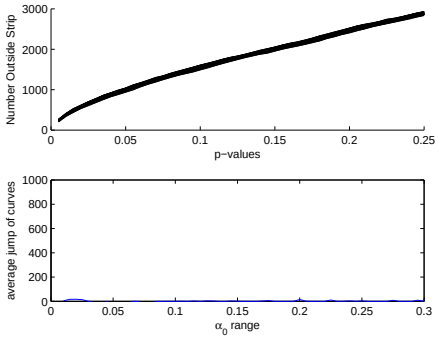
One can extend this argument to a more general setting. In Figure 2.5(c) we describe a sample from a more general trimodal distribution, whereas in Figure 2.5(d) we show that our method detects a jump, around $P_{out} = 0.1$. The argument above also applies to several modes of outliers and explains how one can get few jumps of profile curves. In this case, we prefer the jump with largest value of α_0, which we expect to be the largest. In Figure 2.5(e) we describe a sample from a distribution with 2 layers of outliers (4 modes), whereas in Figure 2.5(d) we show that our method detects two jumps, around the expected fraction of layers of outliers: 5% and 10%. We can generalize this observation to several layers of outliers and conclude that if there are several jumps in the profile curves and the data is generated according to our model, then a good estimator for P_{out} (α_0^* according to our previous notation) is the highest value of α_0 where a jump occurred (it also the jump of highest value). The other jumps indicate transitions between various layers of outliers. Though this is a heuristic method for estimating α_0 it is quite effective in practice when little else is known of the data.

2.6 Restarting the algorithm at stopping time intervals

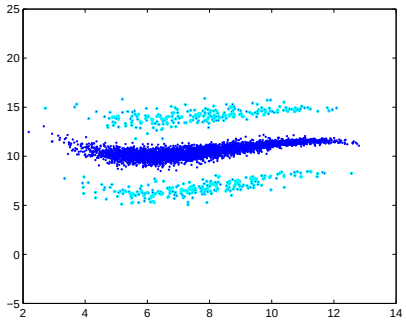
We now present a version of the algorithm that iterates our basic algorithm repeatedly in stopping time intervals. Let us first motivate the development of this extension by a simple example. Assume that data has been generated according to our basic model, where $Q_0^* = [0, 1]$, $\underline{C}$ is the zero function and S is a smooth function that



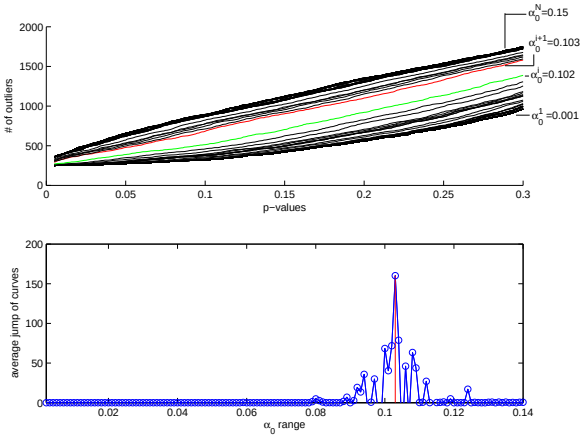
(a) data with no jumps



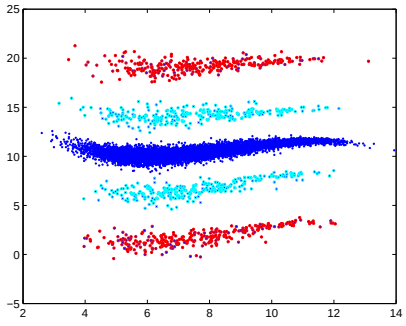
(b) no jumps detected



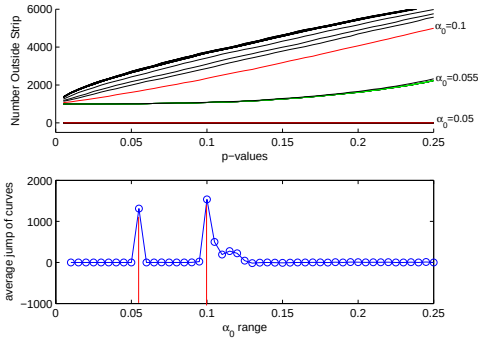
(c) data with 1 layer (either side) of outliers



(d) jump detected



(e) data with 2 layers (either side) of outliers



(f) 2 jumps detected

Figure 2.6: In this case, the data is composed of 2 layers of outliers where the outer-most layer comprises a fraction of 0.05 and the inner comprises a fraction of 0.05 - a cumulative fraction of 0.1.

is zero in a region $\{x_1 : |x_1 - \bar{x}_1| \leq \varepsilon\}$ for some $\varepsilon > 0$ and $x_1 \in [0,1]$, equals 1 on $\{x_1 : |x_1 - \bar{x}_1| \geq 2 \cdot \varepsilon\}$.

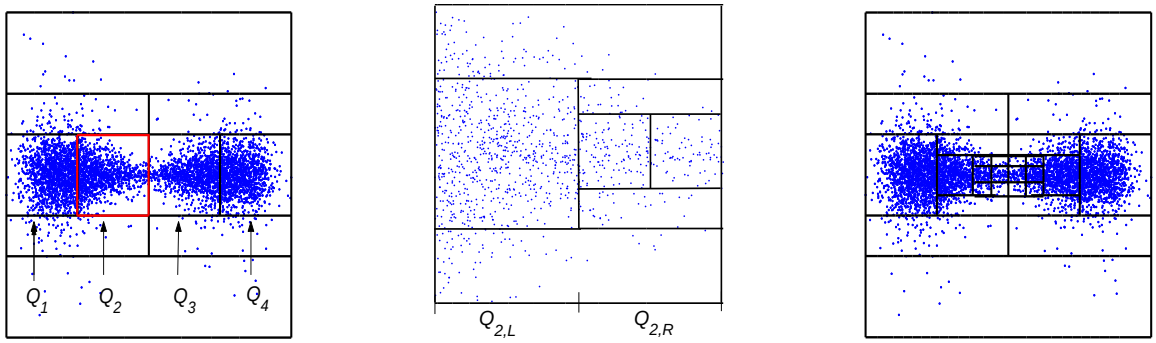
Unless c_0 is sufficiently large, the algorithm will stop early (zeroth or first level) before it arrives at a scale with length comparable to ε. In fact, Proposition 5.4 in the following section points out at a similar obstacle from a different perspective. There are two immediate fixes to that problem which we find undesirable. The first one is to use sufficiently large constant c_0. However, in this way information from larger scales is ignored, not to mention the arbitrariness in choosing such a constant c_0. The other way is to restart the algorithm in each stopping time interval Q by using the data set $\widehat{Q} \cap E$; that is, all points projected on Q. The problem is that this naive way of iterating the stopping time construction may easily introduce large variance of estimation and may also ignore important information from larger scales. We thus try to restart in stopping time intervals using some information from previous scales and also maintaining an estimate (but no strong restriction) on the derivative of $\widetilde{S}_T$. Each time we restart we allow a large constant c_0. We describe the details of the restarting algorithm by pseudocode. For this purpose, first, we modify Algorithm 2.6 slightly, and enumerate some details relevant to this version of the algorithm.

Figure 2.6 depicts the idea of restarting in the stopping time intervals. Figure 2.6(a) is the result of applying the basic algorithm, Algorithm 2.6, or equivalently first iteration of Algorithm 2.6. It is applied to the whole data set with respect to the interval Q_0 and with the parameters $\alpha_0 = 0.15$ and $c_0 = 0.5$. At the zeroth level $f_{Q_0} = 0.007$ and $F_{Q_0} = 0.007$. Level 1 yields two intervals $Q_{1,2} = Q_1 \cup Q_2$ and $Q_{3,4} = Q_3 \cup Q_4$ (the intervals Q_1, Q_2, Q_3, Q_4 are denoted in Figure 2.6). Their corresponding fractions are $f_{Q_{1,2}} = 0.056$ and $F_{Q_{1,2}} = 0.063$ and $f_{Q_{3,4}} = 0.035$ and $F_{Q_{3,4}} = 0.042$. Because both F_Qs are less than α_0 we proceed to level 2 which divides the intervals into Q_1, Q_2, Q_3 and Q_4 as shown. It turns out, each of these is a stopping interval with $f_{Q_1} = 0.275$ and $F_{Q_1} = 0.339$, $f_{Q_2} = 0.148$ and $F_{Q_2} = 0.211$,

Algorithm 2.2: MSC_basic

Input: Interval Q, data E_Q, F_{P_Q}, σ_{P_Q} and parameters
Output: *Stop_int*, and for any $Q' \in$ *Stop_int*: $E_{Q'}$, $L_{Q'}$, $F_{P_{Q'}}$, $\sigma_{P_{Q'}}$, $\widetilde{S}_{Q'}$
Initialization: Set: $l = 0$, $Good\_int = \{Q\}$, $N_{\text{stop}} = 0$
while $l \leq l_0$, $N_{\text{stop}} < N$ **do**
  Apply **Steps 1 and 2** of Algorithm 2.3
  for all $Q' \in Good\_int$ **do**
    $F_{Q'} := F_{Q'} + F_{P_Q}$
  end for
  Apply **Steps 3 and 4** of Algorithm 2.3
  if $Q' \in Stop\_int$ satisfies any one of equations (2.4), (2.5) or (2.6) **then**
    $\widetilde{S}_{Q'} := \sigma_{P_{Q'}}$
  end if
  l:= l+1
end while
for all $Q' \in Stop\_int$ **do**
  $E_{Q'} = E_Q \cap \widetilde{Q}'$
  Record (output): $E_{Q'}$, $L_{Q'}$, $F_{P_{Q'}}$, $\sigma_{P_{Q'}}$, $\widetilde{S}_{Q'}$
end for

$f_{Q_3} = 0.126$ and $F_{Q_3} = 0.168$, $f_{Q_4} = 0.220$ and $F_{Q_4} = 0.262$. We use the interval Q_2 to demonstrate the restarting idea of Algorithm 2.6. We use the data in $\widetilde{Q_2}$ (defined in the previous iteration) and set $c_0 = 1$. In Figure 2.6(b) we see the further division of Q_2 into $Q_{2,L}$ and $Q_{2,R}$. In the cube $Q_{2,L}$, $f_{Q_{2,L}} = 0.188$ and $F_{Q_{2,L}} = f_{Q_{2,L}} + F_{Q_{1,2}} = 0.251$; so we proceed no further in $Q_{2,L}$. In the interval $Q_{2,R}$, $f_{Q_{2,R}} = 0.044$ and $F_{Q_{2,R}} = f_{Q_{2,R}} + F_{Q_{1,2}} = 0.107$, which is less than α_0; so we proceed further by dividing $Q_{2,R}$ itself into two intervals - $Q_{2,RL}$ and $Q_{2,RR}$. Both of the subintervals of $Q_{2,R}$ are stopping time intervals with $f_{Q_{2,R_L}} = 0.282$, $F_{Q_{2,R_L}} = 0.389$ and $f_{Q_{2,R_R}} = 0.129$, $F_{Q_{2,R_R}2} = 0.236$. Figure 2.6(c) displays the final result after restarting the algorithm in all original stopping time cubes. Note that only Q_2 and Q_3 have been subdivided further.



(a) Results of first application of MSC algorithm

(b) Restarting in Q_2

(c) The final results after restarting

Figure 2.7: Subfigure (a) displays the results of first application of MSC algorithm; we will restart the algorithm for the exhausted cube outlined in red where $F_Q = 0.211$. Subfigure (b) displays results after restarting the algorithm in interval Q_2; further dyadic divisions (denoted L (left) and R (right)) of Q_2 are inspected with $F_{Q_{2,L}} = 0.188$ and $F_{Q_{2,R}} = 0.044$. Since $F_{Q_{2,R}} < \alpha_0$ it divides cube R again and then stops. (c) The final results after restarting the algorithm in all stopping time cubes; only intervals Q_2 and Q_3 were subdivided further

Algorithm 2.3: MSC_restart

Input: Data E and parameters
Output: Curve $\widetilde{C}$, standard deviation function $\widetilde{S}$, ranking R, a set of outliers
Initialization:
Approximate E by a line L and apply a rigid transformation to E, so that $L = x$-axis
Set $Q_0 :=$ interval containing the projection of E on L
Set: $repeat = 0$, $Restart\_int = \{Q_0\}$, $P_{Q_0} := Q_0$, $F_{Q_0} = 0$, $Tot\_stop = \emptyset$ and $N_{\text{tot\_stop}} = 0$
while $repeat \leq r_0$, $N_{\text{tot\_stop}} < N$ **do**
$Current\_stop = \emptyset$
for all $Q' \in Restart\_int$ **do**
Apply MSC_main(Q', E'_Q, $F_{P'_Q}$, $\sigma_{P'_Q}$)
Record Output: $Stop\_int(Q')$ and for any $Q'' \in Stop\_int(Q')$: $E_{Q''}$, $F_{P_{Q''}}$, $\sigma_{P_{Q'}}$, $\widetilde{S}_{Q'}$
if $Q'' \in Stop\_int(Q')$ and $|\widetilde{Q}''| > n_0$ **then**
$Tot\_stop := Tot\_stop \cup \{Q''\}$
else
$Current\_stop := Current\_stop \cup \{Q''\}$
end if
end for
$Restart\_int :=$ set of dyadic subintervals of intervals in Tot_stop
repeat:= repeat+1
$N_{\text{tot\_stop}} =$ number of points in Tot_stop
$c_0 = 2 \cdot c_0$
end while
Record Outputs 1-4 as in Algorithm 2.3

2.7 Properties of the Algorithm

The algorithm has several interesting properties following directly from its geometric nature. For instance, we can analytically bound the initial number of points discarded at various scales and then use this bound to control the number of points whose rank exceeds a specific value. Furthermore, we can characterize the smoothness of the output functions. Finally, we estimate the computational speed of the algorithm.

In the following analysis we use $\widetilde{S}$ instead of $\widehat{S}$, in order to simplify our estimates and avoid some tedious, long and technical computations. Nevertheless, numerical experiments show that $\widetilde{S} \cong \widehat{S}$.

2.7.1 bounding number of initial outliers.

We first obtain an upper bound on the local percentages of initial "outliers" discarded at each scale.

Given $Q \in \mathcal{P} \setminus \mathcal{Q}$, we denote by U_Q the subset of $\widetilde{Q}$ containing all discarded points ("initial outliers"). That is,

$$U_Q = \widetilde{Q} \setminus \bigcup_{\substack{Q' \in \mathcal{Q} \\ Q' \subseteq Q}} \widetilde{Q'}$$

Note that

$$U_Q = \bigcup_{\substack{Q' \in \mathcal{P} \\ Q' \subseteq Q}} T(\widetilde{Q'}) \tag{2.16}$$

We control the number of point in U_Q as follows:

Proposition 2.7.1. *For any $Q \in \mathcal{P} \setminus \mathcal{Q}$:*

$$|U_Q| \leq \alpha_0 \cdot |\widehat{Q}|.$$

Proof. We denote by P_L the projection operator from $\mathbb{R}^D$ onto L. We note that

$$\begin{aligned}
|U_Q| &= \sum_{\substack{Q' \in \mathcal{P} \\ Q' \subseteq Q}} |T(\widetilde{Q'})| = \sum_{\substack{Q' \in \mathcal{P} \\ Q' \subseteq Q}} f_{Q'} \cdot |\widehat{Q'}| = \sum_{\substack{Q' \in \mathcal{P} \\ Q' \subseteq Q}} f_{Q'} \cdot \sum_{x \in E \cap \widehat{Q}} \chi_{\widehat{Q'}}(x) \\
&= \sum_{\substack{Q' \in \mathcal{P} \\ Q' \subseteq Q}} f_{Q'} \cdot \sum_{x \in P_L(E \cap \widehat{Q})} \chi_{Q'}(x) = \sum_{x \in P_L(E \cap \widehat{Q})} \sum_{\substack{Q' \in \mathcal{P} \\ Q' \subseteq Q}} f_{Q'} \cdot \chi_{Q'}(x) \leq \sum_{x \in P_L(E \cap \widehat{Q})} \sum_{\substack{Q' \in \mathcal{Q} \\ Q' \subseteq Q}} F_{P_{Q'}}(x) \\
&\leq \sum_{x \in P_L(E \cap \widehat{Q})} \alpha_0 = \alpha_0 \cdot |P_L(E \cap \widehat{Q})| = \alpha_0 \cdot |\widehat{Q}|.
\end{aligned}$$

We remark that the first equality follows from equation (2.16). The second one follows from the definition of $f_{Q'}$. The first inequality follows from the definition of F_Q (it is an equality if $Q = Q_0$). The second inequality follows from the fact that parents of stopping time intervals do not satisfy the first stopping time condition (see equation (2.3)). □

Remark 2.7.1. *It is possible to define the fraction f_Q slightly differently:*

$$f'_Q := \frac{|T(\widetilde{Q})|}{|\widetilde{Q}|}.$$

Note that f'_Q and the corresponding additive sum F'_Q are larger than f_Q and F_Q and will thus result in a coarser stopping time region. It is clear from the above analysis that the use of the latter fraction results in the following property: $|U_Q| \leq \alpha_0 \cdot |\widetilde{Q}|$ for any interval $Q \in \mathcal{P} \setminus \mathcal{Q}$. In the applications we have considered so far, the former quantities f_Q and F_Q suit the task of data analysis better.

We next control the percentages of outliers given by the new ranking. For any fixed constant λ, $\lambda \geq 1$, and for any interval Q in $\mathcal{D}(Q_0)$ we define

$$R_{Q,\lambda} = \{(x, \underline{y}) \in \widehat{Q} : \ \widetilde{R}(x, \underline{y}) \geq \lambda\}.$$

We estimate the size of $R_{Q,\lambda}$ by estimating the size of the following auxiliary set:

$$M_{Q,\lambda} = \{(x, \underline{y}) \in \widehat{Q} \setminus U_Q : \ \widetilde{R}(x, \underline{y}) \geq \lambda\}.$$

The following constant is used when bounding $|M_{Q,\lambda}|$:

$$C^* = \max_{Q \in \mathcal{P} \setminus \mathcal{Q}} \frac{\sum\limits_{\substack{Q' \in \mathcal{B} \\ Q' \subseteq Q}} |\widetilde{P}_{Q'}|}{\sum\limits_{\substack{Q' \in \mathcal{B} \\ Q' \subseteq Q}} |\widetilde{Q}'|}$$

This constant measures the fraction of "number" of points in the cylindrical regions of parents of $\mathcal{B}$ intervals (counting with multiplicities) over the number of points in the cylindrical regions of intervals in $\mathcal{B}$.

Our first estimate for size of outliers is formulated as follows:

Proposition 2.7.2. *For any $Q \in \mathcal{P} \setminus \mathcal{Q}$ and $\lambda > 1$*

$$|M_{Q,\lambda}| \leq \frac{C^* + 1}{\lambda^2} \cdot |\widehat{Q}| \quad \textit{and} \tag{2.17}$$

$$|R_{Q,\lambda}| \leq \left(\frac{C^* + 1}{\lambda^2} + \alpha_0 \right) \cdot |\widehat{Q}|. \tag{2.18}$$

Proof. Equation (2.17) is an immediate consequence of Chebychev's Inequality over a counting measure where $|\cdot|$ indicates the cardinality of the set. Indeed,

$$\begin{aligned}
|M_{Q,\lambda}| &= \{(x,\underline{y}) \in \widehat{Q} \setminus U_Q : \ \widetilde{R}(x,\underline{y}) \geq \lambda\} \\
&= |\{(x,\underline{y}) \in \widehat{Q} \setminus U_Q : \sum_{\substack{Q' \in \mathcal{Q} \\ Q' \subseteq Q}} (\mathrm{dist}(\underline{y}, \underline{\widetilde{C}}(x)) - \lambda \cdot \widetilde{S}(x)) \cdot \chi_Q(x) \geq 0\}| \\
&\leq \sum_{\substack{Q' \in \mathcal{Q} \\ Q' \subseteq Q}} \left|\{(x,\underline{y}) \in \widehat{Q}' \setminus \widetilde{Q}' : x \in Q' \text{ and } \mathrm{dist}(\underline{y}, \underline{\widetilde{C}}(x)) \geq \lambda \cdot \widetilde{S}(x)\}\right| \\
&\quad \text{(by definition of Chebychev for a counting measure)} \\
&\leq \sum_{\substack{Q' \in \mathcal{Q} \\ Q' \subseteq Q}} \sum_{\substack{x \in Q' \\ (x,\underline{y}) \in \widehat{Q} \setminus \widetilde{Q}'}} \frac{\mathrm{dist}^2(\underline{y}, \underline{\widetilde{C}}(x))}{\lambda^2 \cdot \sigma_Q^2} \leq \sum_{\substack{Q' \in \mathcal{Q} \setminus \mathcal{B} \\ Q' \subseteq Q}} \frac{1}{\lambda^2} \cdot |\widetilde{Q}'| + \sum_{\substack{Q' \in \mathcal{B} \\ Q' \subseteq Q}} \frac{1}{\lambda^2} \cdot |\widetilde{P}_{Q'}| \\
&\leq \frac{C^* + 1}{\lambda^2} \cdot |\bigcup_{\substack{Q' \in \mathcal{Q} \\ Q' \subseteq Q}} \widetilde{Q}'| = \frac{C^* + 1}{\lambda^2} \cdot |\widehat{Q} \setminus U_Q| \leq \frac{C^* + 1}{\lambda^2} \cdot |\widehat{Q}| \,.
\end{aligned}$$

At last note that

$$|R_{Q,\lambda}| \leq |M_{Q,\lambda}| + |U_Q| \tag{2.19}$$

and combine it with equation (2.17) and Proposition 2.7.5 in order to that the bound of equation (2.18) holds. □

It can be observed that in many instances of the data sets considered in this paper the decay rate of "outliers" seems to be faster than the one above. The following proposition explains how such a fast rate can be obtained in some cases. The proof of that proposition is an immediate corollary of Proposition 2.7.5.

Proposition 2.7.3. *Assume that there exist constants λ^* and C such that for all $\lambda \geq \lambda^*$ and all $Q \in \mathcal{P} \setminus \mathcal{Q}$*

$$|M_{Q,\lambda}| \leq C \cdot |U_Q|.$$

Then

$$|M_{Q,\lambda}| \ \leq C \cdot \alpha_0 \cdot |\widehat{Q}| \ \ \textit{and} \ \ |R_{Q,\lambda}| \ \leq (C+1) \cdot \alpha_0 \cdot |\widehat{Q}|.$$

2.7.2 The smoothness properties of $\widetilde{S}$

The smoothness properties of the "standard deviation function" $\widetilde{S}$ depend on the parameters of the algorithm. They may be viewed as a qualitative way of characterizing its variance estimation.

We first estimate the "rate of change" of the piecewise constant version of $\widetilde{S}$.

Proposition 2.7.4. *If γ is the graph of the piecewise constant function $\widetilde{S}$ with length $\ell(\gamma)$, then*

$$\ell(\gamma \cap \widehat{Q}) \leq (1 + 2 \cdot \delta_0) \cdot \ell(Q) \quad \textit{for any} \ \ Q \in \mathcal{P} \setminus \mathcal{Q}.$$

The smoothness properties of the estimated function of the curve $\underline{\widetilde{C}}$ and the "standard deviation function" $\widetilde{S}$ are established in length in [46]. Briefly, assume for simplicity, the data is scaled and shifted so that $Q_0^* = [0,1]$. For any shift $0 < \gamma < 1$, we denote $Q_\gamma = [-\gamma, 2-\gamma)$. Let $\widetilde{S}_\gamma$ and $\underline{\widetilde{C}}_\gamma$ be the stepwise constant and linear functions respectively corresponding to the dyadic grid $D(Q_\gamma)$. We define

$$\widetilde{S}_T(x) = \int_0^1 \widetilde{S}_\gamma \, \mathrm{d}\gamma \text{ and } \underline{\widetilde{C}}_T(x) = \int_0^1 \underline{\widetilde{C}}_\gamma \, \mathrm{d}\gamma, \tag{2.20}$$

the functions corresponding to continuously shifted grids on $[0,1)$. Lerman, et al. [46] showed the functions $\widetilde{S}_T(x)$ and $\underline{\widetilde{C}}_T(x)$ are Lipschitz with norms $\|\widetilde{S}_T\|_{Lip} \leq 4 \cdot \delta_0 \cdot (l_0+1)$ and $\|\underline{\widetilde{C}}_T\|_{Lip} \leq 2 \cdot \sqrt{D-1}\theta_0$, where θ_0 is from equation 2.7. Some remarks regarding other properties are also in order, namely:

Remark 2.7.2. *Even if the third stopping time condition (equation (2.5)) is ignored, then we have a restriction on the smoothness of $\widetilde{S}$. Indeed, note that $\sup_{\substack{Q \in \mathcal{D}(Q_\alpha) \\ 0 \leq \alpha \leq \ell(Q_0)}} \beta_{\widetilde{Q}} \leq c_0$, and thus $\|\widetilde{S}_T\|_{Lip} \leq c_0 \cdot \ell_0$.*

Remark 2.7.3. *The current version of the algorithm may not recover some highly nonlinear characteristics or functions with large absolute variation embedded in the data (assume e.g. that* $\|S\|_{Lip} \gg \delta_0 \cdot \ell_0$*). However, it follows from the above proposition that such characteristics could be detected if both the number of levels exploited by the algorithm and the parameter* δ_0 *(or possibly* c_0 *which restricts it) are sufficiently large. In Chapter 2.6 we suggest a fix to that problem, independent of tuning the parameters.*

Remark 2.7.4. *We view the above proposition as a bound on the variance of estimation of the algorithm (at the cost of increasing the bias of approximation; see also Chapter 2.6).*

Remark 2.7.5. *We may view the averaged strip* $\widetilde{S}$ *as adaptive quadrature rule with uniform weights for approximating the integral* $\widetilde{S}_T$*. We may also use nonuniform weights in order to speed up the convergence of* $\widetilde{S}$ *to* $\widetilde{S}_T$*.*

Remark 2.7.6. *Experimentation with data shows that the average version of* $\underline{\widetilde{C}}$ *is practically a Lipschitz function (up to the resolution determined by the number of shifts). In general, we cannot guarantee that* $\underline{\widetilde{C}}_T$ *is a Lipschitz function, especially when the underlying function* $\underline{C}$ *is not Lipschitz. However, we may enforce the Lipschitz property by computing the approximating lines in larger regions (e.g. regions of the form* $3 \cdot \widetilde{Q}$*). In such a case (or in the similar case described in Remark 2.7.7), analogous arguments to those of David and Semmes [20, in particular Chapter 10] can then be used in order to prove the Lipschitz property of* $\underline{\widetilde{C}}$ *and its dependence on the parameter* θ_0*.*

Remark 2.7.7. *It is possible to construct functions* $\underline{\widetilde{C}}$ *and* $\widetilde{S}$ *whose* k*-th derivative is Lipschitz. In order to do so, one may replace the linear regression by polynomial regression of* $(k+1)$ *degree and also replace the indicator functions of the form* $\chi_Q(x)$ *appearing in the definitions of both* $\underline{\widetilde{C}}$ *and* $\widetilde{S}$ *by* $\psi_Q^k(x)$*, a* C^k *function such that*

$\psi_Q^k(x) = 1$ *for all* $x \in Q$ *and* $\psi_Q^k(x) = 0$ *for all* $x \notin 3Q$. *In this case, the averaging over shifted grids is not needed.*

Remark 2.7.8.

2.7.3 Applying Edge Weights

In cases where the data only occupies a fraction of the cube, occurring when the cube contains the minimum or maximum observation on the x-axis (by definition), we limit its effect by giving zero weights to the line fit at these points.

2.7.4 The Influence Function for zero shifts

The *Influence function* is computed for a statistic T

$$IF(T, x) = \frac{\partial}{\partial t} T\left((1-t)P + t\delta_x\right)|_{t=0}$$

[78]. $IF(T, x)$ measures the change in $T(F)$ for an infinitesimally small part of F replaced by a pointmass at x. Unbound influence functions are considered non-robust as a small fraction of the observations may have an inordinate effect on the estimator if their values were greater than or equal to x where the influence function is large.

It is not just a happy coincidence that the MSC version of the curve is robust with increasing level of outliers (see figures 3.1 a-f). Indeed, it was designed with robustness in mind, or at least as a consequence of neglecting "initial" outliers. As the levels increase and the algorithm continues its dyadic division, points with the largest residuals are neglected at the next level. To wit, one can easily infer that if the variance of the independent variable x is constant, we could determine that the influence is bounded by the influence of the point with the largest residual left in the calculation of the curve $\underline{\widetilde{C}}$, recalling $\underline{\widetilde{C}}$ from 2.10.

The influence function *IF* for the least squares estimate is

$$IF(T, x) = \Sigma^{-1}(F)x(y - x^T T(F))$$

[19; 23], where $\Sigma(F)$ is the covariance matrix on F. The influence function is unbounded because $(y - x^T T(F))$ may not be bounded. To mitigate the large effects a single point may have, robust regression re-weights the observations with large values of $(y - x^T T(F))$. By design, in the MSC case, the residual is already bounded by the constant c_0 an the length of the dyadic interval $\frac{\ell(Q_0)}{2^j}$. This can be seen using the empirical equivalent of the influence function called the *empirical influence curve* or

$$EIC_i \;=\; n(X^T X)^{-1} x_i (y_i - x_i \widehat{\beta}).$$

Here $T(\widehat{F}) = \widehat{\beta}$ is the best, unbiased regression coefficient using the empirical distribution $\widehat{F}$ [19].

Consider a non-stopping time cube P_Q with dyadic child Q. The points $\widehat{Q} \cap P_Q$ are used to estimate the new line, L_Q (see figure 2.7.4, for simplicity note that Q_L and Q_R are the dyadic children and Q represents the parent cube in this figure.) The absolute value of the sum of EIC using only points in $\widehat{Q} \cap P_Q$ is

$$|\sum_{i}^{|\widehat{Q}\cap P_Q|} EIC_i| \leq \frac{|\widehat{Q} \cap P_Q|}{\mathbf{x}^T \mathbf{x}} \sum_{i}^{|\widehat{Q}\cap P_Q|} |x_i \; (y_i - L_Q(x_i)|$$

using the triangle inequality for the term in the sum. Here $\mathbf{x} = \left\{x_1, \ldots, x_{|\widehat{Q}\cap P_Q|}\right\}$ and $|\widehat{Q} \cap P_Q|$ is the number of points in $\widehat{Q} \cap P_Q$. Now, points in regions P_Q were used to compute L_{P_Q} but only points in $\widehat{Q} \cap P_Q$ were used to compute L_Q. Thus, naturally, for $(x_i, y_i) \in \widehat{Q} \cap P_Q$,

$$\sum_{i}^{|\widehat{Q}\cap P_Q|} (y_i - L_Q(x_i))^2 \;\leq\; \sum_{i}^{|\widehat{Q}\cap P_Q|} (y_i - L_{P_Q}(x_i))^2 \;\leq\; |\widehat{Q} \cap P_Q| \; (c_0 \ell(P_Q))^2 \qquad (2.21)$$

since L_Q represents the minimum variance, unbiased estimates of the line in $\widetilde{Q}$.

Proposition 2.7.5. *The absolute sum of EIC is bounded in the MSC scheme for stopping time levels greater than 1.*

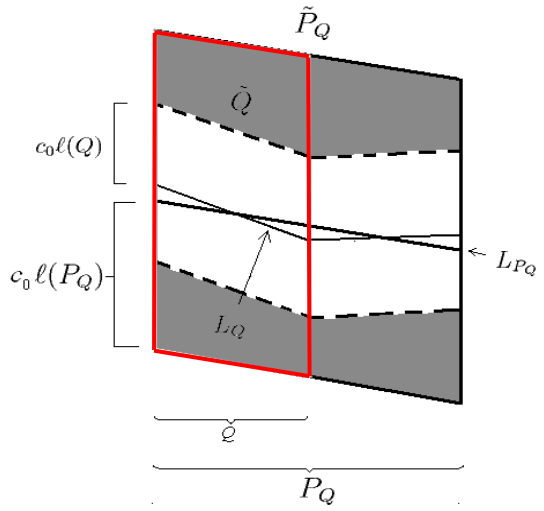


Figure 2.8: Using Q to denote the left child cube, for $\{x_i, y_i\} \in \widehat{Q} \cap P_Q$ (the region outlined in red) we know that $\|y_i - L_Q\| \leq \|y_i - L_{P_Q}\|$.

Proof. For a non-stopping time cube P_Q and its dyadic child Q

$$\sum_i^{|\widehat{Q}\cap P_Q|} |x_i\ (y_i - L_Q(x_i)| \leq (x_Q^T x_Q)^{1/2} \cdot \left(\sum_i^{|\widehat{Q}\cap P_Q|} (y_i - L_Q(x_i))^2 \right)^{1/2} \tag{2.22}$$

$$\leq (\mathbf{x}^T\mathbf{x}\ |\widehat{Q} \cap P_Q|)^{1/2} \cdot c_0\ell(P_Q) \tag{2.23}$$

by the Cauchy-Schwarz theorem and equation 2.21. Thus,

$$|\sum_i^{|\widehat{Q}\cap P_Q|} EIC_Q| \leq |\widehat{Q} \cap P_Q| \left(\frac{|\widehat{Q} \cap P_Q|}{(\mathbf{x}^T\mathbf{x})} \right)^{1/2} \cdot c_0\ell(P_Q).$$

□

The term $\left(\frac{|\widehat{Q}\cap P_Q|}{(\mathbf{x}^T\mathbf{x})}\right)^{1/2}$ is the inverse of the standard deviation of the independent variable in $\widehat{Q} \cap P_Q$ so we are essentially just scaling the bound of the residual of the last iteration to get a bound on the influence for the current iteration.

2.7.5 Speed of the algorithm

We prove the following proposition describing the computational complexity of the algorithm.

Proposition 2.7.6. *The speed of the algorithm for a data set of N points in $\mathbb{R}^D$, when using ℓ_0 levels and n_{sh} shifts and the QR algorithm to solve the linear system of equations is of order $O(N \cdot D^2 \cdot \ell_0 \cdot n_{sh})$.*

Proof. The cost of least squares via the QR factorization is $\sim 2ND^2 - \frac{2}{3}D^3$ flops for an N data points in D dimensions [75]. It is clear that the time complexity of the algorithm is linear in the number of shifts. Therefore, we restrict the algorithm to the grid $\mathcal{D}(Q_0)$. We use the QR decomposition and backsolve to find the regression coefficient for each line since this method is backward stable.

The first initialization step is $O(N \cdot D)$. Indeed, it mainly involves the finding of the first (top) principal axis and applying $(D-1)$ one-dimensional linear regressions using the QR factorization and backsolving. We note that the computation performed in the region $\widehat{Q} \cap P_Q$ associated with any interval $Q \in \mathcal{P}(\mathcal{Q}_\prime)$ is of order $O(|\widehat{Q} \cap P_Q| \cdot D^2)$. Indeed, the main computation at each interval involves finding $(D-1)$ one-dimensional regression lines of $|\widehat{Q} \cap P_Q|$ points. Therefore the speed of the algorithm, when using only one grid, is of order

$$D^2 \cdot \sum_{Q \in \mathcal{P}} |\widehat{Q} \cap P_Q| \leq D^2 \cdot \sum_{j=0}^{\ell_0} \sum_{\substack{Q \in \mathcal{P} \\ \ell(Q) = 2^{-j} \cdot \ell(Q_0)}} |\widehat{Q} \cap P_Q| \leq D^2 \cdot (\ell_0 + 1) \cdot |\widehat{Q}_0| = D^2 \cdot (\ell_0 + 1) \cdot N.$$

□

Chapter 3

Numerical Experiments

In this section, we evaluate our algorithm on different sets of data and make comparisons with other common and powerful techniques when applicable. In this thesis, synthetic data in $\mathbb{R}^2$ is created while in [46] we create synthetic data sets in both $\mathbb{R}^2$ and $\mathbb{R}^D$ as well as analyzing high dimensional data of image patches or pixel neighborhoods (the advantage of the patches data is that outliers of the pixel neighborhoods can be interpreted and visualized as edges of the original image from which they were taken.) Lastly, we generalize the algorithm to data concentrated around a 2-dimensional Lipschitz graph (d-dimensional Lipschitz graph in in [46]). We estimate the 2-dimensional graph and correspondent strip to assess "outliers".

3.1 Synthetic Data in $\mathbb{R}^2$

The data is generated as follows. We assume that $F_{\text{in}}(x, y)$ is a mixture of normal distributions $\{F_{\text{in}}^i(x, y)\}_{i=1}^{I_{\text{in}}}$ with mixture parameters ($\Pi_{\text{in}}^{I_1}, \ldots, \Pi_{\text{in}}^{I_{\text{in}}}$). We fix numbers $x_1 < x_2 < \cdots < x_{I_{\text{in}}}$ (locations of means of mixtures on the x-axis), $0 < \sigma_1, \ldots, \sigma_{I_{\text{in}}}$ (standard deviations of mixtures in the y direction), and $0 < s_1, \ldots, s_{I_{\text{in}}}$ (standard

deviations of mixtures in the x direction) such that $x_{i+1} - x_i \approx (s_{i+1} + s_i)/2$, $i = 1, \ldots, I_{\text{in}} - 1$, and a function $C(x)$ from $\mathbb{R}$ to $\mathbb{R}$ such that

$$\|C\|_{Lip[\mathbb{R}]} \leq 1.$$

The covariance matrices of the mixtures $\{F^i_{\text{in}}\}$, Σ_i, $i = 1 \ldots, I_{\text{in}}$, are 2×2 diagonal matrices with the standard deviations of each mixture on the diagonal. That is, for diagonal element j of the ith mixture, $(\Sigma_i)_{j,j} = s_i$, $i = 1, \ldots, I_{\text{in}}$. The local means of the mixtures F^i_{in}, $i = 1, \ldots, I_{\text{in}}$ are $(x_i, \underline{C}(x_i))$, $i = 1, \ldots, I_{\text{in}}$. The distribution F_{out} is a mixture of the distributions F^i_{out}, $i = 1, \ldots, I_{\text{in}}$ depending on the parameters μ_i, $i = 1, \ldots, I_{\text{in}}$. Each F^i_{out} has mean $(x_i - \mu_i, \underline{C}(x_i))$. They have the same covariance matrix as the distributions F^i_{in}, $i = 1, \ldots, I_{\text{in}}$. Outliers are more "separable" commensurate with the magnitude of μ_i, $i = 1, \ldots, I_{\text{in}}$. We assume that

$$F = \varepsilon \cdot F_{\text{out}} + (1 - \varepsilon) \cdot F_{\text{in}};$$

We go through the trouble of sampling from a mixture of distributions on a curve to guarantee non-constant variance and mean. What's more, the means of the mixture distributions are sufficiently close together to insure an overlap of points. Figure 3.3is a depiction of the two-dimensional data for increasing values of ε.

3.1.1 Comparison of Methods for Finding the Curve

We compare our estimate for the local mean of the stable distribution, the curve C, with the estimate of algorithms of robust regression, when using the two-dimensional synthetic data described above. We measure the quality of both estimation and approximation by the Mean Sum of Square Error (MSSE), which we describe later, for different values of the contamination parameter ε. For higher values of ε, we increase the number of points sampled from the deviating set ("outliers") and reduce those sampled from the stable set. The robust regression methods we have used to

Fraction of outliers = 1%

method	MSC	loess	robust loess	lowess	robust lowess
CPU	10.6653	18.6969	114.5447	18.1962	110.3387
MSE	11.6235	12.7337	12.7253	28.0561	41.5421

Fraction of outliers = 5%

method	MSC	loess	robust loess	lowess	robust lowess
CPU	10.7455	18.4666	124.3288	19.4780	112.3816
MSE	11.4295	17.1251	13.3365	31.3259	40.4673

Fraction of outliers = 10%

method	MSC	loess	robust loess	lowess	robust lowess
CPU	10.3849	18.8371	115.0054	20.2391	121.6549
MSE	13.4478	24.7447	13.1854	36.6408	39.6045

Fraction of outliers = 20%

method	MSC	loess	robust loess	lowess	robust lowess
CPU	9.8442	19.1075	116.4374	19.0774	118.3101
MSE	17.0208	45.8333	13.6805	53.1524	36.4649

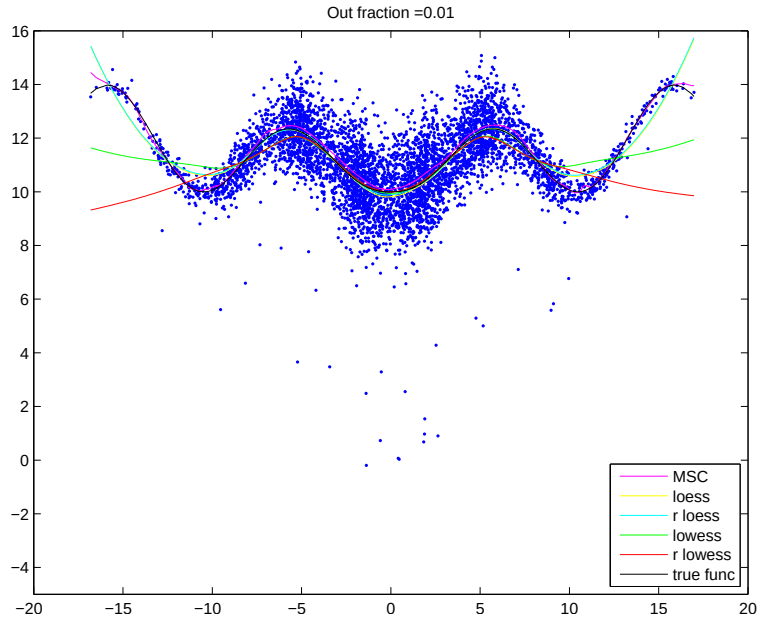
Fraction of outliers = 30%

method	MSC	loess	robust loess	lowess	robust lowess
CPU	7.2705	18.8872	115.5461	18.9773	117.4789
MSE	14.3879	66.3734	17.0366	71.3420	35.5004

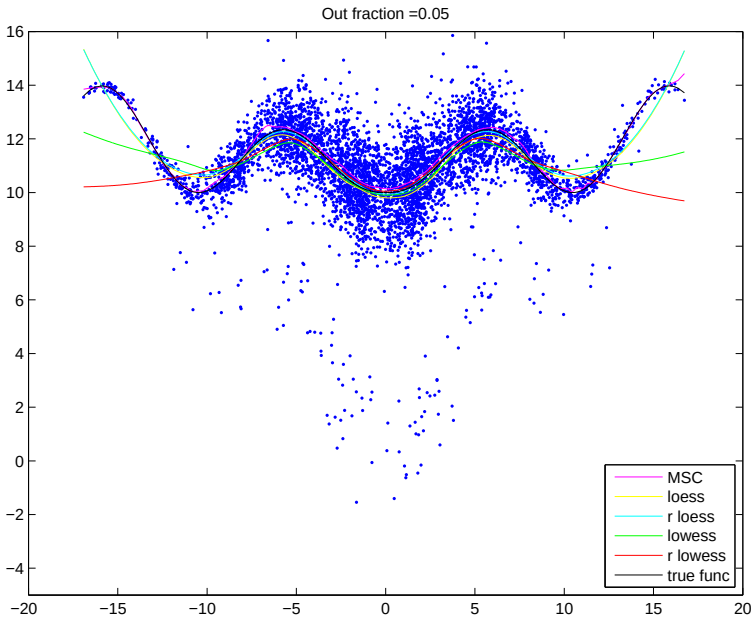
Fraction of outliers = 40%

method	MSC	loess	robust loess	lowess	robust lowess
CPU	6.1188	20.0288	118.5705	21.0102	123.1871
MSE	23.1956	89.0137	16.8871	93.2435	34.9951

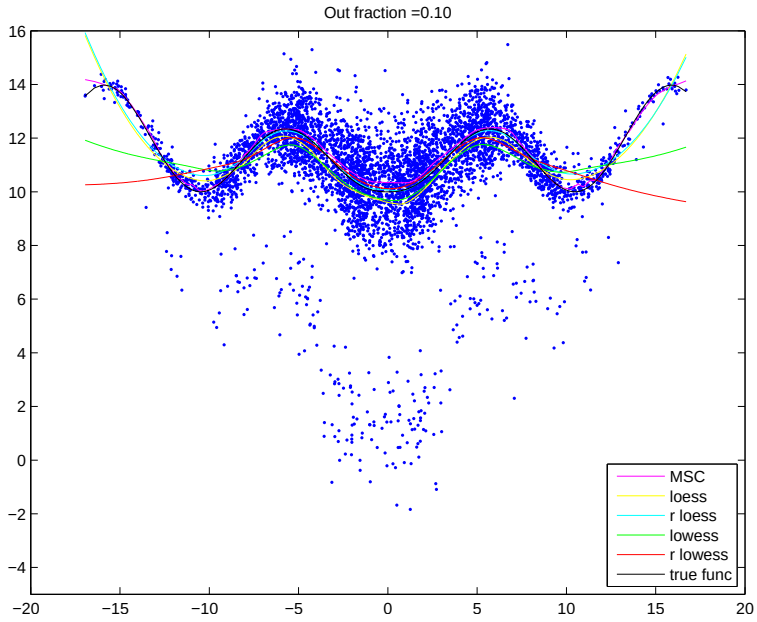
Table 3.1: Comparison of MSC curve and loess, robust loess, lowess and robust lowess curves for synthetically created data. CPU time in seconds recorded in Matlab® using an Intel Pentium M 1600MHz processor with 2 GB of RAM



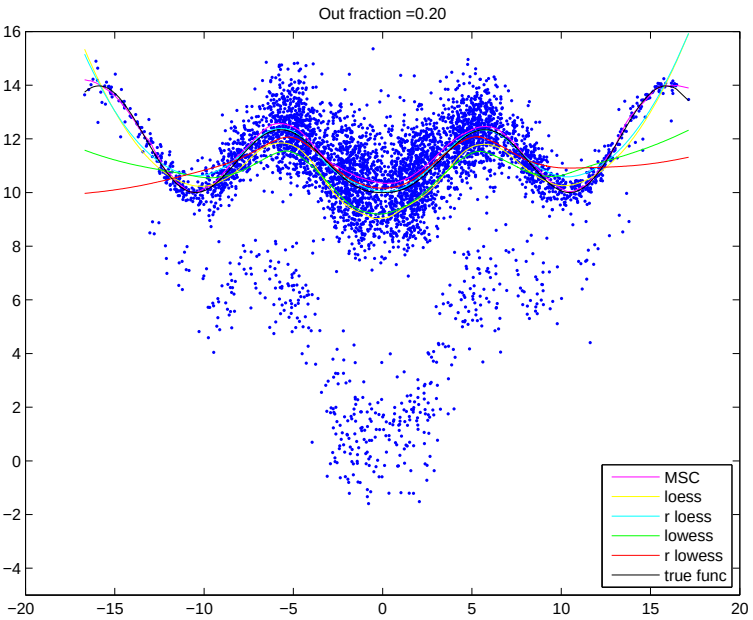
(a) Fraction of outliers = %1



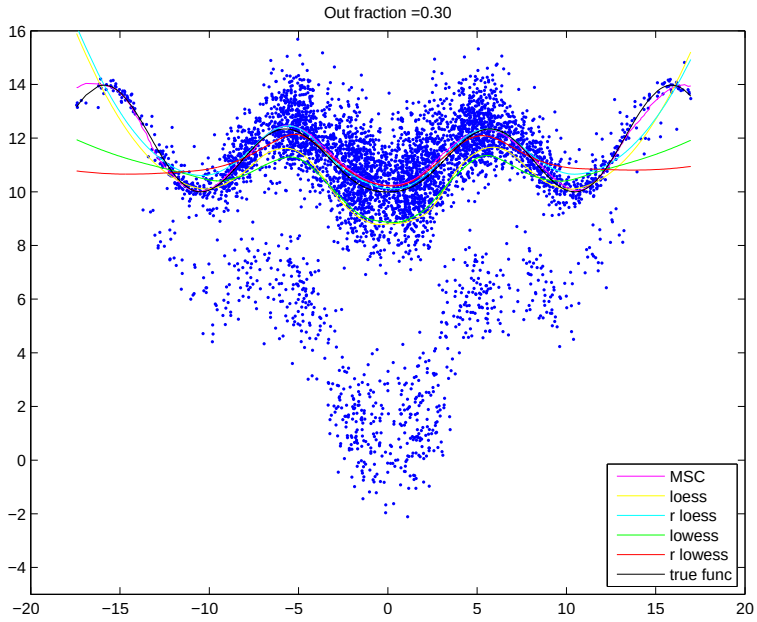
(b) Fraction of outliers = %5



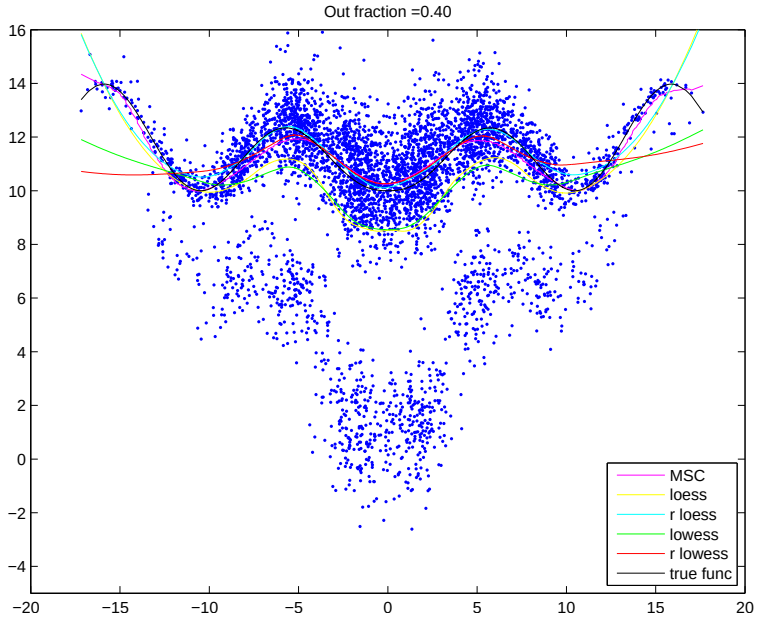
(c) Fraction of outliers = %10



(d) Fraction of outliers = %20

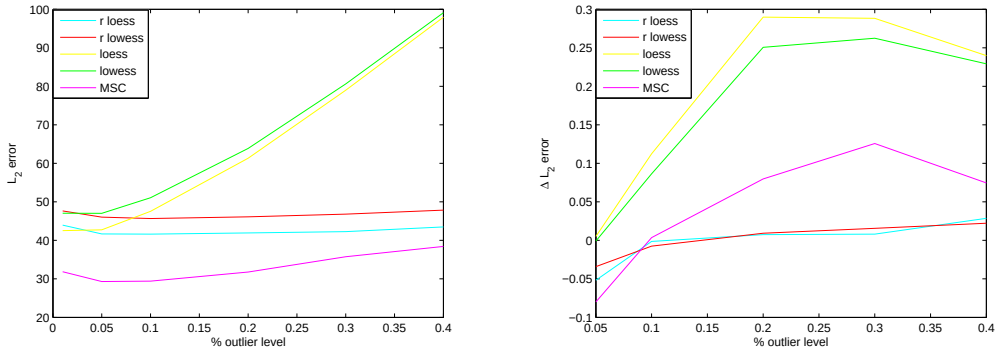


(e) Fraction of outliers = %30



(f) Fraction of outliers = %40

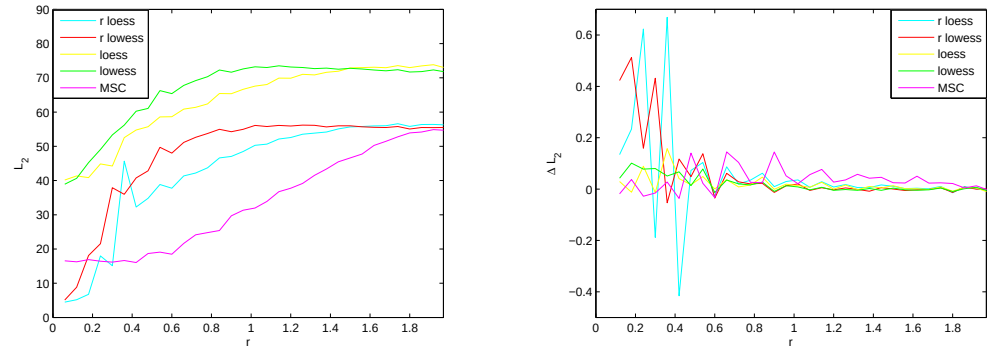
Figure 3.1: Figures 3.1(a)-(f) Illustration of increasing fraction of outliers and its effect on the estimation of the curve $C(X)$ in light blue, hashed line.



(a) L_2 norm error for 5 methods

(b) ΔL_2 norm error for 5 methods

Figure 3.2: Illustration of increasing fraction of outliers and its effect on the estimation of the curve estimations after 32 iterations using 32 sets of artificially created data according $r_i = 3 * \frac{i}{50}$, $i = 1, ...32$. The curve is $f(x) = 10 + \sqrt{x}\sin^2(rx)$



(a) L_2 norm error for 5 methods

(b) ΔL_2 norm error for 5 methods

Figure 3.3: Illustration of increasing "wiggliness" (identified by parameter r) and its effect on the estimation of the curve estimations of 5 methods

compare with our MSC are ©Matlab's *lowess, rlowess* and *loess, rloess* methods of its SMOOTH function. Explanations of these and other kernel methods can be found in [37].

The goal of this exercise was to demonstrate that, given data with multiple mixtures and location dependent, asymmetric noise, the MSC constructed curve is decidedly better than the non-robust local smoothers and comparable to the robust smoothers. That said, the computation expense of local robust smoothers is dramatically higher. Conventional implementations of scatter plot smoothers are $O(N^2h)$, where h is the kernel span [71]. The MSC algorithm is $O(N \cdot \ell_0 \cdot n_{sh})$. Robust versions used in figures 3.1 perform several re-weighted iterations at each point, adding a (possibly) hefty coefficient onto the number of flops. See tables 3.1.

In cases where the outliers are asymmetric or their distribution varies significantly with their location (as in Figure 3.1), then our algorithm will be preferable. We also expect that in this particular example of an underlying cubic polynomial fitting by local cubic polynomials with higher degree of smoothing as described in Remark 2.7.7 may slightly decrease the MSSE of MSC.

We have chosen two-dimensional data in the latter analysis to compare the MSC algorithm against conventional methods of fitting and smoothing. Of course, the virtue of the MSC algorithm is that the data need not be two-dimensional. We may create D-dimensional data analogous to the two-dimensional prescription. In our current version of the algorithm, we have assumed that the local covariance matrices are scalar matrices obtaining the values of the strip function on the diagonal elements. Clearly, one can generalize such an assumption. High dimensional data, including a demonstration using pixel data is more thoroughly explored in [46].

Chapter 4

Bioinformatics: ChIP-on-chip and cDNA Microarray data

4.1 Introduction

Microarray analysis is a high-throughput method to measure abundance of multiple species of target DNA by simultaneous hybridization to an array of DNA probes. When the target DNA is cDNA corresponding to gene expression, it measures the transcriptomic state of a cell under an experimental condition. When the target DNA is sampled from genomic DNA, it measures copy-number variations within a genome as population polymorphisms or as chromosomal aberrations in a tumor genome [58]. Finally, when the target is genomic DNA selected by immunoprecipitation (IP) with a protein, it identifies those regions of genome that interact with proteins, such as transcriptional factors, thus elucidating regulatory genetic controls [63].

All these applications use comparative methods. In a "two-color" scheme, simultaneous array hybridization detects target DNAs of two different experiments, which are labelled with different fluorescent dyes. Target DNAs that have differential behavior from one experiment to the other are called "enriched," the objects sought after in these high-throughput experiments. Enriched targets are found in two steps: First, the measurements are transformed through a "normalization" step in order to assign similar local means (or medians) to "presumed unenriched" targets.

The purpose of this step is to compensate for experimental sources of variation, like dye-specific effects and hybridization unevenness in DNA arrays [72; 12]. Next, the normalized data is used as a basis for statistical identification of enriched targets that truly differ between the two analyzed DNA samples.

In practice, the targets are measured optically in terms of a raw intensity value, and analyzed after being transformed by a logarithmic function. With just two samples involved, the following transformed data representation is standard: for every target, the logarithms of intensities (according to the two samples) are averaged to create an A value (log of their geometric mean) and subtracted to create an M value (log of their ratio), and then plotted in an M vs. A plot. The majority of targets will belong to a stable unenriched set of targets, and after correct normalization their M values will be close to zero. The normalized M values of the enriched targets will lie either significantly above or below zero, but not necessarily with any known distributions, or even symmetry.

This section describes the use of MSC for normalization of microarray data and identification of enriched targets without assuming any prior distribution. Its utility is greatest when the data are difficult to model statistically, for instance, when they contain unavoidable distortion and asymmetry. Such problems are most acute for ChIP-on-chip experimental data.

ChIP-on-chip experiments combine microarrays ("chips") with Chromatin immunoprecipitation (ChIP) assays to identify the genomic loci bound by any given transcription factor [63; 8; 77]. The strucutre of the chromatin fiber at the promoter of a gene is thought to induce the gene's expression and effect the transcription process. The immunoprecipitated sample represents gene fragments attached to the transcription factor, and is compared with a sample not subjected to immunoprecipitation and thus representing all genes equally ("input" sample). The two samples are labelled with different fluorescent dyes and are co-hybridized onto a DNA microarray

representing all gene promoters of the particular species examined. Those spots that show a significant increase in fluorescence in IP sample relative to the input sample are termed enriched spots, and are considered to represent the target genes bound by the transcription factor in the cell nucleus. The goal of so-called ChIP-on-chip experiments is to elucidate the fuzzy picture of transcriptional regulatory networks, whereby tnrascription factors activate responses (expression) from other transcription factors, conveying the signal and, in turn, activating the expression of additional target genes. [9] make the example of the eukaryotic cell cycle where E2F proteins relay extracellular signals to activate genetic expression which modifies *cell cycle effectors* such as cyclin or CKDs and enzymes crucial to biosynthesis or DNA replication. Whatever the case, defining the genetic loci of the transcription factor is one step, but an important one, in the overall picture.

There are currently well-established methods for normalization and identification that work well for many cDNA array data sets [60; 72] ([25; 39; 49; 80]). But, in ChIP-on-chip experiments, enriched target DNA segments deviate in only one direction. That is, the M values (log ratio of input to IP signals) of enriched sites are mostly negative. In this case, the statistical distribution of the corresponding M values is asymmetric, hard to model and thus difficult to estimate. Moreover, in such data the whole M values are frequently skewed, their observed distribution varies locally, and the dependence of their local means on the A values is nonlinear. Consequently, common probabilistic techniques have been difficult to adapt to data arising from ChIP-on-chip experiments [12].

Occasionally, one also encounters cDNA array data with asymmetric distribution of expression values. For example, the sex-biased genes of *D. melanogaster* constitute a subpopulation whose expression values deviate asymmetrically from the bulk of expression values [56].

Three algorithms for ChIP-on-chip experimental data have been developed

very recently. Two of them: ChIPOTle [13] and Mpeak [81] take into account the "neighbor effect", observed in experiments using locus tiling arrays, to improve the identification of targets. However, those methods cannot be used with microarrays, where genes are represented by only one spot. Another method Chipper [31] applies to microarrays. The MSC shows better performance of our algorithm over the latter method for a specific data set.

We perceive the microarray data as arising from a mixture of two distributions, not necessarily parametric. The first one (the "stable" part) represents unenriched targets. In the M vs. A plots introduced earlier, this part is concentrated along a graph of an unknown function f mapping A (mean log-intensity) to M (difference in log-intensities): $M = f(A)+$noise. In the ideal case of perfect correlation between the two intensities, the function f is zero, and the noise is symmetric and homoscedastic (its local variance is independent of A). In reality, the graph is frequently curved due to the systematic sources of variation discussed above, and the local variances around the normalizing curve are not necessarily constant (namely, the data is heteroscedastic). The second component of the mixture distribution represents outliers (enriched targets). The goal of our algorithm is to estimate the conditional mean and variance of the "stable" distribution, ignoring the presence of outliers.

We refer to our algorithm as Multiscale Strip Construction (MSC), as it constructs a normalizing curve (the estimated conditional mean) with an enveloping strip around it (the estimated conditional variance) in a multiscale fashion. The algorithm zooms in adaptively on the "stable" part of the data by constructing parallelograms of decreasing sizes, centered and oriented along the underlying curve. Higher dimensional generalizations of the algorithm for different kinds of data appear elsewhere [46].

Our MSC algorithm performs well on various ChIP-on-chip and cDNA array data, even in problematic cases of significant outliers and strong asymmetries, skewness and curvature of main curve.

4.1.1 Types of Arrays

Proximal Promoters

Arrays fabricated with printed PCR products usually less than 1kb around the transcription start site are called proximal promoter or promoter-specific arrays. They were first used in the pioneering work done on yeast. Due to the simple genome structure of *S. cerevisiae* the promoters were relatively straightforward to design because the ('intergenic') promoter regions were easy to predict and quite short compared to eukaryotes. Amplification of longer PCR products tends to be unstable and often fails. Thus, apart from the issue of designing a probe, there is usually an inescapable rate of failure. The failure of PCR reactions translates to missing spots on the microarray which in turn obfuscates a precise picture and analysis. See [9] for a thorough discussion of the merits of proximal promoters.

Tiling Arrays

Alternatives to printed PCR products involve oligonucleotide arrays who's attractiveness lies in avoiding the necessity to design primers and modeling amplification error. On so-called *genome-tiling* arrays, probes represent equidistant genomic sequences which cover large portions of the genome. With dense enough tiling, the level of finding transcription factors is increased. A "sliding-window" approach is used by [13]. The intensity of the signal should increase until there is an exact match of the probe at the start sight of the transcription factor which should, ideally, then yield the highest signal intensity.

4.2 Algorithm and Methods

We provide a simplified description of the algorithm and leave its more detailed elaborations and analysis to a mathematical paper [46].

The main input to our algorithm is a planar data set E of N points. The algorithm identifies a "stable" set within that data and estimates its local mean and standard deviations. It also uses those estimates in order to assess "outliers" coming from a different model.

In order to simplify the technical description, we assume that the data is normalized along the x-axis so that the M values of the data remain constant at 0. We also assume that the "unenriched" ("stable") part of the data is distributed symmetrically around the x-axis. In this case, the algorithm only estimates the local standard deviations of the "stable" set. We refer to this ideal case as the linear-symmetric case, or LS-Case, and explain its generalization later.

In some cases (e.g. the artificial data exemplifying the algorithm in the supplementary material) the task of the algorithm is well-studied. Indeed, it can be addressed by robust estimation of local means (robust regression) and local standard deviations. However, in many cases, in particular, ChIP-on-chip data, the local percentages of outliers are significantly high, in particular larger than 50% in some regions, and their distribution is asymmetric and hard to model. In order to overcome this obstacle, we suggest a stopping procedure which separates significant "outliers" in a multiscale fashion and forms local regions that cover the "stable" set (excluding those "outliers"). In each local region standard techniques could then be applied to estimate local statistics and then use it to reassess the significance of outliers coming from a different distribution. (See figure 2.1(d) for the asymmetric case.)

The algorithm depends on the following parameters: l_0, c_0, n_0, n_{sh} and α_0. However, the choices for their values are not arbitrary; the optimal values are selected

in a manner to be briefly discussed later, and elaborated further in [46].

4.2.1 Ranking and Identification of Outliers

In order to rank and identify enriched targets, we define a scoring function R for a point (A,M) as follows:

$$R = \begin{cases} \frac{|M-C(A)|}{\widehat{S}(A)} & \text{for cDNA arrays} \\ \max(\frac{-(M-C(A))}{\widehat{S}(A)}, 0) & \text{for ChIP-on-chip .} \end{cases}$$

Initial p-values are obtained from those scores by assuming that the stable distribution is normal. That is,

$$p\text{-val}(A, M) = 1 - \text{ erf } \left(\frac{R}{\sqrt{2}} \right) .$$

Following [62; 4; 27], we have adjusted the p-values in order to control the false discovery rate of the multiple testing procedure. That is, given a false discovery rate level q, we order the computed p-values: $p_{(1)} \leq \ldots \leq p_{(N)}$ and set

$$p^* = p\text{-value}(\max\{i : \ p_{(i)} \leq q \cdot \frac{i}{N}\}). \tag{4.1}$$

We identify the points with p-values less than or equal p^* as enriched.

Choice of Parameters

We fix the values of the following parameters: $l_0 := 10$, $n_0 := 30$, $n_{sh} := 30$ and c_0 is the minimal constant (or almost minimal) for which $E \subseteq \widetilde{Q}$.

The parameter α_0 is crucial for good performance of the algorithm. It describes the global expected percentage of outliers. We have developed an algorithm for estimating this parameter [46, Section 4.8]. The main idea is to apply the MSC

algorithm with different values of α_0 and identify outliers at different fixed levels of FDR. For each value of α_0, we draw the curve of the number of outliers detected by the algorithm as a function of the FDR level. We then observe the jumps between the curves. We expect that the values of α_0 at the jumps reflect cumulative percentages of subgroups of outliers. We have verified this assumption on artificial data sets obeying our outlier model 2.5.

Our ChIP-on-chip data has exhibited several jumps at a full range of values (see Figures), which may reflect local changes in densities (in addition to sublayers of outliers). We thus choose the value of α_0 according to the first significant jump in the profile curves, so that it corresponds to separating the first significant subgroup of outliers. More precisely, we observe the median differences (among 3 replicates) in numbers of detected oultiers and choose the first jump of those, which is usually significant enough to recognize. In cases of ambiguity of first significant jump of a given replicate, we choose the one closest to the median jump. Nevertheless, we show later that our results are not too sensitive to the choice of α_0, but are optimal with the choices suggested by our method.

4.2.2 Complexity for Expression Data

In practice, the CPU time of our algorithm (written in a Matlab code which was not optimized) was 1.11 seconds when computing C and the strip $\widetilde{S}$ and using a data with $N = 5823$ points and a laptop with Intel Pentium processor of 1.60 GHZ and 1 Gigabyte of RAM (the data is replicate A of Myogenin ChIP-on-chip described in Subsection 4.3.2). When also computing the strip $\widehat{S}$, the CPU time was 7.96 seconds. For comparision, the CPU times of LOESS using the same data and pc with the bandwidths parameters 0.1, 0.3 and 0.7 are 8.54, 15.28 and 28.05 seconds respectively.

Clearly, the use of $\widetilde{S}$ instead of $\widehat{S}$ reduces considerably the computational time. Our experience shows that for values of α_0 less than 0.2, the differences between the two functions are not significant.

4.3 Case Studies

We demonstrate results of our algorithm for both cDNA gene expression and ChIP-on-chip data with emphasis on the latter.

4.3.1 *C. acetobutylicum* Gene Expression Data

Using our algorithm, we have analyzed cDNA array data comparing gene expression of megaplasmid pSOL1 deficient *C. acetobutylicum* strain M5 relative to its wild type (WT) strain [80]. The pSOL1 genes are postulated to have expression with a broad range of levels in WT, but unexpressed in M5. Therefore, these genes were expected to be characterized as enriched in the WT strain versus the M5 strain.

To measure the statistical power of our algorithm, we focused on the following quantities: the false positive rate (FPR), the true positive rate (TPR) and the identification error (E_r), all described in [80].

[80] have used the same data in order to compare various algorithms for identification of differentially expressed genes, including their own algorithm: SNN-LERM (segmental nearest neighbor method of logarithmic expression ratios). They concluded that their algorithm performed better than the other algorithms.

We have compared FPR, TPR and E_r of both MSC and SNN-LERM for the six glass arrays of M5 vs. WT in the supplemental material of[80]. We maintained a similar ratio of outliers and summarized our findings in Table 4.1.

The results indicate better identification by MSC in four out of the six ex-

Numerical Results	Slide 422	Slide 424	Slide 783	Slide 784	Slide 786	Slide 805
SNN						
TPR	0.093	0.087	0.059	0.257	0.202	0.176
FPR	0.089	0.073	0.069	0.046	0.057	0.058
E_r	0.498	0.493	0.505	0.394	0.427	0.441
MSC						
TPR	0.11	0.10	0.059	0.236	0.21	0.191
FPR	0.085	0.07	0.069	0.05	0.055	0.055
E_r	0.488	0.484	0.505	0.407	0.423	0.432

Table 4.1: Comparison of SNN-LERM and MSC for identification of C. acetobutylicum pSOL1 genes in six slides of M5 vs. WT experiments (using data where SNN-LERM was shown to be superior to other methods [80]).

periments, though the magnitude of improvement is arguably small. In view of the superiority of SNN-LERM over other existing algorithms for this particular data (as claimed by [80]), we find our results noteworthy.

4.3.2 Mouse DNA microarray from ChIP-on-chip experiment

We have performed ChIP-on-chip experiments using the Mm4.7k mouse promoter DNA microarray, with highly specific antibodies against well-characterized transcription factors. The experiments have been replicated three times. A detailed biological analysis of these experiments is published elsewhere [8].

In the main data described here, the antibodies recognized Myogenin and the experiment was performed in myotubes. In the supplemental material of [47] we have also analyzed ChIP-on-chip experiments where antibodies recognized MyoD in both growing cells and myotubes. Following [8] we have excluded any experiment with more than one replicate with dust speckles on the glass slide or with low spot

fluorescence intensity (65% with respect to background).

With an aim to independently validate observed binding of a transcription factor to a given genomic locus, we have performed confirmatory, gene-specific PCR on immunoprecipitated chromatin in the special case of the Myogenin data. This is a method that does not involve DNA amplification, DNA labeling and microarray hybridization, which are the most prominent sources of error and noise in the ChIP-on-chip procedure. We have chosen microarray spots from different levels of binding ratios. Thirty-five tested genes were determined as unambiguously enriched (and thus considered true targets of the transcription factor under study), while thirty-five were considered unenriched (original data for this comparison is described in [8] and also provided in the supplemental material of this paper).

We have compared the MSC with the binding ratio method (BR) as applied to this data in [8], binding ratios together with LOESS normalization, binding ratios with respect to the principal axis of the data and the recent Chipper algorithm [31]. The binding ratio method identifies enriched sites by selecting (according to a subjective threshold) the points with highest ratios of IP signal to input signal (equivalently, lowest M values). BR can be combined with LOESS by initial application of LOESS normalization and then identifying enriched sites by selecting points with lowest second coordinates. Similarly, BR can be applied with respect to the principal axis by shifting the data so its center of mass is zero, rotating it so the x-axis coincides with the main principal axis and then applying BR. Other approaches for normalization and identification of cDNA arrays yielded even less compelling results than the four methods and are thus not presented here.

A comparison along a full range of true positive and negative rates is described by a ROC curve, the area under which, conveys the algorithms ability to distinguish true positives while limiting false positives. The ranks of the various methods are averaged among un-excluded replicates and their sorted values are used for identifica-

Method	MSC	BR	LOESS+BR	PCA+BR	Chipper
Area	0.913	0.895	0.905	.0.891	0.888

Table 4.2: Areas below ROC curves for the different methods. The LOESS span parameter, 0.3, has been chosen to maximize its area. The MSC parameter α_0 has been chosen according to first significant jumps (see supplemental material in [47]).

α_0	0.1	0.21	0.23	0.26	0.3	0.4	0.5
Area	0.900	0.913	0.915	0.913	0.909	0.907	0.902

Table 4.3: Areas below ROC curves for MSC with different values of α_0.

tion. The areas below the curves for the different methods are recorded in Table 4.2. We have chosen carefully the LOESS span parameter to maximize its area below the ROC curve. In both instances, MSC performs slightly better than the three other algorithm over a full range of false positive rates. However, only 1.19% of the data has been verified to be either enriched or unenriched and therefore the differences between the methods (in particular LOESS and MSC) are not statistically significant (using the methods of [22], noting the area under the ROC is a Mann-Whitney statistic whose distribution converges to a Gaussian). We nevertheless find those results important as we are not aware of ChIP-on-chip experimental data with larger percentage of confirmatory PCRs (it is a time-consuming and an expensive process).

While the ROC curve describes a full range of true positive and negative rates, in practice, there is a specific range which is important for identification. We identify such a range by controlling the FDR level. We have chosen a level of 0.1. In the lack of clear model in some of the other methods, we have maintained the same number of identified enriched points for them (combining all 3 replicates by a weighted score) for the purpose of comparing identification rates. MSC has identified correctly Chrnb1,

Identification for MSC without transformation (FDR 0.1):

α_0 :	0.05	0.1	0.15	0.2	α_0^*	0.25	0.3
TP	0	1	2	18	22	24	28
TN	35	35	35	35	19	34	28

Identification for BR (same % as MSC without transformation):

α_0 :	0.05	0.1	0.15	0.2	α_0^*	0.25	0.3
TP	0	2	3	16	35	25	32
TN	35	35	34	32	33	30	28

Table 4.4: True positives (TP) and true negatives (TN) out of 35 enriched and unenriched confirmed targets for regular MSC (with initial shift and rotation onto main principal axis) and compared with BR with same percentage of identified targets. α_0^* represent the parameters chosen for the three replicates by our jump method (0.2,0.21 and 0.2).

Chrng, Cited2 and Myc as enriched, whereas BR misidentified them. However, BR has identified correctly Sema6c as enriched whereas MSC missed it. BR has falsely identified Hist1h2bc and Cacng1 as enriched, unlike MSC. MSC true positive rate is 0.629, whereas that of BR is 0.542. MSC false positive rate is 0, whereas that of BR is 0.057.

Optimal performance of MSC depends on a correct choice of the parameter α_0 and our algorithm for detecting such a choice is a distinctive advantage of our method. Nevertheless, MSC is not highly sensitive to variations in the parameter α_0. Table 4.3 (see also Table S1) illustrates this point, by recording areas below ROC curve for different values of α_0; the variations in areas is not significant and the optimal area is near our choice of the optimal parameter. Similarly, in Table 4.1 we record identification of true and false positives and negatives for different values of α_0. We have maintained an FDR level of 0.1 and have compared the performance with BR having same percentage of detected enriched points. Tables S9-S13 include

Identification for MSC without transformation (FDR 0.1):

α_0 :	0.05	α_0^*	0.1	0.15	0.2	0.25	0.3
TP	18	24	24	28	31	32	33
TN	34	34	34	32	28	25	25

Identification for BR (same % as MSC without transformation):

α_0 :	0.05	α_0^*	0.1	0.15	0.2	0.25	0.3
TP	16	22	22	26	30	33	33
TN	34	32	32	30	28	26	25

Table 4.5: True positives (TP) and true negatives (TN) out of 35 enriched and unenriched confirmed targets for MSC without initial shift and rotation onto main principal axis and compared with BR with same percentage of identified targets. α_0^* represent the parameters chosen for the three replicates by our jump method (0.1,0.11 and 0.07).

additional values of α_0 and identification of other methods as well.

Our application of MSC includes an initial rotation on principal axis. We have compared it to BR with respect to this axis in order to show that the intial transformation is not enough for good identification (it is worse or at least comparable to regular BR). When applying our method without this initial transformation and choosing α_0 by our method the area below the ROC is 0.904, but it decreases with higher values of α_0. The parameter α_0 has been picked successfully by our method so the differences between the areas under the ROC curve are not significant for those choices. For other values, differences are more significant, but only due to a single spot: Cacng1 which is falsely idenfied by MSC without initial rotation as enriched with very low false positive rate. Our conclusion is that applying MSC without initial rotation can also work well in identifying outliers. It is more convenient to plot the resulted curve and strip, as there is no need to rotate backward. Differences of the two implementations have also been compared for the MyoD data as well [47]. The

Method	MSC	Chipper
Area	0.8514	0.7986

Table 4.6: Areas below ROC curves for MSC and Chipper using only the second and third replicates of the Sko1 data.

good performance of the MSC method independently of the initial transformation is a strong point of its robustness. On the other hand, LOESS did not perform as well when rotated on the principal axis (e.g. its area under ROC is .896).

4.3.3 Yeast DNA microarray from ChIP-on-chip experiment

Proft et al [59] study effects of the TF Sko1 in yeast to examine the expression of genes induced by osmotic stress. Sko1 binds a number of promoters for genes involved in the mitigation of osmotic stress. We compare the now familiar ChIpper algorithm [31] to results using MSC. The first subplot in Figure 4.1 has almost the same number of spots above $\underline{\widetilde{C}}$ as below, indicating poor amplification and/or hybridization. In fact, the ratio of the number of points above the curve to that below the curve is 0.93. If we perform a signed rank test on the data after removing the curve, it is not significantly different from zero at $\alpha = 0.05$ while the other 2 replicates are. The latter two subplots are slightly improved and we concentrate on these for the analysis.

We use the signed test [32] to determine the probability that, given the median of $X - \underline{\widetilde{C}}$ is zero, the probability that we might see a D this large is in the first row of table 4.7. A p-value greater than 0.05 indicates that there was poor hybridization in the experiment. Replicate 1 in 4.1 can be seen to demonstrate poor hybridization. As a result of using only replicates 2 and 3, the area under the ROC is 0.8514 for the MSC algorithm and 0.7986 for the Chipper algorithm, table 4.6.

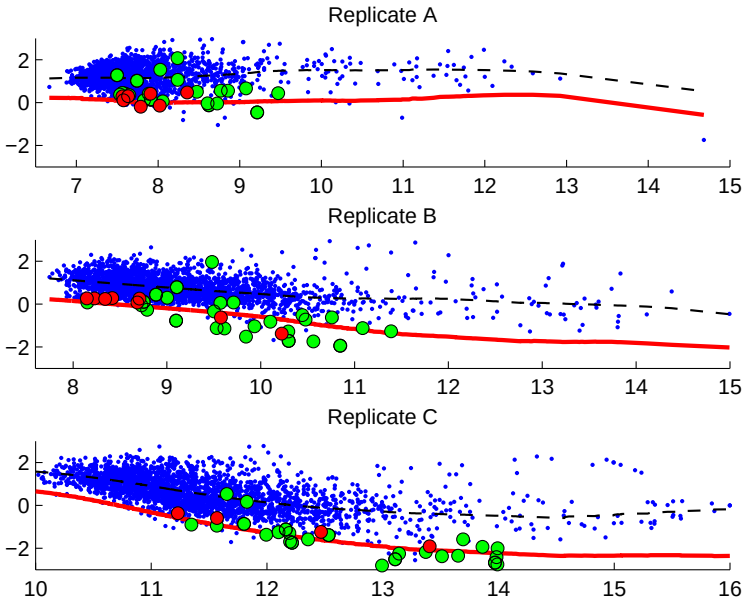


Figure 4.1: MA plots of replicate data using the transcription factor SKO1

Replicate	1	2	3
p-value	0.057	$< 1^{-7}$	$< 1^{-7}$
$\frac{\lvert\{x\colon\ x>\widetilde{C}\}\rvert}{\lvert\{x\colon\ x<\widetilde{C}\}\rvert}$	0.93	0.85	0.82
MSC AUROC	0.4	0.65	0.91

Table 4.7: P-values for signed test (Wilcoxen signed-test) that, differences have mean and median of zero given outcome; in replicate 1, we do not reject the test at a threshold of 0.05

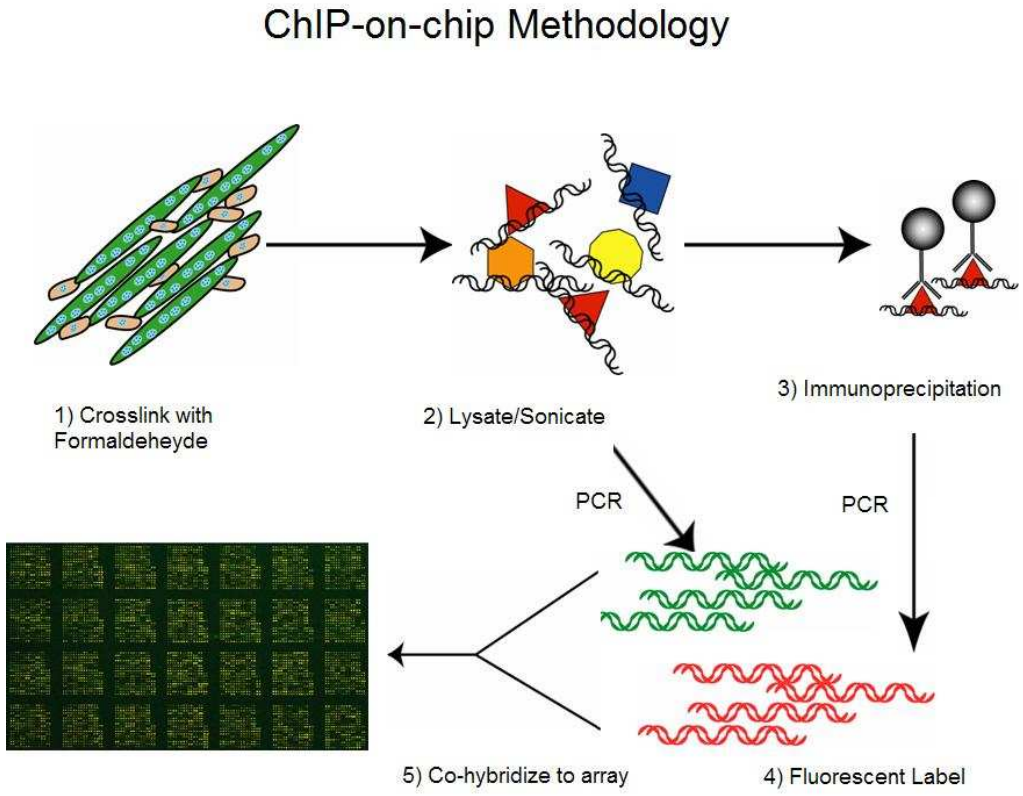


Figure 4.2: Chip-on-chip process for proximal promoters: **1** Formaldyhide is added to DNA to form DNA-protein and protein-protein crosslinks; **2** the new material is lysated or sonicated to shear into roughly equal 1kb specimens; **3** protein-specific antibody is added (immunoprecipitation) in half the specimen; **4** the other half is tagged with Cy65 (green) dye markers and immunopreciptated material is tagged with the Cy35 (red) dye; **5** specimen is co-hybridized to the species specific microarray

4.3.4 Note on Rank Invariance

When analyzing Immunopreciptated data, unlike cDNA, it is hard to determine the possible number of enriched genes [12]. It is possible to adapt the rank invariant methodology used in [76]. [68] uses a variation of this adapted to allow for different more high intensity spots and less low intensity spots. Since we assume only one-sided enrichment we can use the difference in ranks of the control channel versus the IP (immunoprecipitated) channel. Roughly, the algorithm works by determining a subset of genes who's ranked difference is below some threshold multiplied by the number in the subset genes. The algorithm is repeated until the number of rank invariant genes does not change. The fist subset is equal to the entire set at the first iteration. This method can used as another way of determining way of determining α_0. Using the rank invariant method to determine the set used in normalization has serious flaws, however, as it doesn't account for the local variance of the intensities (e.g. in many experiments, high intensity values have relatively less variance than low intensity values). Additionally, we have to come up with a satisfactory rank invariance parameter, essentially exchanging the determination of one parameter for another.

4.4 Conclusion

We believe that our experimental results provide clear indication of the attractive performance of the MSC method, as it allows investigators to extract more meaningful and reliable information from their data sets. See supplemental material in [47] to for instances of failures of some standard techniques to the latter data. There are several marked advantages of our algorithm: its adaptability to regions of lower or higher variance and to areas of the data which exhibit significant nonlinearity between

channels; its model for identifying enriched points under a fixed false discovery rate; its fast implementation; its robustness to transformation of the data and to change of parameters and its ability to choose the main parameter α_0 to improve the identification results. In the ideal case when there is no nonlinearity between channels and the variance is relatively constant throughout the data, we expect MSC to perform similarly to the binding ratio method. However, MSC proves its effectiveness in analyzing many important data sets, because one is frequently confronted with experimental results that stray far from the ideal, as numerous types of artifacts (unequal dye incorporation, unequal background in the two channels, different quantum yield of the dyes, etc) remain difficult to control and confound the analysis in the presence of the inherent asymmetry of ChIP-on-chip experiments.

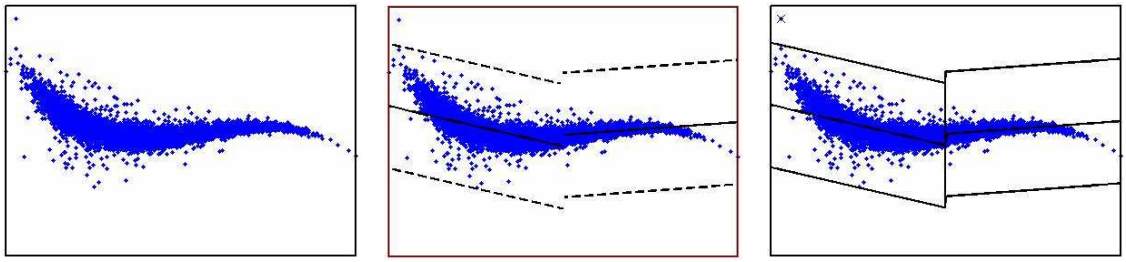
The approach described here fills in a substantial void in the analysis of general DNA arrays, in particular arrays from ChIP-on-chip experiments. Namely, it represents an effective method for identifying enriched targets while handling logarithmic ratios of intensities with asymmetric and heteroscedastic characteristics. Currently, most standard techniques fail to analyze a large fraction of these data and many investigators resort to the simple binding ratio method in order to rank "outliers" (e.g. IP-enriched sites in ChIP-on-chip experiments).

The MSC will prove most advantageous for ChIP-on-chip data sets that display mild or pronounced non-linearity, as well as for data sets where the proportion of enriched spots is very large. However, when working on data sets that are close to the ideal, it still performs as well as other existing methods.

4.5 APPENDIX

Figures 4.1 (a)-(i) through 4.7 (a)-(i) below demonstrate the MSC algorithm on symmetric data along a curve. Listed are the appropriate estimates for determining the stopping region: $T(\widetilde{Q})$, $\widetilde{Q}$, f_Q, and F_Q. This is intended only for a visual appreciation

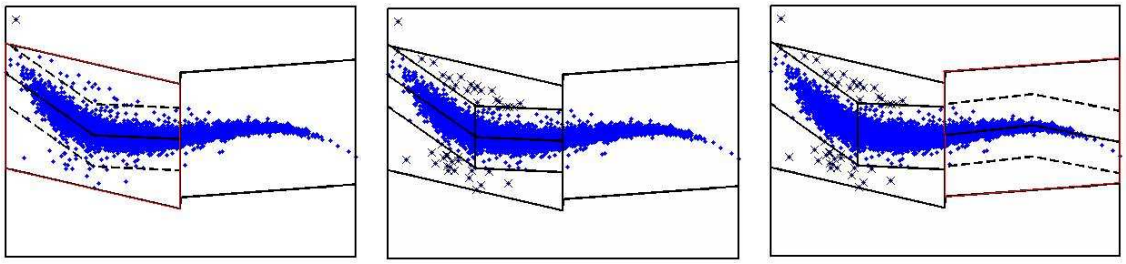
of how the MSC algorithm proceeds, in a recursive manner, removing points from the local estimation given a point's location vis a vis the dyadic region. The solid black lines reference to the regions, the hashed lines represent the potential regions at the next level, and the black crosses represent points excluded from estimation. If, at any interval, at any level, the algorithm proceeds to the next level, the hashed lines become solid lines, etc.



(a) $|T(\widetilde{Q})| = 0$, $|\widetilde{Q}| = 10,000$, $f_Q = 0.00$, $F_Q = 0.00$

(b) Level 0: $|T(\widetilde{Q})| = 1$, $|\widetilde{Q}| = 10,000$, $f_Q = 0.0001$, $F_Q = 0.0001$

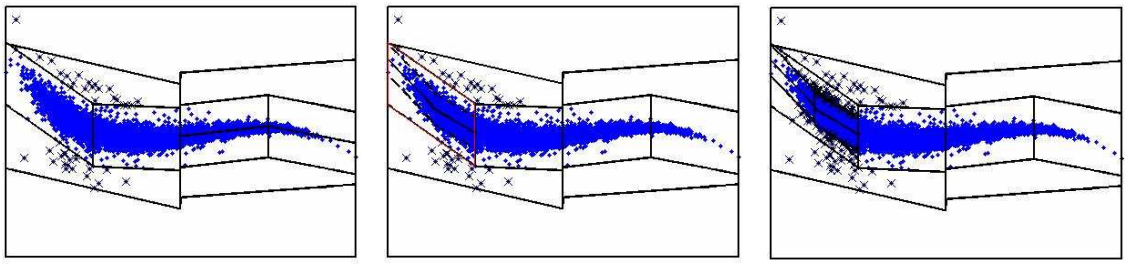
(c) Level 0: $|T(\widetilde{Q})| = 1$, $|\widetilde{Q}| = 10,000$, $f_Q = 0.0001$, $F_Q = 0.0001$



(d) Level 1: $|T(\widetilde{Q})| = 42$, $|\widetilde{Q}| = 7220$, $f_Q = 0.006$, $F_Q = 0.006$

(e) Level 1: $|T(\widetilde{Q})| = 42$, $|\widetilde{Q}| = 7220$, $f_Q = 0.006$, $F_Q = 0.006$

(f) Level 1: $|T(\widetilde{Q})| = 0$, $|\widetilde{Q}| = 2780$, $f_Q = 0.00$, $F_Q = 0.00$

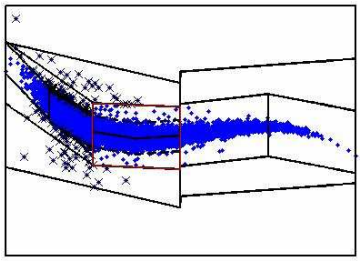


(g) Level 1: $|T(\widetilde{Q})| = 0$, $|\widetilde{Q}| = 2780$, $f_Q = 0.00$, $F_Q = 0.00$

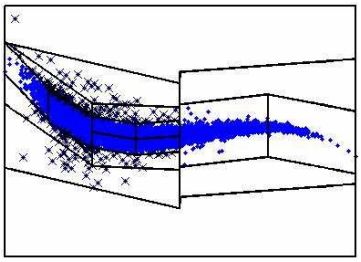
(h) Level 2: $|T(\widetilde{Q})| = 106$, $|\widetilde{Q}| = 2074$, $f_Q = 0.051$, $F_Q = 0.057$

(i) Level 2: $|T(\widetilde{Q})| = 106$, $|\widetilde{Q}| = 2074$, $f_Q = 0.051$, $F_Q = 0.057$

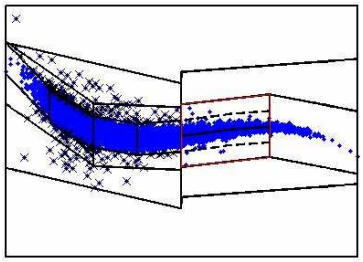
Figure 4.3: Representation of regions constructed for levels 0 to 2



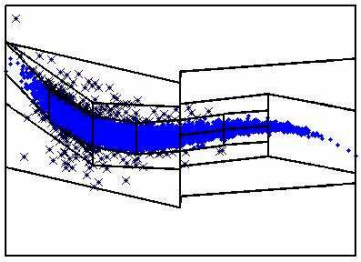
(a) Level 2: $|T(\widetilde{Q})| = 57$, $|\widetilde{Q}| = 5146$, $f_Q = 0.011$, $F_Q = 0.017$



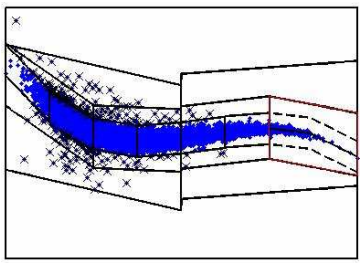
(b) Level 2: $|T(\widetilde{Q})| = 57$, $|\widetilde{Q}| = 5146$, $f_Q = 0.011$, $F_Q = 0.017$



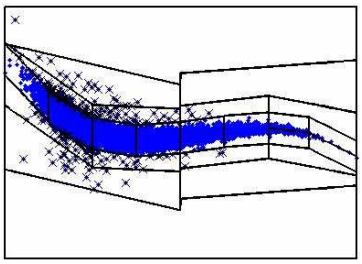
(c) Level 2: $|T(\widetilde{Q})| = 8$, $|\widetilde{Q}| = 2509$, $f_Q = 0.003$, $F_Q = 0.003$



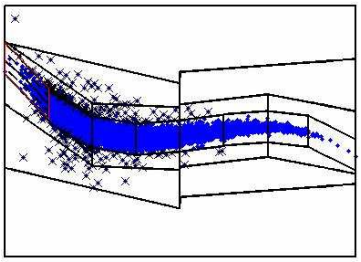
(d) Level 2: $|T(\widetilde{Q})| = 8$, $|\widetilde{Q}| = 2509$, $f_Q = 0.003$, $F_Q = 0.003$



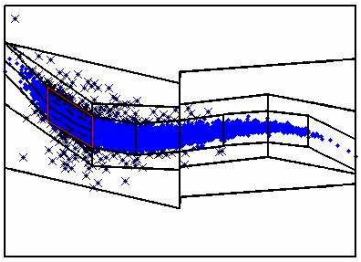
(e) Level 2: $|T(\widetilde{Q})| = 0$, $|\widetilde{Q}| = 271$, $f_Q = 0.00$, $F_Q = 0.00$



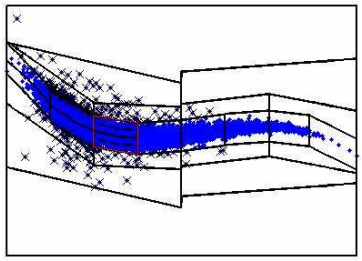
(f) Level 2: $|T(\widetilde{Q})| = 0$, $|\widetilde{Q}| = 271$, $f_Q = 0.00$, $F_Q = 0.00$



(g) Level 3: $|T(\widetilde{Q})| = 81$, $|\widetilde{Q}| = 285$, $f_Q = 0.284$, $F_Q = 0.341$

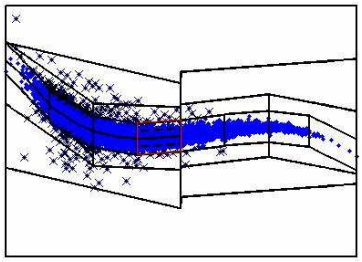


(h) Level 3: $|T(\widetilde{Q})| = 480$, $|\widetilde{Q}| = 1789$, $f_Q = 0.268$, $F_Q = 0.325$

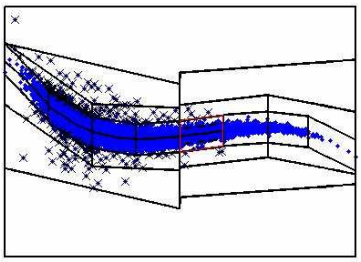


(i) Level 3: $|T(\widetilde{Q})| = 471$, $|\widetilde{Q}| = 2656$, $f_Q = 0.177$, $F_Q = 0.194$

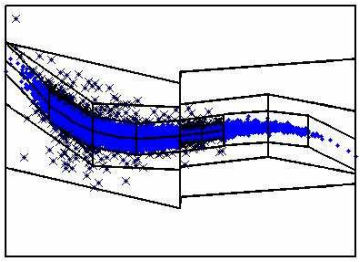
Figure 4.4: Representation of regions constructed for levels 2 to 3



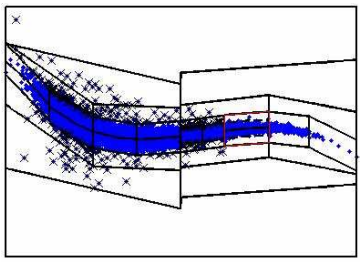
(a) Level 3: $|T(\widetilde{Q})| = 201$, $|\widetilde{Q}| = 2490$, $f_Q = 0.081$, $F_Q = 0.098$



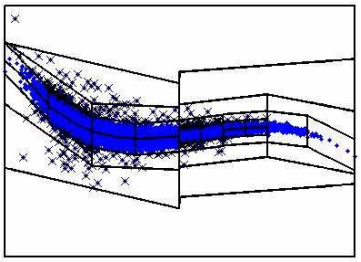
(b) Level 3: $|T(\widetilde{Q})| = 63$, $|\widetilde{Q}| = 1688$, $f_Q = 0.037$, $F_Q = 0.041$



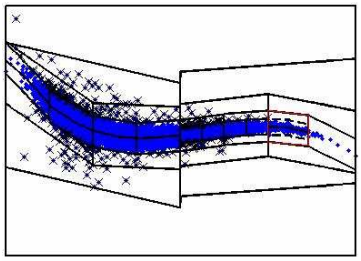
(c) Level 3: $|T(\widetilde{Q})| = 63$, $|\widetilde{Q}| = 1688$, $f_Q = 0.037$, $F_Q = 0.041$



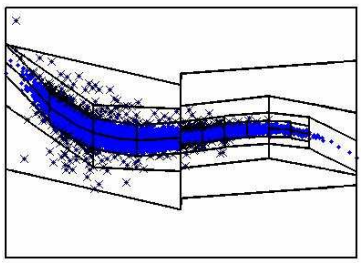
(d) Level 3: $|T(\widetilde{Q})| = 11$, $|\widetilde{Q}| = 821$, $f_Q = 0.013$, $F_Q = 0.017$



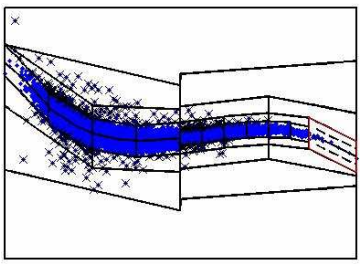
(e) Level 3: $|T(\widetilde{Q})| = 11$, $|\widetilde{Q}| = 821$, $f_Q = 0.013$, $F_Q = 0.017$



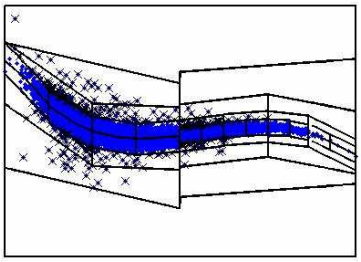
(f) Level 3: $|T(\widetilde{Q})| = 0$, $|\widetilde{Q}| = 260$, $f_Q = 0.00$, $F_Q = 0.00$



(g) Level 3: $|T(\widetilde{Q})| = 0$, $|\widetilde{Q}| = 260$, $f_Q = 0.00$, $F_Q = 0.00$

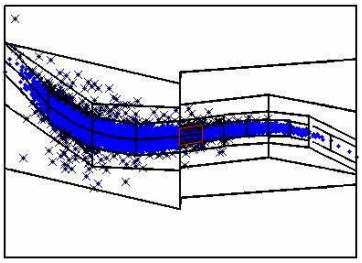


(h) Level 3: $|T(\widetilde{Q})| = 0$, $|\widetilde{Q}| = 11$, $f_Q = 0.00$, $F_Q = 0.00$

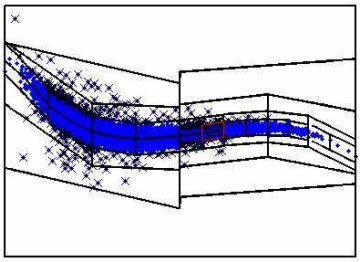


(i) Level 3: $|T(\widetilde{Q})| = 0$, $|\widetilde{Q}| = 11$, $f_Q = 0.00$, $F_Q = 0.00$

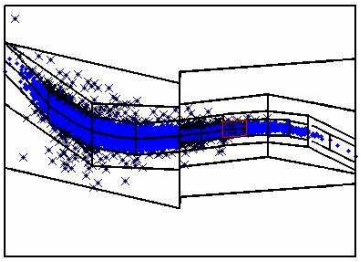
Figure 4.5: Representation of regions constructed for level 3



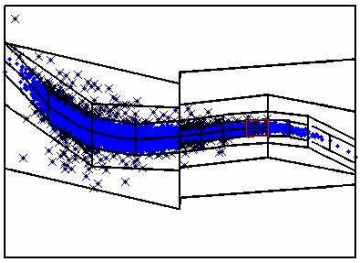
(a) Level 4: $|T(\widetilde{Q})| = 232$, $|\widetilde{Q}| = 942$, $f_Q = 0.246$, $F_Q = 0.287$



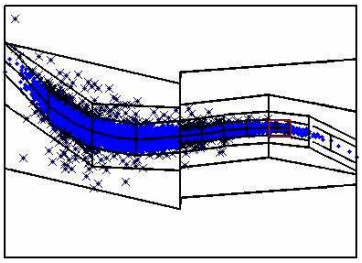
(b) Level 4: $|T(\widetilde{Q})| = 135$, $|\widetilde{Q}| = 746$, $f_Q = 181$, $F_Q = 0.222$



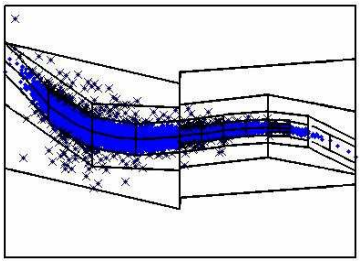
(c) Level 4: $|T(\widetilde{Q})| = 74$, $|\widetilde{Q}| = 522$, $f_Q = 0.142$, $F_Q = 0.158$



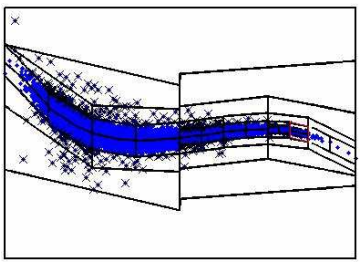
(d) Level 4: $|T(\widetilde{Q})| = 35$, $|\widetilde{Q}| = 299$, $f_Q = 0.117$, $F_Q = 0.134$



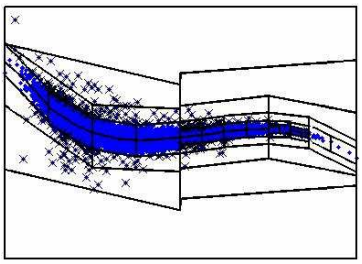
(e) Level 4: $|T(\widetilde{Q})| = 9$, $|\widetilde{Q}| = 190$, $f_Q = 0.047$, $F_Q = 0.047$



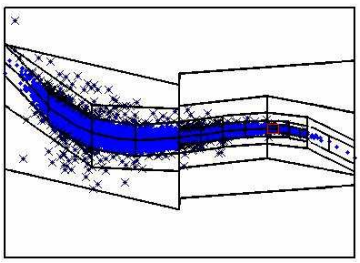
(f) Level 4: $|T(\widetilde{Q})| = 9$, $|\widetilde{Q}| = 190$, $f_Q = 0.047$, $F_Q = 0.047$



(g) Level 4: $|T(\widetilde{Q})| = 2$, $|\widetilde{Q}| = 70$, $f_Q = 0.029$, $F_Q = 0.029$

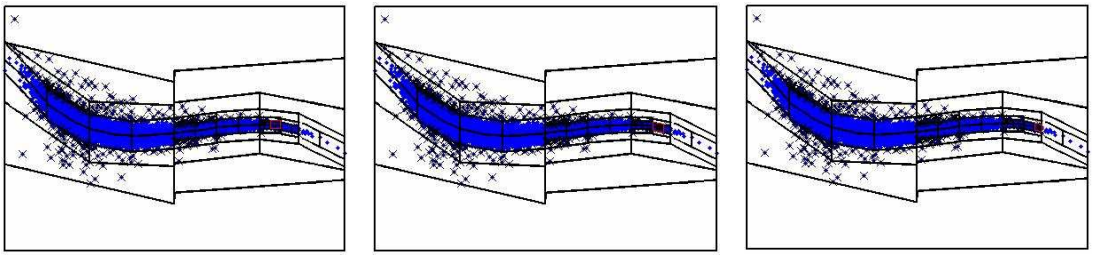


(h) Level 4: $|T(\widetilde{Q})| = 2$, $|\widetilde{Q}| = 70$, $f_Q = 0.029$, $F_Q = 0.029$



(i) Level 5: $|T(\widetilde{Q})| = 2$, $|\widetilde{Q}| = 70$, $f_Q = 0.029$, $F_Q = 0.029$

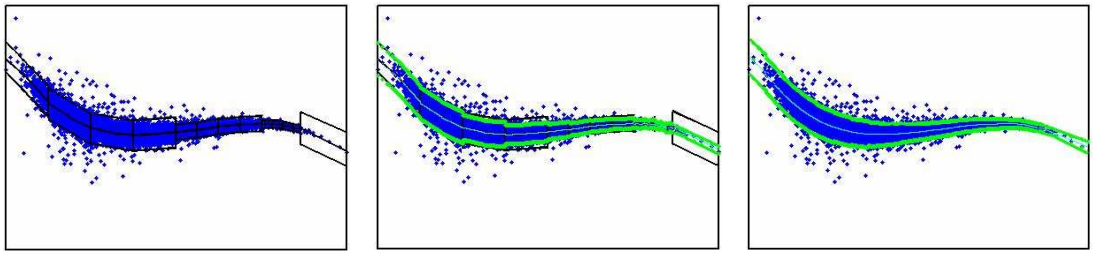
Figure 4.6: Representation of regions constructed for levels 3 to 5



(a) Level 5: $|T(\widetilde{Q})| = 0$, $|\widetilde{Q}| = 8$, $f_Q = 0.00$, $F_Q = 0.00$

(b) Level 5: $|T(\widetilde{Q})| = 0$, $|\widetilde{Q}| = 8$, $f_Q = 0.00$, $F_Q = 0.00$

(c) Level 5: $|T(\widetilde{Q})| = 36$, $|\widetilde{Q}| = 114$, $f_Q = 0.316$, $F_Q = 0.363$



(d) Exhausted cubes $\mathcal{Q}$

(e) Exhausted cubes $\mathcal{Q}$, curve $\underline{\widetilde{C}}$, and strips $\widetilde{S}$

(f) Smoothed $\underline{\widetilde{C}}$ and $\widetilde{S}$ for 5 shifts

Figure 4.7: Representation of regions constructed for level 5

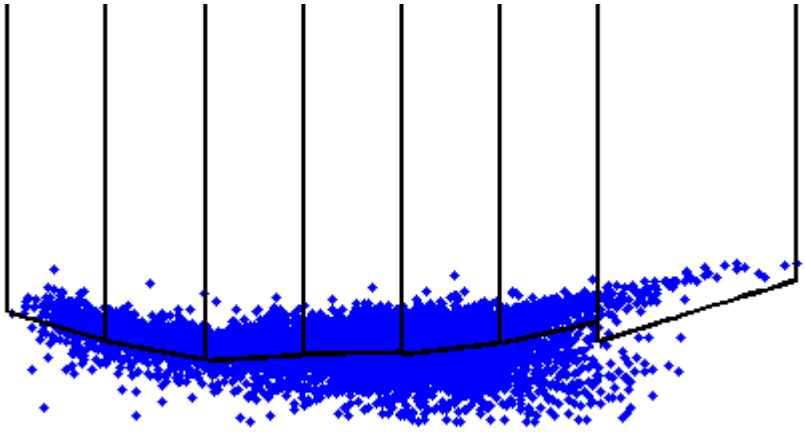


Figure 4.8: Non-symmetric regions ($\widetilde{Q}$) associated with stopping intervals for a ChIP-on-chip data ($\alpha_0 = 0.4$, $n_0 = 20$).

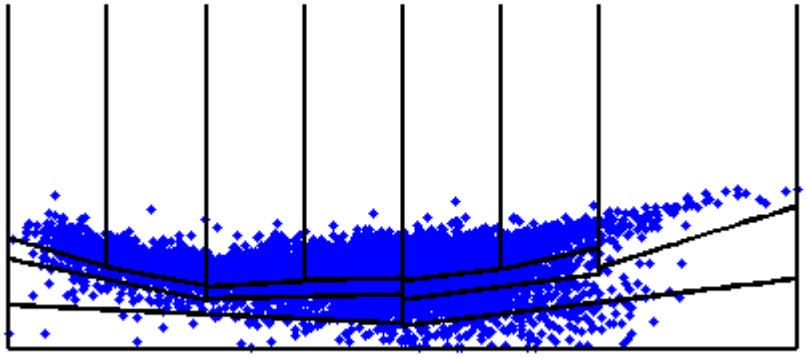


Figure 4.9: Nonsymmetric regions for ChIP-on-chip data associated with all dyadic intervals containing (and including) stopping intervals (same data and parameters as above).

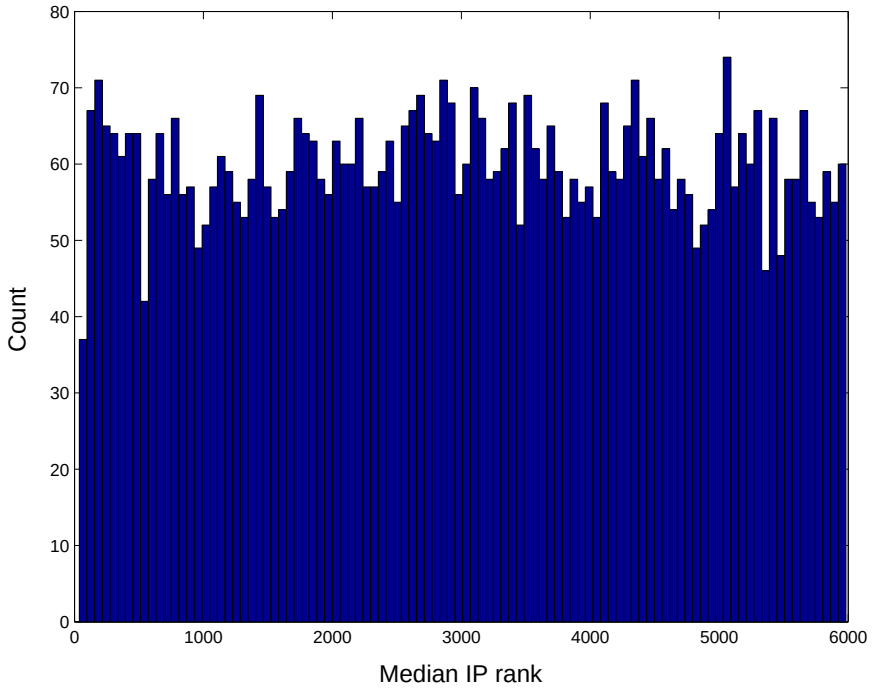


Figure 4.10: Median Rank histogram for 3 Replicates of previous ChIP-on-chip data.

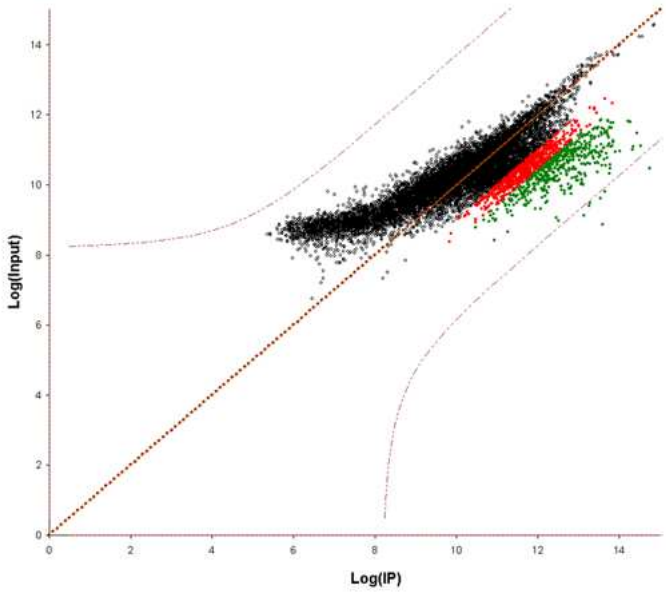


Figure 4.11: Rosetta model applied to previous data. The parameter F=0.691 was used (same as in human Chip-Chip). Purple dotted curve corresponds to p-value = 0.01 (curve p-value = 0.001 is far). Lower values of F shrinks the strip (p-value curve) towards the center line. Data is presented on log intensities' coordinates and $y = x$ is the normalizing line. The wrong shape of the strip is due to both wrong normalization but also wrong parametric assumptions of the Rosetta model. Spots with binding ratio greater than 3 are in green and greater than 2 are in red.

Bibliography

[1] R. Bellman. (1961) "Adaptive Control Processes: A Guided Tour" Princeton University Press.

[2] M. Belkin. (1990) *Problems of Learning on Manifolds* PhD. Thesis, August 2003, University of Chicago.

[3] M. Belkin, M. Niyogi. (2003) "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, 15(6):1373-1396.

[4] Y. Benjamini, Y. Hochberg. (2003) "Controlling the false discovery rate: A practical and powerful approach to multiple testing," *J. Roy. Stat. Soc. Ser. B*, **57** 289-300.

[5] C. Bregler, S.M. Omohundro. (1995) "Nonlinear manifold learning for visual speech recognition," Fifth International Conference on Computer Vision.

[6] M. W. Bern, D. Eppstein, et al. Emerging challenges in computational topology. xxx.lanl.gov e-print archive, September 1999.

[7] C. J. Bishop, P. W. Jones. (1994) Harmonic measure, L^2 estimates and the Schwarzian derivative. *J. Anal. Math.*, 62:77–113, 1994.

[8] A. Blaise, M. Tskilis, D. Acosta, R. Sharan, Y. Kluger, and B. Dynlacht (2005) "An initial blueprint for myogenic differentiation," *Genes and Development*, 19:553–69.

[9] A. Blaise, and B. Dynlacht (2005) "Devising transcriptional regulatory networks operating during the cell cycle and differentiation using ChIP-on-chip," *Chromosome Research*, 13:275–88.

[10] A. Blais. NYU Cancer Institute, division of NYU Medical center (April, 2004) personal communication.

[11] B. Bolstad, R. Izarry, M. Åstrand, T. Speed. (2003) "A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Variance and Bias". *Bioinformatics*, **19**:185-193.

[12] M. J. Buck, J. D. Lieb. (2004) "ChIP-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments" *Genomics*, **83**:349–360.

[13] M. J. Buck, B. Nobel, J. D. Lieb. (2005) "ChIPOTLe: a user friendly tool for the analysis of ChIP-chip data" *Genome Biology*, **6**(11):R97.

[14] E. Arias-Castro, D. Donoho, X. Huo (2005)"Near Optimal Detection of Geometric Objects by Fast Multiscale Methods" *IEEE Transactions on Information Theory* **51** 7, 2402-24

[15] E. Arias-Castro, D. Donoho, X. Huo (2006)"Adaptive multiscale detection of filametary structures in a background of uniform random points" *The Annals of Statistics* **34** 1, 326-24

[16] A. Calderon, A. Zygmund. (1952) "On the existence of certain singular integrals." *Acta Math.* 88, 85-139.

[17] G. Casella, and R. Berger. (2002) *Statstical Inference*, Duxbury.

[18] M. Charikark, S. Khuller, D. Mount, G. Narasimhan. (2001) "Algorithms for facility location problems with outliers", SODA '01: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms,pp 642 - 651.

[19] M. Cook, S. Weisburg. (1982) *Residuals and Influence in Regression*, New York: Chapman & Hall.

[20] David, G. Semmes, S.(1991) "Singular integrals and rectifiable sets in $\mathbb{R}^n$: au-delà des graphes Lipschitziens". *Astérisque*, **193**:1–145.

[21] G. David, S. Semmes.(1993). *Analysis of and on uniformly rectifiable sets*, volume 38 of American Mathematical Society, Providence, RI.

[22] E. Delong, D. Delong, and D Clarke-Pearson. (1988) "Comparing the Areas Under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach", *Biometrics* 44, pp.837-45.

[23] M. Dollinger, R. Staudte (1991) "Influence Functions of Iteratively Reweighted Least Squares Estimators," *J. of the Amer. Stat. Assoc*, **86**(415):709-716.

[24] D. Donoho, C. Grimes. (2003) "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data" *Proceedings of the National Academy of Sciences.*, **100**(10): 5591-96.

[25] S. Dudoit, Y.H. Yang, T.P. Speed, and M.J. Callow. (2002) "Statistical Methods for Identifying Differentially Expressed Genes in Replicated CDNA Microarray Experiments." *Statistica Sinica*, **12**(1):111–139.

[26] B. Durbin, J. Harin, D. Hawkins, D. Rocke . (2002) "Variance stabilization transformation for gene-expression microarray data." *Bioinformatics*, **18 Suppl I**S:105-S:110.

[27] B. Efron . (2007) "Size, Power and False Discovery Rates." *The Annals of Statistics*, **35**(4):1351-77.

[28] B. Efron, R. Tibshirani, J. Storey, and V. Tusher. (2001) 'Empirical Bayes Analysis of a Microarray Experiment." *Journal of the American Statistical Association*, **96**:1151–1160.

[29] T. Ferguson. (1998) *Introduction to Large Sample Theory*, Springer.

[30] Y. Ge, S. Dudoit, and T. P. Speed (2003). "Resampling-based multiple testing for microarray data analysis." *TEST*, **12**(1):1–44.

[31] F. Gibbons, M. Proft, K Struhl, and F. Roth "Chipper: discovering transcription-factor targets from chromatin immunoprecipitation microarrays using variance stabilization" *Genome Biology*,**6**(11) (2005)

[32] J. Gibbons (1985)*Nonparametric Statistical Inference*, 2nd edition, M. Dekker.

[33] J. Glimm, D. Sharp "Multiscale Science" *Siam News* (1997)

[34] J. Glimm. SUNY Stony Brook, Dept. of Applied Mathematics and Statistics (January, 2007) personal communication.

[35] J. Ham, D. Lee, S. Mika,B. Schölkopf. (2004) "A kernel view of dimensionality reduction of manifolds" *Proceedings of the Twenty-First International Conference on Machine Learning*, 369-376.

[36] T. Hastie, W. Stuetzle. (1989) "Principle Curves", *Journal of the American Statistical Association*, **84**(406): 502–516.

[37] T. Hastie, R. Tibshirani, J. Friedman. (2001) *The Elements of Statistical Learning; Data Mining, Inference, and Prediction*, Springer.

[38] T. R. Hu. (2000) "Functional discovery via a compendium of expression profiles." *Cell*, **102**(1): 109–126.

[39] W. Huber, A. von Heydebreck, H. Sültmann, A. Poustka, M. Vingron . (2002) "Variance stabilization applied to microarray data calibration and to the quantification of differenctial expression." *Bioinformatics*, **18 Suppl I**S:96-S:104.
[26]

[40] P. W. Jones. (1990) "Rectifiable sets and the traveling salesman problem." *Invent. Math.*, **102**(1): 1–15.

[41] B. Kégl, A. Krzyzak, T. Linder, K. Zeger. (2000) "Learning and design of principal curves", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(3): 281-297.

[42] S. Lafon. (2004) *Diffusion Maps and Geometric Harmonics.* PhD. Thesis (under the direction of R. R. Coifman), May 2004, Yale University.

[43] E. L. Lawler. *The traveling salesman problem.* Wiley-Interscience, New York, 1985.

[44] E.L. Lehmann. (1951) "Consistency and Unbiasedness of Certain Nonparamtric Tests", *Ann. Math. Stat.* pp 165-79.

[45] G. Lerman. (2003) "Quantifying curvelike structures of measures by using L_2 Jones quantities." *Comm. Pure App. Math.*, **56**(9): 1294–1365.

[46] G. Lerman, J. McQuown and B. Mishra. 2007 "Multiscale Curve and Strip Constructions." *under review*

[47] G. Lerman, J. McQuown, A. Blais, B. Dynlacht, G. Chen and B. Mishra. 2006 "Functional Genomics via Multiscale Analysis: Application to Gene Expression and ChIP-on-chip Data." *Bioinformatics* **23**(3):314-320.

[48] G. Lerman. (2000) How to partition a low-dimensional data set into disjoint cluster of different geometric structures. submitted.

[49] C. Li and W. Wong. (2001) "Model-Based Analysis of Oligonucleotide Arrays: Expression Index Computation and Outlier Detection." *Proceedings of the National Academy of Sciences.*, **98**(1): 31–36.

[50] J. Lieb, X. Liu, D. Botstein, P. Brown. (2001) "Promoter-specific Binding of Rap1 Revealed by Genome-wide Maps of Protein-DNA Association." *Nat. Genet.*. **28**:327-334.

[51] J. S. B. Mitchell, A. Blum, P. Chalasani and S. Vempala. *A constant-factor approximation algorithm for the geometric k-MST problem in the plane.* SIAM J. Computing 28(3):771-781, 1999.

[52] S. Mika, B. Schölkopf, R. Williamson, A. Smola "Regularized Principal Manifolds", *Journal of Machine Learning Research*, **1**(3): 179–209

[53] M. Newton, C. Kendziorski, C.S. Richmond, and F.R. Blattner. (2001) "On Differential Variability of Expression Ratios: Improving Statistical Inference about Gene Expression Changes from Microarray Data." *Journal of Computational Biology*, **8**: 37–52.

[54] M. A. Newton, A. Noueiry, D. Sarkar, and P. Ahlquist. (2003). "Detecting differential gene expression with a semiparametric hierarchical mixture method." *Biostatistics*.**5**155-176.

[55] G. Parmigiani and S. Garrett. (2003) *The Analysis of Gene Expression Data*, chapter 16. Springer-Verlag, New York.

[56] (2003) M. Parisi, R. Nuttall, D. Naiman, G. Bouffard, J. Malley, J. Andrews, S. Eastman, B. Oliver. (2003) "Paucity of Genes on the *Drosophila* X Chromosome Showing Male-Biased Expression." *Science*. **299**(5607):697-700.

[57] V. Pestov On the Geometry of Similarity Search: Dimensionality curse and concentration of measure. *Information Processing Letters*, 73:47-51, 2000.

[58] Pollack, J. et al. (1999) "Genome-wide analysis of DNA copy-number changes using cDNA microarrays." *Nature Genetics*. **23**,41-46.

[59] M. Proft, FD. Gibbons, M. Copeland, FP. Roth, K. Struhl. (2005) "Genomewide identification of Sko1 target promoters reveals a regulatory network that operates in response to osmotic stress in *Saccharamyces cerevisiae* ." *Eukaryotic Cell.*. **4**:1343-52.

[60] J. Quackenbush. (2002) "Microarray data normalization and transformation." Nature Genetics. **32**: 496-501.

[61] J.M. Ranz, C.I. Castillo-Davis, C.D. Meiklejohn, and D.L. Hartl. (2003) "Sex-Dependent Gene Expression and Evolution of the *Drosophila* Transcriptome." Science. **300**(5626):1742-1745.

[62] A.Reiner, D. Yekutieli, Y. Benjamini. (2003) "Identifying differentially expressed genes using false discovery rate controlling procedures." *Bioinformatics* **19**(3):368-75.

[63] B. Ren, F. Robert, J. J. Wyrick, O. Aparicio, E. G. Jennings, I. Simon, J. Zeitlinger, J. Schreiber, N. Hannett, E. Kanin, T. L. Volkert, C. J. Wilson, S. P. Bell, and R. A. Young. (2000) "Genome-wide Location and Function of DNA Binding Proteins." *Science*. **290**:2306-2309.

[64] C. Roberts, B. Nelson, M. Marton, R. Stoughton, M. Meyer, H. Bennett, Y. He, H. Dai, W. Walker, T. Hughes, M. Tyers, C. Boone, and S. Friend. (2000) "Signaling and circuitry of multiple MAPK pathways revealed by matrix of global gene expression profiles." *Science* **287**:873-880

[65] S. Roweiss, S. Saul. (2000) "Nonlinear dimensional reduction by locally linear embedding" *Science*, **290**: 2323-2326.

[66] G. Sapiro, F Mémoli (2005) "A Theoretical and Computational Framework for Isometry Invariant Recognition of Point Cloud Data" *Foundations of Computational Mathematics* **5**(3): 313-47.

[67] L. K. Saul and S. T. Roweis. (2003) "Think globally, fit locally: unsupervised learning of low dimensional manifolds." Journal of Machine Learning Research. **4**:119-155.

[68] E. Schadt, C. Li, B. Ellis, and W. Wong. (2001) "Feature Extraction and Normalization Algorithms for High-Density Oligonucleotide Gene Expression Array Data", *J. Cell. Biochemistry Supp.* **37**:120-125.

[69] B. Schölkopf, A. Smola (2002). *Learning with Kernels Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press.

[70] D. Scott (1985) "Averaged Shifted Histograms: Effective Nonparametric Density Estimators in Several Dimensions", *The Annals of Statistics*,**13**:1024-1040

[71] Seifert, B., Brockman, M., Engel,J., Gasser,T. (1994). *Fast algorithms for non-parametric curve estimation*, *J. Computational and Graphical Statistics.* **2**:192-213.

[72] G. K. Smyth, Y. H. Yang, T. P. Speed. (2003) "Statistical issues in microarray data analysis." In: Functional Genomics: Methods and Protocols, M. J. Brownstein and A. B. Khodursky (eds.), Methods in Molecular Biology. **224**:111-136

[73] J. Storey, and R. Tibshirani. (2003) "Statistical significance for genomewide studies." *Proceedings of the National Academy of Sciences*, **100**(16):9440–9445.

[74] J. Tenenbaum, V. de Silva, J. Langford. (2000) "A global geometric framework for nonlinear dimensionality reduction" *Science*, **290**: 2319-2323.

[75] L. Trefethen, D. Bau. (1997) *Numerical Linear Algebra* SIAM.

[76] G.C. Tseng, M.K. Oh, L. Rohlin, J.C. Liao, W.H. Wong,. (2001) "Issues in cDNA microarray analysis: quality filtering, channel normalization, models of variations and assessment of gene effects". *Nucleic Acids Res.*, **29**:2549-2557.

[77] J. van Steensel. (2005) "Mapping of genetic and epigenetic regulatory networks using microarrays" *Nature Genetics*, **37**: S18-S24.

[78] A. van der Vaart. (1998) "Asymptotic Statistics". *Cambridge Press.*

[79] J. Viaclovsky. (2004) "Calderon-Zygmund decompostion technique: Cube decomposition." *MIT open courseware lecture notes.*

[80] H. Yang, H. Haddad, C. Tomas, K. Alsaker, and E.T. Papoutsakis. (2002) "A Segmental Nearest Neighbor Normalization and Gene Identification Method Gives Superior Results for DNA-Array Analysis." *Proc. Natl. Acad. Sci. USA*. **100**(3):1122-1127.

[81] M. Zheng, L. Barrera, Y. Wu and B. Ren. (2005) "ChIP-chip: data, model and analysis," *Biometrics* **63**: 787-96.

[82] G. Zhu, P. T. Spellman, T. Volpe, P. O. Brown, D. Botstein, T. N. Davis, and B. Futcher. (2000) "Two yeast forkhead genes regulate the cell cycle and pseudohyphal growth." *Nature.* **406**:90-4