PHENOMENOLOGICAL APPROACHES TO THE ANALYSIS OF HIGH-THROUGHPUT BIOLOGICAL EXPERIMENTS

BY

SATWIK RAJARAM

B.Sc., St. Xavier's College, 2001

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Physics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2009

COMMITTEE ON FINAL EXAMINATION*

Professor Karin Dahmen, Chair
Professor Yoshitsugu Oono, Advisor
Professor Yann Chemla
Professor Jim Wiss

Abstract

The analysis and interpretation of the large data sets produced by high-throughput biological experiments are among the most important and challenging problems in science today. The goal of this thesis is to demonstrate the utility of a phenomenological approach to the study of statistics in general, with a particular focus on understanding the results of such experiments.

The Renormalization Group (RG) has been a popular tool in physics for the construction of phenomenological theories. RG may be interpreted as a search for stability, and it is in this form that we make use of it here. We show that an RG stability argument (against addition of new data) may be used to 'derive' standard statistical quantifiers such as the mean and variance. The utility of this principle is also demonstrated in the context of more general quantifiers. In particular, we show how it may be used to guide choices in the development of a novel dimensional reduction scheme.

We have proposed a method, called the ICS Survey, that uses these ideas in the realm of multiple experiments. The ICS survey is a data driven method that exploits the differences between experiments by using them to perturb the system and identify stable parts. By doing so, it successfully identifies the dominant processes (in those experiments) and the genes involved in them, thereby solving some of the more vexing problems faced by exploratory dimensional reduction methods. It is also one of the few methods attempting to make use of the information contained in inter-experiment variability.

We have also discussed various methodological issues faced in the analysis of high-throughput experiments. In particular, novel methods for noise removal and for visualization are presented.

To My Parents.

Acknowledgments

First and foremost, I would like to thank my advisor Yoshi Oono. It would require a separate thesis to list out all the things I have learnt from him. So instead, let me just mention a few of his more stellar qualities that I had hoped to imbibe (but couldn't): his phenomenal work ethic, his polymath-like knowledge, his deep insight, his constant questioning of mainstream ideas and his infinite patience. Despite his incredibly busy schedule, Yoshi makes more time for students in a day than most do in a week; he seemed almost embarrassed whenever our meetings ended in less than hour! These long discussions have greatly shaped my outlook to life (in an outside science). At the same time, Yoshi had the wisdom to let me roam free to explore my interests and make my own mistakes. Without this, I would never have been able to graduate from the bookish student I was, to the independent researcher I hope to become. For all of these things I shall be eternally grateful.

I would also like to thank members of our group past and present. Prasanth Sankar and Bojan Tunguz helped me through my first few years of research. Thanks also to James Cook, Jeremy Tan, Adam Knapp, Aruna Rajan and Mehmet Oz for being great group mates.

I must acknowledge my friends: Rahul Biswas, Vikram Jadhao, Parag Ghosh, Nayana Shah and many others in physics. Kim Hubbard, who is one of my favorite people in Champaign (I promise, the fact that she has two lovely cats, and that all my other friends live in Urbana is just a coincidence!). Vignesh Sethuraman and Eric White with whom I was house and office mates respectively for my first couple of years here. If it weren't for all of these people, I might have finished this thesis a long time ago. More likely though, I would never have found the will to stay the course. It is these people who made my time here worthwhile.

I would also like to make a special mention of the Espresso Royale Cafe. I probably spent more time there than in my office. Since it would not be considered a valid affiliation for scientific papers, I am taking this opportunity to express my appreciation.

Finally, I want to acknowledge the role of my parents. They were my inspiration to take up science to begin with, and it has been their constant encouragement and unwavering faith in my abilities that has given me the strength to come this far. Thank you for everything.

This work was supported in part by teaching assistantship from University of Illinois, and summer research assistance from University of Illinois Campus Research Board and the Institute for Genomic Biology.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1 Scope

Biology is the most rapidly evolving scientific discipline today; our present understanding of biological systems is very different from that just a decade ago. Discoveries being made here have more potential to impact lives than those in any other field. Thus, biological systems represent some of the most exciting and important problems in science today. On the other hand, as is typical of evolving fields, there are many different views that are popular, and it is unclear how best to approach them.

From an organizational point of view, perhaps the most natural and useful way to think of biological systems is as a two tier system [4]

$$\text{biology} = \text{structure} + \text{information}.$$

The structural information consists of, for example, arrangements of molecules that are responsible for carrying out various biological functions. The information part is the encoding of the set of instruction that orchestrate the functioning of the biological molecules.

As a concrete example, consider deoxyribonucleic acid (DNA), the molecule that is central to all known life. DNA is made of two polymers, each consisting of a complementary sequence of molecules known as nucleotides. The chemical and physical properties of the molecules determine the structure and possible functions of DNA. The traditional application of physics to biology has involved the study of the structural part, and much insight has been gained in this way.

However, it is the information of the type contained in the specific sequence of nucleotides that allows organisms to reproduce and is ultimately what distinguishes them from other complex collections of molecules. So the structural part may be thought of as a tool used by the informational part. The properties of the structural part do place many limits on what can be achieved, and understanding them is important. Nevertheless, a true understanding of biology will come only from an understanding the organization of the informational part.

It is instructive to make an analogy with computers. The structural part of biology corresponds to the semi-conductor devices that are used to make computers, while the informational part is like the programs running on them. Is an understanding of semi-conductors really important to do computer science? It is true that without semi-conductor science, we would never have had computers, and if a computer stops working then a knowledge of semi-conductor devices may come in handy. However, computers could have been built on a completely different foundation than semiconductors, and the general principles of computer science would been unchanged. Thus, computer science transcends the materials basis on which it was built.

Much the same can be said for the informational and structural parts of biology. The structural molecules are what makes biological systems possible, and many illnesses are probably related to structural defects. Yet, to truly understand the general design principles of biological organisms, a different level of description (namely, the informational one) is what is needed.

Just like with computers, in biology too, the material properties are in the structural part and this has therefore been the focus of physics approaches. The informational portion has so far not received much attention from physicists. It is our hope that this balance will be redressed eventually, and by using some standard methodology from statistical physics, this thesis attempts to take a small step in that direction.

1.A Physics and Phenomenology

To clarify what we mean by applying physics to understanding informational biology, it is worth pondering what it means to do physics. The Webster's English Dictionary defines Physics as "a science that deals with matter and energy and their interactions". All natural objects are made up of matter and energy, so effectively this means every natural science is physics. While this is true in a sense, it is also true that if a physicist and a biologist were to consider the same phenomena, their approach would be very different.

Faced with a complicated system, a physicist would work under the assumption that there are some deep, yet simple, rules governing it. If he/she is a good physicist, he/she would probably not attempt to explain every single detail of the system, and instead would attempt to identify the rules governing meaningful global variables. In short, he/she would attempt a phenomenological description of the system.

A conventional definition of a phenomenological theory would be one "which expresses mathematically the results of observed phenomena without paying detailed attention to their fundamental significance" [5]. This definition describes a typical trait of phenomenological theories, but misses their deeper significance[1]. The true significance of phenomenology stems from the (completely non-trivial) observation that to understand many systems in nature, we do not need to know their microscopic details.

In fact, there are numerous examples of systems that are completely different microscopically, but show great similarity in the relations between certain observables. The goal of the phenomenology is to capture this universal structure. At the same time, even in these theories, microscopics do play a role. Thus, to construct a phenomenological theory we must describe the system at a level where its essence (or universal structure) can be separated from that which is incidental. We shall have more to say about this in Chapter 2.

Many of the great theories in modern physics are phenomenological theories in this sense. For example, consider the Navier-Stokes equation which describes the slow flow of an ordinary

[1] The ideas in this section are very heavily influenced by the lectures of Y. Oono. See for example [6]

liquid:

$$\rho\Big[\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v}\cdot\nabla)\Big]\boldsymbol{v} = \eta\Delta\boldsymbol{v} - \nabla p,$$

where p is the pressure, $\boldsymbol{v}$ is the flow velocity field with an incompressibility condition $\nabla\cdot\boldsymbol{v} = 0$, ρ is the density, and η is the shear viscosity. Here, ρ and η are the phenomenological parameters that are a function of the system details. This equation applies to fluids as diverse as air and molasses, with just a change in these parameters subsuming all the microscopic differences. The form of the equation on the other hand is the universal description we are after.

It should be clear from the above description that to approach the informational part of biology, a phenomenological approach is the only option. The goal of this thesis is to demonstrate the utility of such a phenomenological approach to the study of biological systems in general, and to the analysis of high-throughput gene expression experiments in particular.

1.B The Renormalization Group

To construct a meaningful phenomenological theory it is necessary to be able to identify which (combination of) observables are truly relevant, while ignoring the many other incidental parameters that are just details. This is in general a non-trivial task, often requiring extra-ordinary ingenuity. Thankfully, there is a technique to guide us (provided we have some insight about the system). It is known as the Renormalization Group (RG) approach.

The basic idea of the Renormalization Group is as follows [6]. Since we are looking to construct a phenomenological theory, we must already have a system we are studying at some macroscopic scale, while the 'fundamental' interactions are at a lower microscopic scale. Macro and micro need not refer to length scales, they could just as well be long time vs short time or large number of particles vs small number. Practically speaking, it is impossible to know all the details, making it virtually pointless to construct a theory that depends on them. Therefore, any meaningful theory (or any quantity it is meaningful to study theoretically) cannot depend on details strongly.

RG is simply a way to identify combinations of variables that are insensitive to microscopic details. It is expected that these can now be modeled theoretically. In actual practice in physics, this is achieved in one of at least two ways

1. Wilson-Kadanoff (WK) type RG [7]: By looking at the system from increasing distance (scaling), with a fixed resolution to find the properties that persist.

2. Stückelberg-Petermann (SP) type RG [8]: Shake the system by changing the microscopic details and removal of parts that change dramatically (i.e., the divergences). The remaining part is expected to be universal.

We will make use of a slight variation on the Stückelberg-Petermann type RG, where instead of removing singularities we look for stable parts, i.e., parts that are not affected by changes in microscopic details. These stable parts are expected to reveal meaningful universal relations. This will be the guiding principle of our approach even when the RG machinery is not used explicitly.

2 The Biology

2.A Bioinformatics and Phenomenology

Over the last two decades the barriers to performing high-throughput experiments have fallen dramatically. Thus, experiments that report simultaneously on a very large number (i.e., on the scale of the whole genome) of biological elements are now performed routinely. This has resulted in an enormous amount of experimental data being generated regarding various aspects of biological systems. With the vast amounts of data available, the challenge has shifted from methods of producing data to techniques to interpret it. Information theoretic techniques have been adopted to handle this torrent of data, giving the resulting field of analyzing and interpreting biological data the name Bioinformatics.

Living organisms are the ultimate complex systems. Data generated by them provides a window into this complexity. If we believe that there are simple laws governing this complexity, then it must be the case that simple relations can be extracted from this data. That is, the high dimensional data should really have a low dimensional structure modulo microscopic parameters. In other words, simple phenomenological relations can be created for this data.

From the biological point of view, high-throughput experiments capture information about a huge number of processes running in parallel within the organism. Some of these may be relevant to the particular experiment, but the majority will not be. Thus, there is a natural way to imagine a phenomenological approach that identifies the information relevant to the experiment under consideration and discards the rest.

Currently there are many dimensional reduction methods purportedly attempting to do this. In reality, their actual goals are much more modest, and they do not respect the biology in any way. Not surprisingly, as we shall see, their performance is still quite poor. As novel techniques are being developed there is presently a lack of clarity about the properties that such techniques should possess. In the course of this thesis we shall attempt to make the argument that an RG motivated approach that attempts to identify stable portions (unaffected by microscopic details) can provide a framework for this. Looking even further, it may be even be hoped that RG could unify various dimensional reduction schemes, in the same way that it has unified singular perturbation techniques.

2.B Understanding the Biology

Before we can even thinking about a phenomenological approach, we must know our system, with a clear understanding of measurable quantities. To this end, let us look at the biological system we shall be considering. While most of what follows is applicable in a much broader context, we shall be restricting ourselves to the study of what is known as gene expression, as measured through microarray based experiments. These terms shall be explained shortly.

Protein molecules are among the most important active players in a cell. They are involved in virtually all functions in a cell from acting as enzymes to facilitate other reactions to acting as antibodies to protect the cell. Thus, in many cases, actions that a cell needs performed are achieved by the production of the appropriate proteins. Each protein is made up of units called amino acids, with the sequence of amino acids uniquely defining a protein. The information based on which amino acids should be assembled into a protein is stored biologically as the sequence of nucleotides on the DNA. The parts of the DNA which encode for these proteins are known as protein coding genes. Although only a minority of genes code for proteins (they may also code for types of RNA), for sake of simplicity when we use the word gene it shall refer to a protein coding gene.

Gene expression is the process by which information from a gene is used in the synthesis of a function gene product (such as a protein). It is an involved process, with many subtleties. However, for the purpose of this discussion, the following gross oversimplification captures the significant steps in higher organisms (Eukaryotes):

1. Transcription: First the DNA sequence for the appropriate protein coding gene is read, and an RNA sequence having a complementary code is constructed. This RNA known as precursor messenger RNA (pre-mRNA) now contains the relevant sequence information to build the protein.

2. Splicing: The pre-mRNA sequence contains parts called exons that will be converted into proteins, but also others called introns which will not. The introns are removed, and the exons spliced together to produce mRNA.

3. RNA transport: The mRNA which is synthesized in the nucleus is now transported out into the cytosol through the nuclear pores.

4. Translation: The mRNA finally reaches the ribosome. Here, the mRNA is read sequentially, and one by one the amino acids prescribed by the sequence are brought and assembled together. Once this process is complete we are left with a single molecule of the required protein.

For a more detailed picture see, for example, [9].

While the process of generating a protein from the DNA is comparatively simple, the sequence of steps involved in deciding/detecting which proteins are required at a given point of time, and sending a signal to produce it is far more complex, especially in Eukaryotes.

For example, if a cell is subjected to a stimulus which needs a response, some proteins detect this information and convey it to another protein which might promote the expression (i.e., transcription) of a gene which produces a single protein. This protein could in turn promote or repress other proteins to produce the appropriate response to this stimulus.

What we are typically interested in understanding is the organization of such a signaling/response mechanism involved with some macroscopically observable biological trait e.g., response to heat shock. Since any such macroscopic property is controlled by the functioning of a much larger number of microscopic genes/proteins, this process is hard to decipher. Additionally, individual proteins/genes are usually responsible for very minor functions. These minor functions are re-used by many different processes. Consequently, a particular protein may be produced under multiple different scenarios and may act in concert with different proteins under different conditions. Since a typical measurement captures many processes, identifying the contribution of a particular gene/protein to a specific process is difficult.

The experimental way this problem is approached is to subject the cell to some external stimulus, and study the production of all the proteins, then to identify the proteins that are generated specifically in response to the stimulus. Unfortunately, proteins tend to be unstable, and it is difficult to measure protein expression levels. Therefore, instead of measuring protein levels directly, the expression levels of various genes are measured in terms of the corresponding mRNA levels. Thus, the gene expression/mRNA levels serve as a surrogate for the proteins. By looking at the expression patterns of genes under different circumstances it is hoped that we can infer which genes function together under what circumstances.

When genome scale measurements are required, the measurement of the mRNA levels are most commonly performed using experimental setups known as microarrays. An explanation of the functioning of microarrays can be found in Appendix B. A single microarray experiment can potentially produce a single expression value for the entire set of genes. To extract relations between genes, typically more information is required. Microarray experiments are therefore repeated across other conditions. We shall henceforth refer to the expression values for a single gene across multiple conditions as its expression profile and such a genome-scale collection of expression profiles as an experiment. The conditions could be different points of time, different tissues, cancerous vs non-cancerous, etc.

3 Traditional Approaches

3.A Problems with Traditional Approaches

Typical exploratory methods of analyzing such data attempt to group genes with similar expression profiles together in the expectation that they will be related. Examples of such methods include cluster analyses, principal component analysis, etc. While there is some merit to this procedure, in practice the results have a number of short-comings:

1. No groups of genes stand out, and limited information on the structure of the population can be gleaned from these results. Thus, biological validation is required for the results to be of any use.

2. When the results are biologically validated (for example, by Gene Ontology based annotation of genes), it is rarely found that groups of genes deemed to be close are biologically related. At best, there may be an over-expression for genes of one type. Thus, the predictive power of these results is minimal.

3. Genes known to belong to a single class are often deemed to be unrelated. Optimistically this may be interpreted as implying that there are sub-classes, showing different expression profiles. Unfortunately, this is usually not the case.

4. Lack of reproducibility: If the entire experiment was performed again, and analyzed in the same way, often the genes which are deemed to be near and far change.

3.B Reasons for Problems

The reasons for the poor performance of these methods are as follows:

1. Co-expression$\neq$ Relation: Related genes sometimes have expression profiles that look quite different. For example, cell cycle related genes will have sinusoidal profiles, but if the genes are expressed in different cell cycle phases, the correlation will be poor. Thus, there is a fundamental problem in looking just for co-expression. Conversely, unrelated genes could accidentally have similar profiles.

2. Small number of measurements: Typically, the number of genes is about two orders of magnitude larger than the number of conditions under which each gene is measured. This reduces the resolution power to differentiate between gene expression profiles.

3. Noisy measurements: Microarray measurements are notoriously noisy, exacerbating the problem of few measurements.

4. High dimensionality of space: The expression profiles capture a very large number of processes in parallel. Thus, it is conceivable that it is quite difficult to perform dimensional reduction and the dimensionally reduced results may be misleading.

5. Methodological biases: Various dimensional reduction methods are biased in non-biological ways (for example, cluster analysis assumes, and then forces, a grouping tendency).

6. Variation of experimental parameters: The results of biological experiments depend on a huge number of parameters, many of which are out of the experimentalists control. It is therefore not surprising that when experiments are repeated, these parameters change, and thus results are not reproducible.

4 Our Approach/Roadmap

In Chapter 2, we introduce the Renormalization Group as a tool to construct phenomenological models, and consider its application to statistics. RG is interpreted as a search for stability, and we propose that stability against addition of data should be a requirement for good statistical quantifiers. In particular, it is shown that the traditional statistical quantifiers such as mean and variance can be 'derived' in this way.

In Chapter 3, we extend this stability principle to more general statistical quantifiers, in particular those produced by dimensional reduction. We introduce and discuss non-Metric Multidimensional Scaling (nMDS), which shall be our dimensional reduction scheme of choice for this thesis. Modified versions of nMDS are considered, and it is found that the one conforming to the RG stability principle performs best.

Chapter 4 is a methodological chapter focusing on current practices and visualization. The agenda of this chapter is essentially to undermine Cluster Analysis, which is by far the most popular exploratory data analysis method. We discuss Cluster Analysis in detail, with a particular emphasis on its abuse and the biases introduced by its use. We believe one of the pillars of its popularity is its use in the clustered heatmap, the dominant visualization method in Bioinformatics. With this in mind, we introduce NeatMap, an R package developed by us to produce heatmap-like-plots using dimensional reduction methods such as nMDS in preference to cluster analysis.

Chapter 5 deals with the problems faced by dimensional reduction methods in the analysis of noisy experiments. We propose a noise reduction procedure based on the phenomenological idea that meaningful relations must be simple, and therefore conform to the dimensionally reduced structure. Data not conforming to this structure must then be related to microscopic details and are discarded as noise. This procedure is found to perform well when applied to a subset of the data corresponding to a single process, but exhibits the same problems as traditional methods when working with the entire data set.

In Chapter 6, we propose a method that solves many of these problems by working with multiple experiments. The traditional approach is to treat variation across experiments as random noise, which is a nuisance to be removed by averaging. Instead, we use these differences as microscopic perturbations to shake the system. By looking for stable well conserved parts, we can identify the dominant processes in the cell and the genes involved in them. Unlike traditional methods, the separation of processes is clear enough to allow us to use it as a method of identifying genes belonging to specific categories. In this way many of the problems with traditional methods are removed.

Finally in Chapter 7, we summarize this work and consider possible future problems. We revisit each of the preceding chapters, identify the important results, and discuss possible future directions.

Chapter 2

Renormalization Group and Data Mining

1 Introduction

We live in a complex world. Physical objects we encounter in our daily lives each composed of a huge ($\simeq 10^{23}$) number of molecules. Each of these molecules is composed of numerous atoms, which in turn are made of more fundamental particles. Each of these fundamental particles interacts (however weakly) with all other fundamental particles. We cannot observe any of these fundamental interactions or details directly. So if observables depended on them, their behavior would seem random and irrational to us. The world would seem completely unpredictable. In such an unpredictable world, it would not make any sense to be rational and intelligence would serve no purpose at all.

We can invoke the anthropic principle to say that if intelligence exists in this world, it is because the world is rational and there are understandable natural laws. The world must be predictable in some sense and the laws cannot depend critically on microscopic details. Therefore, one may expect that most phenomena should be understandable without recourse to microscopic details. Thus, a phenomenological description should be possible.

As an example, let us consider polymers in a dilute solution. A polymer is a long molecule formed by connecting low molecular weight molecules called monomers. In a solvent, the monomers tend to repel each other, and thus the polymer chain performs a self-avoiding walk, i.e., a random walk where a site is never revisited. Irrespective of the monomer and solvent being used, the end to end vector $\boldsymbol{R}$ is found [10] to satisfy the relation

$$\left\langle \boldsymbol{R}^2 \right\rangle = c_0 M^{2\nu},$$

where M is the molecular weight, ν is a universal exponent ($\nu \simeq 0.588$) independent of the choice of polymer-solvent pair, while c_0 is the parameter sensitive to molecular details of both polymer and solvent.

This is an example of a phenomenological relationship. It essentially involves describing a phenomenon as [6]

$$\text{phenomenon} = \text{universal structure} + \text{materials properties}$$

The universal behavior in this case is the scaling law. It is considered universal because it applies to many different compounds. The constant factor contains the material or microscopic properties.

How does one go about constructing a phenomenological theory? We must somehow find a combination of variables that is universal in nature and separate it out from the material details. In general, this is very difficult to do, and would require extraordinary insight. Thankfully there is a framework that provides a guiding principle for this process; it is known as The Renormalization Group (RG). The basic idea of RG is that the universal structure cannot depend on microscopic details, and so if we remove the part that depends critically on the microscopic details, what is left behind must be universal.

How does one decide if a quantity is universal or not? There must be a scale at which we are observing the phenomena L_0. The interactions occur at some scale l. For example, in the polymer case l should be the length of the monomer. If $l/L_0 \ll 1$, then we can say it is microscopic. These scales, in general, need not be length scales. They could be small number of particles vs a large number, genotype vs phenotype, etc.

An observable f of the system is studied as $l/L_0 \to 0$. If the limit exists, then the observable has a definite value in the macroscopic world, and if we are close enough to the limit it does not depend on the microscopic details. Thus, whatever the state of the microscopic world, it has a definite value and is therefore universal. On the other hand, if the limit is not defined, that means the macroscopic observable is affected by microscopic details. Of course, in reality, l/L_0 is not exactly 0 and instead has a small but definite value for each system. Since the limit is not defined, $f(l/L_0)$ takes on differing values depending sensitively on the idiosyncrasies of the microscopics.

If the system allows such divergent parts to be separated out as $l/L_0 \to 0$, then the remainder is universal while the separated divergent part depends on microscopic details. Thus, a phenomenological description has been constructed. A system, which allows such a separation is called renormalizable and this procedure is called renormalization.

Actual implementation of these ideas is achieved in one of two ways:

1. The first is a direct implementation of the ideas above. We take the limit and look for the divergent parts. What is left behind must be universal. This is known as Stückelberg-Petermann (SP) type RG

2. Another way to take the limit is to observe the system from further and further away. The properties that are preserved are global features that we are interested in. This is known as the Wilson-Kadanoff (WK) type RG.

So far, our discussion was completely general. In our case, we are interested in characterizing properties of data-sets or more precisely in finding the 'true' mathematical structure characterizing a population of objects. The population can be thought of as a realization of some statistical distribution. In the asymptotic limit of a large number of samples we assume that we can obtain true structures. Thus, in our case, the macroscopic limit corresponds to a large number of samples.

It is not hard to visualize the utility of the Wilson-Kadanoff type RG in this scenario. We consider more and more points, and look at them from afar and look to see what features are well preserved. In fact, Jona-Lasinio showed [11, 12] that RG is just the extension of the central limit theorem (which applies to independent variable) to strongly correlated variables. For completeness sake, in Appendix A we show how such an approach may be used to 'derive' the central limit theorem.

The utility of the Stuckelberg-Petermann type RG is harder to visualize. We make use of this type of RG in the context of singular perturbations applied to an appropriate dynamical system to derive a Langevin type RG equation. It is then shown independently that an appropriately chosen dynamical system driven by samples from a population produces a very similar result, with the mean and central limit theorem as specific terms. It is known that SP type RG essentially involves a search for stability by removing divergent parts. This motivates a derivation of these statistical quantifiers based on such a stability principle.

This stability principle is one of the guiding principles for the rest of the thesis. In future chapters we shall extend this idea to more general quantifiers such as the results of multivariate dimensional reduction techniques, and we shall also show how such a stability idea may be used to combine multiple experiments.

2 RG and Statistics

2.A Renormalization and Langevin equation

We now consider Stückelberg-Petermann type RG as applied to deal with singular perturbations in differential equations. Its main idea is that the effect of the perturbation causing qualitative changes results in a series that does not converge uniformly with respect to time. To convert the series into uniformly converging series by modifying the constants of motion is the renormalization procedure. The RG equation is the equation summarizing the asymptotic effects of perturbations. Writing down the RG equation is the essence of many singular perturbation methods; we can do so in a fairly abstract fashion [13].

The problem we wish to discuss in order to pave our road to statistical problems from RG theory is the one discussed initially by Hasselmann [14] and mathematically by Kifer and others [15] recently. Consider the following singular perturbation problem:

$$\frac{dx}{dt} = f(x, y), \;\; \frac{dy}{dt} = \frac{1}{\epsilon} g(x, y), \tag{2.1}$$

where x and y are in certain regions of appropriate Euclidean spaces, f and g are well-behaved (e.g., sufficiently smooth) functions on the direct product of the regions for x and y, and ϵ is a small positive real number. The starting point of an RG approach to singular perturbation is to consider the problem at the shortest time scale [16]. Therefore, we rewrite the above system in terms of $\tau = t/\epsilon$:

$$\frac{dx}{d\tau} = \epsilon f(x, y), \;\; \frac{dy}{d\tau} = g(x, y). \tag{2.2}$$

The naive perturbation result reads

$$x = A + \epsilon \int_0^\tau f(A, y_0(A, \sigma)) d\sigma + o[\epsilon], \tag{2.3}$$

where

$$\frac{dy_0}{d\tau} = g(A, y_0). \tag{2.4}$$

If y_0 governed by (2.4) is sufficiently chaotic for each constant A (e.g., an axiom A system), we may assume the physically observable invariant measure μ_A (e.g., an SRB measure) exists, so the naive perturbation result may be rewritten as

$$x = A + t\langle f \rangle(A) + o[1], \tag{2.5}$$

where

$$\langle f \rangle(A) = \int f(A, y) d\mu_A(y). \tag{2.6}$$

The renormalization $A \to A(t)$ cures the nonuniformity of the convergence of the perturbation series (2.5) (to the lowest nontrivial order) and the RG equation reads

$$\frac{dA(t)}{dt} = \langle f \rangle (A(t)). \tag{2.7}$$

Thus, $x = A(t)$ is the solution to order unity[1]. This is also the well-known result of the so-called averaging method (extended to chaotic dynamics).

The error of this leading order result may be estimated by exploiting the shortness of the correlation of $y_0(A, \tau)$. The integral in (2.3) may be interpreted (essentially) as a sum of independently and identically distributed (i.i.d) random variables. We expect the large deviation principle to hold with a sufficiently smooth rate function. Therefore,

$$P\left(\frac{1}{\tau}\int_0^\tau f(A, y_0(A,\sigma))d\sigma - \langle f \rangle (A) \sim \xi\right) \sim \exp\left[-\tau I(\xi)\right], \tag{2.8}$$

where $I(\xi)$ is the rate function. For small enough ξ, the rate function is quadratic, and so we can write:

$$I(\xi) = \frac{\xi^2}{2b(A)},$$

We then make use of the Gaussian property that

$$P\Big(y(x) \sim \xi\Big) = e^{-\xi^2/2b} \Rightarrow \langle y^2 \rangle = b$$

to calculate $b(A)$ as

$$\begin{aligned} b(A) &= \lim_{\tau\to\infty} \tau \langle \Big(\frac{1}{\tau}\int_0^\tau f(A, y_0(A,\sigma))d\sigma - \langle f \rangle (A)\Big)^2 \rangle, \\ &= \lim_{\tau\to\infty} \frac{1}{\tau}\int_0^\tau \mathrm{d}t_1 \int_0^\tau \mathrm{d}t_2 \Big\langle [f(A, y_0(A,t_1)) - \langle f \rangle (A)][f(A, y_0(A,t_2)) - \langle f \rangle (A)] \Big\rangle, \end{aligned}$$

[1] This is only formal. $\langle f \rangle_A$ may not even be continuous, and the reduced equation may not have any solution in some cases. The meaning of the approximation of the true solution $x(t)$ in terms of $A(t)$ is also quite delicate. Mathematically, we expect the modulus of the discrepancy integrated over $[0, t/\epsilon]$ converges to zero as $\epsilon \to 0$.

$$
\begin{aligned}
&= \frac{2}{\tau}\int_0^\tau \mathrm{d}t_1 \Big\langle [f(A, y_0(A,t_1)) - \langle f\rangle(A)][f(A, y_0(A,0)) - \langle f\rangle(A)] \Big\rangle, \\
&= 2\int_0^\infty dt_1 \int d\mu_A(y_0(A,0))[f(A,y_0(A,t_1)) - \langle f\rangle(A)][f(A,y_0(A,0)) - \langle f\rangle(A)].
\end{aligned}
$$

We may write

$$
x(t) = A + t\langle f\rangle(A) + \sqrt{\epsilon b(A)}B(t) + o[\epsilon^{1/2}], \tag{2.9}
$$

where $B(t)$ is the Wiener process. Notice that the fluctuating term is also secular (diverging roughly as $\sqrt{t}$). Renormalization immediately gives the following stochastic differential equation[2]:

$$
dA(t) = \langle f\rangle(A(t))dt + \sqrt{\epsilon b(A(t))}dB. \tag{2.10}
$$

The first term corresponds to the strong law of large numbers, and the second to the central limit theorem for the noise effect. This is almost a trivial observation, so one might conclude it is not even worth mentioning. However, perhaps this is only the tip of a larger picture.

2.B Renormalization and Statistical Features

We now show how all the statistical quantities appearing in the descriptive statistics (e.g., moments, density distribution function, spectrum) can be obtained by perturbing an appropriate dynamical system by the quantity $\{\xi_n\}$ (or its function) whose statistical description we wish to have. The simplest case is

$$
x_{n+1} = x_n + \epsilon\xi_n, \tag{2.11}
$$

where ξ_n are iid random variables. ϵ is put to indicate that the term is a perturbation term, but since the system is linear, ϵ has no actual meaning; we can scale it out. The general

[2] The interpretation of this stochastic equation is delicate. Our interpretation (conjecture) is, under the condition that the averaged equation (2.7) is well-behaved, that $\int dt(\dot{x} - \langle f\rangle)^2/2\epsilon b(A)$ is the rate function for the large deviation of the true solution from $A(t)$. That is, we interpret the Langevin equation (2.10) as a shorthand notation of the large deviation principle. For this point of view, see [17].

solution for the simplest case with the variance reads (compare with (2.9))

$$x_n = A + \epsilon \sum_{k=0}^{n-1} \xi_k = A + \epsilon n \langle \xi_1 \rangle + \epsilon \sqrt{n} \sigma \chi + o[n^{1/2}], \tag{2.12}$$

where the equality is in law, A is a constant, χ is a random variable obeying $N(0,1)$, and σ is the standard deviation. The analogy with the dynamical example in the preceding section is obvious. Renormalizing A as A_n, by absorbing the divergence, we have the renormalization group equation (compare with (2.10))

$$A_{n+1} - A_n = \epsilon \langle \xi_1 \rangle + \epsilon \sigma \chi. \tag{2.13}$$

Here, we have used $\sum_{k=1}^{n} \chi_k = \sqrt{n}\chi$ in law with $\{\chi_k\}$ being a set of iid variables obeying $N(0,1)$. Therefore, not surprisingly, the strong law of large numbers and the central limit theorem are naturally recovered from the RG picture. Needless to say, if the distribution of ξ_k is with a fat tail, we have a different power of n in the second term.

2.C Stability as a Guiding Principle

Intuitively, the SP RG shakes the system and then identifies significantly and persistently perturbed parts. From the knowledge SP RG infers asymptotic behaviors that are insensitive to perturbations. Such an idea is of course common especially in the so-called data driven statistics [18]. For example, the bootstrap methods watch how the statistical results change against sample perturbations (due to resampling); if a result 'shakes' too much, it is rejected. The idea of statistical test in general may be understood as an application of the same idea of stable reproduction.

In the RG procedure sketched above, the renormalized constants were determined to absorb the most dangerous terms that could spoil the series expansion calculation. Thus, to pursue the stability of $x - A$ is the renormalization procedure. This may look a simple restatement of the SP-RG theory, but perhaps it is better to say that the pursuit of stable results against perturbation is the fundamental objective of RG, the SP scheme being its corollary.

Actually, this point of view is consistent with the observation [13] that it is absolutely necessary (and actually sufficient) to absorb the secular behavior (persistent effect of perturbation) to obtain the result correct to order ϵ up to time $1/\epsilon$ from the first order naive perturbation calculation. In other words, to glean the asymptotic effect of the perturbation on a dynamical system we need not accurately compute the perturbation effect, but have only to pay due attention to secular terms. Approximate minimization of the perturbation effect (that is, approximate minimization of $x - A$) through modification of the invariants (that is, $A \to A(t)$) gives the information on asymptotic behaviors of the system.

The example shown earlier may be reinterpreted in the framework of pursuit of stability. The computation of the expectation value is to choose m so that asymptotically

$$\sum_{k=0}^{n-1}(\xi_n - m) = o[n]. \tag{2.14}$$

That we can choose such m is the strong law of large numbers. This is reminiscent of the formulation of the strong law of large numbers in terms of the futility of gambling [19]. In the case of reductive perturbation no condition other than stability (no divergence) is needed to extract asymptotic results, so we can expect the same for statistical feature extractions. This approach will be generalized in following chapters.

This approach makes a few assumption. Firstly, we must assume that the population is fixed; for example, it has no systematic time dependence to ensure statistical uniformity. Secondly, an infinitely large population of objects is an idealization. For example, if we wish to analyze gene activities of an organism, we imagine the population of numerous microarray experimental results under the same condition even if the number of genes under study is not very numerous. These assumptions are not really very restrictive other than statistical uniformity. For example, curve fittings and parameter estimations may be interpreted as prototype examples. Any estimation problem is ultimately justified by the law of large numbers and its refinements. Even the problem of fitting a curve to a finite set of sample points assumes that there are infinitely many such finite sets sampled from the same population. Thus, statistical estimates are always supported by asymptotic results.

3 Concluding Remarks

In this chapter we have introduced the renormalization group and considered its application to statistics. The connection in the case of Wilson-Kadanoff RG is fairly straightforward (and was illustrated with a proof of the Central Limit Theorem in Appendix A).

Our contribution is to make this connection in the case of SP RG. We have started with an observation that statistics and data mining may be regarded as a part of RG theory. If the idea that asymptotic statistical estimation problems may be understood as dynamical system problems whose time variable corresponds to the number of samples and the standard RG are combined, we can conclude that the pursuit of stability against adding new samples allows us to estimate statistically asymptotic features. This point of view seems to give a unified perspective for some other statistical problems, as will be shown in following chapters.

Chapter 3

nMDS

1 Introduction

In the last chapter, we demonstrated the utility of an RG based principle to motivate the standard statistical quantifiers to characterize populations. Such simple characterizations are not very useful or meaningful in the context of microarray experiments. Typically, we are given expression profiles for a large number of genes, and we would like to know their relative relations between the genes based on these profiles. While the number of genes itself can be very large ($\sim 10^4$), and the expression profiles lie in a moderately high dimensional space ($\sim 10^1 - 10^2$) it is generally believed that, for most processes, groups of genes act in concert. Thus, the expression profiles should, in the correct basis, lie in some lower dimensional space that admits a meaningful phenomenological description. The extraction of this correct basis has been the goal of a variety of dimensional reduction methods.

Such dimensional reduction methods thus characterize the population by replacing each gene expression profile by a lower dimensional counterpart in such a way that genes with similar gene expression profiles will be placed close to each other in the lower dimensional space. The lower dimensional embedding may be thought of as a more sophisticated version of the simple statistical quantifiers studied in the last chapter. We shall discuss the utility of the RG stability principle to these more generalized statistical quantifiers. To do this we shall make use of a specific data reduction scheme known as non-Metric Multidimensional Scaling (nMDS). nMDS shall also be our dimensional reduction scheme of choice for the rest of this thesis.

To begin with, we shall discuss the standard implementation of nMDS. The utility of nMDS shall be demonstrated with artificial and biological examples. To illustrate the potential of the RG principle we shall consider improvements to the standard implementation of nMDS. We find that we are faced with two, seemingly very similar, implementations, which one might expect to produce comparable results. Of these, one implementation conforms to the RG stability principle of being maximally consistent with the existing structure, while the other does not. We shall show that the performance of the first implementation is far better than the other, thereby proving the utility of the RG stability principle.

2 Traditional nMDS

Given a set of points lying in some high-dimensional space, the basic goal of dimensional reduction is to find a lower dimensional representation capturing the essence of the relative relations. Based on which aspects are captured, there are many dimensional reduction schemes. In this thesis we primarily use a scheme known as non-Metric Multidimensional Scaling (nMDS) [20, 21, 22, 23]. nMDS is a completely data driven scheme, and in our experience its performance is superior to other methods of its class (except perhaps in terms of computational requirements).

Rather than jumping into the technical details of nMDS, let us get a general idea of what it does using an example [24]. 1000 major cities were considered all around the earth, and the distance between them was calculated as

$$\delta_{ij} = cos\theta_{ij},$$

where θ_{ij} is the angle between cities i and j measured with the origin as the center of the globe. The distances between the different cities were compared, and this information (i.e., the inequalities and not the actual distances) was passed to nMDS. nMDS needs neither

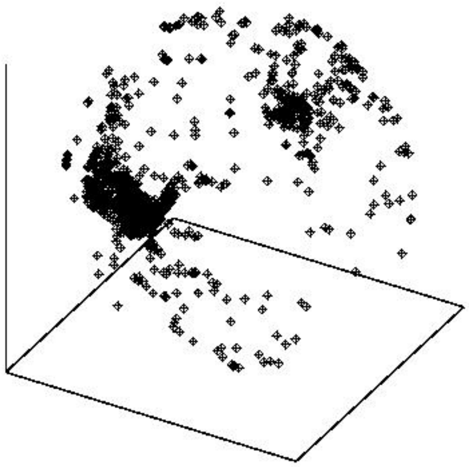


Figure 3.1: nMDS embedding of cities on the surface of a globe

the high dimensional profiles, nor does it need the inter-point distances (just their inequality relations); this is why it is considered non-metric. Based on this information, nMDS constructed a representation of these 1000 points by embedding them in a 3 dimensional Euclidean space. The result is shown in Fig. 3.1.

The spherical structure of the earth is automatically generated. Since the embedding could be any shape in 3D, this is a proof that the earth is in fact round. The shapes of continents can also be discerned, and the positions of the cities seem to be correct within some small error bar. Thus, the low-dimensional representation of data points generated by nMDS captures the underlying geometry (if present) based just on the relations between the pairwise distances. As we shall discuss later, in practice, for computational reasons, rather than passing inequalities the actual distances or data from which the distances can be calculated are usually passed to nMDS.

2.A Implementation

Let us now consider nMDS more rigorously. nMDS is an unsupervised data geometrization method placing N points representing the objects under study, e.g., genes, in a certain metric space E, such that the pairwise distances $d(i,j)$ of the points in E have consistency with the pairwise dissimilarities $\delta(i,j)$ of the corresponding objects in the input data. More precisely, nMDS tries to ensure that if $\delta(i,j) > \delta(k,l)$, then $d(i,j) > d(k,l)$ for all i, j, k and l denoting objects being analyzed. It is considered non-metric because, strictly speaking, the $\delta(i,j)$ values need not be known; only their order relationships, whether $\delta(i,j) > \delta(k,l)$ or not. If we have a reasonable number ($N > 30$, say) of points, this condition is typically strong enough to ensure a unique geometrical pattern for good data, as seen in the example above. There are many ways to implement nMDS [25]. We shall essentially be using the algorithm proposed by Taguchi and Oono [26, 27]. A flowchart explaining it is shown in Fig. 3.2.

A major distinction of this nMDS algorithm from most other implementations is that the comparison of the distance in E and the dissimilarity is performed after converting both into rank orders. If the pairwise dissimilarity $\delta(i,j)$ has ranking R_{ij} in the set of all the available dissimilarities, and $d(i,j)$ has ranking r_{ij} in the set of all the pairwise distances of the points in E, the points in E are so positioned to minimize

$$\Delta \equiv \sum_{i \neq j} (R_{ij} - r_{ij})^2.$$

The minimum of this is clearly when $R_{ij} = r_{ij}$ for all i and j, *i.e.*, when the pairwise rankings in the original and embedded space exactly match. This is achieved through an over-damped dynamics driven by the ranking mismatch. The updating scheme used is:

$$\boldsymbol{x}_i \to \boldsymbol{x}_i + \alpha \sum_{j \neq i} [R_{ij} - r_{ij}] \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{|\boldsymbol{x}_i - \boldsymbol{x}_j|},$$

where the positions of the points in E are given by $\boldsymbol{x}_i$ and α is an appropriately small number to make the relaxation dynamics stable. The $\boldsymbol{x}_i$ are initially chosen randomly, and the positions are updated using the rule above until a fixed point is reached. It should be

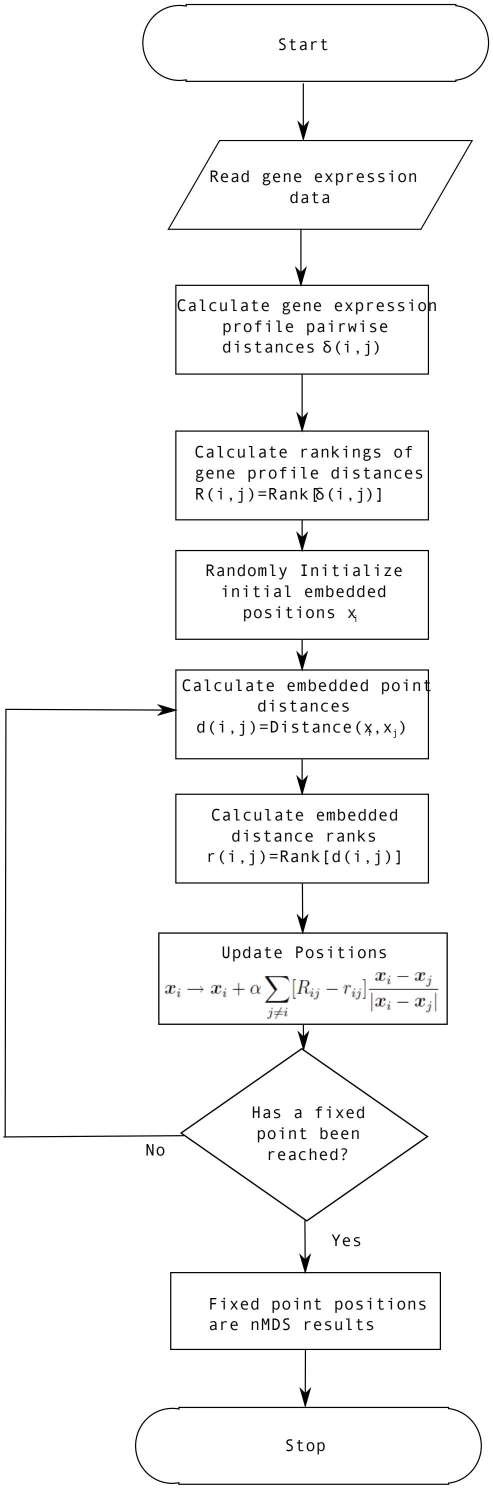

Start
Read gene expression data
Calculate gene expression profile pairwise distances δ(i,j)
Calculate rankings of gene profile distances R(i,j)=Rank[δ(i,j)]
Randomly Initialize initial embedded positions x_i
Calculate embedded point distances d(i,j)=Distance(x_i,x_j)
Calculate embedded distance ranks r(i,j)=Rank[d(i,j)]
Update Positions
$\boldsymbol{x}_i \to \boldsymbol{x}_i + \alpha \sum_{j\neq i} [R_{ij} - r_{ij}] \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{|\boldsymbol{x}_i - \boldsymbol{x}_j|}$
Has a fixed point been reached?
No
Yes
Fixed point positions are nMDS results
Stop

Figure 3.2: nMDS Flowchart

clear that the desired minimum of Δ does correspond to fixed point of the dynamics. Like all non-linear optimization methods, there is a risk of getting trapped in a fixed point that is a local minimum, although in practice the dependence on initial condition seems to be weak compared to other methods. Additionally, the quality of the result can be gauged by the value of Δ as compared to an appropriate null result e.g., one produced by a random ranking, although this will be too lenient.

There are proposals (for example, by Taguchi [28]) about determining the optimal dimension of E, but in this thesis, for simplicity, we discuss only examples for which two (or at most three) dimensional Euclidean spaces supposedly suffice. In any case, the role of dimensional reduction is to produce patterns that can be recognized by humans, and we cannot typically comprehend more than 3 or 4 dimensions.

2.B Example

The utility of nMDS in the analysis of biological data has been shown previously [27], but completeness sake we discuss one example here. We consider a microarray based study of the gene expression in the honey bee (*apis mellifera*) as a function of class and age.

The bee society shows a very pronounced social class structure, being divided into nurses and foragers. Typically, when bees are young (2 to 3 weeks) they act as nurses taking care of the brood, while once they get older they turn into foragers that go outside the hive and forage for food. Under normal circumstances, the transition from nurse to forager is dictated by age. Under special circumstances (such as death of all the foragers), the transitions may occur earlier or later as per the requirements of the hive.

Whitfield *et al.* [29] set about trying to study the genetic basis for these phenotypic changes. To this end, they considered 60 bees, representing all possible combinations of age and class viz. young nurse, old forager, old nurse and young forager. The gene expression of about $5.5k$ genes was measured for these 60 bees using microarrays. Thus, we can either study the relations between the genes (each of which is a 60 dimensional vector) or that among the individuals (who are represented by $5.5k$ dimensional vectors).

Here, we do both. First, nMDS was used to perform a 2D embedding of the individual bees. Bees were compared using the Pearson[1] correlation coefficient of their gene expression vectors. The result of this is shown in in Fig. 3.3 .As can be seen here, there is a fairly clear separation of individuals in terms of class, although age based separation is not as clear (it is better in higher dimensions). Thus, it has been shown that there is a genetic basis to social class. This separation is much clearer than can be achieved using comparable methods such as Principal Component Analysis (PCA).

Then, the 500 genes that showed the greatest variance over the different individuals were considered, and nMDS was used to embed them into 3D. Stereoplots of this result may be seen in Fig. 3.4. Note that the different cells in Fig. 3.4 all represent this same embedding (the positions of the points are all identical), only the coloring is different. Each cell, in fact, shows one representative bee of each age/class combination. We then color each of the 500 genes according to its expression in this individual. Clearly similar genes have been automatically grouped together, and we can roughly say which groups of genes are responsible for change in class, aging etc.

3 Modified Dynamics

In the traditional implementation of nMDS, *all* possible pairwise dissimilarities are considered together, and rankings produced in the original and embedded spaces. Thus, if there are N points, there will be $N(N-1)/2$ dissimilarities. The input to nMDS is, strictly speaking, just the pairwise inequalities, of these pairwise dissimilarities of which there shall be on the order of N^4 (although if rankings were to be passed it would be $O[N^2]$). For a large number of points this is an exorbitant amount of information to store.

[1] For two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, their Pearson correlation is given by $\sum_i \frac{(x_i-\mu_x)(y_i-\mu_y)}{\sigma_x \sigma_y}$, where μ_z and σ_z represent the man and standard deviation of the vector $\boldsymbol{z}$ respectively.

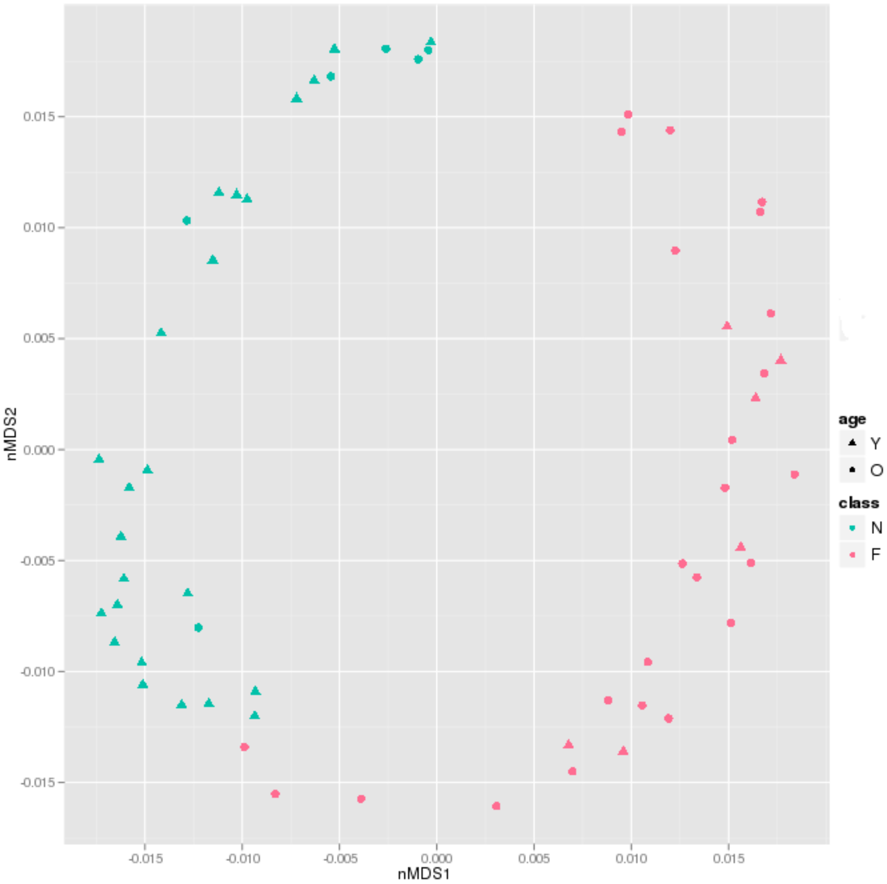


Figure 3.3: nMDS Embedding of Bee Individuals: Individuals are colored according to their class (blue for nurses, and pink for foragers), and the point shapes are decided by their age (triangles for old, and circles for young). Clearly, there is good separation between nurses and foragers.

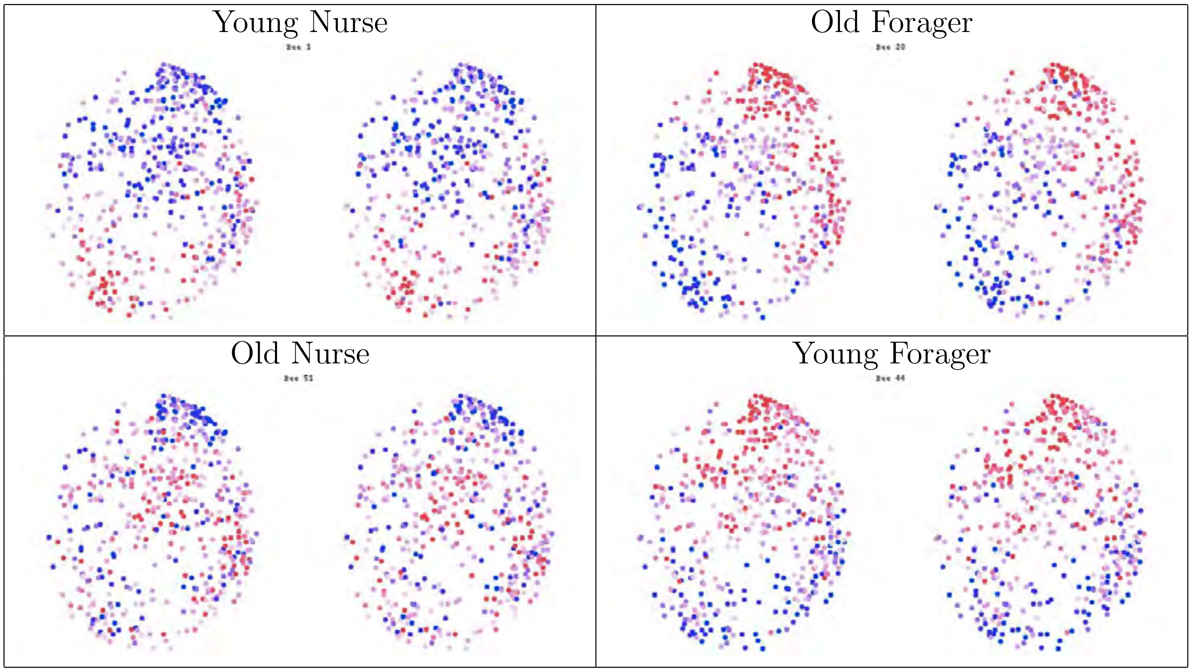


Figure 3.4: 3D nMDS embedding of the 500 genes with the highest variance in the Bee data. Results are colored according to the gene expression levels of 4 individuals, belonging to the age/class combination indicated

3.A Local Schemes

To combat this problem an alternate scheme was considered, where instead of comparing all possible dissimilarities, only those dissimilarities with a point in common would be compared. Thus, for each point we only have information about which points are closer and which are farther, but do not know how these compare to distances with respect to other points. More specifically, in the original scheme, all inequalities $\delta(a, b) < \delta(c, d)$ were considered, while now we shall only be considering those of the form $\delta(a, b) < \delta(a, c)$. This reduces the number of inequalities to be considered from $O(N^4)$ to $O(N^3)$.

The implementation of the dynamics is very similar. However, instead of a single ranking of all the pairwise dissimilarities, there are now different rankings centered around every point. Thus, for each point i, the set of all dissimilarities to that point, viz. $\Big(\delta(i, 1), \delta(i, 2), \ldots, \delta(i, N)\Big)$, are considered. This set of dissimilarities are now ranked from closest to farthest, and $R_i(j)$ is the rank of the j^{th} point in terms of its dissimilarity to i. Similar rankings $r_i(j)$, can be generated in the embedded space E.

As before, the goal of the method is to get the rankings R in the original space and those r in the embedded space to match by minimizing a potential of the form

$$\Delta = \sum_{i \neq j} [R_i(j) - r_i(j)]^2.$$

Although, this looks very similar to the previous form of Δ, there is one big difference. While $R_{ij} = R_{ji}$ in general $R_i(j) \neq R_j(i)$. For example, in Fig. 3.5, point 4 is the farthest from 1, so $R_1(4) = 3$, while 1 is the closest point to 4 meaning $R_4(1) = 1$. This broken symmetry complicates the implementation of the optimization dynamics, and in fact gives us two possible choices:

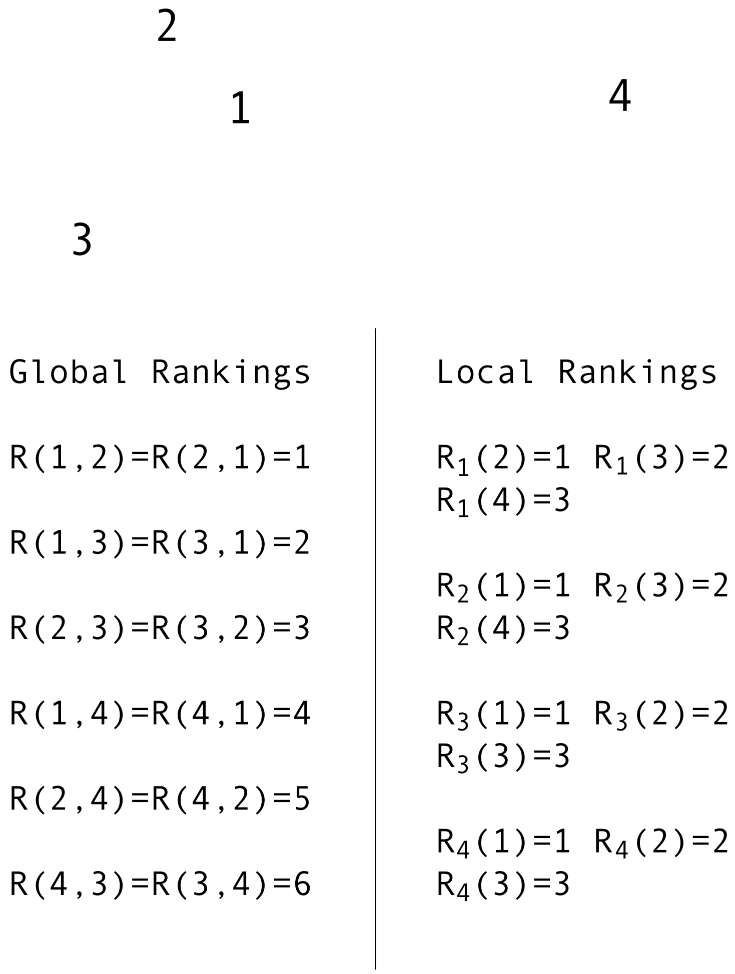
1. Intrinsic Scheme: In this scheme, the position of a point is updated according to *its own* view of the closeness of other points. So for a point i, its position is updated using its own ranking of the other points by the rule

$$\boldsymbol{x}_i \to \boldsymbol{x}_i + \alpha \sum_{j \neq i} [R_i(j) - r_i(j)] \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{|\boldsymbol{x}_i - \boldsymbol{x}_j|}. \tag{3.1}$$

2. Extrinsic Scheme: In this scheme, the position of a point is updated according to *other points* view of its closeness to them. So for a point i, its position is updated using other points ranking of it by the rule

$$\boldsymbol{x}_i \to \boldsymbol{x}_i + \alpha \sum_{j \neq i} [R_j(i) - r_j(i)] \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{|\boldsymbol{x}_i - \boldsymbol{x}_j|}. \tag{3.2}$$

At face value, both of these schemes look quite similar, and the correct configuration will be a fixed point for both of them. The nature of the ‘forces’ at an intermediate step in the dynamics may be seen in Fig. 3.6 . Still, it is unclear *a priori* which one is better. It turns out the RG stability principle provides a way.



Global Rankings

$R(1,2)=R(2,1)=1$

$R(1,3)=R(3,1)=2$

$R(2,3)=R(3,2)=3$

$R(1,4)=R(4,1)=4$

$R(2,4)=R(4,2)=5$

$R(4,3)=R(3,4)=6$

Local Rankings

$R_1(2)=1$ $R_1(3)=2$
$R_1(4)=3$

$R_2(1)=1$ $R_2(3)=2$
$R_2(4)=3$

$R_3(1)=1$ $R_3(2)=2$
$R_3(3)=3$

$R_4(1)=1$ $R_4(2)=2$
$R_4(3)=3$

Figure 3.5: Comparison of Local and global rankings:

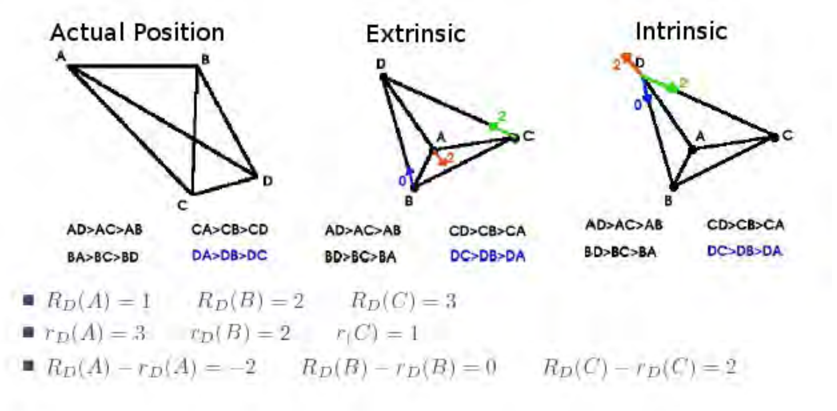


Figure 3.6: Comparison of the Intrinsic and Extrinsic schemes: The figure on the let represents the true configuration that is the goal of the method. The two other figures show the forces cause by point D in an intermediate state using the Intrinsic and Extrinsic schemes

3.B Application of RG Stability

The RG stability principle is the idea that a good statistical quantifier should be stable against addition of data. In other words, if the value of the quantifier with N points should actively pursue being as consistent as possible with that with $N-1$ points. In the context of the two schemes considered above, suppose we have already constructed a configuration $\pi_{N-1} = \{\boldsymbol{x}_i\}_{i=1}^{N-1}$ in E_d from $N-1$ samples. Let us add one more sample and determine the corresponding point $\boldsymbol{x}_n$ in E_d.

To the lowest nontrivial order, we may assume that π_{N-1} is fixed, because we are looking for an asymptotically stable pattern, and try to locate $\boldsymbol{x}_N$ in E_d relative to π_{N-1}. $\boldsymbol{x}_N$ must be consistent with the ranking of dissimilarities around each point in π_{N-1}. We position $\boldsymbol{x}_N$ so that the overall rank mismatch is as small as possible. That is, we require that the change of the already estimated pattern is minimized by judiciously choosing the position corresponding to the newly added sample. This is the stability condition for the already estimated pattern. Notice that the positioning step corresponds to replacing ξ_n with $\xi_n - m$ in (2.14). That is, positioning process may be understood as the subtraction step in the RG.

If we consider the Extrinsic scheme to update the position $\boldsymbol{x}_N$, the rankings used are

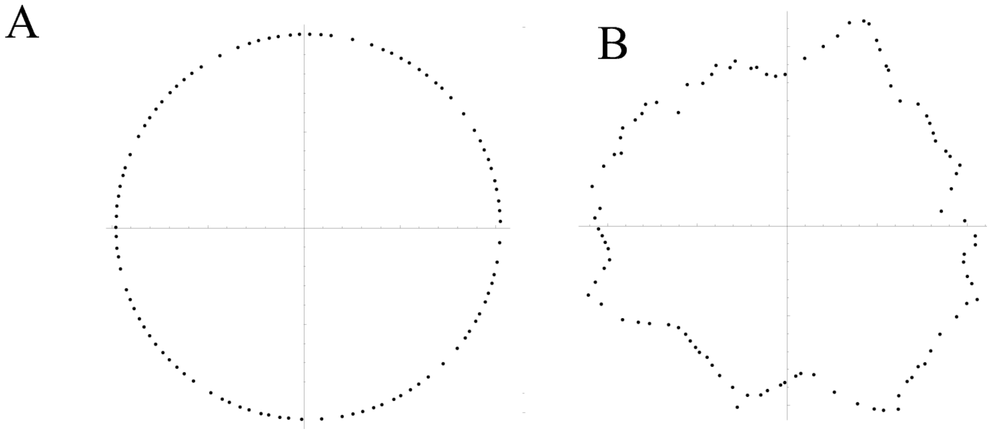


Figure 3.7: Embedding of the 100 points sampled from a circle. (A) is the result obtained with scheme S after 100 iterations; (B) is the result obtained with scheme NS after 1000 iterations. Both schemes used the same $\alpha = 0.001$. Clearly, scheme S converges to the correct configuration quickly, while NS does not. The circle shown in (A) is a stationary pattern. The pattern (B) is actually not the final one; eventually NS results tend to a more symmetric three-leaved clover-like pattern that actually rigidly rotates slowly. That is, the 'ω-limit set' is a sort of limit cycle.

$R_i(N)$ and $r_i(N)$ with $i \in (1, 2, \ldots, N-1)$. Thus, the position $\boldsymbol{x}_N$ is determined with respect to a configuration which is already meaningful (or in analogy with the mean subtraction case, the correct m is being subtracted). Consistency with respect to the old configuration is actively pursued. On the other hand, in the intrinsic scheme the position $\boldsymbol{x}_N$ uses its rankings of the other points viz., $R_N(i)$ and $r_N(i)$ with $i \in (1, 2, \ldots, N-1)$. Since the position of $\boldsymbol{x}_N$ is not correct, all the rankings are with respect to an incorrect reference. Thus, the intrinsic scheme does not conform to the RG stability principle.

Representative results due to the two schemes are illustrated in Fig. 3.7 when the population is a point set consisting of a unit circle. That is, the 'mathematical structure' we must be able to extract is a circle. The input data are only the local ranking information: for each point the input data give only the qualitative information about which point is closer or farther from itself, but not the actual distance to the point. Interestingly, the fate of the initial condition prepared by a slight perturbation of the correct configuration is markedly different between the two schemes as seen in Fig. 3.8. That is, the correct solution is not a stable

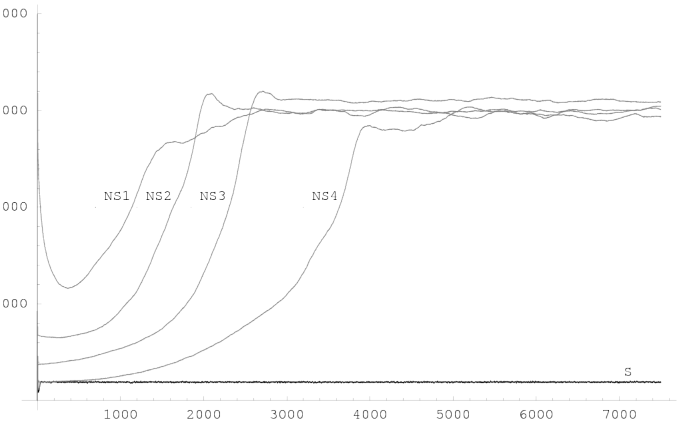


Figure 3.8: For scheme S the circle, i.e., the correct geometrical result, in Fig. 1 is a stable fixed point, but for scheme NS this correct figure is not a stable fixed point. Shown in this figure are residual errors by various schemes as a function of the number of iterations. The initial configurations were prepared from the correct result by displacing the point positions with uniform random noise of various amplitudes. The curves NS1, NS2, NS3 and NS4 show the residual errors for the initial condition with the displacement noise amplitudes 0.25, 0.05, 0.025, and 0.01, respectively. The error level eventually attained corresponds to the pattern similar to (B) in Fig. 1. In comparison, a result due to scheme S for a displacement noise amplitude 0.05 is also given as curve S. Notice that the dynamical system under study is not a continuous dynamical system, but with finite increments, so even for scheme S there is a residual error due to this quantization noise. That is why the curve S does not go to zero even asymptotically.

solution of the Intrinsic scheme. Thus, the pursuit of stability or, in other words, maximal consistency with the already incorporated information can be a good guiding principle for data analysis (and pattern recognition).

As mentioned earlier, the traditional nMDS scheme required $O[N^4]$ inequalities, while these modified schemes require $O[N^3]$ inequalities. The information contained in these inequality sets are actually both of order $N^2 \log N$, so the difference just mentioned may not be important. The number of inequalities is typically not an issue in the analysis of biological data since the input is typically not the inequalities themselves. Instead, either the pairwise distances or raw profiles (from which the pairwise distances can be calculated) is supplied. Then the distances are compared as required to generate the rankings used in the nMDS algorithm. For this reason we do not use the modified schemes in the rest of the thesis, although the results produced by the extrinsic scheme are comparable to those produced by the standard nMDS.

The inequality difference is likely to be an advantage in other fields (such as the humanities) where nMDS is used. For example, if we were trying to understand something about the color space by asking people to judge which colors were closer, then requiring fewer comparison is a big advantage. Also it is easier, and perhaps more meaningful, to ask if red is closer to yellow than it is to green than if yellow is closer to blue than red is to green. One other possible advantage is that it should be easier to parallelize the ranking step when many different rankings of smaller sets need to be performed, than a single ranking of a giant set. Since ranking is the most time consuming step, and multi-node cluster are now the norm, this could offer a significant computational advantage.

4 Conclusion

In this chapter we introduced nMDS, a dimensional reduction scheme we shall repeatedly use throughout this thesis, and demonstrated its utility. We also demonstrated the use of the RG stability principle in a more general setting: we showed that such an argument could be used to guide algorithmic choices in implementations of statistical quantifiers as complex as dimensional reduction schemes. It also led to the development of a modified scheme that could offer significant advantages in fields (such as the humanities) where nMDS is used.

Chapter 4

Visualization and Clustering

1 Introduction

Cluster Analysis is a term used to describe a class of unsupervised data-reduction methods that classify observations into groups known as clusters, such that the observations within a cluster are related in some way (at least more so than those in different clusters). Cluster Analysis has become one of the most popular and widely used unsupervised data-reduction methods. It continues to be used in a diverse range of fields including market research [30], search result grouping [31] and image segmentation [32]. Within biology too, cluster analysis holds a dominant position. It is the method of choice in applications such as constructing phylogenies [33], sequence analysis [34], and what is most relevant to us, as an exploratory tool in the analysis of high-throughput experiments [35].

Undoubtedly, some of this popularity is deserved. Cluster Analysis is fast, its results easy to understand and when there are groups of well separated objects to be identified, its performance is hard to beat. For these reasons, Cluster Analysis is the best choice for dimensional reduction in many contexts. However, in other fields, it is the preferred method among the masses, in spite of (and usually at the expense of) many other methods which are far more suitable. For example, as we shall show nMDS often outperforms cluster analysis, and yet is virtually unknown among biologists.

The simplicity of cluster analysis results belie the many assumptions that are implicitly made (depending on the specific algorithm chosen), which in turn severely dent its claims of being a truly data-driven method. This chapter is primarily meant as a propaganda piece to warn against the dangers of using cluster analysis (and to draw attention to NeatMap, a visualization program created by us) . To this end, we will first provide a quick introduction to cluster analysis. The most popular cluster analysis algorithm types will be surveyed, and we will discuss the underlying assumptions made by them while identifying their relative weaknesses and strengths.

We then turn to the visualization. With the advent of high-throughput experiments, whole genome measurements across multiple conditions have become common. It is almost hopeless to expect humans to comprehend such a large amount of data to infer anything meaningful. On the other hand, human interpretation and pattern recognition ability is still not even close to being matched by computers. Therefore, the best strategy for analyzing data is:

1. First use a dimensional reduction algorithm to produce a simpler reduced representation of the data that can be grasped by humans. The performance of dimensional reduction schemes is still not good enough that this result is biologically meaningful

2. The dimensionally reduced result must be then visualized in a way that takes advantage of human pattern recognition skills and encourages the formulation of biological hypotheses

Just as there are many different dimensional reduction schemes, there are many possible ways of visualizing their results. Much work has been put into understanding the properties of the dimensional reduction schemes and inventing specialized ones. Sadly, much less thought seems to have been put into visualization methods.

For example, the visualization of gene expression data is completely dominated by a single method known as the clustered heatmap. It has been used in thousands of publications spanning a multitude of organisms and a variety of data types [36, 37, 38, 39]; it has even been dubbed [40] a “post genomic visual icon.” As we shall show, the clustered heatmap is a deeply flawed visualization method. By using cluster analysis as its engine not only does it inherit all the associated problems, but it then proceeds to completely misuse those results. Yet, due to a lack of viable alternatives, its dominance has been unchallenged.

It is our belief that the continued popularity of cluster analysis is at least partly due to the lack of associated visualization methods with the visceral impact of the clustered heatmap. Thus, a major step in our agenda to replace cluster analysis is to generate alternatives to the heatmap. To this end, we have created an R package called NeatMap, which shall be the focus of the rest of this chapter. NeatMap offers a variety of novel heatmap-like plot types in two and three dimensions intended to be driven by dimensional reduction methods other than cluster analysis. The superiority of these plots to the traditional clustered heatmaps is shown by using a variety of examples.

2 Cluster Analysis

As mentioned above, cluster analysis is a term used to describe statistical methods that classify a set of observations into groups in a way that the observations in a group are related in some meaningful way. Cluster Analysis includes a variety of different algorithms and approaches. In order to focus this discussion we will restrict ourselves to some of the most popular types. In particular, we will only discuss hard-clustering, where each observation is associated with one and only one cluster. Thus, methods like fuzzy clustering [41], that allow observations to be assigned to multiple groups, will not be covered. While such approaches are useful, they are not very popular for the kind of applications we are interested in. Additionally, they typically require further assumptions about the probabilities of belonging to different groups, making them conceptually closer to parametric approaches such as Gaussian Mixture Models (GMM) [42].

On the other hand, we do not want to restrict the possible application types, and so we would like to be as general as possible when we define observations. They are just elements we wish to classify. In the case of gene expression experiments, when we are interested in understanding the relationships between genes, each gene (or more accurately a vector consisting of it expression levels under different conditions) would be considered an observation. In the case of the 1000 cities on the globe we considered in the last chapter, each city would correspond to an observation.

Thus, as far as we are concerned, we will restrict our discussion to the set of algorithms that conform to the following definition: Let us assume we are given a set of N observations $S = \{\boldsymbol{x_1}, \dots, \boldsymbol{x_N}\}$, that either a) are objects existing in space with a well defined distance measure (such as the gene expression profiles, allied with an appropriate distance measure) or b) have a known set of pairwise distances d_{ij} (like the distances between cities on the globe). A cluster analysis algorithm is one that generates a partition $P = \{P_1, \dots, P_k\}$ of S in a way that the patterns in each P_i are maximally alike, while those in different ones are less similar.

This definition is deliberately vague in order to allow us to encompass the different algorithms that are considered to perform cluster analysis. To be more specific, it is instructive to classify cluster analysis algorithms in two ways:

1. By the structure of their algorithmic operation (operational classification)

2. By the property they aim to optimize (optimization-function based classification)

2.A Operational Classification

Here, algorithms are classified in terms of the general structure of the procedure by which clusters are constructed from the raw data. For this, we follow the general classification proposed by Jain *et al.* [43]. In terms of this classification, clustering algorithms can broadly be assigned to two categories, Agglomerative/Hierarchical vs Partitional/Divisional.

Agglomerative/Hierarchical Algorithms

In an agglomerative algorithm, initially each element is in its own singleton cluster. Then at each step the closest clusters are merged. In this way larger and larger clusters are formed. The process can be stopped at an appropriate point to get a desired number of clusters. Based on the order in which clusters are merged, a hierarchy of cluster relations is established. This is typically represented in dendrogram form. Thus, apart from the final assignment into clusters, hierarchical algorithms also provide information about inter-pattern relationships. Since at each step, all pairwise distance between cluster have to be calculated, computational complexity goes as $O(N^2)$, where N is the number of patterns. The various hierarchical algorithms differ in terms of their linkage, i.e., the way in which distances between two intermediate clusters are defined. Some of the most prominent ones are

1. Single Linkage Hierarchical Clustering: The distance between two clusters A and B is the *smallest* pairwise distances with one element in the pair from each of the two clusters:
$$d_{AB} = \min\{d(a,b) | a \in A, b \in B\}.$$

2. Complete Linkage Hierarchical Clustering: The distance between two clusters A and B is the *maximum* of the pairwise distances with one element in the pair from each of the two clusters:
$$d_{AB} = \max\{d(a,b) | a \in A, b \in B\}.$$

3. Average Linkage Hierarchical Clustering: The distance between two clusters A and B is the *average* of the pairwise distances with one element in the pair from each of the two clusters:
$$d_{AB} = \frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} d(a,b).$$

Partitional Algorithms

In the case of partitional algorithms, rather than providing the entire hierarchy of relations, the output is just the assignment of patterns into clusters. This usually means that the number of clusters k needs to be specified by the user in advance. Assignment into clusters is usually achieved by optimizing a scoring function. Since the time requirements of an exhaustive combinatorial search are prohibitive, usually some kind of iterative scheme starting with a random configuration is implemented, and the best scoring configuration over multiple runs is selected. Unlike the hierarchical schemes, partitional algorithms do not compare all possible pairs and therefore implementations with computational complexity of $O(N)$ are possible. The most popular of the partitional schemes is k-means clustering.

Rather than work with the pattern dissimilarities, k-means works with the multidimensional pattern vectors directly. To begin with, centroids for the k clusters are chosen randomly in the space containing the pattern vectors. Then, each pattern is assigned to its nearest centroid (or more precisely, the cluster represented by the centroid). The centroids are then recomputed with the updated membership, and this process is repeated till the memberships of the various clusters reach a fixed point. k-means attempts to minimize the Within Cluster Sum of Squares (WCSS) given by:

$$WCSS = \sum_{j=1}^{k} \sum_{i=1}^{n_j} ||\boldsymbol{x_i^{(j)}} - \boldsymbol{c_j}||^2,$$

where $\boldsymbol{x_i^{(j)}}$ represents the i^{th} pattern in the j^{th} cluster, with centroid $\boldsymbol{c_j}$. It should be clear that a configuration minimizing this will be a fixed point of the dynamics. However, the converse is not true. k-means clustering has a tendency to get caught in local minima, making the initial configuration very important.

A comparison of the agglomerative and paritional algorithms is shown in Table 4.1. Due to its great speed, k-means is the best choice for applications such as computer vision problems, where there are a very large number of points, or where high performance is critical.

Table 4.1: Comparision of Agglomerative and partitional schemes

Criterion	Agglomerative	Partitional
Output	Hierarchical Tree showing relations among all patterns	Assignment of patterns into Clusters
Computational Complexity	$O(N^2)$	$O(N)$
Convergence to True Solution	Guaranteed	Uncertain
Reproducability	Absolute	Depends on Initial Conditions
Example	Complete Linkage	k-means

For the problems of interest to us, we do not know the number of clusters beforehand, and we are interested in the relations between all the elements, not just their assignment into clusters. Given the advances in computing, gene expression data sets, which typically contain $10k$ points, can easily be analyzed with the agglomerative/hierarchical algorithms. We shall therefore primarily focus on these for the rest of this thesis.

n

2.B Optimization Function Based Classification

Another instructive way to classify clustering algorithms is in terms of the cluster properties they aim to optimize:

1. Compactness: The goal is to get the patterns in each cluster to be as similar as possible, and to produce compact and tight clusters. This approach is ideally suited to spherically shaped clusters, but can fail for more complicated cluster shapes, see for example, Fig. 4.1. Examples of Cluster Analyses with this goal include, k-means clustering, average and complete linkage hierarchical clustering.

2. Connectedness: The aim here is the elements close to each other should belong to the same cluster in order to produce a more contiguous structure. Since this criterion is essentially local, it is strongly affected by the properties of a few data points rather than the cluster as a whole, making the results unstable. On the other hand, it is well

suited to arbitrary cluster shapes. Single linkage hierarchical clustering is an algorithm of this type.

2.C Problems with Cluster Analysis

Cluster analyses are the most popular class of unsupervised, data-driven dimensional reduction algorithms. They owe this popularity to their speed, ability to deal with large data sets, and the intuitive and (seemingly) easy to interpret nature of their results. Given the popularity of these methods, it is important to critically evaluate the nature of the results they produce. In this section, we will discuss some of the problems associated with the use of cluster analysis as an exploratory data analysis method. These problems are, for the most part, one of

1. Biases introduced by the underlying assumptions of cluster analysis

2. Lack of stability of cluster analysis results

3. Biases in cluster analysis results caused by discarding of information

We elaborate on each of these below.

Assumed Cluster Properties

As mentioned above, the various cluster analysis algorithms are best suited to different types of clusters. In an exploratory analysis, we usually do not know the structure of the data before hand. Therefore, to explore the utility of cluster analysis as an exploratory tool, it is worth studying its performance when the data does not conform to expectations.

The algorithms that value compactness work best for tight spherical clusters. We therefore consider their performance on more extended distributions as shown in Fig. 4.1. Here two clusters were created, in two dimensions, by sampling points from two highly asymmetric ($\sigma_x/\sigma_y = 20$) normal distributions with the length of the clusters far exceeding their

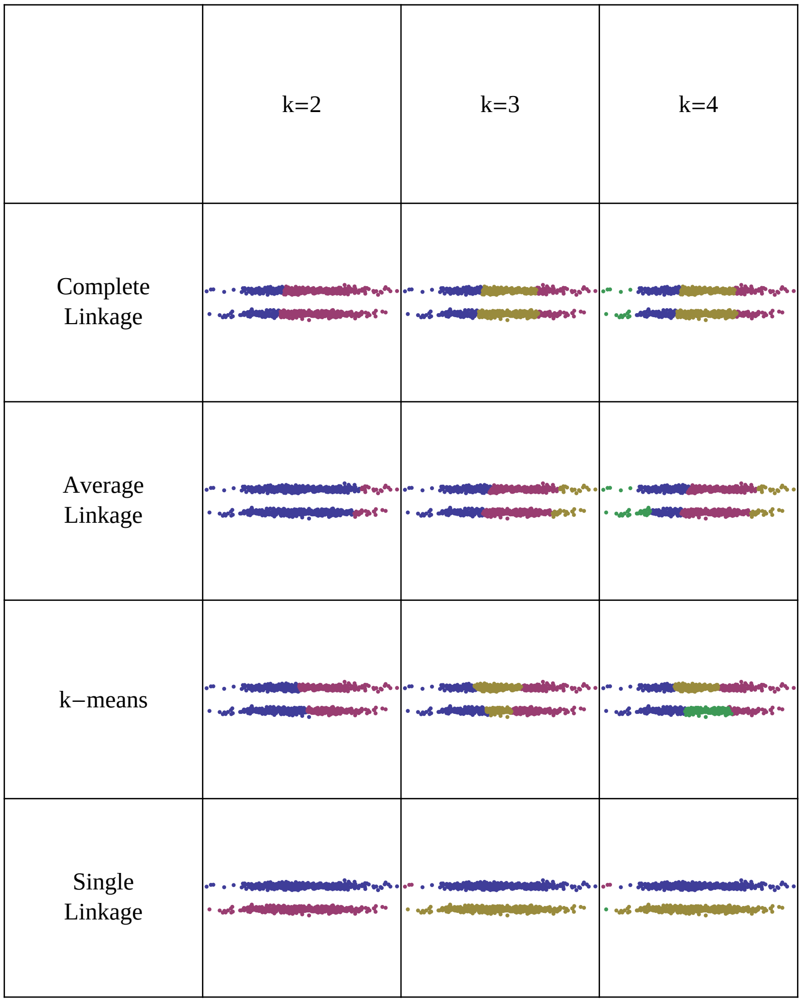


Figure 4.1: Performance of different cluster analysis algorithms for highly asymmetric distributions: The data is sampled from two highly skewed two dimensional normal distributions. Different cluster analysis techniques were applied to this data for multiple choices of number of clusters k. In each plot, points are colored according to the cluster to which they belong

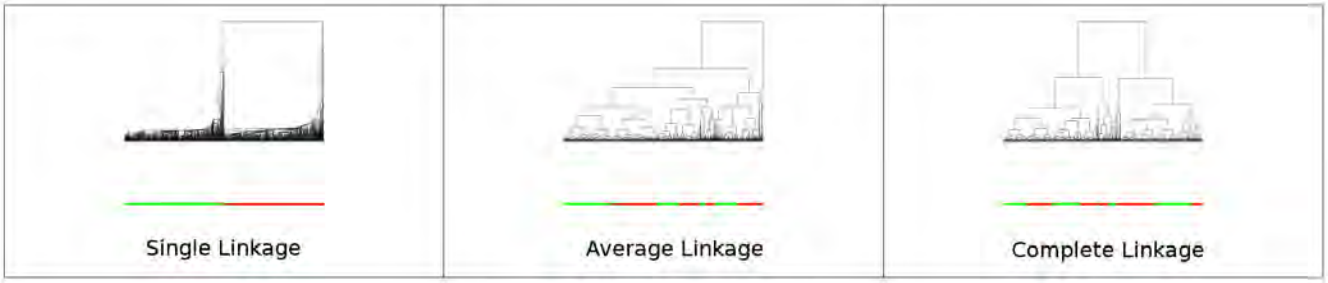


Figure 4.2: Performance of the agglomerative cluster analysis algorithms for highly asymmetric distributions: The data is sampled from two highly skewed two dimensional normal distributions, as in Fig. 4.1. The dendrograms for the three methods are shown. Leaves are colored according to the identity of the Gaussian to which they belong.

separation. It is clear that the algorithms which value compactness fare poorly, while single linkage clustering which aims for connected clusters does much better. This can also be seen from the dendrograms generated by the agglomerative schemes (see Fig. 4.2). In this case leaves are colored according to the true cluster (i.e., Gaussian) that they belong to.

In a real data analysis scenario, without the aid of some other method, we would not know the true distribution of points in their native high dimensional space. The classification of points into the various clusters, or the structure of the dendrogram is all we would have to go by. Neither of these, by themselves, suggest the complete mis-classification that took place in this example. Thus, cluster analysis schemes could potentially give very misleading results if applied to data that does not support the type of clusters that they expect.

Lack of Stability

While it may seem that single linkage hierarchical clustering seems to perform quite well, it is almost never used in real world application on account of its extreme instability. This point is illustrated in Fig. 4.3. Cluster Analysis is applied to four isotropic, well separated, 2D Gaussians, and this is repeated on four instances of points sampled from the same distributions. Average and complete linkage hierarchical clustering along with k-means clustering place points from the four Gaussian in different clusters in all 4 instances. However, single

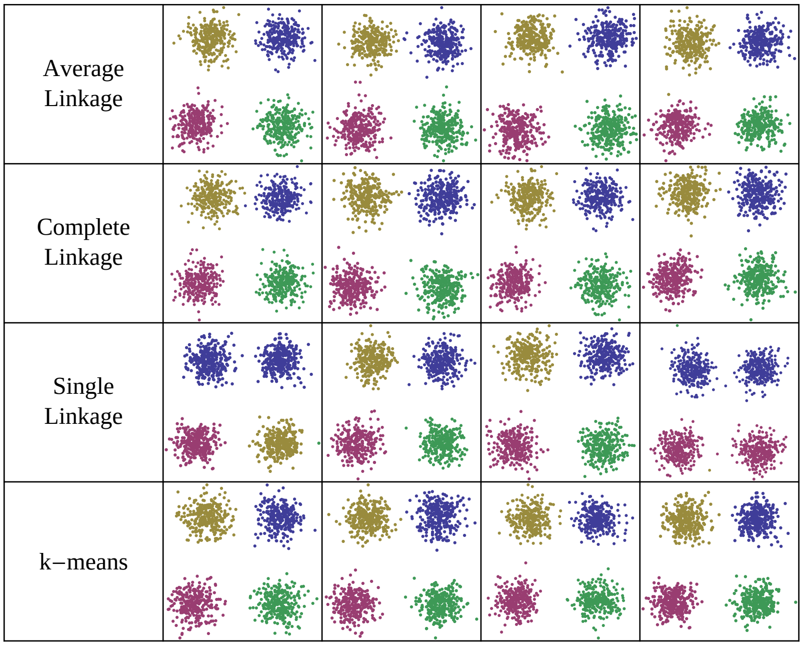


Figure 4.3: Stability of cluster analysis algorithms: Different cluster analysis algorithms are applied to data sampled from 4 equally spaced, symmetric two dimensional Gaussian distributions, and cluster analysis is applied to them, assuming there are 4 clusters ($k = 4$). The different columns correspond to different instances of sampling from the same distributions

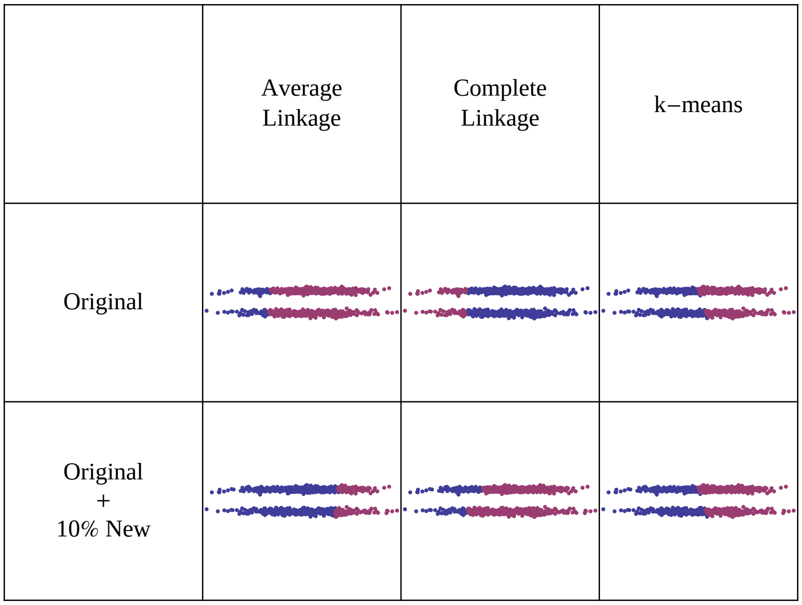


Figure 4.4: Stability Against Addition of New Data: In the upper row, cluster analysis is applied to data sampled from the two asymmetric Gaussians used in Fig. 4.1. In the second row, 10% new points are added sampled from the same distribution, and clustering is repeated.

linkage hierarchical clustering is badly affected by outlying points and ends up joining 2 of the Gaussians in all but one instance.

The good performance of the other methods is due to the fact that they are designed to faithfully identify clusters of this type. If we work with elongated clusters, these algorithms too are unstable. For example, if we consider the two asymmetric Gaussians in Fig. 4.1 and add 10% new points sampled from the same distribution, the cluster results change dramatically (see Fig. 4.4). Thus, cluster analysis does not pass the RG based stability criterion in this case.

Discarded Intra-Cluster Information

The partitional algorithms do not provide information about the relations between patterns, except the cluster they have been assigned to. It may seem that using the dendrogram produced by agglomerative schemes, we may infer the relations between arbitrary patterns,

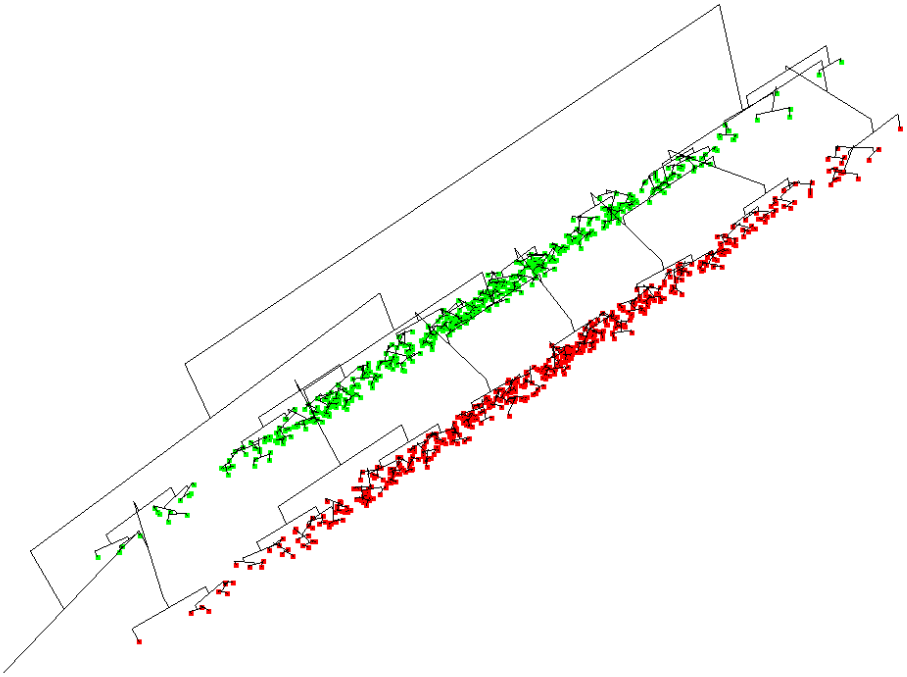


Figure 4.5: Discarded Clustering Information: The complete linkage dendrogram result is superimposed on the data. cluster analysis is applied to data sampled from the two asymmetric Gaussians used in Fig. 4.1. In the second row, 10% new points are added sampled from the same distribution, and clustering is repeated.

but this is misleading. Consider an intermediate stage of agglomeration. At this stage, many patterns have already been combined into clusters, and the algorithm will proceed by coalescing the two closest clusters. The definition of closest used is determined by the kind of linkage used. All of the linkage schemes make use of some pre-determined statistical relation between the two clusters as a whole, and ignore all other information regarding the relations between individual elements. More seriously, when the two clusters are combined, there is no way of knowing how close elements from one of the original clusters is to those in the other clusters. Only relations at the cluster level are preserved, and within cluster data is lost. This essentially comes from the belief that intra cluster distances are much smaller than inter cluster ones, making such relations meaningless.

However, this approach can have unintended consequences, when the above assumption is not true. For example, Fig. 4.5 shows the cluster analysis dendrogram superimposed on the data points for the same data that was used in Fig. 4.1 (this was generated using the NeatMap package to be discussed later in the chapter). At various locations, the dendrogram shows bridges connecting the two Gaussians. However, it is obvious from the picture that there are points within the same Gaussian that are closer to the bridge head, and which one might expect should be connected first. This is caused precisely by this effect described above. Looking at the dendrogram itself, the information about the nearness of such points is completely lost. The biological consequences of this will be discussed in conjunction with NeatMap.

2.D Correct Usage of Cluster Analysis

Although we have pointed out various problems faced while using cluster analysis, these are largely due to the use of cluster analysis to perform tasks it wasn't intended to rather than due to inherent problems with cluster analysis itself. The goal of cluster analysis is to assign patterns to groups. It is therefore implicitly assumed that these groups are meaningful, and sufficiently different from each other. Cluster Analysis was not intended to provide a compact depiction of the relations between the various patterns, and it is ill-suited for this task. However, for exploratory analysis of biological data, often this is precisely what we need. This makes cluster analysis a poor choice for such tasks. Yet, because of their simplicity and speed it is very tempting to attempt to glean some information from the application of cluster analyses. While this is a potentially dangerous approach, there are ways of minimizing the risk of producing misleading results [44].

1. Check for clustering tendency: Possibly using PCA/MDS or specialized measures for this task. If no clustering tendency is seen (which is the case for most gene expression data), cluster analysis should not be applied.

2. Selection of algorithm: The appropriate clustering algorithm for desired task must be

chosen.

3. Validation of results: There are a variety of approaches that validate the results produced by cluster analysis. Some of these repeat the clustering and check for robustness, others provide measures of cluster quality. Results that are poor should not be used.

4. Comparison to other methods: Cluster Analysis should be compared using different algorithms and their results should be validated using other methods such as nMDS. The NeatMap package provides an easy way to do this.

5. Do not overextend the results. The cluster analysis results should primarily be used to group patterns into clusters. Any information extracted beyond this should be considered unreliable.

Unfortunately, it is hard to find bioinformatics papers that use cluster analysis without violating at least a few of these points.

3 Heatmaps

3.A What is a Heatmap?

Results of experiments are often measurements of properties of objects under different conditions. For example, in gene expression experiments, the expression levels of multiple genes are measured across different conditions (e.g., times, tissues, etc.). Such data may be easily represented in matrix form, with each row corresponding to a single gene, and each column a specific condition under which the expression levels of genes are measured. The heatmap is just a visualization of this matrix in color form, with each matrix cell colored according to its expression level.

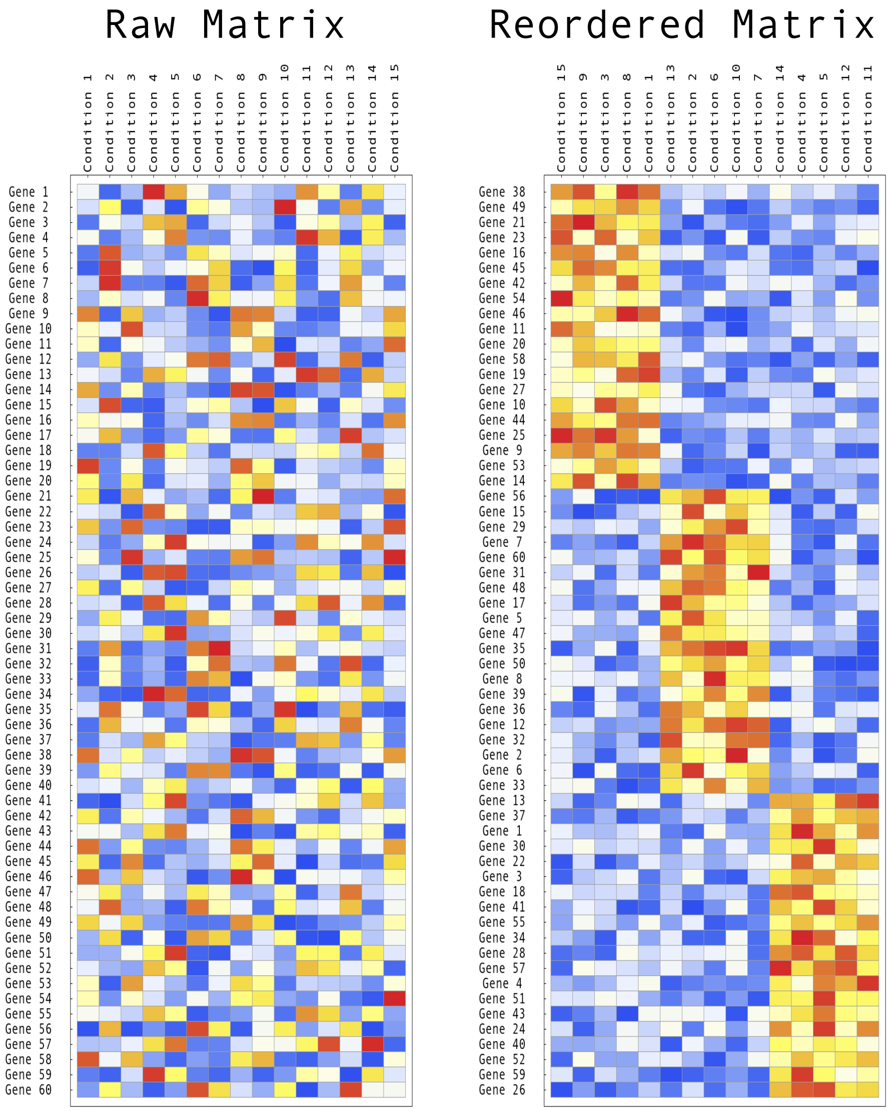
Figure 4.6 shows two such matrices. As the heatmap on the left shows, such a visualization by itself typically does not provide much insight. However, if the rows and columns of the matrix are re-ordered so that similar ones are placed close to each other, then the patterns supported by the data are far clearer. In the example in Fig. 4.6, such reordering has been performed to produce the heatmap the right. Now, it is clear that there are three groups of genes each showing high-expression in a different set of genes.

As mentioned in the introduction, the (clustered) heatmap is the most popular visualization scheme for gene expression data. There are good reasons for this popularity:

- Unlike the methods we used to visualize the nMDS and cluster analysis, both the relations between the genes and those between conditions can be seen at the same time (in the sense that similar genes/conditions should be closer to each other in the heatmap ordering).

- It allows us to visualize the entire data set, not just the dimensionally reduced results. In the traditional visualization, the profiles underlying the dimensionally reduced results are not seen.

- The large contiguous bands of color produced by heatmaps encourage the formulation



Raw Matrix

Reordered Matrix

Figure 4.6: A schematic representation of a heatmap: The matrices above show the expression of 60 genes across 15 conditions. Blue and red represents low and high gene expression respectively. The figure on the left is meant to be a matrix of data drawn from some experiment. The figure on the right is the same matrix with the rows and columns reordered to best represent the data. In this case, after reordering it is clear there are 3 different groups of genes, each of which is over-expressed in a different set of conditions. This insight was not evident before reordering.

of general relations between the variables being measured (for example, see Fig. 4.6). This is, after all, the goal of visualization methods.

- It is visually striking (the importance of this should not be underestimated)

3.B Problems with the Clustered Heatmap

When the reordering of the rows and columns of a heatmap is achieved using cluster analysis, it is known as a clustered heatmap. Since the relations between all elements are important, some kind of agglomerative scheme is usually chosen. While we believe heatmap type visualizations are very powerful, there are reasons to be sceptical about their implementation via cluster analysis:

1. As the astute reader would have noticed that cluster analysis does not produce an ordering of elements. This is the fundamental problem of clustered heatmaps. The result of an agglomerative cluster analysis is a dendrogram. In a clustered heatmap, the ordering of dendrogram leaves for the genes (conditions) is used to order the rows (columns) of the heatmap. However, this leaf ordering is not specified by cluster analysis. In fact, for a given cluster analysis result, the ordering of leaves may be changed by swinging the arms of the tree at each bifurcation. Closeness in leaf order is not the same as closeness in dendrogram. The nearness of two leaves in a dendrogram is defined as the distance between them *along the tree*, not along the branch tip ordering. While these measures are related (especially for very similar elements), they could be very different [45]. Thus, the ordering of the branch tips does not respect the intrinsic topology (if any) of the data, making it a poor choice for use in a heat map.

2. To compensate for these failings 'swinging' based reordering using an independent method is often required, post-clustering, to capture the structure of the data. However, these methods are quite dangerous because

- Unlike the clustering schemes, the reordering algorithms, while complex enough to warrant dedicated software packages, are often not elaborated upon or even stated, thereby reducing the reproducibility of the result.
- Such procedures could potentially place (deliberately or otherwise) objects that are distant along the tree in close proximity in the row/column order. Heat maps are commonly read in this order rather than by their dendrogram structure (if this were not the case, such reordering schemes would not be needed). Effectively a spurious pattern could be created, leading to incorrect results (e.g., see clustered heat map for Spellman data in Results)

3. On top of this, all the problems inherent in cluster analysis (as discussed earlier) are also present:

   - During clustering, when objects are assigned to different clusters, further analysis essentially involves these clusters as a whole, and the relationship between the elements themselves is lost (see analysis of human gene atlas in Results)
   - When the assumptions made by cluster analysis are violated the results become unpredictable as shown earlier. In particular cluster analysis results are unreliable when there is no clear clustering/grouping tendency.
   - As suggested earlier, it is considered good practice to test for clustering tendency before performing clustering or to perform bootstrap-like methods to estimate cluster quality post-clustering [44]. Unfortunately, this kind of information is not typically provided in a heat map. Thus, validation is only by visual inspection of the color patterns, and this may be misleading.

For all these reasons, we would like to have a heatmap-like visualization method that does not make use of cluster analysis.

3.C Replacing the Clustered Heatmap

Biological data often has a low dimensional structure that may be visualized as a spatial pattern, so direct use of a suitable dimension-reducing algorithm could, in many cases, be more natural and better characterize the data than the current combination of structure destroying clustering + restoring algorithm. There are many such algorithms whose utility in the analysis of biological data has been demonstrated [46, 24]. Multiple packages (for example, in R [47]) implement them. Despite this, we believe their use has been limited, at least partially, by the lack of associated visualization methods with the visceral impact of the clustered heat map.

Here, we present an R package called NeatMap to meet this need while addressing some of the deficiencies of the clustered heat map. It consists of novel plot-types in two and three dimensions intended to be used in conjunction with any dimension-reduction scheme capable of embedding results in low dimensional Euclidean space (e.g., Principal Component Analysis (PCA) and Multi-Dimensional Scaling (MDS)). This places weaker constraints on the data than does cluster analysis, which requires the data to exist in an tree space. Like the heat map, and unlike typical visualization schemes for these methods, NeatMap displays the entire dataset underlying the result. It also has provisions to superimpose the cluster analysis results, for mutual validation. This feature is not commonly implemented in software packages, and our implementation is more informative about individual points than existing implementations [48]. Also note that unlike the clustered heat map, the layout of the plot is almost entirely determined by the output of the dimension-reduction scheme, thereby respecting the intrinsic structure in the data more than would a clustering based reordering.

There are a number of alternatives to hierarchical clustering (see, for example, the R package **seriation** [2]) designed specifically to produce an ordering that reflects the relative relations between elements. NeatMap is a visualization method, and in general it is not intended to compete with these (in fact they can easily be used in conjunction). However, some of these techniques involve ordering by the first component of PCA/MDS. Unless, this component captures most of the relevant information, NeatMap, which uses 2D embeddings, is likely to better utilize the dimensional reduction results.

3.D Availability and Requirements

Project name: NeatMap

Project home page: http://cran.r-project.org/web/packages/NeatMap/index.html

Operating system(s): Platform independent

Programming language: R

Other requirements: R, R packages(ggplot2 and rgl)

License: GPL-3

3.E Neatmap Implementation

The general class of data considered involves factors (e.g., genes) being measured across multiple conditions (e.g., samples, times, tissues, etc.). For each factor, these measurements will be referred to as its profile. It is assumed here that some dimension-reduction scheme, (e.g., PCA) has been used to depict the relationship between factors by embedding them into a 2D Euclidean space. The plots described here allow us to visualize these relationships, while simultaneously showing the profiles underlying them. non-Metric MultiDimensional Scaling (nMDS) [24] was used as the dimension-reduction scheme for the demonstrations in this chapter, since, generally speaking, the embedding produced by nMDS is more informative than the corresponding PCA result. An R implementation of nMDS is included for convenience in the package. There are multiple plots in this package, each emphasizing different aspects of the factor-condition relationship:

1. **heatmap1**: This is the traditional heat map, except a dimension-reduction scheme other than clustering (for examples see [2]) may be used for ordering of rows and/or columns. Neatmap itself provides a novel way to do this from a 2D embedding method: normalize the data, or use an amplitude neutral distance measure such as the Pearson correlation. Then, the embedded result produced by PCA, nMDS, etc., is often annular and can be parameterized, approximately, by a single variable, viz., the angular position (Fig. 4.7d). This is a better option than using the ordering based on a single component. The standard cluster dendrogram may be superimposed on the heat map for mutual validation.

2. **circularmap**: Similar to **heatmap1** except the arrangement is circular (Fig. 4.7e) rather than linear to emphasize the periodicity of the angular positions obtained as above (or using other methods [3] that produce annular results). It is easy to make comparisons across conditions and factors. The factor clustering result may be superimposed on this plot.

3. **lineplot**: The 2D dimensionally-reduced factor relationship result is gridded, and the profiles of all the factors within each grid cell are displayed together as line graphs (Fig. 4.7c). This provides a global understanding of the nature of the data and its embedding. However, individual factors are harder to pick out, and comparison across conditions is more difficult.

4. **profileplot3d**: Addresses the inability of **heatmap1** and **circularmap** to depict radial information by visualizing the profiles in a 3rd dimension using a rotatable 3D environment (Fig. 4.10c).

5. **draw.dendrogram3d**: Cluster validation of the 2D embedding result for factors (Fig. 4.10b) in a 3D environment. The clustering result for both factors and conditions may be superimposed on **profileplot3d**.

The functions above are dimension-reduction method neutral; dimensionally-reduced results provided by the user are plotted. Convenience wrapper functions **make.heatmap1**, **make.circularmap**, and **make.profileplot3d** are also provided. They take just the raw data as input, perform dimension-reduction using either nMDS or PCA, and finally produce the appropriate plots.

All 2D plots were implemented by using **ggplot2** [49] and 3D plots using **rgl** [50]. These libraries have numerous functions for additional customization and modification of the plots produced by NeatMap.

3.F Results

The utility of the plots described above are demonstrated by using two different micro-array-based datasets. The 2D plots are illustrated using the Spellman *et al.* [1] dataset identifying cell cycle related genes in yeast, while micro-array data from the human gene atlas study [51], profiling gene expression across multiple tissues, is used for the 3D plots.

2D plots

Spellman *et al.* [1] produced genome-wide time course profiles in yeast using micro-arrays under different synchronization methods. Fourier analysis was then used to identify genes with the correct periodicity as cell cycle related. We consider these cell cycle related genes and study their profiles under *alpha* synchronization. Since a natural time ordering of the measurements exists, we are only interested in the relationship between genes.

For comparison to the plots produced by **NeatMap** we used the Multiexperiment Viewer (MeV) software to generate the standard clustered heat map for this data (Fig. 4.7a). Average linkage hierarchical clustering of the Pearson correlation, followed by MeV's function for optimal reordering of genes were used. Although the periodicity of these genes is clear, and locally good groupings are seen, the pattern as a whole appears quite jagged. This is because a cluster like topology was forced on an essentially continuous distribution. Closely related

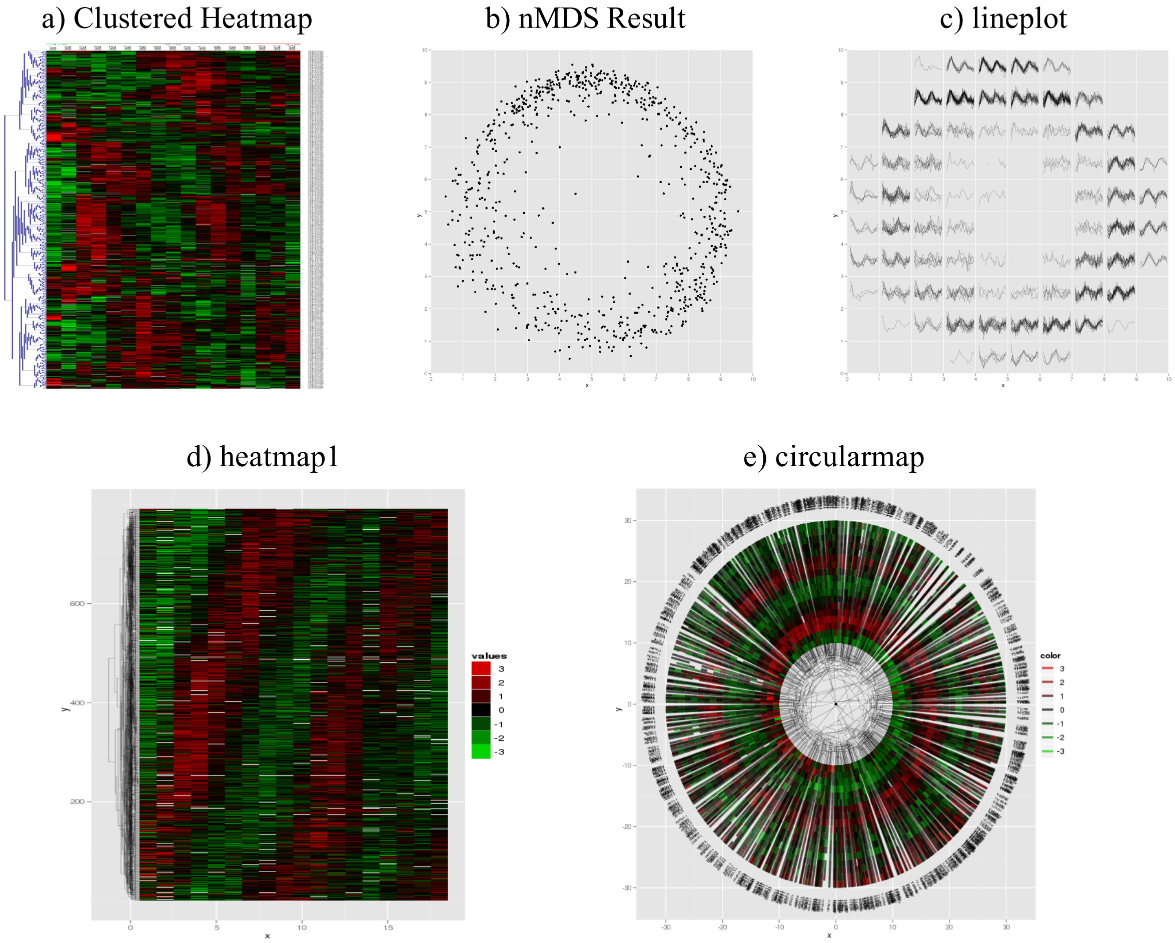


Figure 4.7: Different ways of representing the cyclic genes for the alpha experiment in Spellman *et al.* [1]. (a) is the standard heat map using average-linkage hierarchical clustering (+rearrangement) in MeV, shown here for comparison. (b) is the result of 2D nMDS. The profiles for all the genes in each grid cell in (b) are shown using **lineplot** in the corresponding grid cell in (c). (d) shows **heatmap1** with the angular positions of genes in (b) used to reorder the rows in (a). (e) is **circularmap** using the angular positions of points in (b).

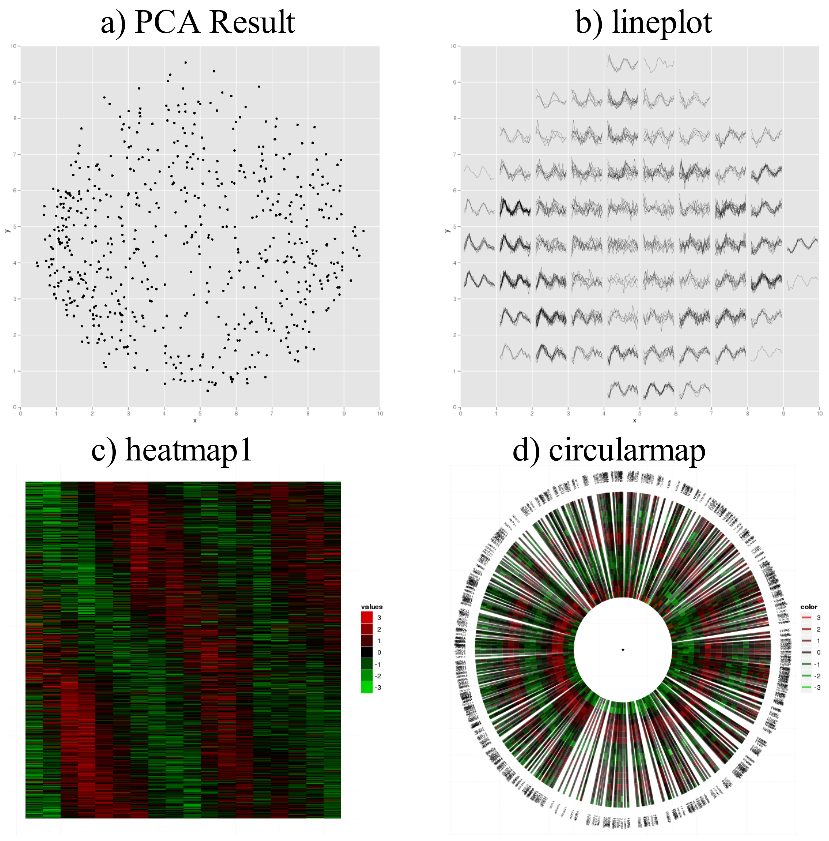


Figure 4.8: The NeatMap plots in Fig. 4.7 produced using PCA instead of nMDS. Spellman *et al.* data using *alpha* synchronization was visualized using PCA and NeatMap. The profiles were normalized to have zero mean and unit variance, and all profiles with missing data were discarded. (a) is the standard PCA result, (b), (c) and (d) show the **lineplot**, **heatmap1** and **circularmap** functions respectively applied to (a).

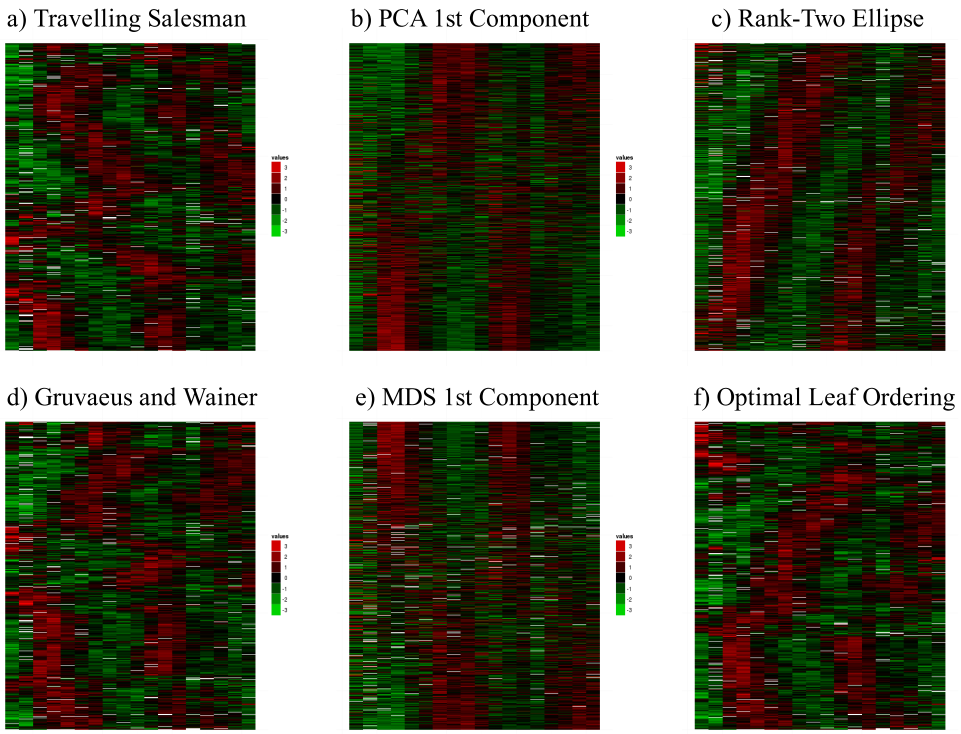
groups of genes are correctly clustered together but the global relations between genes in different clusters (which is essential for complete ordering) are lost. Fig. 4.7b shows the result produced by a 2D embedding of the gene profiles using nMDS, again with the Pearson correlation. A clear continuous ring like pattern emerges naturally. (PCA, with normalized profiles, shows a similar result although the ring structure is more diffuse; see Fig. 4.8)

Such a ring-like structure is very common when an amplitude-normalized distance measure such as the Pearson correlation is used. In this situation, it is natural to parameterize the position of a gene by a single angle. This is what **heatmap1** does. For each gene, its angular position in the nMDS result (Fig. 4.7b), with respect to its center of mass, is determined, and the profiles are placed (Fig. 4.7d) in a standard heat map ordered according to this angle. The periodic nature of the profiles is now clear, and it is evident that points are arranged by time of up-regulation; essentially the cell cycle phase in which the gene is expressed. Note that **heatmap1** also accepts orderings produced by other methods. The R package **seriation** [2] offers a variety of these, and **heatmap1** plots using them for the Spellman data set are shown in Fig. 4.9. In general, the NeatMap ordering is superior, except for the case of Rank Two Ellipse [3]. This method, like NeatMap, uses angular ordering based on normalized profiles (the correlation matrix itself in this case). **heatmap1** also allows the superimposition of clustering results. Evidently, the local arrangements in nMDS and clustering are consistent. Large scale rearrangement, produced by incorrect ‘swinging’, however, makes the clustered heat map result seem poor.

There are some long lines in the gene clustering result in Fig. 4.7c spanning the entire length of the heat map. This is a consequence of the periodicity of the angular variable, which results in the two opposite ends of the heat map being almost identical. To avoid artifacts from this periodicity, one may use **circularmap** (Fig. 4.7e). The ordering of profiles is identical to **heatmap1**, except they are placed along a circle according to their angular positions in Fig. 4.7b. One additional advantage of this format is that the non-uniformity in the phase distribution stands out more clearly. It is much harder to gain this type of information from a traditional heat map display.

Fig. 4.7c shows the **lineplot** based on the nMDS result in Fig. 4.7b. As explained earlier, each cell in the grid in Fig. 4.7c shows the time course profiles of all the genes in the corresponding cell in Fig. 1b. The sinusoidal nature of the profiles is much clearer in this plot. It also emerges that the radial coordinate in this case is a measure of ‘cyclicity’, with the genes close to the centre being less cyclic.

The **lineplot** emphasizes the overall nature and change in profiles with position. However, compared to **heatmap1** and **circularmap**, comparison of expression at a fixed time



a) Travelling Salesman

b) PCA 1st Component

c) Rank-Two Ellipse

d) Gruvaeus and Wainer

e) MDS 1st Component

f) Optimal Leaf Ordering

Figure 4.9: **heatmap1** may be used in conjunction with orderings produced using external algorithms. The R package seriate [2] contains a number of these. **heatmap1** using the Spellman data [1] and different ordering schemes using seriate are shown in the figure. a) uses the Travelling Salesman Algorithm, b) orders rows according to the first component of the PCA embedding of the rows, c) is ordering according to elliptic ordering method proposed by Chen [3], d) by the method proposed by Gruvaeus and Wainer, e) by the 1st component of the MDS embedding of rows, f) by the Optimal Leaf Ordering algorithm.

across genes is more difficult. It is also more difficult to quickly look up a specific gene. On the other hand, **heatmap1** and **circularmap** are intended for essentially one dimensional results. To deal with the more general case we must use 3D rotatable plots.

Assuming the profiles are stored in matrix form in `alpha.profiles`, the code to produce Fig. 4.7 c, d, and e (except for specific graphics options) is:

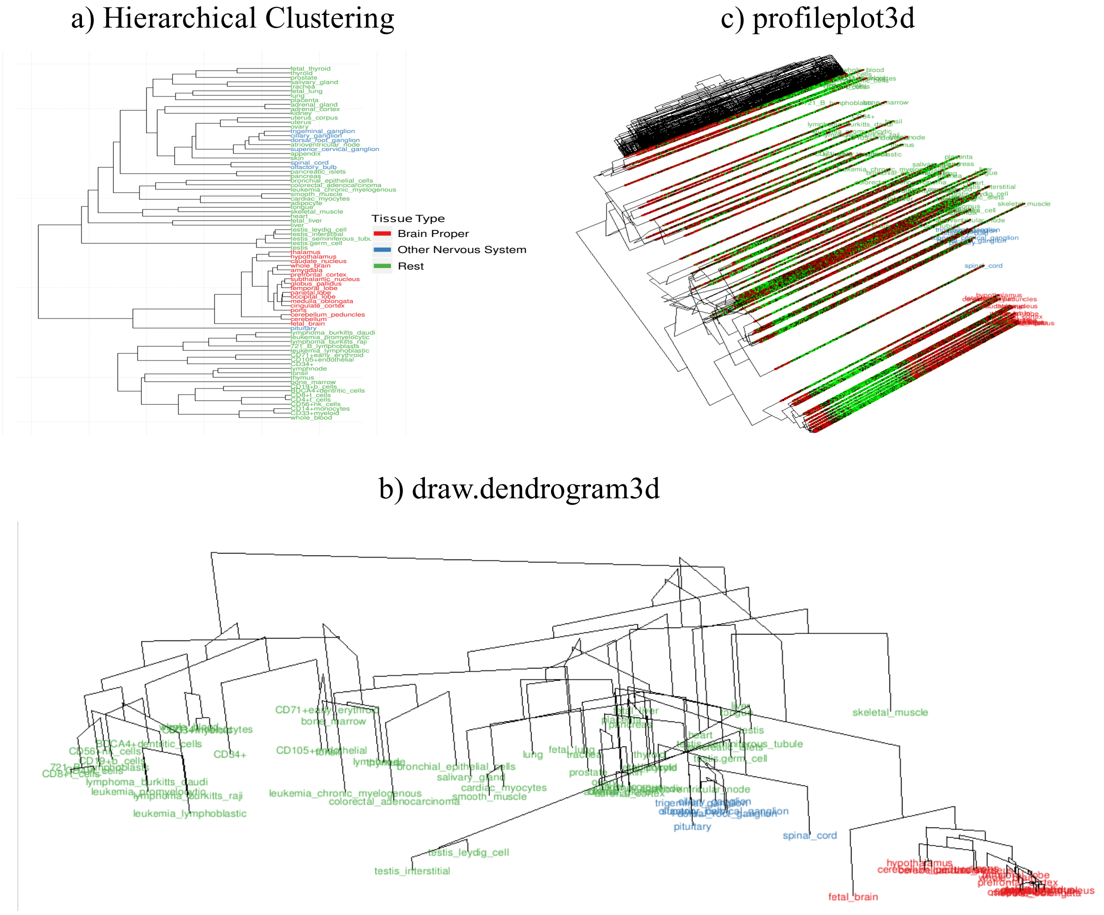
```
pos.nMDS<-nMDS(alpha.profiles)$x;# Perform nMDS embedding
lineplot(pos.nMDS,alpha.profiles,normalize=T); #1c
make.heatmap1(alpha.profiles,row.normalize=T); #1d
make.circularmap(alpha.profiles); #1e
```

To use PCA instead of nMDS, a single parameter specifying this would need to be added to each of these plots.

3D plots

We illustrate the 3D plots using the gene atlas dataset. Su *et al.* [51] used microarrays to analyze the expression profiles of genes in a variety of tissues in both humans and mouse. There is no natural ordering of the genes or tissues, but the relationships between tissues are more easily understood. We therefore primarily focus on these.

Since, in the present context, we are not interested in cross-species comparison, for this demonstration only human data was used (mouse gives similar results). The 1000 genes on the HG-U133A array showing largest variance across the 79 tissues were analyzed. Functionally, there are broadly 3 groups of tissues: those from the brain proper, some nervous system related tissues, and those from other parts of the body. The result of applying hierarchical clustering (average-linkage) using the Pearson correlation to the tissues is shown in Fig. 4.10a. Three distinct clusters are seen, one of which is composed solely of brain tissues. However, the nervous tissues are mixed with the other non-brain tissues in the second cluster and no relation to the brain can be gleaned from the leaf order or distance along the tree.



a) Hierarchical Clustering

c) profileplot3d

b) draw.dendrogram3d

Figure 4.10: Representations of the human gene atlas data: (a) is the average-linkage hierarchical clustering (using pearson correlation) result applied to the tissues, (b) shows the superimposition of the clustering result on a 2D nMDS embedding of tissues using **draw.dendrogram3d**, (c) shows the expression profiles underlying (b) using **profileplot3d**.

A 2D embedding of the same data using nMDS with Pearson correlation was also performed. The cluster analysis result was superimposed on the 2D nMDS result in a rotatable 3D environment using **draw.dendrogram3d** (Fig. 4.10b). The same 3 clusters are present, and there is broad agreement between the clustering and nMDS results. Unlike the clustering result, however, the relationship between the brain and nervous system tissues is much clearer. The nervous system genes are also quite similar to the central cluster of tissues in Fig. 4.10b. Apparently, cluster analysis assigns them to this cluster, and in doing so their relationship to the proper brain tissues is lost.

The profiles underlying the nMDS result may be displayed in a rotatable 3D environment by using **profileplot3d**. Fig. 4.10c shows this with the cluster analysis results for genes and tissues superimposed on it. The genes were ordered according to their angular positions in a ring-like nMDS embedding using the Pearson correlation, much like **heatmap1**. The separation between the 3 groups of tissues can be seen as before. However, **profileplot3d** makes it clear that there are different set of genes up-regulated in these groups.

Assuming the data is stored in matrix form (with genes along the rows and tissues along columns) in `atlas.profiles`, the cluster analysis result for tissues in `alpha.cluster`, and the three groups are color coded in `alpha.group.colors` the code to produce the plots in Fig. 4.10 are:

```
atlas.nMDS<-nMDS(profiles)$x;

draw.dendrogram3d(atlas.nMDS,atlas.cluster,labels=colnames(atlas.profiles),

label.colors=alpha.group.colors);

make.profileplot3d(alpha.profiles,column.method=‘‘nMDS’’,

labels=colnames(atlas.profiles),label.colors=alpha.group.colors);
```

4 Conclusions

Cluster Analysis is by far the most popular exploratory data analysis method for analysis of gene expression. In this chapter, we have critically studied its performance and found that, except when there is a very clear grouping tendency in the data and within group relations are not of interest, cluster analysis is a poor choice. Despite this, cluster analysis has enjoyed a great deal of popularity, at least partly because of the popularity of the clustered heatmap as a visualization method. We have shown that while heatmaps are a powerful visualization technique, the use of cluster analysis to drive them is deeply flawed. To encourage the adoption of alternate data-driven dimensional techniques such as nMDS, we have created an R package called NeatMap where heatmap like plots are generated using such techniques in preference to cluster analysis. Using the well-known Spellman yeast cell-cycle and human gene atlas microarray datasets, we have shown that a dimension-reduction method (nMDS was used in this paper for illustration) in conjunction with **NeatMap** is more informative than the clustered heat map. It is hoped that this package will increase the popularity of these methods and spur the development of novel visualization schemes.

Chapter 5

Noise Reduction

1 Introduction

In the last two chapters we have discussed how dimensional reduction techniques can be used to produce low dimensional representations of biological data that are then amenable to human interpretation. In working with a low dimensional representation, rather than the raw data, one potentially runs the risk that the results are biased by the idiosyncrasies of the dimensional reduction algorithm. Most popular algorithms have been tested enough that we can be reasonably sure that if there is a very clear low dimensional structure present it will be extracted correctly. However, if this structure is hazy, things are more problematic. In the context of biological data, we expect related genes to act in concord and thus to be well captured by a low dimensional representation. Thus, if a low dimensional representation is not found, it is likely that either a) the genes represent multiple processes and/or b) the data is noisy. In the analysis of high-throughput experiments, both problems are ever-present; a whole genome study undoubtedly will capture genes involved in multiple processes, and microarray type experiments are notoriously noisy. The way to deal with the first problem is to perform some kind of pre-selection to isolate genes belonging to a single process before performing dimensional reduction. This best way to do this will be discussed in the next chapter where we show a data driven method to identify genes involved in different processes. The focus of this chapter will be on dealing with noisy data, and for the most part we shall assume that pre-selection has already been performed.

High-throughput biological experiments are very noisy. So it is natural to expect that not all data points carry the same amount of biological information. When applying dimensional reduction to such a noisy data set, it is quite possible that the noise makes the low dimensional law diffuse, thereby hampering performance. Our idea is that if there were some way to remove the more noisy/less informative points preferentially, then the dimensional analysis result with the remaining points would be more meaningful. This point is illustrated in Fig. 1.

Our method to identify noisy points makes use of the belief (assuming pre-selection has been performed reasonably well) that genes will act in concert and so biologically meaningful points should satisfy the extracted low dimensional law. Thus, points that are not consistent with the low dimensional law are far more likely to be noisy ones. In this way, noisy points may be identified and removed. From the point of view of constructing a phenomenological theory, one may say that the goal of dimensional reduction is to extract the universal structure. This should not depend on details, and must therefore be low dimensional. Thus, aspects which are not low dimensional are likely to be microscopic details we are not interested in.

This procedure pre-supposes that we have the correct low dimensional law. If the data is very noisy to begin with, this may not be the case. We therefore implement our noise removal procedure iteratively. The dimensional reduction is performed, and the most noisy points are removed with respect to this result. The data sans the removed points is dimensionally reduced again, and the noise removal procedure is performed once more. This is repeated until no improvement is possible by noise removal.

Since the goal of the procedure is to improve the dimensionally reduced result, we shall refer to it as *data honing*. The utility of honing is first demonstrated using the artificial noisy example shown in Fig. 1, where the true low dimensional structure is known. The nMDS based honing procedure successfully identifies this structure. Then we show how nMDS + honing may be used to improve real biological data illustrated with the micro-array data produced by Cho *et al.* [52] .

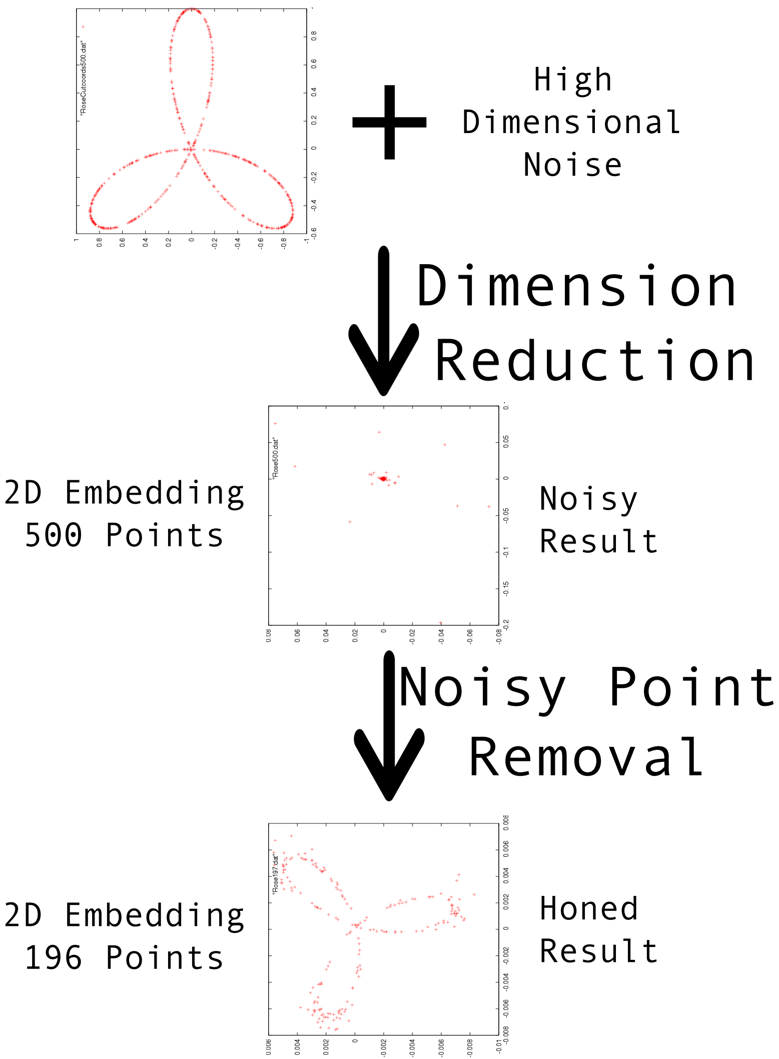


Figure 5.1: Basic Idea of the Honing Procedure: We start with a 3 leaf clover shaped 'law' in 2D, to which we add a high-dimensional noise giving us 500 points in a 10 dimensional space. This is embedded back into 2D, but the correct pattern is not recovered. A noise removal procedure is applied to remove noisy points, and the 197 point pattern resembling the 3 leaf clover is recovered

The reader may wonder, if the pre-selection is meaningful, why we do not simply perform pre-selection with a more stringent cutoff, thereby removing noisy data points. The answer is that except for the ICS Survey method we propose in the next chapter, there are very few data-driven pre-selection methods. The methods that do exist risk strongly biasing the results. Even the ICS Survey only works in a multiple experiment setting, and that too for only the most prominent processes. For the cases when biological information is available we propose a more data and biology driven method to perform this pre-selection, and show how honing can be used for further improvement and reduction of possible biases. This is applied to the well known cell cycle related micro-array data set by Spellman *et al.* [1]. We show that honing improves the biological interpretability of the data, and allows us to determine the set of cell cycle related genes in a more reliable fashion.

2 Method

2.A Basic Idea

The basic assumptions here are that genes in related processes work collectively and cooperatively. Thus, the patterns detectable from the gene expression data correspond to biologically meaningful collective behaviors. These assumptions are likely to be violated if we consider whole genome observations. In this case, some means of pre-selection must be used to extract a subset where the behavior of interest is the dominant feature. In the following discussion we assume this has already been done. Therefore, the basic idea of (data) honing is as follows:

(I) Applying a pattern extraction method to the data set, the pattern supported by the data is extracted. Then,

(II) The consistency of the individual data points to the extracted pattern is determined.

(III) Less consistent data points (called diffuse points) are removed, and the pattern becomes sharpened (honed).

An iterative scheme can be devised removing only a fraction of the diffuse points at a time and restarting the pattern extraction step with the remaining data points until no honing is needed (as determined by some statistical criterion).

Step (I) is accomplished by expressing the mutual relationships among genes geometrically in a certain (low-dimensional) space. We use Pearson correlation coefficient of their expression profiles to quantify the similarity between two genes. nMDS, as described in Chapter 3, is used to find a configuration of points in an Euclidean space E such that the distance between the points in E is consistent with the similarity of the corresponding genes. In short, nMDS gives a visual representation of the closeness of gene expression profiles.

Although honing is discussed in the context of nMDS, the basic procedure is essentially independent of the data reduction method. In Appendix C, we illustrate the honing method with PCA instead of nMDS. Experience suggests that although nMDS is computationally more demanding than PCA, it is generally a more powerful pattern extractor/dimensional reducer than PCA; if PCA works, nMDS certainly works, but not vice versa.

In Step (II) for each gene, we use a measure of its consistency with the nMDS result, in terms of its local ranking mismatch. Step (III) requires a criterion to judge what mismatch is sufficiently bad. This is realized with the aid of a kind of bootstrap method in a data driven fashion. Genes with significant mismatches are removed by this statistical method. The procedure is iterated to extract robust patterns supported by the original data.

2.B Identification of Noisy Points

nMDS is used to embed the gene expression profiles in a Euclidean space E of desired dimensionality d as discussed in the last chapter. In practice, it is found that certain points are consistently embedded much more poorly than others. To identify such points, a local quantity called the (rank) mismatch is used. The mismatch $\Delta(i)$ for a point i is defined by

$$\Delta(i) = \sum_{j \neq i} [R_i(j) - r_i(j)]^2,$$

where $R_i(j)$, as defined in the last chapter, is the ranking of $\delta(i,j)$ (i.e., the dissimilarity between the i^{th} and j^{th} points) among the dissimilarities of all the points to the i^{th} point. $r_i(j)$ is the corresponding ranking in the embedded result. With this definition, steps (I) and (II) outlined above can be implemented.

A gene with a large rank mismatch implies that it is difficult to place the corresponding point in a low dimensional space consistently with the given data. If we assume that the obtained pattern captures important features of the data set, it is sensible to discard poorly embedded points (diffuse points). To this end, an objective criteria is required to remove them. Typically, there are no points that are perfectly embeddable; a continuum of mismatch values is seen. Therefore, appropriate statistical tests must be formulated in order to discard only points failing them. Here, we outline an iterative method, the *bootstrap scheme*.

2.C Determining How Many Points to Discard

Bootstrap Scheme

Suppose we wish to discard $s\%$ of the points with the worst rank mismatches. Since the rank mismatch is a stochastic variable, even if a particular gene has a mismatch ranking within $s\%$ (s should not be too small; usually 5 or 10) from the worst end, we cannot immediately be sure that we may discard this gene. To judge the reliability of its mismatch ranking, $(100-s)\%$ of genes are sampled randomly and embedded together with the gene. Repeating this procedure, a bootstrap distribution of the mismatch ranking of the particular gene is generated. If the mismatch ranking distribution of the worst $s\%$ genes and that of the next worst $s\%$ are not statistically distinguishable, there is no reason to discard the worst $s\%$ as

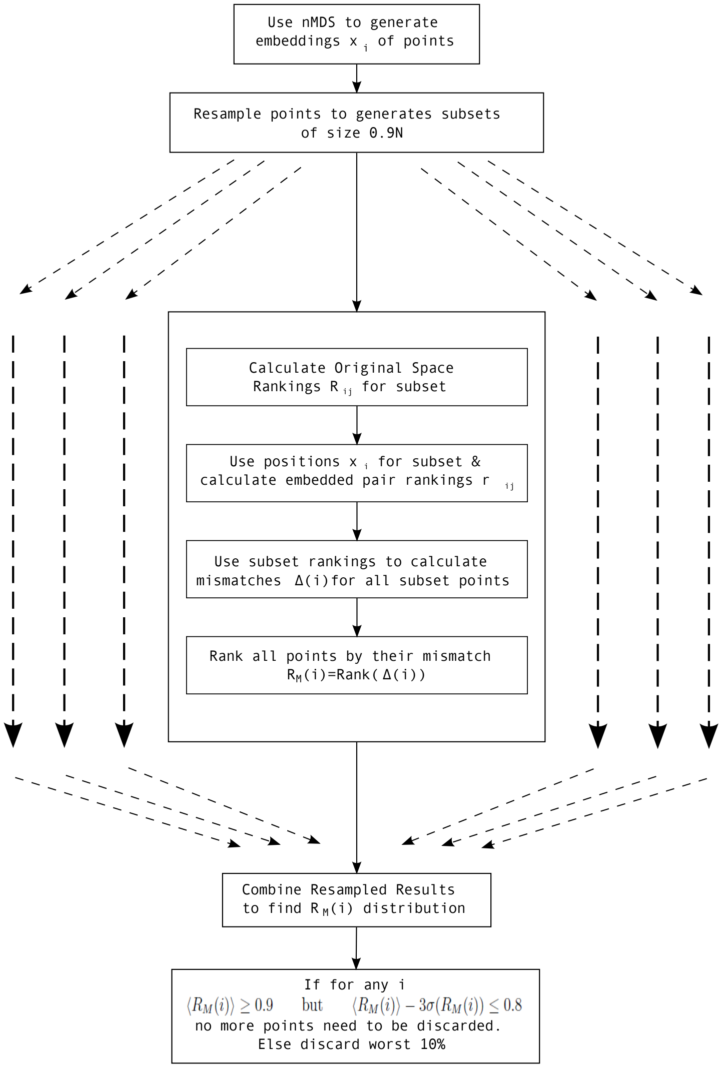

Use nMDS to generate embeddings x_i of points
Resample points to generates subsets of size 0.9N
Calculate Original Space Rankings R_{ij} for subset
Use positions x_i for subset & calculate embedded pair rankings r_{ij}
Use subset rankings to calculate mismatches Δ(i)for all subset points
Rank all points by their mismatch $R_M(i)$=Rank(Δ(i))
Combine Resampled Results to find $R_M(i)$ distribution
If for any i
$\langle R_M(i) \rangle \geq 0.9$ but $\langle R_M(i) \rangle - 3\sigma(R_M(i)) \leq 0.8$
no more points need to be discarded.
Else discard worst 10%

Figure 5.2: Bootstrap Scheme: The sequence of steps used to decide if more honing is needed

such. Furthermore, discarding them will not significantly improve the clarity of the analysis result. Thus, the discarding process (the recursive honing process) is stopped when this 'improvability limit' is reached. For a schematic diagram of this procedure with $s = 10$, see Fig. 5.2. One may wonder why we do not use traditional, less computationally intensive methods such as a comparison to randomized or shuffled profiles. As it turns out (especially for long profiles), nMDS sees shuffled profiles as being so different from realistic profiles, that they represent an excessively weak null hypothesis. That is to say, nearly all points invariably pass such tests.

Algorithm with nMDS

We are now in a position to consider an algorithm to implement the honing procedure. For actual implementation the choice $s = 10$ was made.

1. Analyze the data set by nMDS to extract the structure supported by the data.
2. Calculate the rank mismatch of each data point and identify the worst $s\%$ of points in terms of rank mismatch. These are candidates for removal (called the candidate set).
3. Calculate the empirical distribution of rank mismatch of each gene after randomly sampling $(100 - s)\%$ of the data set many times. Now, focus on the distribution of rank mismatch for the members of the candidate set in (2).
4. If the distribution obtained in (3) strongly supports the results of (2), remove the candidate set in (2). See below for an explicit criterion.
5. With the remaining genes as the starting set for the next iteration repeat the procedure from (1) to (4) until (3) no longer supports (2).

Cutoff criterion

In Step 4 of the algorithm above, to decide if the bootstrap results support the set of points we have selected for removal, the following criterion is used.

As indicated in the algorithm, for each point, the bootstrapping procedure gives us a distribution for its normalized rank, i.e., Rank/N. This distribution resembles a Gaussian centered close to its pre-bootstrap rank. The uncertainty in rank is characterized by the 3σ value of this distribution. More specifically (for a choice of $s = 10\%$), when the distribution becomes so broad that for some point i

$$\langle R_M(i) \rangle \geq 0.9 \qquad \text{but} \qquad \langle R_M(i) \rangle - 3\sigma(R_M(i)) \leq 0.8$$

the honing process is stopped, and the result is accepted. Here, $\langle R_M(i) \rangle$ and $\sigma(R_M(i))$ are the mean and standard deviation, respectively, of the normalized rank distribution of the point i.

3 Results

3.A Artificial Data

Let us first consider an example where the correct underlying noise-subtracted result is known. A 3-leaf clover like structure shown in Fig. 5.3 (a 2D pattern) is corrupted by noise: to each point we added an 8 dimensional noise orthogonal to the given pattern, so that the input data is now 10 dimensional. The noise amplitude for each point is chosen randomly from a random distribution which is Gaussian. Then, each component of the noise is sampled from a uniform distribution with this amplitude. This helps to increase non-uniformity. The average noise amplitude in each of the noise directions is about 0.6 times the average amplitude in the 2D pattern dimensions. Thus, we are dealing with a very noisy object. This 10 dimensional data is then embedded back into 2D Euclidean space via

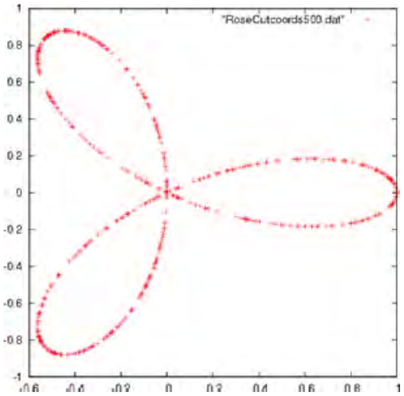


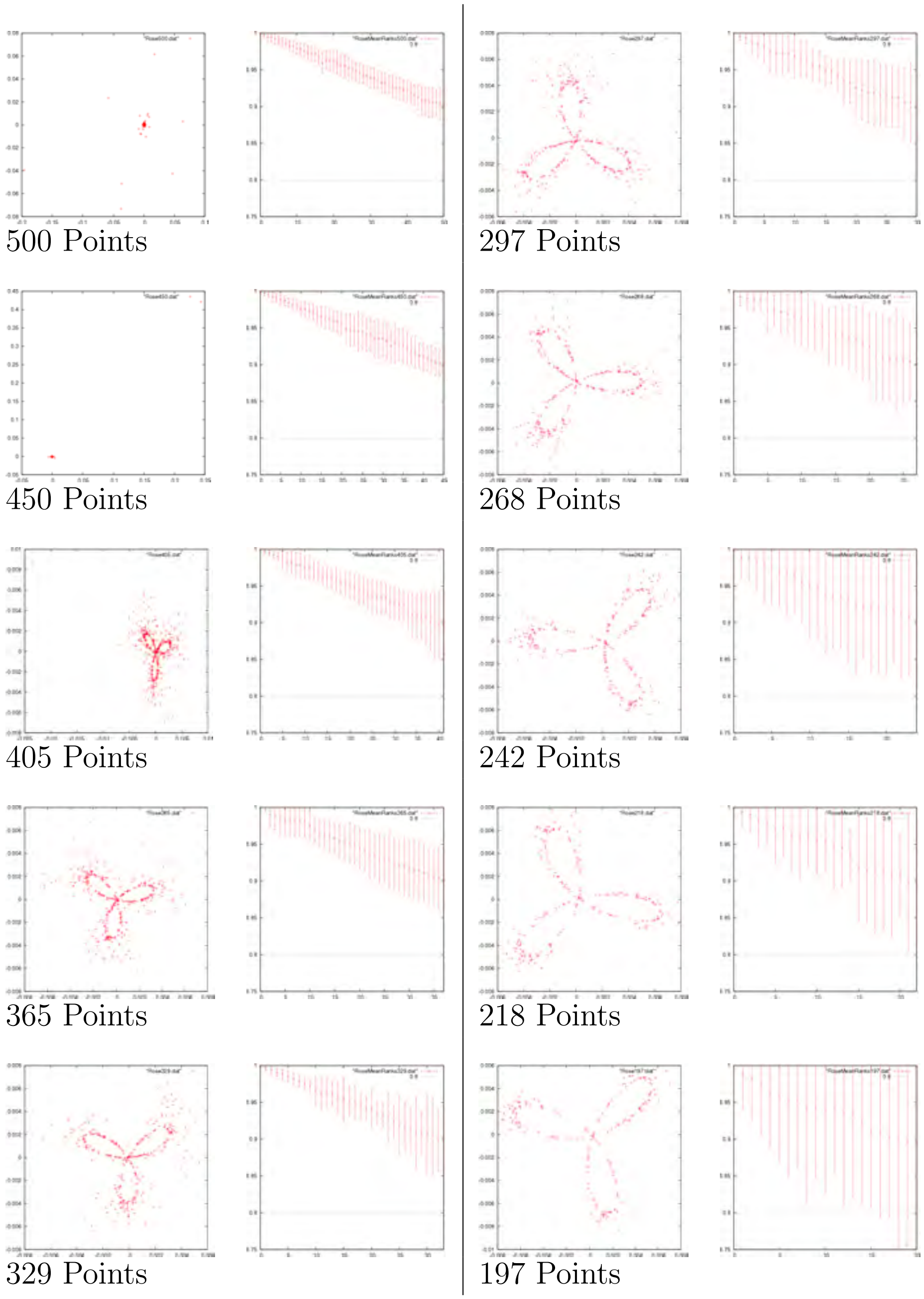
Figure 5.3: Underlying Structure

nMDS. Ideally the original 2D pattern would be recovered. In Fig. 5.4, for a given number of points, the left hand side shows the nMDS results. The right hand side exhibits the bootstrap normalized rank distributions for the worst 10% of embedded points. The plus marks denote their average values and the bars indicate the 3σ width. To begin with, the worst points are much worse than the others, and hence the distribution of their ranks is very narrow, as can be seen from the small sizes of the error bars. As more and more points are discarded, the worst points become more like the majority, and hence their distribution (quantified by the 3σ error-bars) broadens. Honing is stopped when the error bars go below 0.8.

As can be seen from the figure, the nMDS result before polishing looks nothing like it should. Its quality gradually improves as points are discarded iteratively, the most pronounced improvements appearing in the first few steps. The final result as suggested by the cutoff criterion is clearly recognizable as the correct law.

3.B Micro-array Data for Fibroblast by Cho *et al.*

Let us now consider the analysis of actual microarray based data of primary fibroblasts prepared from human foreskin produce by Cho *et al.* [52]. For each gene, microarrays are used to find the expression levels as a function of time. This was done twice, giving us



500 Points

297 Points

450 Points

268 Points

405 Points

242 Points

365 Points

218 Points

329 Points

197 Points

Figure 5.4: Recursive honing results with $s = 10$: For a given number of points, the figure on the left shows the embedding result, while the one on the right shows the result of bootstrap procedure for the bottom 10% of embedded genes. The plus marks show their mean ranks, while the error bars are the 3σ width. The bootstrap procedure (with $s = 10$) requires us to stop when any of these goes below 0.8

two experiments N2 and N3. Cho *et al.* averaged the profiles across the two experiments and identified cell cycle related genes as those showing a strong Fourier component with the time period of the cell cycle. Shedden and Cooper [53] bitterly criticized this procedure. They noted that the cell cycle related genes identified using the average of profiles from two experiments were not necessarily supported by the individual experiments separately, thereby casting doubt on the validity of the results.

The use of a small subset of genes selected by Fourier analysis, instead of the whole set, for the analysis does have some problems. There has been some controversy over the exact identity of the set of cell cycle related genes, and this has been seen in many different species. Thus, it is worthwhile developing a general methodology to deal with it. Different studies (for example, in yeast see [54, 55, 56]) come up with their own sets of cell cycle related genes. These sets have varying sizes and show a less than impressive overlap with each other (although there is perhaps a small core set of genes common to all studies). Also, in selecting based on some externally imposed criterion, we run the risk of ignoring the underlying biology supported by the data set. However, since cell-cycle related genes form a minority of the genome, and behave quite differently from the other, pre-selection is essential to study cell cycle related behavior. We will elaborate on this further later in this chapter.

A nMDS embedding, without honing, of the average (of experiments N2 and N3) gene expression profiles of the cell cycle related genes, as identified in Cho *et al.*'s paper, yields a configuration of points arranged along a ring in conformity with the identification by Cho *et al.* (Fig. 5.5(a)); the genes corresponding to different cell cycle phases separate out nicely, showing that gene positions are directly connected to the time in the cell cycle when they are up-regulated. We now repeat this procedure, i.e., nMDS embedding without honing, for the time series generated by experiment N3 alone. In this case (Fig. 5.5(b)), the cyclic pattern reflecting the cell cycle is still recognizable but diffuse. This is in conformity with the observation by Shedden and Cooper. After honing the N3 data, (Fig. 5.5(c)) we can see that the cell cycle phases are nicely separated. Honing indeed sharpens the majority

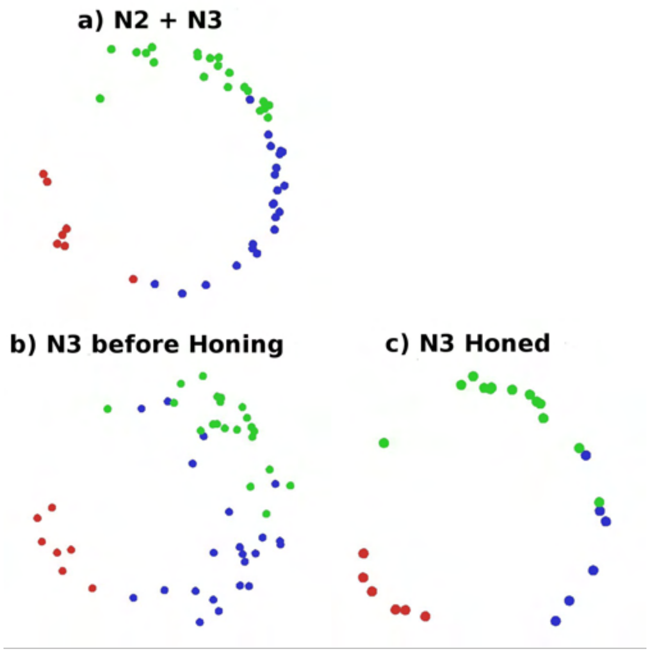


Figure 5.5: nMDS positions for genes identified by Cho *et al.* to be up-regulated in specific cell cycle phases: (a) exhibits the result using the average of experiments N2 and N3; (b) exhibits the result produced by the data from N3 only; (c) is the result obtained after honing the N3 data.

behavior that is periodic. Thus, we may conclude that there is a genuine periodic pattern supported by each data set, and the averaging effect, due to combining two data sets, could enhance periodicity by reducing the noise. That is, the averaging method adopted by Cho *et al.* has been justified.

3.C Yeast Cell Cycle Data by Spellman *et al.*

In the case discussed above, even without honing the averaged result seems quite good. This is because we only considered the cell cycle related genes; a fraction of the entire data set. Without this pre-selection, there would be genes corresponding to many different processes, and the data would not be easily described in a single low dimensional space. In this section, we investigate the effect of such pre-selection procedures using microarray based time course profiles of genes in yeast produced by Spellman *et al.* [1].

In their paper, six different synchronization methods were used to ensure that all cells started in the same cell cycle phase before gene expression measurements were made. Of these, the profiles produced by Cln3 and Clb2 are too short to be used. Spellman *et al.* have pointed out (and we, and other authors, have confirmed this independently) that Elutriation behaves very differently from the other synchronization method. We therefore consider only the data produced by cdc15, cdc28 and alpha synchronization. For each gene, the time course profiles for these three experiments were concatenated to produce 59 dimensional time expression vectors. This allows us to simultaneously treat all three synchronization methods without any need for fine tuning, and yet focus on genes which are consistent over all of them. On the other hand, information contained in the differences between the experiments is discarded. A method to exploit these differences will be the focus of the next chapter.

First, we demonstrate the need for pre-selection. All the genes missing less than 5% of their data were embedded into 3D using nMDS. The (negative of the) Pearson correlation coefficient was used as the dissimilarity measure. The result is shown as a stereo-plot in Fig.

5.6. A set of 84 genes previously identified as being cell cycle related (based on small scale experiments) are marked in color according to the cell cycle phase in which they are known to be upregulated. Since this set is large and represents all cell cycle phases, we expect cell cycle related genes, in general, to be placed fairly close to these (this has been confirmed using the set of 800 genes proposed by Spellman based on a Fourier based identification).

It is clear that this set of cyclic genes is differently distributed than the bulk of genes, being concentrated roughly at two opposite poles, while the region in between, where the majority of genes are located, shows a very low density of cell cycle related genes. Thus, the properties of these genes are different from the bulk, and without pre-selection one would expect those properties would be swamped out, especially if honing were to be performed.

That said, it does seem as if the nMDS embedding captures some cell cycle related information. It is possible to define an angle along a great circle, which roughly corresponds to the time of up-regulation of these cell cycle related genes, and not too many few cell cycle related genes are present away from this plane. Thus, the closeness to this plane seems to be a rough measure of cylicity. In this sense we have managed to identify some cell-cycle related behavior without pre-selection. Even though the predictive power is poor, to the best of our knowledge not even this has been achieved previously. A much better data-driven method will be proposed in the next chapter.

When honing is applied to this data set cell-cycle related genes seem to be preferentially removed. The results before and after honing are shown in Fig. B1 (a) and (b), respectively. The cell cycle phases do not separate out, even after honing. Since cell cycle related behavior no longer represents the majority, it seems these genes are preferentially removed by the honing procedure. Here, only about a quarter of the genes have been discarded, and already a dramatic reduction of marked genes is seen. With further honing, the remaining marked genes disappear rapidly. Local enrichments of specific Gene Ontological categories are seen, although these results are not consistent over experiments and synchronization methods, and therefore other sources of validation are required. So, this result is at best of dubious

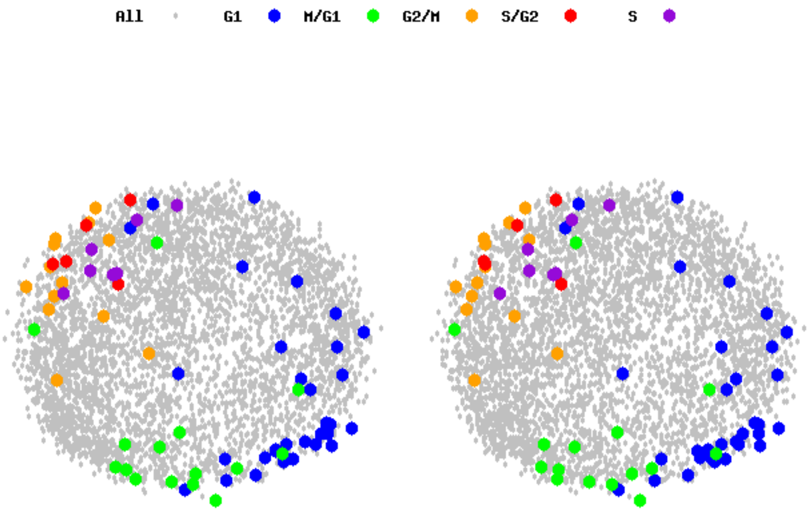


Figure 5.6: nMDS positions for the genes missing less than 5% of their profiles. Those that are known biologically to be cell-cycle related are marked in black.

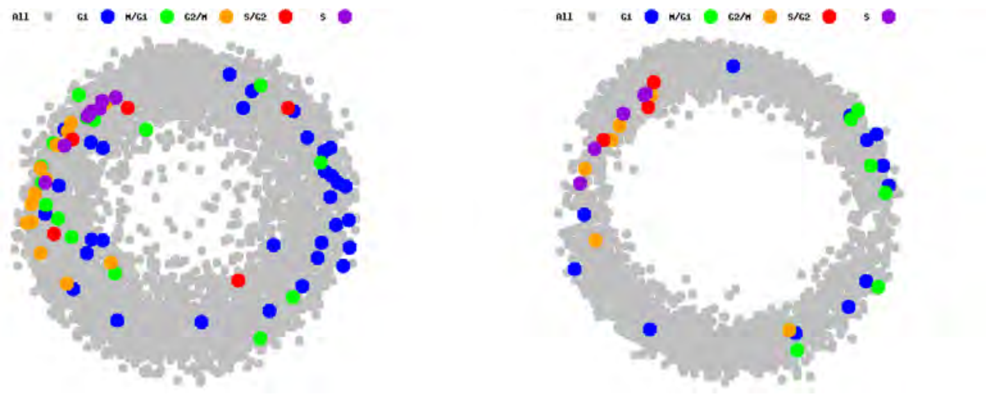
reliability. Thus, we have to conclude that, even with honing, pre-selection is necessary to see a reliable cell-cycle related behavior.

One may be tempted to believe that a clever pre-selection removes the need for honing. For example, instead of applying honing to reduce the number of genes in the Cho data, perhaps we could have simply chosen fewer (and consequently more strongly periodic) genes using Fourier analysis. This approach is dangerous because:

1. Pre-selection is only possible when we know precisely what kind of gene expression profiles we are looking for (in this case periodic). It would not work in exploratory studies.

2. Even if the assumptions about the nature of gene expression profiles used for pre-selection seem reasonable, non-data-driven pre-selection risks imposing an un-natural behavior on the data set.

3. Pre-selection methods usually rank genes according to their quality. The cutoff is often



(a) (b)

Figure 5.7: Embedding of data without pre-selection: (a) All genes embedded into 2D. Only the genes known, biologically, to be up-regulated in specific cell cycle phases are colored (b) The honed result, after removing a quarter of the genes, of (a).

arbitrary, and even in those cases where p-values are found the null results used involve straw-man type scenarios casting doubts on their validity.

That said, for a single experiment, in most cases, pre-selection is inevitable (a data driven way to perform pre-selection in the context of multiple experiments is shown in the next chapter). There are very few properties that are expressed genome-wide. So, if we use all genes or genome-wide data, the property of interest is likely to get swamped out. We now show how honing can be used in conjunction with known biological information to avoid the dangers listed above.

1. First identify a set of genes known *biologically* to possess the property we are interested in. This set of genes shall henceforth be known as anchor genes.

2. Then, choose a large set of genes having profiles similar to this anchor set.

3. This set is deliberately chosen to be large enough that the pre-selection step does not unduly constrain the selection. Yet, the size of the set must be small enough that the genes are reasonably coherent with respect to the property of interest.

4. We then apply the honing procedure to reduce this set to a smaller set of genes that are truly coherent and consistent with the anchor gene behavior.

Thus, pre-selection produces a candidate set, and honing is used to arrive at the true list of core genes. It is hoped that this will remove the effect of any idiosyncrasies of the pre-selection method. The process of pre-selection is clearly biology driven, meaning fewer assumptions need to be made. Honing stops discarding genes when we can't be sure if the genes to be discarded are any worse than the remaining ones, thereby providing a statistically meaningful way of determining how many genes should be preserved. In this way, such a procedure could help avoid some of the possible dangers outlined above.

If a biology driven means of pre-selection is not possible, one may use standard pre-selection methods (such as Fourier analysis). Let us now apply these ideas to the Spellman *et al.* [1] data set. Since pre-selection is used simple as a means of producing a candidate set of genes from which we hope to extract the core set of cell cycle genes, it is advantageous to keep the pre-selection methods as simple as possible. With this in mind we consider three methods of pre-selection. The set of 84 genes mentioned above are the anchors referred to below.

1. NN1: For each gene, find its Pearson correlation coefficient with respect to all the anchor gene profiles, and pick out the anchor gene it is closest to. We then rank all the genes in terms of their closeness to anchor gene nearest to them, in the sense described above. This is the most naive means of pre-selection. It is easily affected by anchor genes of poor quality (some anchor genes in Fig. 5.6 are seen to be distributed differently from the others).

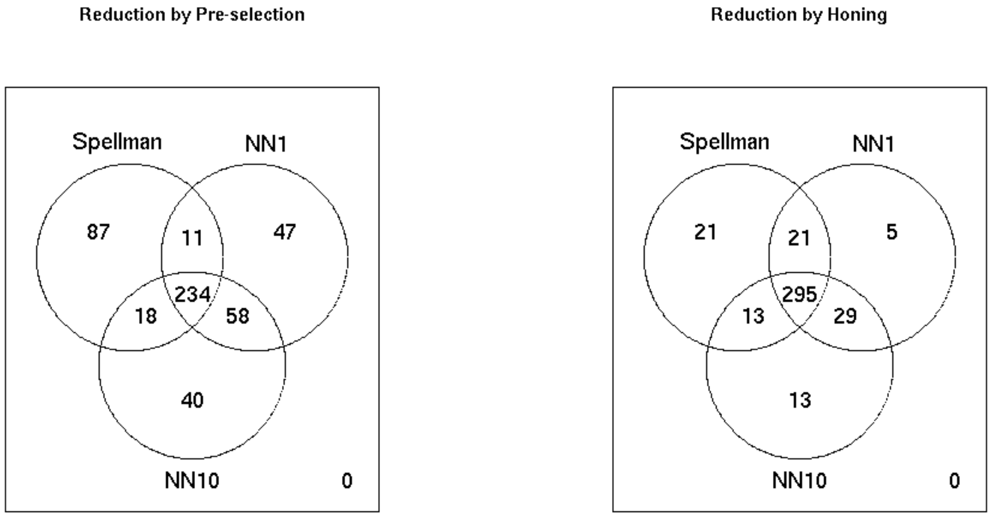
2. NN10: This is a more sophisticated version of NN1. Instead of ranking genes by the distance of the closest anchor gene, the average of the distances to the 10 closest anchor genes is used. This reduces fluctuations produced by bad anchor gene data. The choice of 10 was based on the number of aberrant anchor genes seen in Fig. 5.6, and is thus

still data driven.

3. Spellman: Finally, for purpose of comparison we consider the set of 800 cell cycle related genes produced by Spellman. They performed a great deal of fine tuning and tweaking to align the profiles corresponding to different synchronization methods, and using Fourier analysis, they arrive at a cyclicity score for all the genes. Based on a somewhat arbitrary criterion they decided that 800 of these were cyclic.

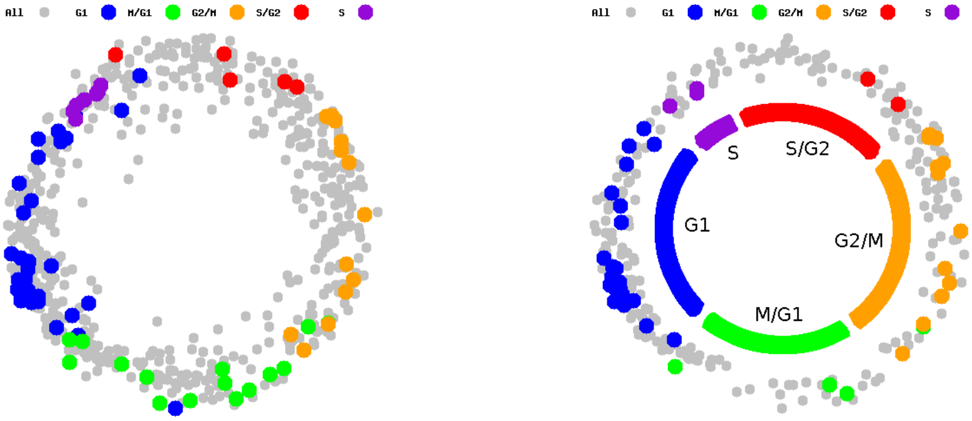
For NN1 and NN10 the best 1000 genes were selected. With this choice clear separation of cell cycle phases is seen, much in the same way as for Cho's data. They look very much like Fig. 5.9(a), which is the result for Spellman's selection. As the size is increased beyond this, the behavior worsens rapidly. For Spellman's data set, 625 genes were used. These are all the genes selected by them and present in our set after removal of missing data. Honing applied to these different sets suggests that between 480 and 330 genes be preserved. As a compromise for easy comparison, we selected the best 350 genes by honing for the 3 pre-selection methods. we also used just the pre-selection criterion themselves, without honing, to select sets of the top 350 genes. The overlaps for these two methods are compared in Fig. 5.8. It is clear that applying honing to larger pre-selected sets leads to a significant improvement in the overlap of these sets, thereby minimizing the effect of pre-selection. As a conservative estimate, we select the 295 genes common to all three honing selected sets, as being cell-cycle related.

To understand the effect of honing, we compare the nMDS results on the 625 gene set based on Spellman's selection to our proposed set of 295 cell cycle related genes. This can be seen in Fig. 5.9. Honing concentrates genes which are known, biologically, to be up-regulated in specific cell cycle phases. We may now use this result to classify the remaining genes into cell cycle phases.



(a) (b)

Figure 5.8: Overlaps between the top 350 genes selected using different pre-selection methods: (a) exhibits the result using only pre-selection (b) exhibits the result produced by applying honing to larger pre-selected sets



(a) Spellman Selection

(b) Honed Selection

Figure 5.9: (a) exhibits the result using 650 genes selected by Spellman before honing; (b) exhibits the result produced after honing down to 295 genes. The colors of the arcs in (b) are chosen to be the same as the that of genes belonging to the corresponding phase

4 Conclusions

The success of pattern or structure extracting algorithms is often limited by the presence of points in data sets which are corrupted by noise. In this chapter we have proposed the idea that this problem can be alleviated by identifying such points and removing them, a process called data honing. If the biologically interesting pattern represents majority behavior, we propose that the noisy points are those that are inconsistent with the extracted structure (supported by the majority of data points). Their removal therefore enhances the extracted structure in a fully data-driven fashion. This general idea may be used by any pattern extracting multivariate analysis method. A concrete implementation was provided in the context of nMDS (a discussion of PCA based honing is illustrated in the appendix.). The utility of this method was illustrated by extracting the (known) correct pattern from a noisy artificial data set, and by validating the controversial data set produced by Cho *et al.*

The idea proposed above assumes that the genes of biological interest are in a majority. However, in real data where there are typically multiple processes at work this is hardly ever the case. In the context of the much cited study by Spellman *et al.* [1], attempting to identify the cell cycle related genes in yeast it is shown that a) when considering all genes, it is difficult to extract any biologically meaningful results by standard methods and b) honing fails to improve the results. Consequently, some means of pre-selection to identify only genes belonging to a single process is required. Unfortunately, pre-selection methods are typically not biology driven and risk imposing their artificial biases on the data. A method to overcome this problem by using honing in conjunction with a biology driven pre-selection method is proposed. It is found that honing reduces the effect of pre-selection, and allows us to arrive at a core of genes believed to be cell cycle related. This core of cell cycle related genes shows improved clustering of biological properties such a time of up-regulation.

Chapter 6

Variability and Internal Consistency

1 Introduction

Living organisms are the ultimate complex systems. At any given point of time, there are thousands of processes running in parallel, and intertwined in ways we will not understand for years to come. It should therefore not be surprising that the outcome of experimental measurements on cells can depend on a huge number of factors. Apart from the specific biology that the experiment is trying to capture, some of the important classes of factors affecting expression of genes are found to be:

- Genetics: (Non-clone) individuals have different genomes, which means the instructions they use for control of gene regulation are different

- Epigenetics: Even if individuals are very similar genetically, there are a number of other sources of control that are not genetic in origin. For example, methylation causes methyl groups to be added to the DNA, and methylated regions are less transcriptionally active. Thus, even cloned populations, under identical conditions, could show different expression.

- Organism State: The state of an organism varies as a function of time. Different processes change in different ways. While it is perhaps possible to ensure that genes involved in a single process are in sync across cells, this is impossible genome-wide

- Environmental: The state of a cell is at least partly a product of its environment (not just its present state, but also the history)

- Noise: Many cellular functions are carried out by molecules in very low concentrations, so that stochastic fluctuations become significant.
- Experimental Protocol: Even slight differences in the way an experiment is performed can have a significant impact on the outcome. For example, a recent survey [57] of microarray results on identical sample from multiple groups should surprisingly poor reproducibility.

Some of these factors we can know and control, but the majority we cannot. It is therefore inevitable that when experiments are repeated, the results are somewhat different. The question we are going to attempt to address in this chapter is whether there is any biological information in this variability of gene expression across experiments. Such variability is seen across various other quantifiers in a population, and there are various techniques to take advantage of them [58]. However, in the context of gene expression, such variation is typically assumed to be caused by random, meaningless, uncorrelated noise to be removed by averaging. In this chapter, we present a method that instead makes use of the variability of gene expression across experiments and uses it to solve many of the problems we faced in the previous chapter.

Given the complexity of biological systems, gene expression could potentially depend critically on the state of just a few molecules in the cell, making prediction of biological behavior impossible. In practice though, we find that even when they are subjected to vastly different environments, living organisms behave (at least at a macro level) in a reliable, reproducible fashion. This is a testament to their robustness.

For this to happen there must be design systems in place which ensure that, irrespective of the microscopic details of the state of the cell, the important functions are performed normally. One might use this idea in reverse to identify the important functions; look at the cell under many different conditions and the functions that are always being performed the same way are likely to be important. This is the guiding principle of this chapter. Thus, instead of perturbing the system directly, we make use of noise and fluctuations to infer properties of the system.

In physics terms this is essentially the RG principle that the quantities worth studying cannot depend on microscopic details, and so by shaking the system (or more specifically, its microscopic details) and looking for parts that do not move, we can identify the right variables to study.

The macro/phenotypic behavior is generated by the complex and microscopic 'network' of gene interactions. High-throughput experiments, such as microarrays, provide us with a snapshot of the state of the genetic machinery by telling us about the expression levels of thousands of genes at a given point in time. Since the individual gene expression profiles themselves represent raw microscopic details we expect them to be affected by the change in experimental parameters. What one might expect to stay fixed is the groups of genes involved in a particular process, and consequently the relations between their gene expression profiles.

With this in mind, we have designed a method that looks for groups of genes whose relations (irrespective of their strengths) are well preserved in the expectation that these groups are related. The basic idea of this method is summarized schematically in Fig. 6.1. Each point in the picture corresponds to a single gene, and its position a representation of its expression profile. Many of our beliefs/assumptions about biological systems are captured by the picture:

- The gene expression profile for a single gene can vary unpredictably from experiment to experiment.
- Biologically related genes can have significantly different expression profiles.
- Closeness in expression profile for a single experiment does not necessarily imply a biological relationship.
- Related genes respond in correlated ways to changes in experimental parameters.

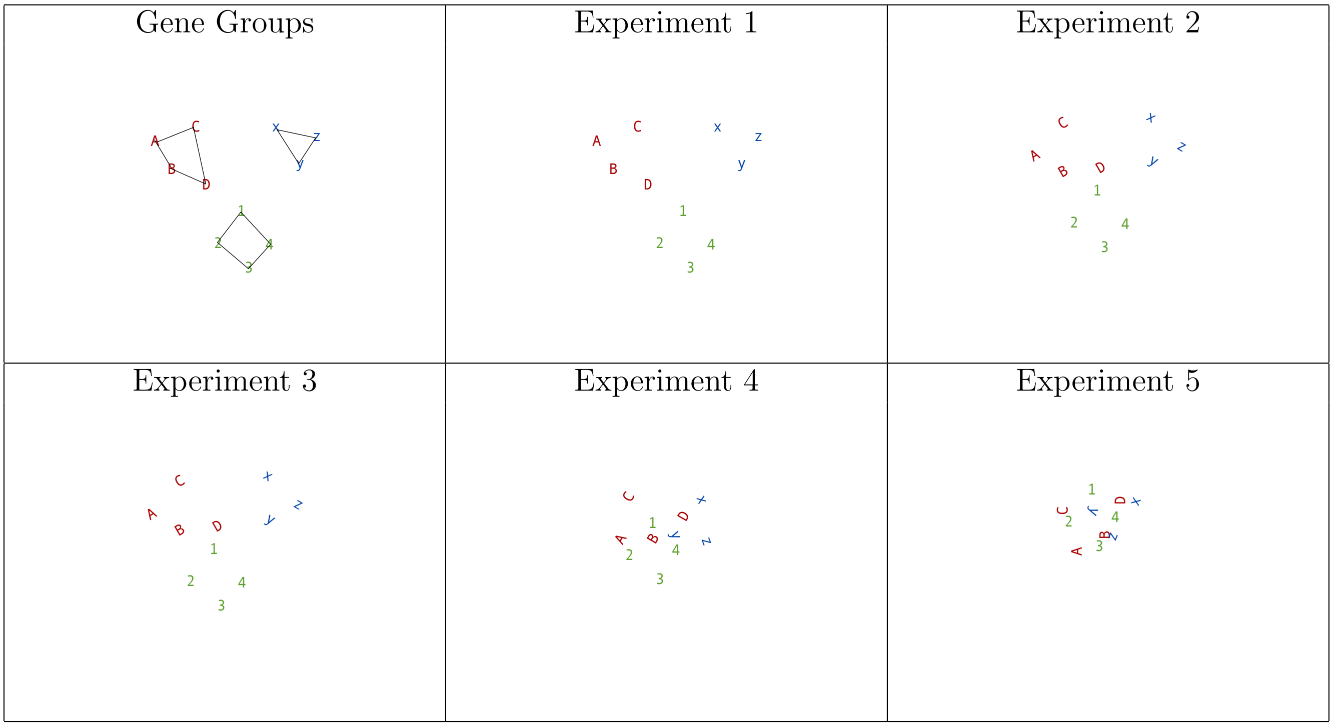


Figure 6.1: Basic Idea of the ICS Survey: The different points here are supposed to reflect the positions of genes in the high dimensional space of their expression profiles. As parameters change between experiments, the expression profiles change too, altering the positions of the various genes. However, the relative positions of groups of related genes (same color) are preserved, while those of unrelated ones are not.

- Unrelated genes respond differently.
- Therefore relations between the gene expression profiles of related genes are better preserved, i.e., they act as rigid bodies in gene expression space.
- We can identify related groups of genes by looking for such rigid bodies.
- There may be many such rigid bodies, by identifying them we can identify the dominant processes captured by the experiments.

This idea was applied to the microarray-based study of gene expression in yeast performed by Spellman *et al.* [1] that we studied in the last chapter. Unlike in the last chapter, where the experiments were combined to look for the averaged patterns, here we explicitly look for the differences in gene expression across experiments. It was found that there are essentially two rigid gene groups supported. One of them corresponds to periodic genes which were found to be related to the cell cycle, while the other group seems to consist of genes that showed a strong response to the cell cycle arrest methods employed. The latter group showed a strong over-representation of genes related to the ribosome. Our method, despite being completely data-driven, shows very clear separation of groups with classification accuracy rivalling that of methods specifically designed to identify cell-cycle/ribosomal genes. This unequivocal separation of groups is the key advantage of this method and solves many of the shortcomings of traditional methods noted in the introductory chapter.

2 Method

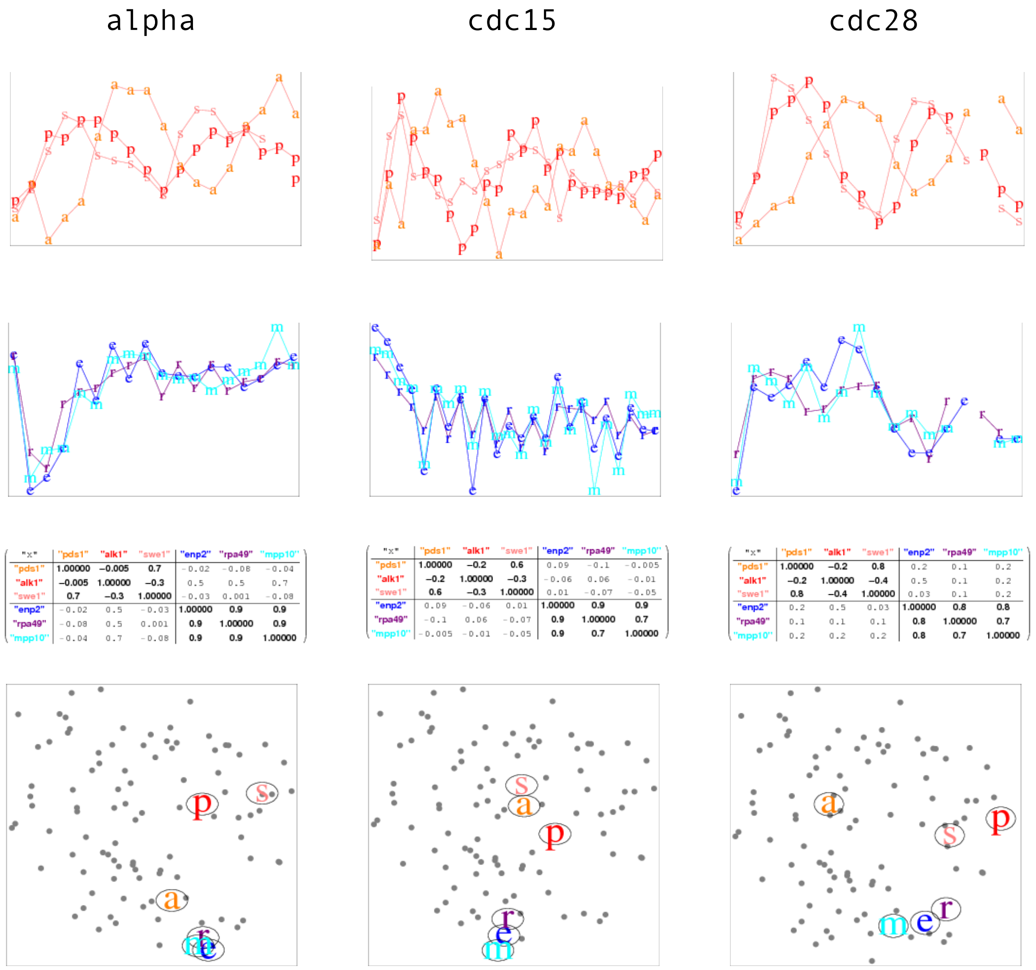
2.A Basic Idea

Before explaining the details of our approach, let us motivate it with an example. Fig. 6.2 shows the expression profiles of a few selected genes based on the Spellman data. Profiles for the three different cell-cycle arrest methods used by Spellman *et al.* [1] are shown separately.

Two groups of genes were considered, one related to ribosome biosynthesis and the other to the cell-cycle. Comparison of their time course expression profiles across the synchronization methods shows that the cell-cycle-related genes roughly preserve their relative relations by phase shifting all the profiles equally. The ribosomal genes all have similar expression profiles for each synchronization method. However, the change in expression levels from protocol to protocol is unclear. Thus, each group maintains robust intra-group (anti)correlation patterns across protocols, but the intergroup relations exhibit no regularity. This difference between inter- and intra-group relations can clearly be seen in the correlation matrices of the gene expression profiles in Fig. 6.2. This observation motivates the already mentioned idea that the extent of robustness of relations among gene expression levels can be a means to unravel biologically important features of the system.

As mentioned in the introduction, geometrically if we think of genes as points lying in a high dimensional space, closely (e.g., functionally) related genes could form robust constellations that are stable against modification of experimental factors, even though mutual relations between these constellations are strongly affected. A method shall now be proposed to identify groups of genes that form robust constellations of this type (i.e., groups of genes among which relative relations are well preserved). Each constellation may then expected to represent one class of genes related in a biologically significant way. It should be noted here that moderate but consistently reproducible relations among genes may be exploited in this analysis, while typically it is only the strongest relations that are considered.

The geometrical analogy above might lead the reader to think it may be better to perform a dimensional reduction procedure to embed the genes in a low dimensional space first and then look for consistency in this low dimensional result. An illustration for PCA can be seen in the fourth row of Fig. 6.2. Here, the 6 (cell cycle related + ribosomal) genes were combined with 94 other randomly chosen genes. PCA was then performed on these 100 genes. In the PCA result, neither do the cell cycle genes appear to be related, nor is the robustness of their phase relationship captured by PCA. Although PCA places the ribosomal



"x"	"pds1"	"alk1"	"swe1"	"enp2"	"rpa49"	"mpp10"
"pds1"	1.00000	–0.005	0.7	-0.02	-0.08	-0.04
"alk1"	–0.005	1.00000	–0.3	0.5	0.5	0.7
"swe1"	0.7	–0.3	1.00000	-0.03	0.001	-0.08
"enp2"	-0.02	0.5	-0.03	1.00000	0.9	0.9
"rpa49"	-0.08	0.5	0.001	0.9	1.00000	0.9
"mpp10"	-0.04	0.7	-0.08	0.9	0.9	1.00000

"x"	"pds1"	"alk1"	"swe1"	"enp2"	"rpa49"	"mpp10"
"pds1"	1.00000	–0.2	0.6	0.09	-0.1	-0.005
"alk1"	–0.2	1.00000	–0.3	-0.06	0.06	-0.01
"swe1"	0.6	–0.3	1.00000	0.01	-0.07	-0.05
"enp2"	0.09	-0.06	0.01	1.00000	0.9	0.9
"rpa49"	-0.1	0.06	-0.07	0.9	1.00000	0.7
"mpp10"	-0.005	-0.01	-0.05	0.9	0.7	1.00000

"x"	"pds1"	"alk1"	"swe1"	"enp2"	"rpa49"	"mpp10"
"pds1"	1.00000	–0.2	0.8	0.2	0.1	0.2
"alk1"	–0.2	1.00000	–0.4	0.5	0.1	0.2
"swe1"	0.8	–0.4	1.00000	0.03	0.1	0.2
"enp2"	0.2	0.5	0.03	1.00000	0.8	0.8
"rpa49"	0.1	0.1	0.1	0.8	1.00000	0.7
"mpp10"	0.2	0.2	0.2	0.8	0.7	1.00000

Figure 6.2: Each column represents data from a different experiment in [1]. The first row exhibits the profiles for three cell-cycle-related genes *pds1*(p), *alk1*(a), *swe1*(s). Clearly, the profiles preserve their shapes (and relative relations) over experiments. The second row exhibits three ribosomal genes *enp2*(e), *rpa49*(r) and *mpp10*(m). In each experiment, the ribosomal genes have nearly identical profiles, but they change dramatically from experiment to experiment. There is little conserved relation between the profiles of genes in these different groups. As shown in the third row, the inter- and intra-group correlation coefficients reflect the same patterns. However, the results of PCA do not. These six genes were combined with 94 randomly selected genes, and PCA was performed. The positions of these genes based on the first two principal components is shown in the third row (the first letter of each gene name is used to denote its position). The consistency of phase relationships between the cell cycle related genes is not captured by PCA. Although the ribosomal genes consistently cluster, without biological knowledge, they don't stand out in any way.

genes close together, they do not stand out as being special without additional biological information. Thus, the correlation matrices capture the consistency information better than PCA.

That said, the PCA result is quite instructive. It suggests why traditional exploratory methods that attempt to find co-expressed groups of genes cannot always identify biologically related groups of elements. Firstly, co-expression for a single experiment could be accidental, and need not imply biological relation. Different processes running in a cell give different and often contradictory signals depending on various conditions. Thus, if all the mutual relations among genes are treated evenly, it is often unclear if the genes deemed to be close are in any way functionally related. Secondly, just like the cell-cycle-related genes, there could be groups of related genes that are not co-expressed. Such groups of genes are not readily recognized as related by most dimension-reducing or information distilling techniques such as cluster analyses and principal component analysis (PCA).

3 Implementation

The important message from the PCA result is that when different processes are considered simultaneously, the desired consistency information has a tendency to get swamped out. Therefore, the individual processes must be considered separately. Our strategy centers on the existence of constellations of genes unambiguously associated with certain biological-processes whose internal relations will be preserved across experiments. Such a set of genes will be called an *Internal Consistency Core* (ICC) associated with the process. The method adopted to characterize the data can then be considered to have the following steps:

1. Construct an ICC corresponding to a single process.

2. Create a method to measure the consistency of the gene expression profiles of an arbitrary gene with respect to the ICC.

3. Rank all the genes in terms of this consistency score.

4. Use some statistical test to determine how many of the top ranked genes should be considered at the desired level of confidence. This extended set of genes should now all correspond to the same process as the ICC, and will be called an Internally Consistent Set (ICS). The ICS is the result of the method.

Going back to the geometrical analogy, the ICC genes may be considered as the core of the rigid object while the ICS genes are the halo of other objects around this core that also share this rigidness property.

The actual implementation of these steps shall be explained below. Based on the way the ICC is selected, we have devised two methods. The first, called the Steered ICS, is a semi-supervised method. It is assumed that a set of related genes are known to be biologically related, and these are used as the ICC. The other, a data driven method known as ICS Survey, initially starts with a random set of genes as the candidate ICC. This set is then iteratively updated to improve consistency. Once no improvement is possible, the best ICC candidate is treated as a true ICC, and the corresponding ICS is generated. By using different randomly selected initial ICC candidates and repeating this method, a sampling of all the ICCs (and consequently the significant processes) supported by the data is found.

3.A Calculating Consistency With Respect to a Set of Genes

This section describes the implementation of Step 2, and is the central engine of our method. Suppose we have N_e experiments, measuring the gene expression profiles of N_g genes. Here, we shall primarily consider time course profiles, but measurements across multiple tissues, etc. can also be used. For a given experiment e and for each gene g, suppose the raw data is its time expression profile $\boldsymbol{x_g^e}$. Let $C^e(g, h)$ be a correlation between the profiles of genes g and h for experiment e. The Pearson correlation coefficient was used for the results in the paper, but other similarity measures by Spearman, Kendall, etc., also work. Let us assume that by some means (methods to do this will be shown later), we have a set of N genes which are believed to constitute an ICC. Then to find the consistency of the gene g, with respect to this ICC, the following steps are followed (a schematic version can be seen in Fig. 6.3):

1. Consider the gene expression profiles of the gene g and of the N ICC genes $(h_1 \dots h_N)$ across all N_e experiments

2. For each experiment, find correlation of g with each ICC gene, and construct a vector of length N with these correlations (one such vector will be constructed for each experiment)

$$(C^e(g, h_1), \cdots, C^e(g, h_N))$$

3. Normalize each vector appropriately (mean zero, unit standard deviation), to allow comparison over experiments on equal footing

$$\boldsymbol{c_g^e} = \frac{1}{\sigma_g^e} \left(C^e(g, h_1) - m_g^e, \cdots, C^e(g, h_N) - m_g^e \right)$$

4. Compare vectors across experiments, finding variance of each component across experiments

5. Inconsistency score (larger the score, less the consistency) for the gene is given by sum of these variances

$$S_g = \sum_{i=1}^{N} Var_e(c_{g,i}^e),$$

Here $c_{g,i}^e$ is the i-th component of $\boldsymbol{c_g^e}$ and Var_e denotes the variance over experiments. The use of variance ensures that the strength of (anti)correlation *per se* is not considered, just its consistency.

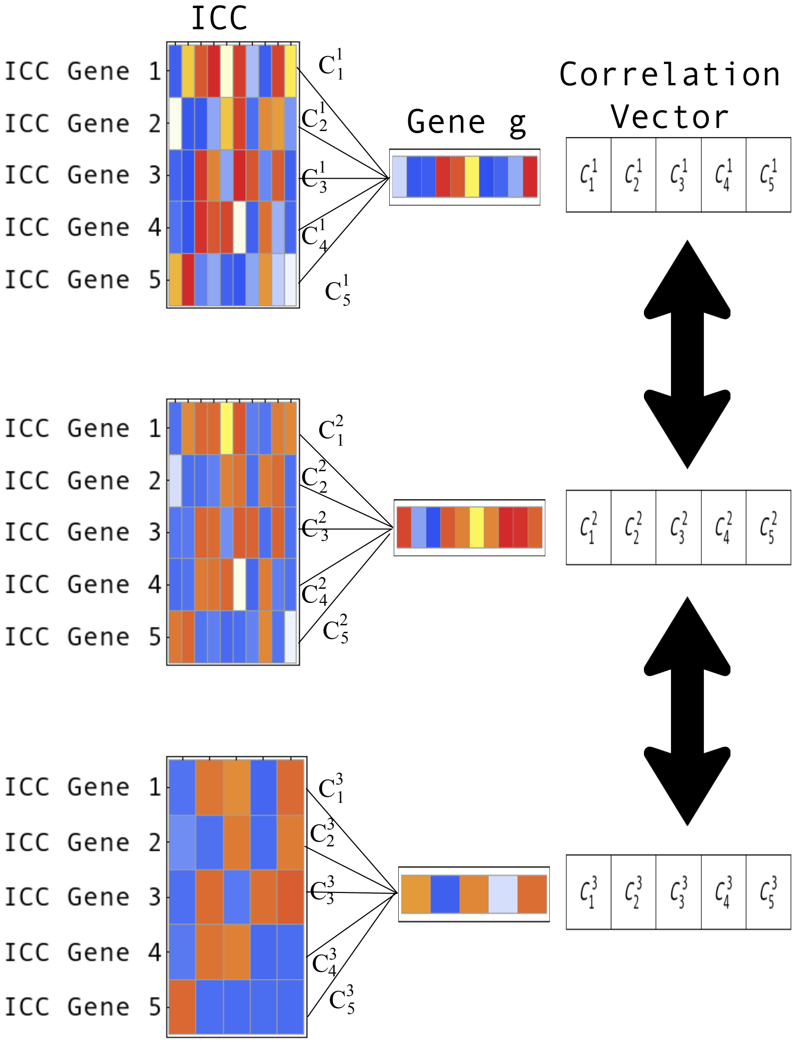


Figure 6.3: Sequence of Steps to Find consistency of a gene g with respect to a known ICC. Here, we assume there are 3 experiments and 5 genes in the ICC. Note that that the lengths of the gene expression profiles in each experiment need not be the same.

3.B Steered ICS

Steered ICS is a biology driven method to be used when we have a set of representative genes belonging to a process we are interested in. In terms of its goals and inputs it is quite similar to the NN1 and NN10, except that those methods are essentially single experiment methodologies and they are limited to finding genes that are strongly correlated with at least a few of the genes in the training set. Consequently, they need a much more complete training set. Like these methods the steered ICS is clearly not data driven, and is limited to only identifying genes related to the training set.

The steps involved in performing the Steered ICS are as follows:

1. Use the set of genes known to be biologically related as ICC.
2. Calculate the inconsistency scores S_g of *all* genes, as outlined in the previous section.
3. Rank all genes according to their S_g score.
4. Apply Random ICC test (described later) to determine how many of the top genes to preserve at desired level of confidence.

This list of ICS genes is the result of the method and is expected to be functionally related to the genes in the training set.

3.C ICS Survey

The ICS Survey is a truly data driven method which, unlike the Steered ICS, does not require any additional biological input. Instead, it starts from a random set of genes which it treats as its candidate ICC. The ICC is then updated to improve self-consistency, till a fixed point is reached. The fixed point ICC may then be used to construct an ICS. By repeating this procedure, a sampling of the different ICSs supported by the data may be found.

The steps involved in performing an ICS Survey are:

1. Select a random set of N genes as the ICC candidate.
2. Calculate inconsistency scores S_g for all genes (including the ICC genes themselves) with respect to the ICC.
3. Rank genes according to S_g.
4. N top ranked genes are new candidate ICC.
5. If new and old candidate ICCs are not the same, go to step 2.
6. Else, if the fixed point, ICC has been reached use top ranked genes based on this fixed point ICC. Number of genes to be preserved may be decided using Random ICC test.
7. Repeat from step 1 with another randomly selected set of genes to get sampling of the ICCs supported by the data set.

Thus, essentially the ICS Survey involves starting with a random set of genes and using them as an ICC to perform a Steered ICS. Then, using the top genes as the ICC, this process is repeated until a fixed point is reached.

To avoid a situation in which the ICC update gets caught in a loop, if the algorithm does not converge within a certain number of steps (chosen to be 50 for our implementation), then that run is terminated.

The result of the ICS Survey when implemented for many random initial conditions is a series of rankings (the top genes among which are the prominent ICS supported by the data). It is quite likely that many of these rankings represent the same process, and therefore the rankings must be analyzed as a whole to identify the distinct ICSs.

ICC Size

So far, we have simply assumed that the number of genes N in an ICC was known. It is not dictated by the method, and shall be decided by the user. Let us therefore consider the effects of varying ICC size.

- In the case of the Steered ICS, the larger the set, the better the results. Thus, ICC size will be decided by our confidence in the set of gold standard genes being used (we do not want to use genes we are unsure of).
- In the case of the ICS Survey, if N is too large, we run the risk of not having processes with containing so many genes.
- Even for moderately large sizes, some of the less dominant processes could be excluded, and it could take longer to converge to a fixed point.
- If ICC size is too small, fluctuations could become important, and we cannot have much confidence in the results produced.

Thus, for the ICS Survey an intermediate ICC size should be used. In practice, we have found that an ICC of size between 20 and 50 genes strikes the right balance.

ICS Size (Random ICC Test)

To determine how many genes should be considered as an ICS, their inconsistency scores based on a given ICC are compared to the distribution of lowest inconsistency scores constructed as follows. N genes are chosen randomly and treated as an ICC to calculate inconsistency scores for all the other genes. The lowest inconsistency score for this random ICC is recorded. Repeat this procedure for many randomly chosen ICCs to obtain an empirical distribution of the lowest inconsistency score. Then, the inconsistency score corresponding to an appropriate p-value can be selected as a cutoff: the genes whose inconsistency score is less than this cutoff are included in the ICS.

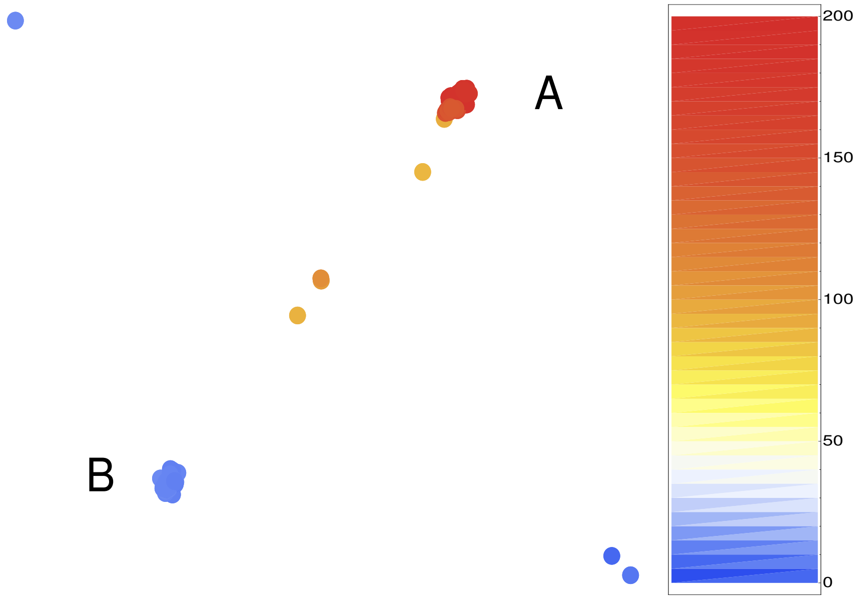


Figure 6.4: 2D nMDS embedding of gene rankings for different seed ICCs: Each point represents a single ranking of genes. The points are colored according to the overlap of the top 200 genes with the Spellman *et al.* set of cell-cycle-related genes. A completely random ranking of genes would give an overlap of about 30 genes.

4 Results

4.A *Saccharomyces cerevisiae*

We first analyzed the same data set we considered in the last chapter, namely the microarray data by Spellman *et al.* [1] used to identify the cell cycle related genes in *Saccharomyces cerevisiae*. As discussed earlier, there are multiple data sets corresponding to different methods used for cell synchronization. Here, as in the last chapter, we only considered alpha, cdc15, and cdc28. Unlike then, here the experiments are compared not averaged. Genes that were missing more than 25% of their time expression profiles in any of the 3 experiments were discarded. This reduces the size of the gene set, from the 6178 studied by Spellman *et al.*, to 5239. Spellman *et al.* had proposed a set of 800 cell-cycle-related genes based on Fourier analysis. 717 of these are contained in our reduced set. We shall use this set of genes as our gold standard for cell cycle related genes.

ICS survey

First, the ICS Survey was performed on this data set as descrbed above. 300 ICC candidates of size $N = 30$ were selected randomly. Of these the 258 that reached a fixed point within 50 iterations were used to construct ICSs. In order to classify ICCs, inconsistency score rankings of all the genes were produced for each ICC (as in Step 3 of the algorithm). Thus, each ICC is represented by a particular ranking of genes. ICCs were embedded into 2D Euclidean space with the aid of nMDS [24] according to the dissimilarity of the corresponding gene rankings measured by the Spearman rank correlation.

The results are shown in Fig. 6.4. Each point represents an ICC. Despite selecting the initial guesses in a completely random and unbiased fashion, essentially only two types of ICC were found. Since the ICS are the top ranked genes according to the ranking being used, this means that there are just two dominant ICS exhibited by these experiments.

In Fig. 6.4, the points are colored according to the overlap of the top 200 genes in the ICC rankings and those in the gold standard list proposed by Spellman *et al.* If the two lists were uncorrelated, an overlap of about 30 genes would be expected. Thus, Cluster A with typical overlaps of 180 is very highly cell-cycle-related. Cluster B, on the other hand, shows a worse than random overlap with the Spellman list, suggesting it is ordered according to a very different criterion.

Cell-Cycle-Related Cluster

As is clear from Fig. 6.4, the ICC rankings in Cluster A are quite similar. The inconsistency scores were therefore averaged to arrive at a single ranking of genes for this cluster. The top 180 averaged scores were found to pass the random ICC test (see *Methods*) at the $p = 0.01$ level of confidence. These top 180 genes therefore constitute an ICS. Of these 176 were on the Spellman list. The effect of choosing a larger number of genes can be seen in Fig. 4.A.

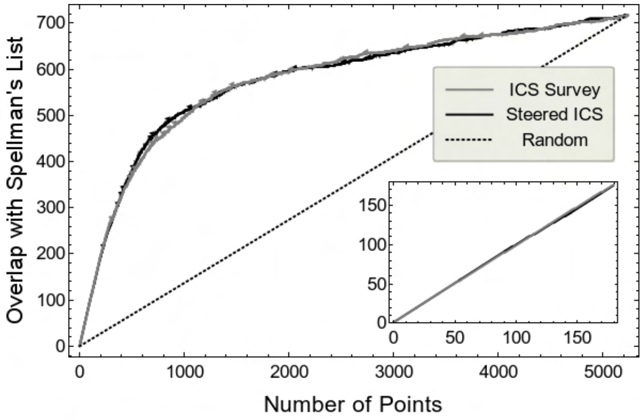


Figure 6.5: Comparison of steered ICS and the ICS survey results: Overlap of the top ranked genes with the Spellman *et al.* list are shown. The black curve shows the results due to the steered ICC, the gray curve from ICS survey while the dotted line shows the expected overlap for a randomized ranking. The Inset figure is the result for just the top 180 genes passing the random ICC test at the $p = 0.01$ level of confidence.

Clearly, this ICS is cell-cycle related.

This may also be independently inferred from observing the profiles themselves (first column of Fig. 6.6) across the three experiments. It should be clear that the profiles are all periodic, containing about two periods worth of data. The profiles are arranged[1] in a ring-like configuration with order roughly corresponding to their time of peak expression. The angular position in the ring is essentially the cell-cycle phase. The non-uniformity of distribution is an intrinsic property of the genes (the Spellman *et al.* set shows a similar distribution of phases). Note that genes up-regulated at different cell-cycle phases show poor correlations between their profiles, and would therefore be considered unrelated by methods that look just for strong pair-wise correlations (such as traditional meta-analysis methods [59, 60]).

[1] This plot can be generate using the NeatMap package to be discussed later. Essentially nMDS was applied to the 300 profiles selected as being cell cycle related using their Pearson correlation as dissimilarity measure. It is separate from the use of nMDS to determine the relations between ICSs

Non-Cell-Cycle-Related Cluster

Cluster B is not related to the cell-cycle. Again, all ICC rankings are similar so the inconsistency scores were averaged and a single ranking was calculated. The average scores of the top 650 genes pass the random ICC test at the $p = 0.01$ level of confidence. The expression profiles of these genes, for the 3 experiments, are shown in the second column of Fig. 6.6. It should be clear that all the profiles exhibit an initial jump, before relaxing into a more stable behavior. On the basis of the direction of this jump, it should also be evident that there are two distinct classes of behavior.

The coloring for these classes, for all three experiments, is based on the *alpha* results. Clearly, the identities of the genes in the two behavioral classes are conserved. This suggests that they are biologically meaningful, reproducible effects, but undoubtedly induced by arresting procedures needed for synchronization. For the genes with profiles colored black, no single dominant Gene Ontological (GO) category was found. The gray group, on the other hand, shows a remarkable over-representation of genes related to the ribosome (Table 6.1). The results for ribosome biogenesis are particularly noteworthy. There are 298 genes out of 5186 annotated genes that belong to this GO category, but of these, 163 are in the 361 genes in the gray group. Thus, the majority of the ribosomal biogenesis may be selected in this way. In addition, the 5 GO categories shown in Table 1 account for 211 of the 361 gray genes. The extent of agreement suggests that this technique could be used to identify genes related to the ribosome.

The existing method to identify ribosome related genes [61] takes advantage of the fact that under perturbations that halt the production of ribosomes, all ribosome related genes seem to behave in the same way (although this typical behavior may differ greatly depending on the perturbation). They therefore attempted to find the typical ribosomal gene profile, and find the genes with profiles matching this as closely as possible. The typical profile was found by taking the average of the expression profiles of a set of genes they previously found

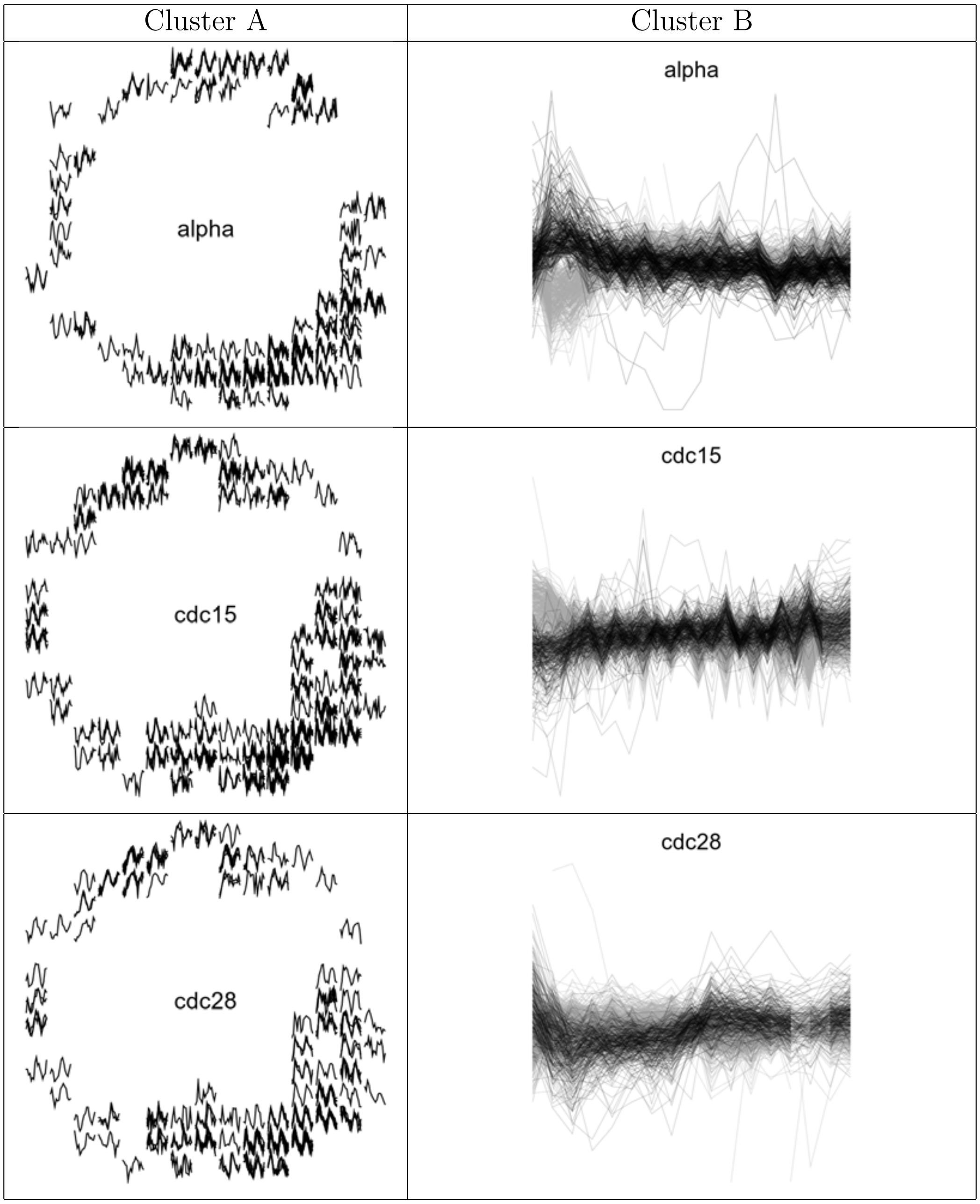


Figure 6.6: Profiles of selected genes: The first column shows 2D nMDS embedding across experiments of the 180 genes in the Cell Cycle related ICS (Cluster A) based on the Pearson correlation of their profiles. After embedding, the result was divided into a 15 by 15 grid, and the profiles of the genes within each cell are displayed. The second column shows the time course expression profiles for the 650 genes in non cell-cycle-related ICS (Cluster B) over 3 different experiments. Two distinct kinds of behaviors are seen. The profiles are colored according to their behavior in the *alpha* experiment.

Table 6.1: GO categories over-expressed in the gray genes in Cluster B from Fig. 6.6: Total # is the number of genes in a given GO category present among all the genes, whereas Gray Group # is the number of these in 365 the gray genes. All p-values were calculated using the GO Term Finder (Oct 29th 2008) tool at http://www.yeastgenome.org/ .

GO category	Total #	Gray Group #	*p value*
ribosome biogenesis	298	163	1.21e-119
rRNA processing	168	96	4.27e-68
maturation of 5.8S rRNA	55	35	3.69e-25
ribosome assembly	55	27	6.04e-15
ribosome localization	29	16	2.97e-09

[62] to be ribosome related. When this method was applied to the present data set, the results were found to be comparable, especially for the top ranked genes. Note that unlike the ICS Survey this method needs biological information in the form of a set of genes known to be related to the ribosome.

4.B Steered ICS

In the Spellman *et al.* case the purpose of experiments was to study cell-cycles, and as mentioned in the last chapter there is a set of 80 gene known, on the base of small scale experiments, to be cell cycle related. These genes were used to drive a steered ICS analysis.

Of the 80 genes, the 72 that survived our culling procedure were used as ICC. All the genes under consideration were then ranked according to their consistency with this ICC. The overlap of the top ranking genes in the Spellman *et al.* list and the top ICC ranking genes is shown in Fig. 4.A. The agreement is almost identical to the data driven ICS method.

4.C *Schizosaccharomyces pombe*

If we consider data sets produced by different groups, then not only do we have to contend with different synchronization methods, but also with differing experimental protocols. A striking example of this is the identification of cell-cycle-related genes in *S. pombe*. Three different groups [63, 64, 65] have attempted to do this using a variety of synchronization

Gene	*Description*
SPBP4H10.16C	related gene in S.c involved in negative regulation of G1 cyclin
SPBC106.20	cell separation during cytokinesis
SPAC644.06C	positive regulation of progression through mitotic cell cycle
SPBC725.16	G1/S transition of mitotic cell cycle
SPAC144.17C	fructose 2,6-bisphosphate metabolic process
SPAC4F10.03C	rRNA processing
SPBC11B10.09	G1/S transition of mitotic cell cycle
SPAC6B12.10C	cell cycle
SPBC211.03C	ER to Golgi vesicle-mediated transport
SPAC637.13c	actin cytoskeleton organization and biogenesis
SPAC17C9.01c	cell cycle arrest
SPAC589.06C	phosphate transport
SPBC18H10.08C	ubiquitin thiolesterase activity
SPAC637.13C	phosphoinositide binding
SPAC1952.08C	FMN binding
SPAC6G10.02C	selection of site for barrier septum formation
SPAC23D3.10C	cell wall catabolic process
SPAC144.04C	ornithine catabolic process, by decarboxylation
SPAC31G5.16C	GPI anchor biosynthetic process

Table 6.2: Genes included in my top 125 list for *S.pombe*, but not in the Marguerat list

methods. Each group proposed their own set of cell-cycle-related genes. Unfortunately, the sizes and overlaps between these sets were quite poor. Attempts were made by Marguerat *et al.* [66] to explain this discrepancy. They proposed a set of 500 cell-cycle-related genes based on the combination of experiments.

A total of 10 experiments from 3 different groups [63, 64, 65] were considered. Only genes that were common to all experiments and were missing less than 25% of their expression profiles in each experiment were preserved. This gave us a set of 2239 genes. In this case, the data-driven ICS approach gave only periodic, cell-cycle-related behavior with 125 genes surviving the random ICC test at the $p = 0.01$ level of confidence. To construct the steered ICS, as the core we used 33 genes from the set cited by Marguerat as previously being known to be cell-cycle-related. Of the top 125 genes selected in this way, 107 were also deemed cell-cycle-related by Marguerat *et al.* Among the genes on our list but missing from the Marguerat *et al.* list, many are cell-cycle-related. See Table 6.2 for details.

4.D Number of Cell Cycle Related Genes

There are numerous examples [66, 67] of different groups proposing sets of cell cycle related genes on the same species, but with poor overlap. This is caused [66] by the inclusion of genes, with weak evidence of expression modulation in concord with cell cycle, passing statistical tests with dubious (i.e., trivial) null hypotheses. Thus, the number of cell cycle related genes present in an organism remains an open question.

This may be addresses in two ways: extrinsic and intrinsic. Firstly, as suggested above it is to be expected that the top ranked genes are considered cell cycle related by all methods. Thus, the point at which the ICS results start diverging from Fourier based ones is a natural cutoff point. For *S. cerevisiae*, the top 250 or so genes in the ICS list are in almost complete agreement with the Spellman results. It is beyond 400 genes that a divergence is seen, suggesting there are 250 to 400 cell cycle related genes. A similar conclusion may be reached using the random ICS test proposed here. It indicates that at a $p = 0.05$ (0.01) level of confidence at least the top 250(180) genes are cell cycle related. For *S. pombe* we only considered the 2239 genes that survived our culling procedure. The criteria above suggest that about 100 to 150 of these are cell cycle related. Thus, for the full set of 6000 genes, 300 to 400 are expected to be cell cycle related.

4.E Computational Details

The computational time for the scheme presented here scales linearly with the number of genes, ICC size and number of experiments. The runs for different seed ICCs are independent, and may be computed in parallel. It is therefore expected that there will be no difficulty in extending this scheme to larger whole genome data sets. The computation time for the Spellman results are less than 30 seconds per ICC iteration on an Intel Core 2 Duo laptop with 2GB RAM. Thus, even on a laptop, a single ICS calculation (involving no more than 50 iterations) is possible in less than 25 minutes. There must be enough (~ 30) runs corresponding to each distinct ICS so that statistical quantities can be calculated reliably.

5 Discussion

5.A Comparable Methods

Although they are not very popular, there are a few existing methods that share some common features and goals with the ICS Survey:

1. There are some methods which attempt to account for the change in profiles over experiments by explicitly generating models to transform profiles between experiments. This is fairly restrictive, and does not always work.

2. In the context of gene expression analysis there have been previous attempts [59, 60] to make use of the consistency of the gene-gene correlation matrix over multiple experiments. However, for each experiment they restrict themselves to considering the reproducibility of strong correlations between pairs of genes, making them essentially a binary (correlated/not correlated) approach. Such drastic coarse-graining of information may be admissible if there are a large number of experiments, but is problematic if there are just a few. In order to obtain a more global gene relation picture from pairwise relations, they make use of clustering/network construction algorithms. Given the limited information being used, however, the separation of clusters corresponding to distinct biological features is not sufficiently pronounced to be used as a means of classification.

3. As an extension to the work by Lee *et al.* [59], one might also conceivably calculate the variance of the correlation matrix across experiments, and use this as a distance measure between genes. This approaches would be the closest to ours. However, the separation between groups is much worse than ours because only consistency of pairwise relations are considered instead of multi-factor relations like us. This approach might

work if there are enough experiments, but in the typical case where there are only a few experiments, our approach seems superior.

4. It may be argued that Fourier analysis does a very good job of identifying cell-cycle-related genes. This is only because it is not an exploratory method, and looks for a specific behavior. It thereby avoids the problems that arise from the mixing of different behavioral classes during whole genome analyses. That said, even for this specific purpose, there are reasons [53] to be skeptical about it. Beyond the most cyclic genes, the ranking of genes itself could be suspect, especially for periodic profiles that differ greatly from a sinusoidal shape. Also, if we simply average the Fourier strength, we are discarding valuable phase information: if a gene is periodic over multiple experiments, but is up-regulated in different cell-cycle phases, the results are not likely to be meaningful. The steered ICS provides a method that may be used to just identify factors of a specific type. We have shown that in the case of cell-cycle-related genes, the results agree well with those produced by Fourier analysis. Unlike Fourier analysis though, it is easy to apply it to identify other classes of genes.

5.B Conclusions

The vast data produced by high throughput experiments provide a snapshot of a large number of processes running in parallel within the cell. Each of these can critically depend on experimental protocol, hidden variables out of our control, random noise, etc. This makes it very difficult for methods based on single experiments to distinguish between different groups of factors, and direct comparisons of multiple sets of experiments are not always informative. Consequently, traditional single-experiment-based methods have not taken full advantage of high throughput experiments.

We proposed a method to identify groups of elements (e.g., genes) that represent biologically meaningful processes. The method relies on the expectation that intrinsic relations between related elements relevant to a particular process should be better preserved across different experiments than those between unrelated elements. Thus, the proposed methodology is inherently multi-experimental and is for meta-analysis.

Listed below are salient features that distinguish this method.

- The ICS Survey is completely data driven, no biological information is needed except for the last step of biological validation.

- Unlike traditional methods, it is not restricted to identifying strongly correlated genes. Weak but reproducible relations may also be identified. It therefore does not make co-expression implies relation assumption (or its converse). This is why cell cycle related genes could be identified.

- Despite being an exploratory method, the performance in identifying cell cycle and ribosomal genes compares favorably with methods specifically designed for these purposes, while making far fewer assumptions

- Unlike traditional dimensional reduction methods (as in the honing chapter), there is no mixing of processes. The genes belonging to the different classes are very well separated. This is the big advantage of our method

- The uncertainty is now transferred from demarcating groups in a result to deciding how many genes to keep in each group (for example, by the Random ICC test).

- The superior performance of this method stems from use of group instead of pairwise relations, and the fact that the different processes are separated out explicitly in the algorithm.

- Our method turns inter-experiment variability from a problem into the solution. If all the experiments were identical, this method would fail completely. Thus, it explicitly depends on variability.

Commonly used methods usually involve applying the same single-experiment-based methodology to all the experiments, and averaging the results. This implicitly assumes that the differences between experiments are meaningless noise, and random enough to be removable by averaging. The success of the method proposed here demonstrates these assumptions to be untrue.

Thus, the ICS survey may be thought of as a first attempt to make use of this information contained in experimental variability. This is a rich and as yet untapped source of information, and it is hoped that our method encourages future work in this direction.

Chapter 7

Summary and Future Work

We now summarize the major results in each chapter and discuss possible avenues for future work.

1 RG and Statistics

In the second chapter, we introduced the Renormalization Group as a means of constructing a phenomenological description of a system of interest. We explained how the strong law of large numbers and central limit theorem emerge naturally by applying RG to an appropriately chosen dynamical system. The RG process can be interpreted as a search for stability. In the case of statistics, we suggest that the appropriate choice is a stability against addition of more data. Thus, the basic statistical quantifiers can be 'derived' through such a stability argument. The hope is that this stability can be extended beyond these obvious cases and can offer novel insight about more complicated statistical quantifiers. We followed through on this in various ways in the rest of the thesis.

One direct (although perhaps not simple) possibility that was not fully explored is to continue the sequence that yielded the strong law of large numbers and central limit theorem as the first two terms. The next term in the series should be a non-standard statistical quantifier. The first term in the series, namely the law of large numbers is a descriptor of samples. The next term, corresponding to the central limit theorem describes the distribution of the previous term, viz. the sample means, which for a finite sample size, deviate from the true mean. Thus, the third term should describe the distribution of the finite size CLT type distributions themselves.

The problem of pursuing a mathematical structure supported by a population may be interpreted as a problem of arranging the sample points appropriately in a certain space (for example, as performed by a dimensional reduction algorithm). In Chapter 3, we extended the use of the idea of stability (against addition of new data) to such quantifiers. In particular, we applied it to a dimensional reduction method known as non-Metric Multidimensional Scaling (nMDS). While proposing modifications to the implementation of nMDS dynamics, we found that we were faced with two very similar dynamics, which one might naively expect to give very similar results. One of these dynamics conformed to the stability idea, in that new points being added to the structure maximally respect the existing structure, while the other one did not. It was found in practice that the first scheme performs much better than the other, thereby vindicating our idea. This modified dynamics requires much less inequality information, and could be very advantageous in fields (such as the humanities) where inequalities are the direct input to nMDS.

2 Methods to Analyze Gene Expression Data Sets

In Chapters 3, 4, and 5 we discussed various aspects of the dimensional reduction, analysis and visualization of microarray-based gene expression data sets. In Chapter 3, nMDS was introduced as a dimensional reduction method and its superiority to other more sophisticated schemes was demonstrated in the analysis of a microarray based data set studying the developmental stages of *Drosophila*.

The agenda of Chapter 4 was to replace cluster analysis, which is the dominant method in the field, despite the existence of other methods more suited to the analysis of gene expression data. A major reason for the continued use of cluster analysis is its role in the construction of clustered heatmaps, presently by far the most popular visualization method. With a view to undermine this we demonstrated the superiority of NeatMap, a package created by us to create heatmap like plots by using more suitable dimensional reduction methods (in preference to cluster analysis).

In Chapter 5, we found that despite using the best possible dimensional reduction schemes, the performance, in biological terms, was quite poor. This is attributable to two effects: a) biological experiments are very noisy, with many points carrying very little biological information b) biological data sets capture a large number of processes, and are consequently very high dimensional. Thus, if the results of a whole genome data set are projected onto a low dimensional space, the results are not very meaningful. The solution to these problems involved two steps. First, rather than analyze all the genes at the same time, some method for pre-selection must be used to separate out the individual processes which can then be analyzed. Secondly, a noise reduction procedure must be applied to remove less meaningful points. Unfortunately, (until the emergence of the ICS Survey which we described in Chapter 6) there were very few data driven pre-selection methods and all of them risked biasing the results in non-biological ways. We therefore proposed that some biology driven pre-selection method be used to produce a candidate set of genes (even a little) likely to be involved in the process of interest. The selection can then be improved by using an appropriate noise reduction method. For this noise reduction method, we proposed that since genes are expected to work together, meaningful points must be consistent with the low dimensional structure identified by (say) nMDS. Therefore, by removing points inconsistent with this, the noise content may be reduced. This method was shown to improve biological meaning and reduce pre-selection biases.

Our method of choice for dimensional reduction has been nMDS because it is completely data driven, non-linear and produces better results than comparable methods like PCA. One of the primary drawbacks of nMDS as compared to other methods are its demanding computational needs. Since its traditional implementation requires all pairwise distances to be ranked, it scales quite poorly with the number of points. Thus, if the computational speed were increased, without sacrificing performance, it would greatly increase the usability of nMDS. We have conceived two ways in which this might be done

1. Instead of considering all pairs at once, partition the points into related groups, and perform nMDS honestly on each of them. Then perform some appropriate joining technique to stitch the different groups back together to generate an nMDS embedding for the whole set. An approach of this type has been adopted by Tzeng *et al.* [68]. We could potentially adopt this to our algorithm

2. The time consuming step in nMDS is the ranking step. Thus, if we could invent a scheme where ranking is not required at each step, it would be much faster. One possibility that has shown early promise is as follows. It is possible to move the embedded points around to match some set of target pairwise distances D_{ij}, provided these are not too far away from the present set of distances d_{ij}. The idea is to have a sequence $\{D^1_{ij}, D^2_{ij}, \cdots\}$ which converges to a set of distances consistent with the rankings. At any intermediate step k, the dynamics pushes the points around to match D^k_{ij}, and once a fixed point is reached, the target is updated.

3 ICS Survey

The main idea in Chapter 6 was to extend the stability idea to the realm of multiple experiments. The difference in experimental parameters (known and unknown) were used as perturbations to shake the system. Stable parts, in this case groups of genes with well preserved relations, were identified and found to be biologically related.

Biologically, the results were found to be very significant. The ICS Survey method, despite being completely data driven, identifies the important classes of genes in a data set, with performance comparable to specialized methods designed to identify specific classes of genes. It also provides a way to solve the pre-selection problem that dogs most exploratory methods like nMDS.

Ideologically too, the ICS survey represents a major shift. It looks for consistency of correlation rather than strength. Also, it explicitly depends on experimental variation; unlike other methods whose results would improve if experiments were identical, the ICS survey would fail completely. Thus, the ICS Survey is one of the few methods to attempt to make use of the information contained in experimental variation. It is our belief that there is much untapped potential in this information, and the ICS Survey should be considered as a first attempt at utilizing this (that too only in a very specific way, i.e., through consistency).

3.A Possible ICS Survey Related Projects

The ICS Survey is our newest and most significant work. Consequently, it offers more interesting avenues for future work than our other projects. Below a few options are ordered according to the ease with they may be achieved.

Defining Consistency/Stability

Our basic idea was to look for features that are well preserved across experiments. Essentially, we looked for rigid objects in expression space, i.e., those whose relative positions across experiments were related by affine transformations. This definition could be extended.

The easiest way to implement this idea is by changing the normalization scheme used before combining experiments. As discussed in the last chapter, for each gene g and experiment e, we calculate a vector consisting of the correlations of the profiles of the ICC genes $\{h_1, \ldots, h_N\}$ with that of g, which I shall call the raw correlation profiles:

$$(C^e(g, h_1), \cdots, C^e(g, h_N))$$

The present scheme of normalizing this to have mean zero and unit variance is the most naive one imaginable.

$$\boldsymbol{c_g^e} = \frac{1}{\sigma_g^e} \left(C^e(g, h_1) - m_g^e, \cdots, C^e(g, h_N) - m_g^e \right)$$

Here, m_g^e and σ_g^e are the mean and standard deviation of the raw vector respectively. This normalization scheme has some clear drawbacks:

1. It only works with bounded measures (such as correlations). For a moderate ICC size, if one of the distances were to become very large, then this scheme would strongly affect all the others, making it sensitive to outliers. These are more pronounced in un-bounded schemes

2. It is biased against ICCs consisting of genes with very similar profiles. If the genes $h_1, \ldots, h_N$ have very similar profiles for a given experiment e, then it follows that the correlations $C^e(g, h_1), \cdots, C^e(g, h_N)$ too will be nearly identical. It follows that σ_g^e becomes tiny, thereby greatly magnifying any differences in $\boldsymbol{c_g^e}$. This makes it very difficult for $\boldsymbol{c_g^e}$'s to be consistent across experiments.

3. The normalization doesn't take into account the distribution of correlations. It is found to perform poorly for cases when the distribution of correlations is skewed.

Keeping these points in mind, a novel normalization scheme has been developed that replaces the raw correlations by their corresponding cumulative distribution function (CDF) values. More specifically, for gene g and experiment e the cumulative distribution of the set $\{C^e(g, 1), \cdots, C^e(g, N_g)\}$ is used. Here, N_g is the total number of genes (not the ICC size). The CDF is bounded, insensitive to outliers, and automatically accounts for the distribution of correlations. No further normalization is found to be necessary, and the problems mentioned above are solved. The details of this will be part of a future publications, but the results so far seem very promising

Moving Beyond Microarray Based Time Course Experiments

In this thesis, we primarily discussed applying the ICS survey to microarray based time course experiments with experiments differing in protocol. There are few obstacles to extending the methodology to other types of microarray experiments. For example, striking results have been observed where measurements are in different tissues instead of times.

Looking further, it should be possible to consider experiments that capture different types of biological information. In such a case the ICS survey will identify the true conserved features, and will allow us to go beyond the idiosyncrasies of the individual experiment.

Perhaps the most interesting applications are where the different experiments correspond to different species, and striking results have already been produced in human-chimpanzee comparisons. It is also possible to extend this methodology to experiments other than microarrays, e.g., protein interaction data, and sequence information. One of the primary obstacles to doing this was the inability of the ICS survey to work with non-bounded distance measures. As mentioned in the section above, this problem has been solved using the cumulative distribution based normalization. Some thought also needs to be given to what it means to be consistent when such different types of information are being combined.

Identifying Less Prominent ICCs

In most of the data sets we have worked with, only a few distinct ICCs have been identified with a significant number of runs converging to them. It is conceivable that these are just the most prominent ICCs, and there are many others that the dynamics happens to visit less frequently. To effectively sample these ICCs one would need to perform many more runs. Multiple runs corresponding to the same ICC are not used except for averaging (see next section for alternatives) and thus the majority of runs are wasted. Thus, it might be advantageous to bias the dynamics in a way that less prominent ICCs are visited more frequently. One easy way is to terminate runs that start to converge towards existing ICCs. A more sophisticated alternative would be to choose the initial seed genes in a biased fashion (perhaps by using the results of inferior methods).

Non-overlapping Data

Another important feature that needs to be added to the ICS Survey for dealing with multiple data types is the ability to deal with non-overlapping data. Experiments are often performed on different sets of genes, and in its present form the ICS Survey would only provide results for those genes common to all experiments. Instead, a scoring scheme that appropriately awards genes for being present in more experiments (but does not completely discard genes not present in all of them) is needed.

Combining Large Numbers of Experiments

The present implementation of ICS looks for relations that are preserved across *all* experiments. This is a reasonable strategy when one is considering a few related experiments. However, if one wants to build databases by combining a huge number of diverse experiments, this could be problematic. The same gene often plays roles in multiple processes. Thus, experiments performed under different conditions, that invoke many different processes, may fail to show any relations that are preserved across all of them.

For example, consider the situation where genes A and B perform some very basic function, e.g., methylation. Suppose A and B function together very closely in the response to heat shock. Then, the ICS Survey applied to a group of heat-shock experiments would identify the connection between A and B. However, if A is also used in some other process, say spermatogenesis, but B is not, then ICS survey applied to such experiments would not connect A and B. With the present implementation of ICS, if we combined both sets of experiments, the connection between A and B would not be detected, because of the lack of consistency in a subset of experiments. This behavior is sub-optimal.

The most obvious solution would be to look for relations that are very strongly preserved across at least a few experiments. Unfortunately, as the number of experiments grows, the computational effort in this grows combinatorially. One possible way to deal with this would be to follow an approach similar to that taken by Hibbs *et al.* [69] in combining experiments. Before data integration, they identified the significance of specific experiments for certain classes of processes (based on Gene Ontology). Then, for any interaction, only the relevant subset of experiments were considered. Perhaps we too could use such an approach.

Beyond Simple Consistency

As mentioned earlier, the ICS Survey is an initial attempt at making use of the information contained in inter-experiment variability. Consistency provides just one window into this kind of information, and there are likely to be many other approaches (although speculation about them is outside the scope of this thesis). However, even within the scope of our method, there are sources of information we have not exploited.

Figure 6.4 in Chapter 6 shows the ICCs corresponding to different runs. We concluded that there were only two kinds of ICCs and averaged over all the runs within each ICC, thereby discarding any higher order information in their distributions. Fig. 7.1 shows a similar plot for a different data set. This data [70] was taken across different tissues in *Drosophila melanogaster*, and the different experiments correspond to technical replicates. Technical replicates are nearly identical experiments, and are therefore expected to show far less variation. In this case, instead of showing sharply defined ICCs, we also see a continuum among a subset of the runs. Thus, as one might expect, as the amount of variation decreases, the performance degrades. However, this result also suggests that there is information about the variation of the experiments in the distribution of runs. It is as yet unclear how best one might use this information. Application to artificially generated data sets with different models of variation could be a good first step to build intuition.

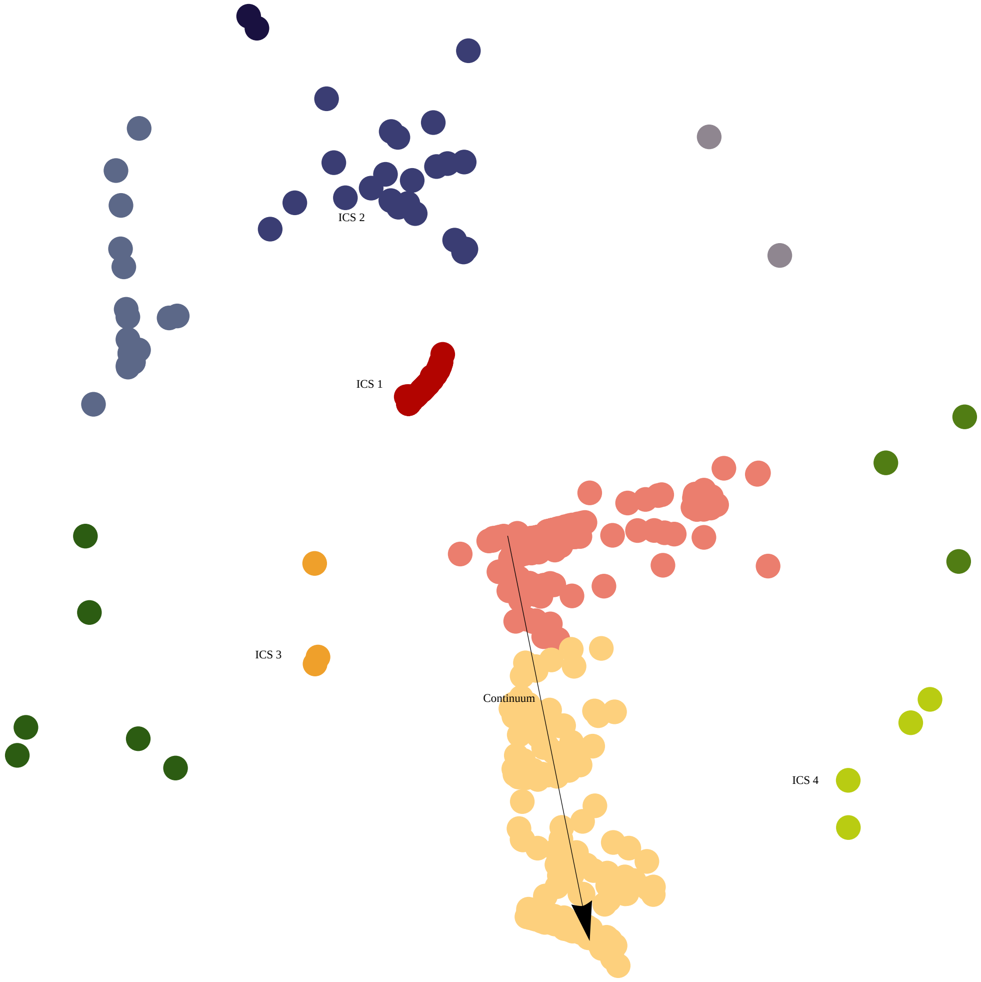


Figure 7.1: The results of ICS Survey applied to a *Drosophila* data set. Each point here corresponds to the fixed point ranking of a single ICS survey run. The different rankings were compared using their Spearman rank correlation and embedded into 2D using nMDS. The 2D results were the clustered and k-means clustering was performed to color the points.

4 Conclusion

The goal of this thesis was to promote a phenomenological approach to statistics and data analysis, particularly in the context of bioinformatics. We believe that the renormalization group is the right tool to construct such a phenomenology. RG can be thought of as a search for stability, and thus stability is the constant motif in our approaches. We proposed that in the context of single experiments, stability against addition of additional data points should be required. This approach was used to derive the standard statistical quantifiers, and offered useful non-trivial guidance in the choice of algorithms for more complicated statistical quantifiers. In the context of multiple experiments, we use the difference in experiments as perturbations and search for stable/well-preserved relations. Based on this idea, we proposed a data-driven method called the ICS Survey, that identifies related groups of genes far more reliably than existing methods. Given the significance of these results, we believe that the usefulness of phenomenological approaches to statistics has been demonstrated, and we hope to spur future work in this direction.

Appendix A

Central Limit Theorem from Wilson-Kadanoff type RG

The Central Limit Theorem (CLT) states that if $X_1, X_2, \ldots, X_n$ are a sequence of independent and identically distributed (i.i.d) variable each having finite values and expectation value μ and variance $\sigma^2 > 0$, then in the limit $n \rightarrow \infty$, the partial sum $S_n = X_1 + \ldots + X_n$ approaches a Normal distribution:

$$\lim_{n\rightarrow\infty} \frac{\sum_{i=1}^{n}(X_i - \mu)}{\sigma n^{1/2}} = N(0,1),$$

where convergence is in distribution and $N(0,1)$ is a normal distribution of mean zero and unit variance.

In the Wilson-Kadanoff type RG, we begin with a small system at the microscopic scale, and look at some quantity of interest. We then consider bigger and bigger systems and look at how this quantity changes. Ultimately, we are interested in the fixed-point behavior of this procedure as the system size tends to infinity. To derive the CLT using Wilson-Kadanoff type RG we follow Jona-Lasinio [12]. Spins (corresponding to the X_i) are constructed on an infinite one dimensional lattice. This lattice is divided into a hierarchical structure of blocks of increasing size. The probability distribution of the total spin in a block is studied as a function of block size, particularly in the asymptotic limit of infinite block size. It is found that a normal distribution is a fixed point distribution, and that even if the distributions for smaller block size show deviation from a Gaussian, in the limit of large block size they shall become Gaussian. This constitutes a proof of the CLT.

Let $\xi_i = \frac{X_i - \mu}{\sigma}$ be a spin on a one dimensional lattice. ξ_i is therefore effectively drawn from a distribution of mean zero and unit standard deviation. Let us define a hierarchy of blocks where each block may be subdivided into two smaller ones. This division may be repeated till we arrive at the individual spins. We may then define spin variables as:

$$\chi_n^1 = 2^{-n/2} \sum_{i=1}^{2^n} \xi_i \text{ and } \chi_n^2 = 2^{-n/2} \sum_{i=2^n+1}^{2^{n+1}} \xi_i$$

There are 2^n spins in each such block, and since the spins themselves have zero mean and unit variance, these variables are of the form $\sum_{i=1}^{n}(X_i - \mu)/\sigma n^{1/2}$, which is the form of the central limit theorem. Thus, to prove the CLT, we only need to prove that as $n \to \infty$ the distribution of these spin variables approaches a normal distribution.

The spin variable for a large block can be related to that of its constituent sub-blocks by the formula:

$$\chi_{n+1} = \frac{1}{\sqrt{2}}(\chi_n^1 + \chi_n^2).$$

Let $p_n(x)$ represent the probability distribution of χ_n. The equation above implies the recursive relation

$$p_{n+1}(x) = \sqrt{2} \int p_n(\sqrt{2}x - y) p_n(y) \mathrm{d}y = (\mathcal{R} p_n(x)).$$

The transformation $\mathcal{R}$ is called a renormalization group transformation. It relates the value of a variable before and after scaling. Notice that $\mathcal{R}$ preserves the normalization, mean and variance of the distribution. We are interested in what happens in the limit $n \to \infty$, and in particular we want to prove that $p_n(x)$ converges to a normal distribution. Since the above formula is a convolution, we may prove this using Fourier transforms. However, we choose to outline a 'proof' using a method that is closer to the traditional RG application with which the reader may be familiar.

Knowing this it is easy to check that a normal distribution $p_G(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ is a fixed point of the dynamical system defined by $\mathcal{R}$. To prove the CLT we must show that it is an attractive fixed point, and that irrespective of the probability distributions of the spins themselves there is convergence to a normal distribution after sufficient coarse graining. To prove this, let us study the evolution of the p_n under the RG coarse graining dynamics for an arbitrary distribution. We expand the arbitrary distribution around the Gaussian fixed point as:

$$p_\eta(x) = p_G(g)\Big(1 + \eta h(x)\Big),$$

where η is a parameter measuring the deformation from a normal distribution. Applying the RG transformation

$$(\mathcal{R}p_\eta) = \mathcal{R}\Big(p_G(x)(1 + \eta h(x))\Big)$$

If η is small we may linearize around the Gaussian by only keeping terms upto first order in η

$$\begin{aligned}
(\mathcal{R}p_\eta)(x) &= \mathcal{R}(p_G) + \sqrt{2}\eta \int p_G(y)p_G(\sqrt{2}x - y)[h(\sqrt{2}x - y) + h(y)]\mathrm{d}y \\
&= p_G(x) + \frac{2\eta}{\sqrt{\pi}}p_G(x) \int e^{-z^2} h(\sqrt{2}x + z)\mathrm{d}z \\
&\equiv p_G\Big(1 + \eta(\mathcal{L}h)\Big)(x)
\end{aligned}$$

The integral operator $\mathcal{L}$ has the Hermite polynomials H_k as eigenfunctions with corresponding eigenvalues:

$$\lambda_k = 2^{1-k/2}$$

It is natural therefore to expand $h(x)$ in the basis of Hermite polynomials. The normalization, mean and variance preserving nature of $\mathcal{R}$ imposes the conditions that

$$\int p_G(x)h(x)\mathrm{d}x = 0$$
$$\int p_G(x)xh(x)\mathrm{d}x = 0$$
$$\int p_G(x)x^2h(x)\mathrm{d}x = 0$$

This implies that it has zero projections on the first three Hermite polynomials. Thus, only the Hermite polynomials with $k > 3$ show up in the expansion of h.

$$h(x) = \sum_{i=4}^{\infty} c_i H_i(x).$$

So, if we repeatedly apply $\mathcal{R}$, we find that

$$\begin{aligned}
(\mathcal{R}^n p_\eta)(x) &= p_G\Big(1 + \eta(\mathcal{L}^n h)\Big)(x), \\
&= p_G\Big(1 + \eta \mathcal{L}^n \sum_{i=4}^{\infty} c_i H_i(x)\Big)(x), \\
&= p_G\Big(1 + \eta \sum_{i=4}^{\infty} c_i \lambda_i^n H_i(x)\Big)(x).
\end{aligned}$$

For all $i \geq 4$ corresponding eigenvalues λ_i are less than $1/2$. Thus, as $n \to \infty$, each of these terms goes to zero. And hence we have the result that

$$\lim_{n\to\infty} \mathcal{R}^n p_\eta = N(0, 1)$$

which is the central limit therem.

In this case the spins on different sites were i.i.d and hence uncorrelated. The typical application of RG (for example, consider the Ising model) is to cases when there are strong correlations between sites. Thus, as shown by Jona-Lasinio [11, 12], RG may be thought of as an attempt to expand the realm of the central limit theorem to (strongly) correlated variables.

Appendix B

Microarray Experiments

DNA Microarrays are tools used to simultaneously measure the expression levels of a large number of genes at a given point of time. Since it is believed that mRNA levels represent the amount of transcription taking place, it is these levels that microarrays attempt to measure. Strictly speaking, what one is typically interested in is differential expression, i.e., the change in expression of a gene. Therefore, two such measurements are usually made (one is often a baseline measurement), and these are compared. Thus, the result of a microarray experiment is usually a ratio of expression levels rather than an absolute level. This is thought to reduce experimental artifacts since both measurements used to construct the ratio should be affected in the same way.

As it turns out, mRNA is fairly unstable (especially in prokaryotes). Therefore, in practice instead of measuring the mRNA levels directly, the strands of mRNA are used to synthesize complementary strands of DNA (cDNA) using the enzyme reverse transcriptase. cDNA is far more stable, and it is in fact the cDNA levels which are directly measured in a microarray experiment.

The microarray itself is typically a glass slide or nylon membrane marked with a very large number of tiny spots arranged in an array (hence the name). Each spot is designed so that only a specific cDNA attaches to it. This is achieved in different ways, according to the type of microarray, but will always contain a sequence of base pairs present in (and hopefully unique to) the cDNA of interest. The microarray is exposed to the cDNA (synthesized from mRNA) from a sample of interest, and by measuring the relative amounts of cDNA attached at the different spots, we can get an idea of the relative mRNA levels.

Steps Involved in a Typical Microarray Experiment

There are many different ways in which microarray experiments can be performed. Most of them involve the following steps:

1. Prepare the biological sample for which the gene expression levels are desired.

2. Purify the sample for RNA.

3. Generate a cDNA strand corresponding to each RNA, using reverse transcriptase. The amounts of cDNA may be amplified using PCR amplification. A label/fluorescent dye is usually attached to the cDNA to facilitate measurement. The nature of this dye depends on the type of microarray being used.

4. The labelled sample is placed on the microarray and hybridized. The cDNA attach to the appropriate spots on the microarray.

5. The spots are excited using lasers, and the fluorescent intensity at each spot is measured.

6. This is repeated for all the spots, and the intensity levels are supposed to be a rough measure of mRNA level, and by extension, gene expression.

7. The measurements are appropriately normalized, first by taking the appropriate ratio as mentioned earlier, and subsequently to account for dye, array and other artefactual experimental effects.

Popular Microarray Types

There are many different types of microarrays, but in general most experiments correspond to one of two types [71]:

- cDNA spotted microarrays: cDNA (usually from a cDNA library) are generated and printed onto the slides as spots at defined locations. Typically the cDNA from the two populations to be compared are labelled with fluorescent dyes known as Cy3 (which is green) and Cy5 (which is red). Spots corresponding to both dyes must be on the same array (to avoid inter-array effects), and it is the ratio of these intensities that is the result of the experiment.

- Oligonucleotide microarrays : In this case, rather than using cDNA, a much smaller DNA sequence (20-25mer) is synthesized *in situ*.The high-reproducibility of this process allows accurate comparison of signals hybridized to different arrays. cDNA is tagged with an appropriate fluorescent marker and the ratio of the intensities of the corresponding spots on the two arrays is the result of the experiment.

Statistical Analysis

The result of the microarray experiment is the ratio of the intensities at two spots. We are usually interested in the (ratio of) the mRNA levels. Unfortunately these are not the same because of a number of experimental effects. A number of fairly sophisticated models [72, 73, 74] have been developed to infer the mRNA levels from the experimental measurements.

In particular, it is found that even for the same gene, the expression level depends on dye, array, array position and other such effects. To account for these, typically the measured level is modelled as a sum of these effects in a linear model such as [75]:

$$y_{ijgr} = \mu + A_i + D_j + (AD)_{ij} + G_g + (AG)_{igr} + (DG)_{jg} + \epsilon_{ijgr}.$$

Here, μ represents the average signal across all factors. The global effects A_i, D_j, and $(AD)_{ij}$ account for overall variation in arrays and dyes. The gene effect G_g accounts for average signal for gene g across arrays dyes and varieties. $(AG)_{igr}$ refers to the spot effects which are a function of array position. $(DG)_{jg}$ terms are gene-specific dye effects and so on.

ANOVA type analysis is performed, and by comparing different spots having the same properties (dye/array positions/array) these individual effects are estimated and subtracted out. Many of these assumptions are *ad hoc*, and there is a lack of consensus on the appropriate statistical model to use.

Reliability of Microarray Experiments

One of the foundations of science is the reproducibility of experiments. Microarray experiments often depend on detailed experimental parameters and their reproducibility is notoriously bad. To combat this, a standard known as the Minimum Information About A Microarray Experiment (MIAME) was formulated which provides a checklist of details that need to be supplied when reporting microarray results. Despite this it has been shown that microarray results show very poor reproducibility and could even depend on factors such as lab quality [57].

Thus, even after processing, microarray results are likely to contain many artefactual effects. Thus data analysis methods to analyze the should be able to deal with such noise.

Appendix C

PCA honing

The honing procedure can be used with any multivariate pattern extraction technique. We have shown its usefulness with nMDS. nMDS is arguably the most powerful unsupervised pattern extraction method, but even our efficient algorithm used in this paper is much slower, although sufficiently fast to be practical, than the linear algebraic methods such as PCA. Ordinary PCA is not very useful in gene analysis, because the dynamic range of the data values is large. In such cases, it is best to perform PCA on the correlation matrix generated from the normalized, mean-subtracted profiles [76]. Let us call this normalized PCA. Indeed, this normalized PCA can give roughly the same result as nMDS if the data quality is excellent and low dimensional enough. Therefore, we describe a honing procedure for normalized PCA here.

The procedure may be summarized as follows (a similar procedure should work for kernel PCA):

1. Apply (normalized) PCA to the whole data set (missing data must be treated appropriately).

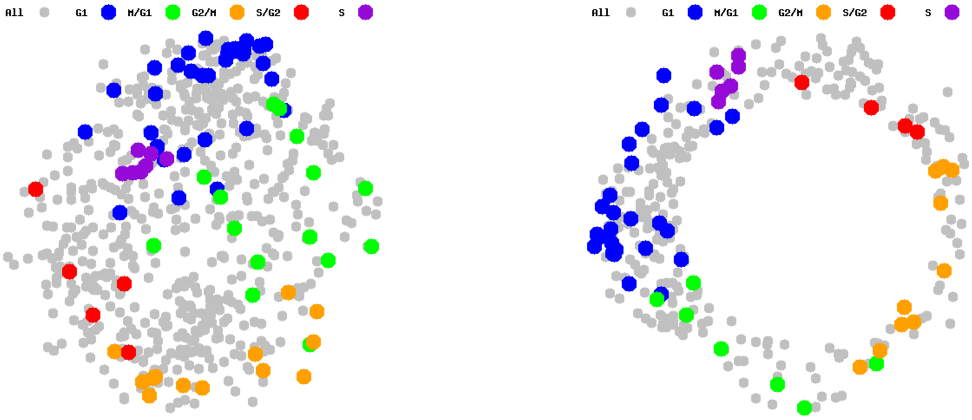
2. Pick an embedding dimension D, i.e., the number of PCA components to be used.

3. For each point (gene), find the percentage of variance captured by the D-dimensional PCA subspace.

4. Arrange the vectors in decreasing order of this percentage.

5. Discard appropriately using the analogue of the bootstrap scheme proposed for nMDS above or comparison to random/shuffled data.

It is systematically found that normalized PCA based honing requires more points to



(a) Spellman Selection

(b) Honed

Figure C.1: Analysis of Spellman's *cdc15* data (the normalized PCA version of Fig. 5.7; color codings are the same). (a) shows the result using Spellman's choice of cyclic genes; (b) shows the result produced after honing down to 334 genes (chosen to allow comparison to nMDS).

be discarded than its nMDS counterpart. This is consistent with the resolving power of these methods; nMDS sees a bigger difference between true and random data of the rank mismatch. Despite such differences, there is a rough qualitative agreement between the results. It should be noted incidentally that the PCA honing procedure is very similar to that of "gene shaving" [77]. However, "gene shaving" was not intended to be used for noise reduction, and the method of deciding the number of points to discard is quite different (and seemingly inappropriate for this purpose).

In Fig. C.1 the results obtained by PCA are compared with those by nMDS for Spellman *et al.* As can be surmised from the figure, the agreement in the angular positions determined by nMDS and PCA improve dramatically with honing. Before honing, although some clustering of genes up-regulated in specific cell cycle phases is seen, it is very unclear if we can consider this a good starting point for the honing procedure. After honing however, the correct time ordering is very clear. Thus, although PCA+honing can extract the correct cell cycle behavior it is too week a method to use for determining the correct number of cell cycle related genes. The effect of PCA honing may look more impressive than its nMDS counterpart, but it is simply because PCA is so weak that it cannot extract a significant structure by itself.

References

[1] G.R. Fink, P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular biology of the cell*, 9(12):3273–3297, 1998.

[2] Michael Hahsler, Christian Buchta, and Kurt Hornik. *seriation: Infrastructure for seriation*, 2009. R package version 1.0-0.

[3] C.H. Chen. Generalized association plots: Information visualization via iteratively generated correlation matrices. *Statistica Sinica*, 12(1):7–30, 2002.

[4] Y Oono. Integrative Natural History. Lecture Notes, 2009.

[5] J. Thewlis. Concise dictionary of physics. *Concise dictionary of physics., by Thewlis, J.. Oxford (UK): Pergamon Press, 8+ 366 p.*, 1973.

[6] Y Oono. Informal Lecture Notes on Renormalization and Phase Transitions. Lecture Notes, 2008.

[7] K.G. Wilson. Renormalization group and critical phenomena. I. Renormalization group and the Kadanoff scaling picture. *Physical Review B*, 4(9):3174–3183, 1971.

[8] ECG Stueckelberg and A. Petermann. La normalisation des constantes dans la theorie des quanta. *Helv. Phys. Acta*, 26:499, 1953.

[9] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular biology of the cell.* Garland, 2002.

[10] Y. Oono and M. Kohmoto. Renormalization group theory of transport properties of polymer solutions. I. Dilute solutions. *The Journal of Chemical Physics*, 78:520, 1983.

[11] G. Jona-Lasinio. The renormalization group: A probabilistic view. *Il Nuovo Cimento B (1971-1996)*, 26(1):99–119, 1975.

[12] G. Jona-Lasinio. Renormalization group and probability theory. *Arxiv preprint cond-mat/0009219*, 2000.

[13] Y. Oono. Renormalization and asymptotics. *International Journal of Modern Physics B*, 14(12/13):1327–1362, 2000.

[14] K. Hasselmann. Stochastic climate models. *Tellus*, 28:473–485, 1976.

[15] V. Bakhtin and Y. Kifer. Diffusion approximation for slow motion in fully coupled averaging. *Probability Theory and Related Fields*, 129(2):157–181, 2004.

[16] L.Y. Chen, N. Goldenfeld, and Y. Oono. Renormalization group and singular perturbations: Multiple scales, boundary layers, and reductive perturbation theory. *Physical Review E*, 54(1):376–394, 1996.

[17] Y. Oono. Onsager's Principle from Large Deviation Point of View. *Progress of Theoretical Physics*, 89(5):973–983, 1993.

[18] P. Sprent and P. Sprent. *Data driven statistical methods.* Chapman & Hall London, 1998.

[19] G. Shafer and V. Vovk. *Probability and finance: it's only a game!* Wiley-Interscience, 2005.

[20] R. N. Shepard. The analysis of proximities: multidimensional scaling with an unknown distance function I. *Psychometrika*, 27:125–139, 1962.

[21] R. N. Shepard. The analysis of proximities: multidimensional scaling with an unknown distance function II. *Psychometrika*, 27:219–246, 1962.

[22] J. B. Kruskal. Multidimensional scaling by optimizing goodness-of-fit to a nonmetric hypothesis. *Psychometrika*, 29:1–29, 1964.

[23] J. B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29:115–129, 1964.

[24] Y. Taguchi and Y. Oono. Relational patterns of gene expression via non-metric multidimensional scaling analysis. *Bioinformatics*, 21(6):730–740, 2005.

[25] Ingwer Borg and Patrick J. F. Groenen. *Modern Multidimensional Scaling : Theory and Applications (Springer Series in Statistics)*. Springer, 2005.

[26] Y.-h. Taguchi and Y. Oono. Nonmetric multidimensional scaling as a data-mining Tool: new algorithm and new targets. *Advances in Chemical Physics*, 130B:315–351, 2005.

[27] Y.-h. Taguchi and Y. Oono. Relational patterns of gene expression via non-metric multidimensional scaling analysis. *Bioinformatics*, 21(6):730–740, 2005.

[28] Y-h Taguchi. private communication, 2005.

[29] C.W. Whitfield, A.M. Cziko, and G.E. Robinson. Gene expression profiles in the brain predict behavior in individual honey bees, 2003.

[30] G. Punj and D.W. Stewart. Cluster analysis in marketing research: review and suggestions for application. *Journal of Marketing Research*, pages 134–148, 1983.

[31] H.J. Zeng, Q.C. He, Z. Chen, W.Y. Ma, and J. Ma. Learning to cluster web search results. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 210–217. ACM New York, NY, USA, 2004.

[32] GB Coleman and HC Andrews. Image segmentation by clustering. *Proceedings of the IEEE*, 67(5):773–785, 1979.

[33] GE Fox, E. Stackebrandt, RB Hespell, J. Gibson, J. Maniloff, TA Dyer, RS Wolfe, WE Balch, RS Tanner, LJ Magrum, et al. The phylogeny of prokaryotes. *Science*, 209(4455):457–463, 1980.

[34] J.D. Thompson, D.G. Higgins, and T.J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–4680, 1994.

[35] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns, 1998.

[36] J.N. Weinstein, T.G. Myers, P.M. O'Connor, S.H. Friend, A.J. Fornace Jr, K.W. Kohn, T. Fojo, S.E. Bates, L.V. Rubinstein, N.L. Anderson, et al. An information-intensive approach to the molecular pharmacology of cancer. *Science*, 275(5298):343, 1997.

[37] Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS*, 95(25):14863–14868, 1998.

[38] J. Bobe, J. Montfort, T. Nguyen, and A. Fostier. Identification of new participants in the rainbow trout(Oncorhynchus mykiss) oocyte maturation and ovulation processes using cDNA microarrays. *Reproductive Biology and Endocrinology*, 4(1):39, 2006.

[39] Y. Le Priol, D. Puthier, C. Lecureuil, C. Combadiere, P. Debre, C. Nguyen, and B. Combadiere. High cytotoxic and specific migratory potencies of senescent CD8+ CD57+ cells in HIV-infected and uninfected individuals. *The Journal of Immunology*, 177(8):5145, 2006.

[40] J.N. Weinstein. BIOCHEMISTRY: A Postgenomic Visual Icon. *Science*, 319(5871):1772, 2008.

[41] F. Hoeppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy cluster analysis: methods for classification, data analysis, and image recognition.* Wiley, 1999.

[42] DA Reynolds and RC Rose. Robust text-independent speaker identification using gaussianmixture speaker models. *IEEE transactions on Speech and Audio Processing*, 3(1):72–83, 1995.

[43] AK Jain, MN Murty, and PJ Flynn. Data clustering: a review. *ACM computing surveys*, 31(3), 1999.

[44] J. Handl, J. Knowles, and D.B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201–3212, 2005.

[45] D.A. Baum, S.D.W. Smith, and S.S.S. Donovan. The tree-thinking challenge. *Science(Washington)*, 310(5750):979–980, 2005.

[46] S. Raychaudhuri, J.M. Stuart, and R.B. Altman. Principal components analysis to summarize microarray experiments: application to sporulation time series. In *Pac Symp Biocomput*, volume 5, pages 455–466, 2000.

[47] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.

[48] M.A. Hibbs, N.C. Dirksen, K. Li, and O.G. Troyanskaya. Visualization methods for statistical analysis of microarray clusters. *BMC bioinformatics*, 6(1):115, 2005.

[49] Hadley Wickham. *ggplot2: An implementation of the Grammar of Graphics*, 2008. R package version 0.8.

[50] Daniel Adler and Duncan Murdoch. *rgl: 3D visualization device system (OpenGL)*, 2009. R package version 0.84.

[51] A.I. Su, T. Wiltshire, S. Batalov, H. Lapp, K.A. Ching, D. Block, J. Zhang, R. Soden, M. Hayakawa, G. Kreiman, et al. A gene atlas of the mouse and human protein-encoding transcriptomes. *Proceedings of the National Academy of Sciences*, 101(16):6062–6067, 2004.

[52] R.J. Cho, M. Huang, M.J. Campbell, H. Dong, L. Steinmetz, L. Sapinoso, G. Hampton, S.J. Elledge, R.W. Davis, and D.J. Lockhart. Transcriptional regulation and function during the human cell cycle. *Nature Genetics*, 27:48–54, 2001.

[53] Kerby Shedden and Stephen Cooper. Analysis of cell-cycle-specific gene expression in human cells as determined by microarrays and double-thymidine block synchronization. *PNAS*, 99(7):4379–4384, 2002.

[54] R.J. Cho, M.J. Campbell, E.A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T.G. Wolfsberg, A.E. Gabrielian, D. Landsman, D.J. Lockhart, et al. A genome-wide transcriptional analysis of the mitotic cell cycle. *Current Biology*, 13(19):1740–1745, 2003.

[55] L.L. Breeden. Periodic Transcription: A Cycle within a Cycle. *Current Biology*, 13(1):31–38, 2003.

[56] U. de Lichtenberg, R. Wernersson, T.S. Jensen, H.B. Nielsen, A. Fausbøll, P. Schmidt, F.B. Hansen, S. Knudsen, and S. Brunak. New weakly expressed cell cycle-regulated genes in yeast. *YEAST*, 22(15):1191–1201, 2005.

[57] T. Bammler, RP Beyer, S. Bhattacharya, GA Boorman, A. Boyles, BU Bradford, RE Bumgarner, PR Bushel, K. Chaturvedi, D. Choi, et al. Members of the Toxicogenomics Research Consortium: Standardizing global gene expression analysis between laboratories and across platforms. *Nat Methods*, 2(5):351–356, 2005.

[58] M.D. Slack, E.D. Martinez, L.F. Wu, and S.J. Altschuler. Characterizing heterogeneous cellular responses to perturbations. *Proceedings of the National Academy of Sciences*, 105(49):19306, 2008.

[59] H.K. Lee, A.K. Hsu, J. Sajdak, J. Qin, and P. Pavlidis. Coexpression Analysis of Human Genes Across Many Microarray Data Sets. *Genome Research*, 14:1085–1094, 2004.

[60] X. Yan, M.R. Mehan, Y. Huang, M.S. Waterman, P.S. Yu, and X.J. Zhou. A graph-based approach to systematically reconstruct human transcriptional regulatory modules. *Bioinformatics*, 23(13):i577, 2007.

[61] C.H. Wade, M.A. Umbarger, and M.A. McAlear. The budding yeast rRNA and ribosome biosynthesis (RRB) regulon contains over 200 genes. *Yeast*, 23(4):293–306, 2006.

[62] C. Wade, K.A. Shea, R.V. Jensen, and M.A. McAlear. EBP2 Is a Member of the Yeast RRB Regulon, a Transcriptionally Coregulated Set of Genes That Are Required for Ribosome and rRNA Biosynthesis. *Molecular and Cellular Biology*, 21(24):8638–8650, 2001.

[63] A. Oliva, A. Rosebrock, F. Ferrezuelo, S. Pyne, H. Chen, S. Skiena, B. Futcher, and J. Leatherwood. The cell cycle-regulated genes of Schizosaccharomyces pombe. *PLoS Biol*, 3(7):e225, 2005.

[64] G. Rustici, J. Mata, K. Kivinen, P. Lio, C.J. Penkett, G. Burns, J. Hayles, A. Brazma, P. Nurse, and J. Baehler. Periodic gene expression program of the fission yeast cell cycle. *Nature Genetics*, 36(8):809–817, 2004.

[65] X. Peng, R.K.M. Karuturi, L.D. Miller, K. Lin, Y. Jia, P. Kondu, L. Wang, L.S. Wong, E.T. Liu, M.K. Balasubramanian, et al. Identification of Cell Cycle-regulated Genes in Fission Yeast. *Molecular Biology of the Cell*, 16(3):1026–1042, 2005.

[66] S. Marguerat, T.S. Jensen, U. de Lichtenberg, B.T. Wilhelm, L.J. Jensen, and J. Bahler. The more the merrier: comparative analysis of microarray studies on cell cycle-regulated genes in fission yeast. *YEAST*, 23:261–277, 2006.

[67] KP Keegan, S. Pradhan, JP Wang, and R. Allada. Meta-analysis of Drosophila Circadian Microarray Studies Identifies a Novel Set of Rhythmically Expressed Genes. PLoS Comput Biol. *press. doi*, 10, 2007.

[68] J. Tzeng, H.H.S. Lu, and W.H. Li. Multidimensional scaling for large genomic data sets. *BMC bioinformatics*, 9(1):179, 2008.

[69] M.A. Hibbs, D.C. Hess, C.L. Myers, C. Huttenhower, K. Li, and O.G. Troyanskaya. Exploring the functional landscape of gene expression: directed search of large microarray compendia. *Bioinformatics*, 23(20):2692, 2007.

[70] V.R. Chintapalli, J. Wang, and J.A.T. Dow. Using FlyAtlas to identify better Drosophila melanogaster models of human disease. *Nature genetics*, 39(6):715–720, 2007.

[71] A. Schulze and J. Downward. Navigating gene expression using microarraysâĂŤa technology review. *Nature Cell Biology*, 3(8):E190–E195, 2001.

[72] R.A. Irizarry, B. Hobbs, F. Collin, Y.D. Beazer-Barclay, K.J. Antonellis, U. Scherf, and T.P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, 2003.

[73] W. Huber, A. von Heydebreck, H. Sultmann, A. Poustka, and M. Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18(Suppl 1):S96–S104, 2002.

[74] BM Bolstad, RA Irizarry, M. Astrand, and TP Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias, 2003.

[75] M. KATHLEEN KERR and G. A. CHURCHILL. Statistical design and the analysis of gene expression microarray data. *Genetics Research*, 77(02):123–128, 2001.

[76] Alvin C. Rencher. *Methods of Multivariate Analysis (Wiley Series in Probability and Statistics)*. Wiley-Interscience, 2002.

[77] Trevor Hastie, Robert Tibshirani, Michael B. Eisen, Ash Alizadeh, Ronald Levy, Louis Staudt, Wing C. Chan, David Botstein, and Patrick Brown. 'gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, 1(2):0003.1–0003.21, 2000.

Author's Biography

Satwik Rajaram was born on Jan. 12, 1981 in Mumbai, India. He received his BSc. degree in Physics from St. Xavier's College, Mumbai University, India in 2001. In August 2001 he joined the graduate studies program in Physics at the University of Illinois at Urbana-Champaign.