TWO METHODS TO ESTIMATE PROTEIN COPY NUMBER FROM DROSOPHILA EMBRYO IMAGE DATA

Lee Zamparo

A thesis
in
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science
Concordia University
Montréal, Québec, Canada

September 2009

Your file Votre référence
ISBN: 978-0-494-63020-4
Our file Notre référence
ISBN: 978-0-494-63020-4

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Lee Zamparo**

Entitled: **Two methods to estimate protein copy number from Drosophila embryo image data**

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

_____ Chair

_____ Examiner

_____ Examiner

_____ Supervisor

Approved _____
Chair of Department or Graduate Program Director

_____ 20 _____ _____

Dr. Robin A.L. Drew, Dean
Faculty of Engineering and Computer Science

Abstract

Two methods to estimate protein copy number from Drosophila embryo image data

Lee Zamparo

Experiments using microscopy which measure gene expression data usually do so indirectly, by recording the intensity of messenger RNA or proteins tagged with fluorescent agents, to produce semi-quantitative data measured by fluorescent intensity. However, quantitative measurements of mRNA or protein concentrations are imperative for developing predictive models of gene regulation networks. In the absence of experimental procedures designed to calibrate the conversion from intensity to concentration, a statistical model of the intensity values may be used to estimate this relationship. In this thesis, two different estimators are developed to estimate the relationship between intensity and protein copy number. The methods were applied to a data set of time-lapse protein expression data taken from embryos of *Drosophila melanogaster*. Both methods assume a linear relationship between intensity and concentration. When restricted to a specific protein, the methods produce very consistent results, and are in general agreement with other methods applied to similar data. The software used to generate the estimates is implemented as a series of scripts in R. The data is all drawn from FlyEx, and is available at `http://flyex.ams.sunysb.edu/flyex/`

Acknowledgments

I would like to thank my supervisor, Gregory Butler, for his guidance and support during my time spent in his lab. I would also like to thank my fellow colleagues in Dr. Butler's lab for offering me their encouragement and advice. Also, I would like to thank Theodore Perkins of the department of computer science at McGill University for introducing me to this problem, and for his advice and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem statement

Fluorescent imaging is widespread in the analysis of mRNA and protein expression in single cells, in populations of cells, and within tissues or organisms. This type of data is used in many different types of genomic studies including:

- determining the effects of gene knockouts in organisms such as yeast, e. coli, and other bacteria and fungi.
- over or under-expression of genes when the organism under study is subjected to different environmental stresses or conditions such as heat shock, change of pH, or change of available nutrients.
- promoter manipulations, understanding regulatory networks.
- determining co-expression of genes or co-location of gene products, which may indicate complexing.
- identifying markers for tissue types or processes.
- inferring protein function.

As useful as such data is, one limitation is that it usually only gives relative expression values. Greater fluorescent intensity implies greater expression, but the actual concentrations or copy numbers of molecules being imaged are usually not known. Knowing expression in absolute terms can be important for detailed quantitative modeling ([BS03]) and for extracting biologically meaningful parameters. For example, many current fitting methods express reaction rates in fluorescence units (e.g., [TXGB07]), rather than in molecules or moles.

Absolute expression is also relevant for understanding the sources of noise or variability in gene expression ([RWA02]; [SES02]; [Swa04]; [RPA+06]; [RO05]; [BEPM+06]). This is because many studies in gene expression variability or expression noise observe isogenic populations of cells, and

try to quantify gene expression while explaining the expression variation as either intrinsic or extrinsic noise. Intrinsic noise involves processes within the cell, such as fluctuations in signal transduction or noisy ligands. Extrinsic noise are changes in the cell's transcription or translation routines based on stimuli from the cell's environment. Given that expression noise is defined in terms of the ratio of standard deviation over mean expression, accurately quantifying gene expression is crucial to understanding this variability. Absolute expression is also important for recent analyses of information processing in gene regulatory networks ([GTWB07], [LPS07], [TCB08]).

1.2 Other approaches to quantification

Measuring concentration is a difficult problem in molecular biology. One approach that has many advantages is mass spectrometry. Current mass spectrometry technology has the capability to measure the number of different masses in a biological sample as well as the intensity of each mass. Advancing technology and protocols for MS technology have improved the accuracy and reproducibility of experiments. However, several obstacles remain. Depending on the complexity of the sample under analysis, experimental runs may produce inconsistent results [LPDA08]. Measurements may be confounded due to strong sequence similarity among sample peptides, unexpected ions produced as a result of the protein digestion process [PAD07], as well as other different types of noise which remain poorly understood. Furthermore, mass spectrometry is unable to measure any spatial expression information, due to the sample preparation process.

This means that any study that depends on measuring expression with spatial information is dependent (currently) on interpreting cell images. There are imaging techniques that can result in concentration or copy number information, such as fluorescence- and image-correlation spectroscopy, fluorescence-intensity distribution analysis ([KPUG99]), and photon-counting histogram analysis ([CMRG02]). Alternatively, the intensity signal can be calibrated by measuring expression of mRNAs or proteins that are at known concentrations (e.g., [BEPM$^+$06]; [GTWB07]). However, these methods suffer from some practical drawbacks.

Gregor et al. [GTWB07] investigated the limits of the information a cell can extract from its position. They also use the segmentation process in drosophila as their model. As part of four questions relating to noise, they try to measure the absolute concentration of Bicoid (Bcd), a maternally expressed gene that regulates other genes during the segmentation process. One question they want to investigate is how reproducible are the absolute Bcd concentrations at corresponding locations in different embryos. They construct an experiment using altered embryos that contain bicoid-eGFP fusion protein, and use time series images to estimate the rate of diffusion from the anterior towards the posterior of the embryo. This method relies heavily upon the previously well characterized migratory pattern that is specific to Bcd, and may not be widely applicable.

Kask et al. developed fluorescence-intensity distribution analysis, to measure concentration of fluorescently tagged molecules in confocal microscopy studies [KPUG99]. By modeling the empirical spatial brighness distribution, and using generating functions to calculation the theoretical photon

count number distributions, they can simultaneously determine the correspondance between concentration and observed brightess values. However, the authors admit their method rests on two assumptions: that coordinates of molecules do not change significantly during a counting time interval and that the brightness of each molecule can be expressed as a product of a spatial brightness function, which is common to all molecules in the sample, and specific brightness, which has a characteristic value for each molecule. So the procedure requires sampling images at a rapid rate, as well as calibration expertise for the particular microscope.

Similar in principle to the work of Kask et al., Chen et al. developed a method which they call fluorescence fluctuation spectroscopy[CMRG02]. They apply it to images of solutions of fluorescent dyes. The fluorescent image data is used to obtain information about the number of fluorescent particles in a small volume and the diffusion coefficient from the autocorrelation function of the fluorescence signal. They measure the empirical distribution of detected photons, called the photon counting histogram, and compare it to a theoretical distribution. While the authors demonstrate the effectiveness of their method for samples containing one or two different fluorescently labeled dyes, the question of how it would scale to more complex biological samples is unanswered.

Tian et al. develop a method to relax a differential equations model into a stochastic model using poisson processes to represent the birth and death of molecules involved in biochemical reactions, in a bid to estimate kinetic parameters [TXGB07]. However, their methods needs to fit a large number of parameters, and requires both a large set of time series data, as well as a high abundance of molecules in the samples.

Currently, none of these approaches are nearly as common in practice as direct, un-calibrated measurement of fluorescent intensity. This is likely due to a number of factors, not the least of which is a considerable amount of required expertise, precise calibration of the microscope instruments, or various assumptions about the composition and behaviour of the observed samples. Therefore, a method that uses only intensity data should be both widely applicable and useful when constructing systems level models of gene product interactions. As I will show, the two methods described herein show close agreement with that of Gregor et al. for Bicoid, which is encouraging.

Motivated by a well-annotated publicly available data set and some previous work on modeling regulation of genes in the segmentation network of *Drosophila melanogaster* ([PJRG06]), I chose to focus on that particular system to test the two methods. They are both applicable to similar fluorescence data obtained from other organisms.

1.3 My work

Based on a method developed by Rosenfeld and colleagues to estimate protein concentration in growing colonies of E. coli in [RPA+06], I developed two methods for estimating copy numbers of fluorescently labeled molecules. Each method takes as input the intensity values in the three channels of a single image. Each method makes assumptions about the underlying biological state of the embryos in each image, and is designed to exploit that state, along with the underlying variability of expression.

Both methods for estimating protein copy number assume that the observed expression intensity in a nucleus for a particular channel is proportional to the copy number of the corresponding protein. If we denote the observed expression intensity as O_i, and the true concentration of the protein as N_i, then under this assumption these two quantities are related by a proportionality factor ν:

$$O_i = \nu * N_i \tag{1}$$

The proportionality factor ν may vary for different proteins, but is assumed to be constant for embryos imaged under similar conditions. Each method estimates ν by fitting a probability model to the concentration values, and then exploiting relations between the expression values of local nuclei. From these estimates of ν, we can solve equation 1 for the protein concentrations N_i. The two methods make different assumptions about the data, and so they apply in different situations.

The first method uses a binomial model for concentration. It is most accurate for embryos where the nuclei have just undergone division, and where the sibling pairs can be easily identified. The binomial method assumes that proteins present in the mother nucleus are passed to the siblings independently and with equal probability. The difference in intensities of the siblings can then be related to absolute concentration.

The second method is for embryos that are near the end of segmentation. It models protein concentration as a stochastic process, where each protein is being produced in the nucleus at a rate μ, and degraded at a rate δ. This is known as a birth-death process. This model is biologically plausible as proteins are created and degraded by different processes as needed by the cell. The method then exploits the fact that the first two moments of the poisson distribution are the same, recovering an estimate of ν. Detailed descriptions of both methods appear in chapter 3. Figure 1 displays how both of the methods are applied from loading the data to producing the estimates.

1.4 Summary of the Results

The results of the estimates are very consistent within each transcription factor, and within a factor of 4 when comparing the score for the same factors across the two methods. Results are quite consistent for the the maternal and gap genes.

1.5 Outline

This thesis is organized into five chapters. This first chapter introduced the problem of estimating concentrations from image data, some contemporary methods to do so, and summarizes the two methods developed as part of this work. The second chapter presents some background knowledge about the data set and the segmentation network of drosophila, as well as a brief review of some of the relevant molecular biology. A review of systems biology is included, to motivate this work and situate the work. The third chapter describes the both of the methods in detail, the two models used (binomial model and poisson model), as well as some justification for both models. The fourth chapter presents the results of the method on a selection of embryos, and a discussion of noise. The

fifth chapter concludes this work with a discussion of the results, some comparison with current works, and a discussion of potential improvements and impasses. Figures appearing in the first five chapters are re-printed in colour in the first appendix. Further appendices containing the code used to run the experiments appear after the bibliography.

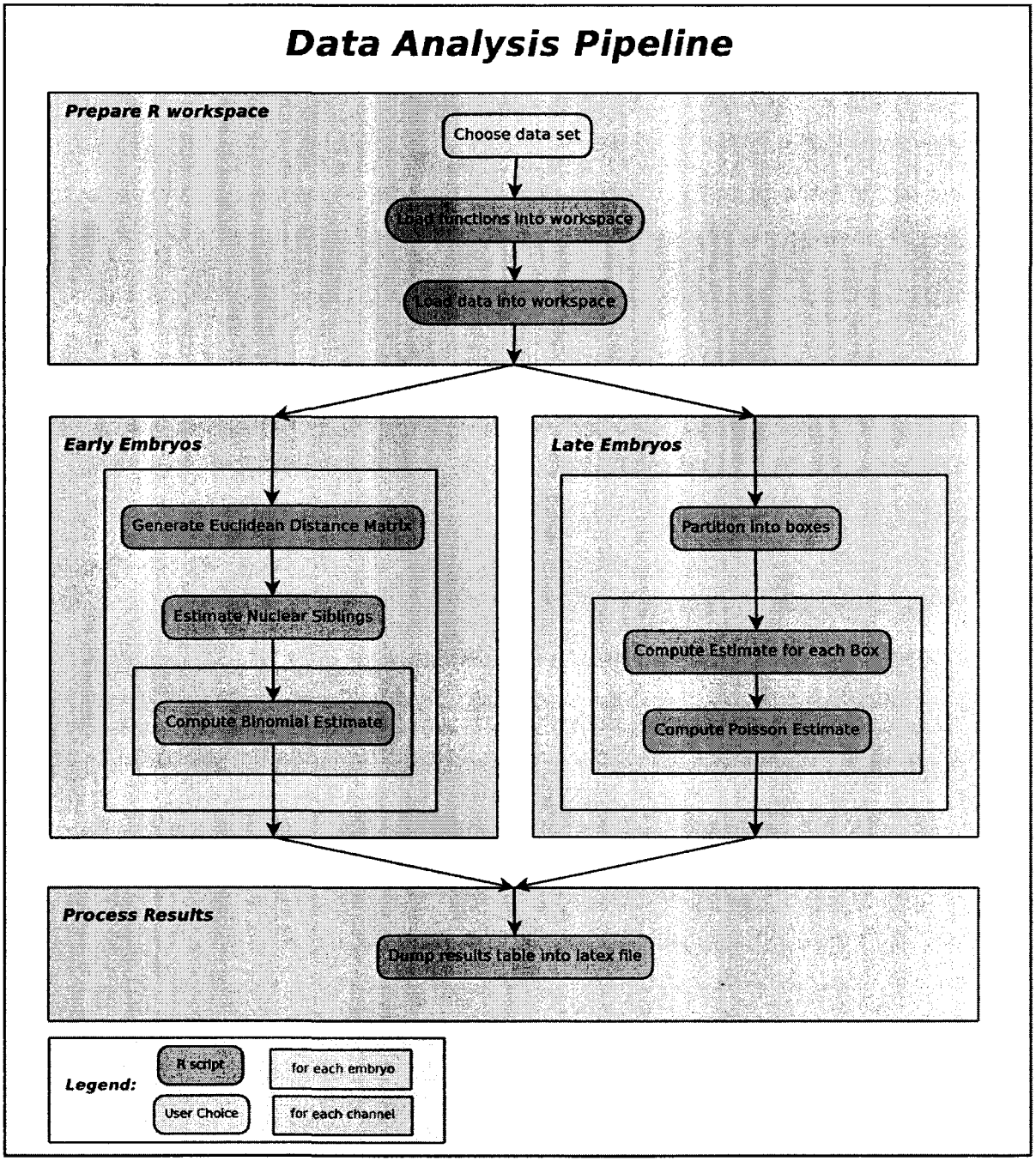

Figure 1: An activity diagram summarizing the analysis pipeline for both the poisson and binomial methods.

Chapter 2

Background and Data Set

This chapter presents some relevant background knowledge. It is split into two parts: a biology section and a statistics section. The biology section is composed of a short introduction to systems biology, a primer on the central dogma of molecular biology with a focus on transcription factors, and an overview of the *drosophila* gene network, along with the motivations that drive research in this system. The statistics section contains reviews of the two relevant probability distributions used, the binomial and Poisson, along with the relation between the two. The method of maximum likelihood estimation concludes the statistics section. Finally, the data set used in the work is presented.

2.1 Biology background

At the molecular level, different types of biological knowledge (DNA, genes, proteins) can be modeled as annotated sequences of symbols drawn from various alphabets. For DNA, the alphabet consists of the four deoxyribonucleic acids adenine, cytosine, guanine, and thymine. A, C, G, T. For RNA the set of ribonucleic acids shares A, G, C with the DNA alphabet, but thymine is replaced by uracil, forming the set A, U, G, C. This strings and alphabets representation of molecular biology allows for the conception of molecular biology as computation. This also leads to a natural collaboration with computing science, with whom biologists have developed tools to gather, store, visualize and annotate biological information. This task is at the core of systems biology, which is explored below.

2.1.1 Systems biology

Systems biology is an approach to understanding biological science at a systems level perspective ([Kit02]). Since at least the middle of the 20th century, systems biology has been proposed and discussed, but not until present day has it been even possible. Yet today, three principle factors have made systems biology possible. The first is great advances in the field of molecular biology. While molecular biology has proven to be tremendously difficult, with every discovery revealing new exceptions and levels of complexity, technology has enabled much more efficient representation and dissemination of biological knowledge, accelerating the rate of new discoveries. The second is

the development of fast and relatively inexpensive whole genome sequencing techniques, as well as technologies to simultaneously perform measurements on a genome wide scale for a wide variety of bio molecules. The third is the willingness of physicists, computer scientists and biologists undertaking new collaborative efforts, with each contributing expertise previously inaccessible to biologists alone, with the overall goal of gaining a more fine-grained knowledge of biology. These three factors make it possible for biology to be understood as a system operating at the molecular level. According to Hood et al., Three principal technological advances have made a systems approach to biology practical ([IGH01]). The first is high-throughput techniques for genetic manipulation, which are now affordable, standardized and automated. The second is the availability of complete genomic sequences, complete with annotated lists of genes and known functions, which has made systematic mutagenesis projects possible. The third is technologies for disrupting genes in trans, which allows the application of genetic perturbations to a wide range of eukaryotic organisms.

In addition to technological breakthroughs, systems biology required a change in the philosophy of biological science. Ideker et al. say that systems biology is the result of the integration of two different approaches to biology: hypothesis driven science and discovery science ([IGH01]). Hypothesis-driven science is where researchers survey what is known, create hypotheses to explain the unknown, and attempt to distinguish among them by interpreting the results of appropriate experiments. Discovery science is where researchers seek to define all of the elements in a system, and to create a database containing that information. Ideker et al. claim that the integration of these two approaches is one of the main goals of systems biology. According to Kitano, systems biology required a shift in the notion of what to look for in biological experiments ([Kit02]). While the basic units of genes and proteins remain important, experiments should aim to reveal instead the structure and dynamics of a system of genes or proteins.

Because a system is more than just a series of genes or proteins, few of its properties can be discovered by simply creating a diagram of their interactions. So while this is an important first step, Kitano uses the analogy of a roadmap. What we really want to understand is how traffic travels along roads, and why traffic patterns occur. A system level understanding of biology depends upon knowledge of the following four aspects of the system ([Kit02]):

1. System structures. Examples include gene interaction networks, pathways, and the knowledge of how interactions change the physical properties of cellular structure.

2. System dynamics. This is how the system behaves over time, under different conditions and in response to various stimuli. Examples of techniques to understand system dynamics include metabolic analysis, sensitivity analysis, or bifurcation analysis.

3. Control methods. Mechanisms that systematically control the state of the cell to deal with challenges or malfunctions. Examples include transcriptional regulation, protein folding regulation, apoptosis,

4. Design methods. These include strategies to modify existing cellular control methods to engineer certain types of cellular behaviour in a principled manner.

As previously stated, advances in the above will require advances in computational models for biology, more plentiful and reliable genomic data, cheaper and more accurate measurement technologies including microscopy, and easier integration with existing knowledge. In addition, it must be made accessible to everyone who is able to make a contribution: biologists, computer scientists, chemists. and others. These processes have already been undertaken by a wide variety of groups. Knowledge of gene regulatory logic for various organisms are stored in databases available through the interned (e.g SCPD, TRANSFAC) ([IGH01]). Similarly, biochemical pathways in various organisms are available and (e.g KEGG, EcoCyc) ([IGH01]).

2.1.2 Elementary Molecular Biology

Western science views biological systems as being part of two types of information: genes, which encode protein machine parts that combine to create life, and networks or regulating interactions, which specify the complex patterns in which genes are expressed. This information can be modeled in a hierarchy or pathway:

DNA $\Longrightarrow$ mRNA $\Longrightarrow$ protein $\Longrightarrow$ protein interactions $\Longrightarrow$ pathways $\Longrightarrow$ networks $\Longrightarrow$ cells $\Longrightarrow$ tissues

The first part of this pathway, from DNA through to proteins, is known as the central dogma of molecular biology. It posits that all the information about our cells is bound in DNA, which is transcribed to RNA, and then the RNA is translated into amino acid chains and folded into proteins. Proteins then play a central role in both the structure and function of the cell. Not included in this extensive list are any of the many small molecules and macro-molecules that are present in the cell and which also play a large role in many cellular processes.

The relevant part of the above pathway for this study begins at the DNA level and ends in protein interactions. The proteins in the segmentation network of *drosophila* bind to DNA to influence the process of transcription, and are part of a class of proteins known as transcription factors. Below is a review of the mechanisms of transcription, the process by which DNA sequences are read and reproduced as RNA, as well as how cells regulate gene expression through transcription.

Overview of Transcription

The complexity of each of the steps in the pathway from gene to protein has required that they be studied in isolation, and most of our knowledge in this area has been generated using classical biochemistry. Using this approach, the biochemist first obtains experimental conditions in which the process of interest (e.g., transcription or translation) can be reconstituted in vitro in a cell-free extract. The protein machineries involved are then purified from the protein extract, allowing the process to be recapitulated using purified proteins and allowing the role of each player to be analyzed mechanistically. While this type of approach has been very useful, it forces the scientist to take a reductionist view. Each step in the pathway (e.g., transcription, pre-mRNA processing, and translation) is studied separately, often with little thought being given to the connections between steps. Consequently, the different steps have traditionally been viewed as discrete, unconnected events. This separation of events is also most compatible with human thought process, which can

most easily visualize complex processes as a linear series of events, with each step going to completion before the next begins.

In recent years, the way in which we view gene expression has changed significantly, and decade-old observations suggesting that consecutive steps in the pathway are interdependent or are influenced by one another have taken on new meaning. A growing number of genetic studies have revealed functional links between the protein factors that carry out the different steps in the gene expression pathway. Similarly, conventional biochemical approaches and large-scale mapping of protein-protein interaction networks have uncovered physical interactions between the various machineries. In combination, these studies suggest that each stage is a subdivision of a continuous process, with each phase physically and functionally connected to the next ([OR02]). It has now been demonstrated, for example, that the transcriptional apparatus plays an active role in recruiting the machinery that caps and processes the nascent RNA transcript, and that pre-mRNA splicing promotes transcription elongation ([OR02]) and is required for efficient export of the resulting mRNA into the cytoplasm. The temporal separation of each step has also been questioned: pre-mRNA splicing and packaging of the mRNA for export occurs even as the transcript is spooling off of the transcribing RNA Polymerase II (RNAP II). The picture that is emerging is one in which most steps are physically and functionally connected, like a factory assembly line, ensuring efficient transfer from one manipulation to the next. This organization of events may also introduce a series of quality control mechanisms, as it ensures that no individual step is omitted.

The results of a large body of work have revealed at least three general principles:

- The protein factors responsible for each individual step in the pathway from gene to protein are functionally, and sometimes physically, connected.
- Regulation of the pathway is controlled at multiple stages.
- Different classes of gene are regulated at different stages.

The next few subsections introduce ideas of how DNA structure is implicated in transcription, and how transcription factors act to regulate expression.

Chromatin in Gene Expression

The DNA in our cells is not floating freely. Most of the time, it is packaged into a highly organized and compact protein structure in the nucleus known as chromatin. The basic organizational unit of chromatin is the nucleosome, which consists of 146 base pairs of DNA wrapped almost twice around a protein core containing two copies each of four histone proteins: H2A, H2B, H3, and H4. These small, positively charged proteins are the building blocks of our chromosomes. In terms of evolution, they are highly conserved among eukaryotes. Further compaction of our genes is achieved by nucleosome folding, a process which remains poorly understood.

Once thought of as being a static organizational framework for DNA, it is now apparent that chromatin plays a pivotal role in regulating gene transcription by providing access of the cellular machinery responsible for transcription to genes ([OR02]). Chromatin seems to come in one of two

classes. Untranscribed regions of the genome are packaged into highly condensed heterochromatin. Transcribed genes appear in euchromatin, which is less condensed and therefore more available. Different types of chromatin help explain how it is that each cell in our body has the same DNA, yet has very different structure and function. It is now known that each cell type packages its genes into a unique pattern of heterochromatin and euchromatin, and that this pattern is maintained after cell division. The pattern of packaging into these alternative chromatin states determines which genes are available to be transcribed in newly divided cells, thus allowing tissue specific cells to divide without change of function. Therefore, to initiate gene expression, transcriptional activator proteins must find some way to access DNA that may be tightly packaged in chromatin, and presently inaccessible. As is explained below, they do this by working with other agents in the nucleus to set in motion events which result in a change in chromatin conformation, leading to increased DNA accessibility.

How do Transcription Factors Regulate Each Other

Higher eukaryotes have developed sophisticated mechanisms for controlling the rate of gene transcription. Many of these mechanisms can be seen as sequences of reactions acting to produce a change in the biochemical state of the cell, or cell compartment. These are referred to as signal transduction cascades. The result of many signal transduction cascades is the activation of transcriptional regulator proteins that bind to short sequence motifs found in the promoter and enhancer regions of genes. These regions appear on the DNA, usually upstream of the transcription start site of a gene. The regulatory sequences of most eukaryotic genes contain binding sites for multiple transcription factors, allowing each gene to respond to multiple signaling pathways and facilitating the fine-tuning of transcript levels. The activities of many transcription factors depend upon many factors including the presence and concentration of various other proteins, and can be modulated by other regulators bound nearby. These complex relationships allow transcription factors to take on multiple roles: a single activated transcription factor can induce transcription of one gene while repressing that of another. This combinatorial and context-dependent regulation of transcription allows metazoan cells to respond to many different stimuli using the same factors, but in different combinations. This is in contrast to transcriptional control in prokaryotes, where metabolically related genes are coregulated in common transcription units called operons by a single transcriptional activator or repressor.

Considering the diversity of physiological signals that regulate gene expression, it is not surprising that the activities of transcription regulators are subject to multiple modes of regulation. A common theme in their regulation is the transport of a protein between the nuclear and cytoplasmic cell compartments. This occurs through nuclear pores, which are specialized gateways that span the nuclear membrane and control the passage of macromolecules through the membrane. A family of transport factors that recognize short amino acid motifs found in proteins mediates the movement of proteins through these pores. Two types of transport motif exist: nuclear export signals (NES) are found in proteins transported from the nucleus, while nuclear localization signals (NLS) label a protein for nuclear import. Transcription factors, which act on DNA that is found in the nucleus, possess NLS labels.

To respond to changes in stimulus, a cell must be able to change the state of a transcriptional activator as quickly as it is induced. A number of recent reports have linked the ubiquitin protease system to both transcription factor activation and degradation ([OR02]). Ubiquitin is a small, highly conserved protein that modifies proteins by forming a covalent bond with its targets. This bond may change various biochemical properties of the target protein, affecting it's function or localization within the cell. It may also mark the target protein for degradation. The tight coupling of transcription factor activation and degradation seems to act as a control against prolonged transcription factor activity. Thus, ubiquitination can be though of as a activity license for transcription factors by linking their activity to their inactivation. The rapid turnover of promoter bound activated transcription factors resets the signaling pathway and allows the cell to continuously monitor its environment. If signaling is prolonged, a newly activated protein will replace the degraded transcription factor. In the absence of signal, the degraded factor is not replaced and transcription ceases. The role of the ubiquitin ligase system may extend to other phases of transcription. For instance, it is known that the largest subunit of RNAP II is ubiquitinated during transcription in vitro ([OR02]). Taken together, this suggests that the transcriptional process is closely linked to the cellular processes that degrade proteins, allowing the rapid termination of transcription at multiple stages in response to various cellular signals. This allows the cell to respond to rapid changes in its environment, and to remain viable in the face of different challenges such as infection, changes in temperature, changes in pH, and others.

In addition to protein ubiquitination, a number of other post-translational modifications play important roles in regulating transcription factor activity. The next most common modification is protein phosphorylation, and is carried out by a family of proteins known as kinases. These proteins transfer phosphate groups from a donor protein to a substrate protein. Often the donors are molecules such as ATP. As with ubiquitination, this transfer alters the biochemical properties of the accepting protein, and induces a change in either cellular localization or function. In addition, transcription factors are subject to many other modifications, including acetylation on lysine residues and methylation on arginine and lysine amino acid residues. See [ZR01] for more details. Many of the enzymes that catalyze these modifications have been identified only recently. The p53 tumor suppressor protein, which responds to stress signals and coordinates a wide variety of cellular processes, was among the first transcription factors shown to be acetylated with functional consequences.

How Transcription Factors Regulate Gene Expression

Transcriptional activator proteins must bind to and decompact repressive chromatin structures to induce transcription. The way in which they do this is gradually becoming clear. To elicit their effects on gene expression, activators require the cooperation of a diverse family of coregulator proteins ([MO02]). The function of these ancillary proteins was obscure until it was found that many were subunits of protein complexes that alter chromatin structure, or were themselves chromatin-modifying enzymes. Thus, the recruitment of coactivators by DNA bound transcription factors leads to local chromatin decompaction and allows access of RNAP II and the general transcription

machinery to the promoter.

2.1.3 The segmentation network in *drosophila*

The segmentation network in drosophila is a system of genes that are most active during embryogenesis. During the process, cell membranes have not yet been formed to strictly divide the nuclei[1].

Depending on the phenotype induced by the expression of the various genes active during segmentation, they may be classified into one of four groups:

- gap genes are those in which mutations cause multiple adjacent segments to be missing from the embryo.
- pair-rule genes are those in which mutations cause multiple alternate segment size units to be missing from the embryo.
- segment polarity genes are those in which mutations cause deletions in part of each segment
- maternal genes are genes that are not transcribed in the embryo, but whose transcripts are donated by the mother via the oocyte[2].

Segmentation is governed by a program of sequential gene expression. It begins by three maternal gene regulatory proteins - including Bicoid, Hunchback and Caudal - which specify an initial 'pre-segmentation' pattern along the anterior-posterior axis, while the anterior and posterior ends of the body are specified independently by the localized activation of the maternal receptor tyrosine kinase Torso. The principal target genes of these maternal factors in the embryo's genome are known as gap genes, as their lack leads to gaps in the body pattern. The gap genes, such as Kruppel, Knirps and Giant, encode sequence-specific transcriptional repressors ([Lev08]).

The whole process of drosophila embryogenesis takes place in less than one day. The blastoderm stage, during which the segmentation and homeotic genes become active, is reached in just the first few hours. During this early period, expression patterns change rapidly. Some early research proposed that the interaction between these genes could be characterized as a cascade ([SC87]). The first group regulates the expression the following group, and then the following group regulates the expression of the next group. However, more recent work shows that the process is more complex. There is evidence of auto-regulation of genes, as well as regulatory interactions between non-adjacent groups in the cascade ([SPF+04]).

The interplay of the maternal factors and the gap repressors constitutes one of the leading paradigms for the combinatorial control of gene expression in development ([Lev08]). These regulatory factors bind to the enhancers of the segmentation genes to produce precisely positioned on/off repeating transverse stripes of expression for each gene, foreshadowing the subdivision of the embryo into a repeating series of body segments. The segmentation genes typically have highly complex enhancers, with multiple binding sites for each gene regulatory protein. Each enhancer contains a

[1] In this stage, the embryo is referred to as a syncytium: a single cell with multiple nuclei, which divide regularly and roughly at the same time

[2] An oocyte is a female germ cell involved in reproduction. In other words, it is an immature ovum, or egg cell.

specific constellation of binding sites for maternal and gap proteins, and within each nucleus there is a particular combination of transcriptional activators and repressors that can bind to these sites.

This process is of general interest because it is perhaps the best characterized example of a *morphogenetic field* ([SKK+08]). The morphogenetic field is a fundamental object in developmental biology. It was shown in the late 19th century that groups of cells underwent collective determination events (morphallaxis) in which cell fate was stably assigned to individual cells with exquisite spatial precision.

Success in elucidating genomes, proteomes, and so on suggests the importance of understanding the *morphome*, by which we mean the complete set of determinants of a morphogenetic field. In general, the morphome will consist of a description of the quantities of morphogenetic determinants at a resolution in space and time sufficient to uniquely determine the biological trajectory of the system. Because of the central role of cells and their genomes, the information about the morphome must be of at least cellular resolution in space, must include the expression levels of all the genes encoding cell fate determinants, and must be of a time resolution shorter than the time in which significant changes in the levels of these determinants can take place.

During segmentation, the embryo is syncytial and only a very limited number of zygotic genes are expressed. Only 14 of these genes act in the blastoderm as determinants of the segmentation morphogenetic field ([SKK+08]). All of these genes code for transcription factors. Together with the syncytial nature of the blastoderm suggests that cell-cell communication by means of signaling pathways does not occur in the segmentation morphogenetic field, but rather that spatial interactions occur through diffusion of these transcription factors. Mechanical forces and cell migration appear to be uncoupled from the segment determination process as well, since mutations in segmentation genes do not affect morphology until after gastrulation ([SKK+08]).

Therefore, estimating the spatial concentration of these gene products during the segmentation process is a prerequisite to constructing a working model of the morpheome, and to a quantitative systems biology understanding of fundamental developmental processes in animals. The data from Reinitz et al. cited in this thesis and in Surkova et al. ([SKK+08], [KMP+02]) claims to be quantitative, but in fact only the fluorescent intensity is reported. The correspondance between this intensity and the concentration of the associated transcription factor is unknown.

Given that this system has been actively studied for over twenty years, much of the biological knowledge is beyond the scope of this thesis, and is not directly relevant to the central aspect of estimating protein concentration. For an in-depth review of the system from a biology perspective, see the review by Scott et al. ([SC87]). Some more recent studies are collected in the supplementary mateials in Surkova et al. ([SKK+08]).

2.2 Statistics background

This section contains a brief summary of the statistical background needed to discuss the estimators. The binomial and Poisson distributions are introduced as well as some of their more important properties. The connection between them, which is important for understanding the development

of the Poisson estimator, is presented. Finally, the concept of maximum likelihood estimation is presented since all the parameters in the model are estimated in this way.

2.2.1 The binomial distribution

A binomial distribution possesses the following properties:

1. It is a sum of a finite number, say n, of independent Bernoulli(p) random variables. Each Bernoulli random variable is sometimes called a trial.

2. Each trial results in one of two outcomes.

3. The probability of success for one trial is p, and is the same in each trial. Correspondingly, the probability of failure is always $q = 1 - p$.

The probability density function is

$$P(Y = k) = \binom{n}{k} p^k q^{n-k}$$

The expected value μ is $n * p$. This can be verified by the properties of expectation: since the binomial is a sum of n independent Bernoulli(p) random variables, $E(Y) = E(X_1 + ... + X_n) = E(X_1) + E(X_2) + ... + E(X_n) = n * E(X_1) = n * p$.

2.2.2 The Poisson distribution

The Poisson distribution is a discrete distribution that is designed to model the number of relatively rare events in a certain dimension (space, time, or volume). This can be represented by a number N. For the purposes of a formal definition, the span of the dimension is subdivided into small sized intervals. For a given subinteraval of the dimension, the Poisson distribution possesses the following properties:

1. $P(N = 0) = 1 - p$ for a given p

2. $P(N = 1) = p$

3. $P(N > 1) = 0$

The subintervals are considered to be independent, and so the distribution over the whole interval is a sum of Bernoulli(p) random variables, which is binomially distributed. However, usually the parameters n, the number of sub-intervals and p, the probability of observing an event in one sub-interval are unknown. So the Poisson distribution is usually paramaterized and described by the parameter $\lambda = np$. Taking the limit as $n \to \infty$, we get the probability mass funciton as $P(Y = y) = \frac{\lambda^y}{y!} \exp(-\lambda)$.

The expected value of a Poisson distributed random variable is λ. Interestingly enough, the variance of the Poisson distribution is also λ. This property is exploited when fitting the Poisson expression model.

The Poisson as the limiting distribution of the binomial

Let Y be a random variable that is binomially distributed with parameters (n, p). If we write $\lambda = np$. and let n get large, the probability mass function of Y takes on the form of a Poisson distribution:

$$\lim_{n\to\infty} P(Y) = \lim_{n\to\infty} \frac{n(n-1)\dots(n-y+1)}{y!} \left(\frac{\lambda}{n}\right)^y \left(1-\frac{\lambda}{n}\right)^{n-y} \tag{2}$$

$$= \lim_{n\to\infty} \frac{\lambda^y}{y!} \left(1-\frac{\lambda}{n}\right)^n \frac{n(n-1)\dots(n-y+1)}{n^y} \left(1-\frac{\lambda}{n}\right)^{-y} \tag{3}$$

$$= \frac{\lambda^y}{y!} \lim_{n\to\infty} \left(1-\frac{\lambda}{n}\right)^n \left(1-\frac{\lambda}{n}\right)^{-y} \left(1-\frac{1}{n}\right) \times \left(1-\frac{2}{n}\right) \times \cdots \times \left(1-\frac{y-1}{n}\right) \tag{4}$$

After applying the identity

$$\lim_{n\to\infty} \left(1-\frac{\lambda}{n}\right)^n = \exp(-\lambda)$$

and noting that all the remaining terms to the right of the limit converge to 1, we see that

$$P(Y=k) = \frac{\lambda^k}{k!} \exp(-\lambda)$$

which is the probability mass function of a Poisson random variable with parameter λ.

2.2.3 Maximum likelihood estimation

The method of maximum likelihood estimation is a standard method in statistical inference. Central in this theory is a model for the data and the data itself. These two are combined in a clever way to give rise to a function of the model parameters, which is called the likelihood function. The values that maximize this function are deemed to be the maximum likelihood estimates for the model parameters.

Likelihood function

Suppose that we have a statistical model where each P_θ is discrete, with it's probability mass function denoted $f_\theta(\cdot)$. That is, each outcome o of the process P_θ has $f_\theta(o)$ chance of occurring. Also suppose that we have collected a set of data from the process that P_θ describes, call it $D = (o_1, o_2, \dots, o_n)$. If these observations are independent, then their joint probability is given by the function $L(\theta \mid D)$, which is defined on the possible values of the parameter(s) θ, and $L(\theta \mid D) = \prod_{i=1..n} f_\theta(o_i)$. This joint probability is a function of the parameter values of the model P, and is called the likelihood function. The value of $L(\theta)$ is called the likelihood of θ. Remember that the value of θ varies, but the data D are fixed.

The likelihood function allows us to compare different values of θ by quantifying how likely we were to have observed D if the true value of the parameter were in fact θ_1, or θ_2, or any other possible value, because $L(\theta \mid D)$ is just the probability of observing D when the parameters of process P has the value θ. Thus, when $L(\theta_1 \mid D) > L(\theta_2 \mid D)$, θ_1 is considered to be *more likely* to be the true value of θ than θ_2.

Finding Maximum likelihood estimates

Once familiar with the idea of likelihood, and with a properly formed likelihood function, the idea of obtaining an estimate using it makes intuitive sense. As before, assume we have a likelyhood function $L(\theta \mid D)$. If we want to obtain a point estimate of θ, then a value $\hat{\theta}(D)$ that maximizes $L(\theta \mid D)$ makes sense, as it is this value that best explains observing the data D:

$$L\left(\hat{\theta}(D) \mid D\right) \geq L\left(\theta \mid D\right) \tag{5}$$

the value(s) $\hat{\theta}$ that satisfy this equation are called *maximum likelyhood estimates* for the parameter θ.

2.3 Data Set

The data used was a set of images of fruit fly embryos that were undergoing a process called segmentation. The segmentation process is briefly described below, and the aims of the experiments are then summarized.

The determination of the semgent pattern takes place during embryogenesis in *Drosophila.* At this stage of development, the embryo is a syncytiuma single cell with multiple nuclei that divide periodically and roughly simultaneously. The data was collected from experiments performed by Reinitz and Samsonova, and is available online at FlyEx ([PPB+04]). Each image captures a *Drosophila melanogaster* embryo. The body of the fruit fly *Drosophila melanogaster* is made up of repeated units called segments. Before the segments morphologically differentiate, their pattern is marked out chemically during a process called determination, or segment determination ([MSK+01]). Post fertilization, the zygotic nucleus undergoes many rapid nucleic divisions. After the 8th division, the nuclei migrate to the outside of the embryo, which marks the beginning of the segmentation process. Each subsequent nuclear division is referred to as a *cleavage cycle.* At this stage of development, the embryo is a hollow shell of nuclei which are not yet separated by cell membranes. The embryo will undergo six more cleavage cycles (9 through 14A) before cell membranes emerge to separate the nuclei, which marks the end of segmentation. Cleavage cycle 14 is much longer than the previous cycles, and is further subdivided into eight time classes. Each embryo in cleavage cycle 14 is assigned to a time class based on expert human observation of the characteristic expression pattern of the *even-skipped* gene.

The aim of this series of experiments was to document the spatial expression patterns of a 14 genes in a network which controls segment determination, by observing and capturing a large number of images of gene expression *in situ.* In the experiments, protein expression was measured using fluorescence tagged antibodies ([JKVA+05]). Each gene product was detected in a single channel on a confocal microscope, and for each channel two raw images were made corresponding to two optical sections fo an embryo separated by two microns. The gain on the microscope was calibrated so that each pixel in the images varies in intensity from 0 to 255 on an 8-bit scale. The images were averaged, cropped and rotated to output an embryo image that displays the expression pattern of a single gene in a given embryo. Each embryo was scanned for the expression of three

genes at a time. Unfortunately, the sample preparation process kills the embryo, so each embryo is only observed once. Three embryo images were combined and the resulting image is segmented to make a nuclear mask. This mask is used to determine the average coordinates of the nucleus along the AP and DV axes, and then to estimate the average fluorescence level of each of three gene products within the nucleus. At the centroid of each nucleus, the fluorescence data in three channels is measured several times, and a mean estimate for each channel is recorded. Each of three channels in the image records the intensity of a different tagged protein. Taken together, this procedure produces a table of nuclear records. Each nuclear record consists of a nuclear identification number, the x, y coordinates of its centroid, and the average fluorescence levels estimated for each of the three proteins. See table 1 for an example. There are currently 1355 embryos available in the database. Of these, 1100 are from cleavage cycle 14A (including all time classes), 135 are from cycles 11 to 13, 170 are mutants (which were excluded from this study ([PPB+04])).

Table 1: A sample of the quantitative data table available for download. The three channels in this embryo record the average estimated expression of *even-skipped*,*kruppel*, and *bicoid* respectively.

Nucleus	AP	DV	Eve	Kr	Bcd
0	2.35	52.21	18.11	10.72	149.08
1	3.21	37.92	13.72	8.93	99.83
2	4.52	56.28	27.59	14.49	163.20
3	5.24	49.02	29.90	16.84	182.01
4	6.31	62.60	25.93	13.04	135.61

The embryo is oriented so that the x-axis corresponds to the anterior-posterior axis of the embryo, and the y-axis to the dorsal-ventral axis. The x, y coordinates are scaled so they represent the percent of the maximum size of the embryo in the x and y directions. Figure 3 shows one such embryo. Each embryo was stained for the gap gene *even-skipped*, as it is expressed throughout the segmentation process, as well as two other genes. The embryos used in this study were stained for *bicoid, even-skipped* and *caudal*.

As mentioned in this section, each embryo is observed only once, as the process of confocal scanning destroys the embryo.

2.3.1 Data set types

The Flyex website offers different types of quantitative data for individual embryos. Each type of quantitative data is formatted as shown in table 1, but they differ in that some types have undergone filtering techniques to try and improve the accuracy of both the positions of the nuclei, and the intensity levels in the nuclei. Each of the different data sets in FlyEx are summarized below.

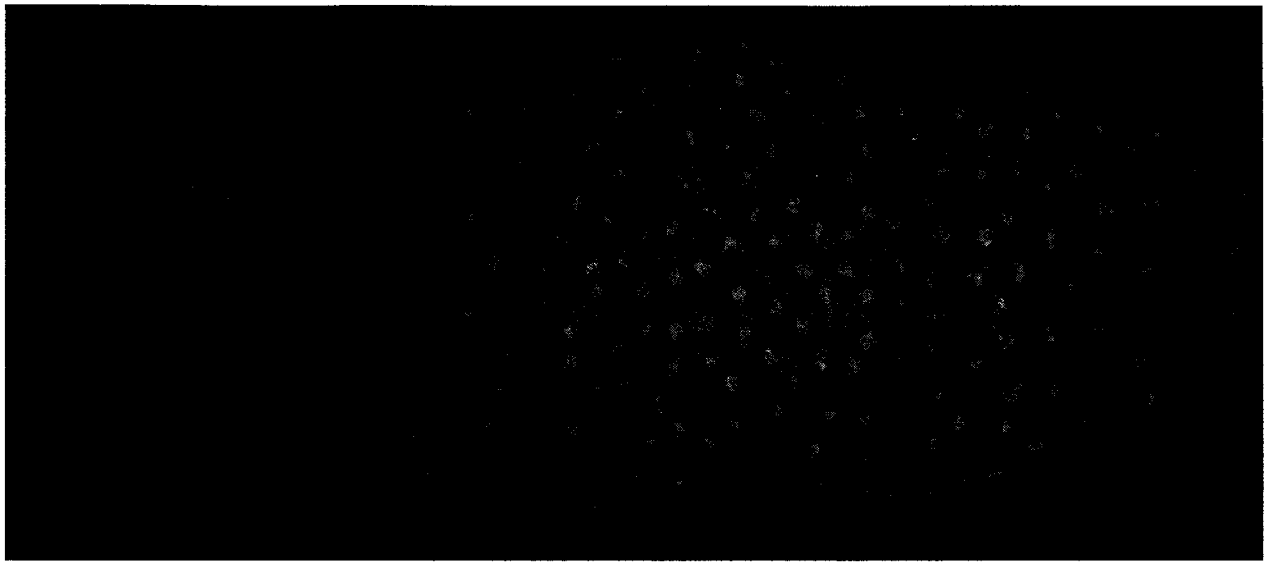

Figure 2: A sample of the images from the data set, embryo ad24 from cleavage cycle 11.

Quantitative data

This data set is the basis of all the others. Embryos ranging from cleavage cycles 10 through 14A are available. The quantitative data embryos are by far the most abundant in the database. While it is the 'noisiest' of all the data sets, tests for noise performed by myself did not reveal significant additive or multiplicative noise (see chapter 4 for details).

The criteria for choosing embryos for the binomial data set were the following:

- The embryo had an image in the database, not merely a data table.
- The embryo was stained for *Eve,Cad,Bcd*, which is the most common trio of proteins in the database. In the exploratory stages of the project, only embryos from this subset were considered to minimize the potential for variation due to the affinity of antibody-protein staining.
- The embryo displayed a large subset of easily identifyable sibling nuclei. This was required both for the construction of the gold standard data set of hand-paired sibling nuclei. Also, it greatly affects the quality of the greedy pairing algorithm (see below).

The embryos used in the binomial data set appear in table 2.

For the Poisson estimator, the following criteria were employed when selecting embryos for the data set:

- The embryo had an image in the database, not merely a data table.
- The embryos were from cleavage cycle 14, and ideally time class 8.

The embryos used in the Poisson data set appear in table 3.

Table 2: Embryos used for binomial data set. The stage column refers to cleavage cycle, while the channels column lists the abbreviations of the three proteins stained in that embryo.

embryo	stage	channels
ab18	11	Eve, Cad, Bcd
ad14	11	Eve, Cad, Bcd
ad33	11	Eve, Cad, Bcd
ac22	11	Eve, Cad, Bcd
ad22	11	Eve, Cad, Bcd
ad23	11	Eve, Cad, Bcd
ad24	11	Eve, Cad, Bcd
cb16	11	Eve, Cad, Bcd
iz3	12	Eve, Kr, Bcd
ms21	12	Eve, Kr, Bcd
ms9	12	Eve, Kr, Bcd
ms3	12	Eve, Kr, Bcd
hz12	12	Eve, Hun, Bcd
FESce05	11	Fsh, Eve, Slp
HETae10	11	Hun, Eve, Tls

Data without background

This data set consists of a subset of the quantitative data set for which a background subtraction algorithm has been applied. For each embryo, the non-specific binding distribution, or background signal, is approximated by a 2D parabola. The background parabola was fitted based on expression in areas of wild-type embryos in which a given gene is not expressed, and used to remove background from the entire embryo. The details are reported in ([MSKR05]), and are discussed later in this section. Not all embroys in the database have undergone background subtraction, and so some parts of the data set used for my work were not available in this data set. So I chose not to use it. Furthermore, in some cases the subtraction of the estimated background expression resulted in a measured expression of zero at some nuclei. This introduces numerical instability into my estimators, and in some cases the quantity being estimated is not well defined. See the methods section for details.

Embryos used for binomial data set for the data without background appear in table 4

Embryos used for binomial data set for the data without background appear in table 5

For the data without background, both the binomial data set and Poisson data set were chosen to try and match the corresponding embryos in the quantitative data sets. A sample from one of the data files appears in table 6.

Table 3: Embryos used for Poisson data set. The stage column refers to cleavage cycle and the time class if applicable, while the channels column lists the abbreviations of the three proteins stained in that embryo.

embryo	stage (time class)	channels
ab11	14 (1)	Eve, Cad, Bcd
dq2	14 (7)	Eve, Cad, Bcd
tu7	14 (7)	Cad, Eve, Bcd
bd5	14 (8)	Cad, Eve, Bcd
cb15	14 (1)	Eve, Cad, Bcd
ac10	13	Eve, Cad, Bcd
ms14	14 (7)	Eve, Kr, Bcd
ms36	14 (7)	Eve, Kr, Bcd
dm14	14 (8)	Eve, Kn, Hun
tn2	14 (8)	Eve, Kn, Hun
hne8	14 (8)	Eve, Kn, Hun
fq4	14 (8)	Eve, Kn, Hun
kf9	14 (8)	Eve, Kr, Hun
ba3	14 (8)	Eve, Kr, Hun
rf11	14 (8)	Eve, Kr, Hun
rf6	14 (8)	Eve, Kr, Hun

The formula used to perform background subtraction:

$$intensity_{norm} = 255 * \frac{(intensity - (a_0 * x^2 + a_1 * y^2 + a_2 * x + a_3 * y + a_4 + a_5 * x * y))}{(255 - (a_0 * x^2 + a_1 * y^2 + a_2 * x + a_3 * y + a_4 + a_5 * x * y))} \quad (6)$$

An example of the coefficients used in the normalization formula is in table 7.

Registered Data

This data set consists of embryos that have undergone registration by one of two methods: spline approximation, or a wavelet procedure outlined in ([KMP+02]). The data is also available after background subtraction has taken place. Registration means a transformation in coordinates of an image. The confocal microscope used to capture these images can only reveal the expression of up to three genes in any one image. The registration procedure transformed the coordinates of the different embryo images so that they all adhered to the same scale, and could then be overlaid onto each other. By overlaying enough images, the three gene limitation of the instrument can be overcome, and all 14 genes in the network can be visualized in one image. Since this was not of interest for our project, we chose not to use registered data.

Table 4: Embryos used for binomial data set, without background. The stage column refers to cleavage cycle, while the channels column lists the abbreviations of the three proteins stained in that embryo.

embryo	stage	channels
ab18	11	Eve, Cad, Bcd
ad14	11	Eve, Cad, Bcd
ad33	11	Eve, Cad, Bcd
ac22	11	Eve, Cad, Bcd
ad22	11	Eve, Cad, Bcd
ad23	11	Eve, Cad, Bcd
ad24	11	Eve, Cad, Bcd
cb16	11	Eve, Cad, Bcd
iz3	12	Eve, Kr, Bcd
ms21	12	Eve, Kr, Bcd
ms9	12	Eve, Kr, Bcd
ms3	12	Eve, Kr, Bcd
hz12	12	Eve, Hun, Bcd

Table 5: Embryos used for Poisson data set. The stage column refers to cleavage cycle and the time class if applicable, while the channels column lists the abbreviations of the three proteins stained in that embryo.

embryo	stage (time class)	channels
ab11	14 (1)	Eve, Cad, Bcd
dq2	14 (7)	Eve, Cad, Bcd
tu7	14 (7)	Cad, Eve, Bcd
bd5	14 (8)	Cad, Eve, Bcd
cb15	14 (1)	Eve, Cad, Bcd
ac10	13	Eve, Cad, Bcd
ms14	14 (7)	Eve, Kr, Bcd
ms36	14 (7)	Eve, Kr, Bcd
dm14	14 (8)	Eve, Kn, Hun
tn2	14 (8)	Eve, Kn, Hun
hne8	14 (8)	Eve, Kn, Hun
fq4	14 (8)	Eve, Kn, Hun
kf9	14 (8)	Eve, Kr, Hun
ba3	14 (8)	Eve, Kr, Hun
rf11	14 (8)	Eve, Kr, Hun
rf6	14 (8)	Eve, Kr, Hun

Table 6: A sample from the background corrected data for embryo ab18. In the file header, the coefficients for the paraboloid are listed, as is the normalization formula.

0	3.34	41.98	0.00	1.92	119.49
1	5.77	52.07	5.49	5.17	166.44
2	5.34	41.38	0.31	0.00	124.84
3	8.49	62.06	8.93	5.33	153.42

Table 7: The normalization coefficients for each of the proteins in the embryo ab18, used to remove the background noise.

gene	a0	a1	a2	a3	a4	a5
eve	−0.016079	−0.016327	1.659246	1.690695	−11.644875	−0.003831
cad	−0.032261	−0.018883	3.419556	1.828046	−46.015909	$-8.09E-4$
bcd	−0.003283	−0.003577	0.329193	0.277711	0.022621	$3.62E-4$

Chapter 3

Methods

This chapter describes the two methods for estimating protein copy number. Both methods assume that the observed fluorescent intensity of a nucleus in a particular channel is proportional to the copy number of the corresponding protein in that nucleus. The proportionality factor, ν, may vary for different proteins, but is assumed to be consistent across different embryos imaged in the same manner. Both methods work by estimating the factor ν , from which protein copy numbers in each nucleus can be derived based on their intensity. Each makes different assumptions about the types of images to which they are applied, and so the two methods do not compete against each other. Rather, they can each be applied depending on the type of data available, which hopefully will provide flexibility.

Both methods employ maximum likelihood inference applied to a statistical model to estimate ν. They each assume a probabilistic model to explain how the intensity distributions are generated by the data. Each of them attribute all variability in protein expression to intrinsic stochastic chemical processes. In a more realistic model, the data include other sources of variability, or noise. However, for various reasons I chose not to model these other sources of noise explicitly, which are discussed in chapters 4 and 5. A consequence of this is that the true variability in protein expression tends to be overestimated by the methods, which results in copy number estimates that are biased low. In chapter 4, I analytically explore the effects that different types of noise would be present in the intensity data, based on our models, and compare with empirical estimates.

3.1 Binomial Estimator

Since the proteins of interest in this system are transcription factors, they will be concentrated primarily in the nucleus. As discussed in sections 2.1.2 and 2.1.2 many of these transcription factors bind the DNA to play their role in the transcriptional program of segmentation. These locations cannot be known based on this experiment, and so may be though of as "random". Before nuclear division, each protein can thus be thought of as equally likely to be on the DNA copy that goes to one daughter as it is to be on the DNA copy that goes to the other daughter. Proteins that are not bound to the DNA, assuming they are uniformly distributed throughout the nucleoplasm, are

Figure 3: An embryo that has just undergone a recent nuclear division. The developmental program in drosophila makes the nuclei divide roughly simultaneously. The recent division makes the task of estimating sibling pairs much easier.

also equally likely to end up in either daughter nucleus. This is because each daughter receives close to half of the nucleoplasm of the mother. This reasoning motivates the use of an even binomial distribution.

The binomial estimation method applies to embryos in which nuclei have recently undergone division, and sibling pairs can be easily identified (figure 3).

The method assumes that proteins in the mother of each sibling pair pass independently and with equal probability to each daughter nucleus. The difference in intensities of the siblings can then be related to absolute concentration.

More formally, let i_1 and i_2 denote the members of a sibling pair, and i denote their (unobserved) mother nucleus. Let N_i be the unknown number of fluorescent molecules in the mother, and N_{i1} and N_{i2} be the unknown numbers of molecules in the daughters. Assume that N_{i1} and N_{i2} are both distributed as binomial(N_i, $p = 1/2$). N_{i1} and N_{i2} are dependent binomial random variables, related by the unobserved random variable N_i. Assuming no protein is lost in the division process, then $N_{i1} + N_{i2} = N_i$ is a good estimator for N_i.

The model assumes that the observed fluorescence is proportional to the number of molecules present: $O_{ij} = \nu N_{ij}$ for nucleus i and transcription factor j. If I can estimate ν , then I can estimate protein copy number in each nucleus simply by solving the previous equation for N_{ij}. Each sibling pair provides us with one estimate for ν as:

$$\hat{\nu}_i = \frac{(O_{i1} - O_{i2})^2}{O_{i1} + O_{i2}} \tag{7}$$

This is very similar to the estimator used by Rosenfeld and colleagues ([RPA+06]), who applied it to sibling data from fluorescent image sequences of growing colonies of E. coli. The main difference is that Rosenfeld et al. were working with time series data, so in addition to observed intensity values

for sibling cells, they also observed intensity measurements for the mother cells prior to division. This allows them to use observed values for the mother nucleus directly, instead of estimating it as I do in the denominator above. One mathematical rationale behind the estimator is that its expected value, treating N_i as given and N_{i1}, N_{i2}, O_{i1} and O_{i2} as dependent random variables, is:

$$E(\hat{\nu}_i) = E\left[\frac{(O_{i1} - O_{i2})^2}{O_{i1} + O_{i2}}\right] \tag{8}$$

$$= E\left[\frac{(\nu N_{i1} - \nu N_{i2})^2}{\nu N_{i1} + \nu N_{i2}}\right] \tag{9}$$

$$= \frac{\nu^2 E\left[(N_{i1} - N_{i2})^2\right]}{\nu N_i} \tag{10}$$

$$= \nu \frac{E(2N_{i1} - N_i)^2}{N_i} \tag{11}$$

$$= \frac{\nu 4 Var(N_{i1})}{N_i} \tag{12}$$

$$= \nu \tag{13}$$

The above derivation makes use of the substitutions $N_i - E[N_{i1}] = E[N_{i2}]$ and $E[N_{i1}^2] = Var(N_{i1}) - (E[N_{i1}])^2$.

Once the sibling pairs have been estimated for each of the nuclei in the given embryo (see 3.1.1), the method is applied to each pair, to produce an estimate of ν in each of the three channels. These per-channel estimates from each sibling pair are then combined into an overall channel (or transcription factor) specific estimate for ν based on simple averaging:

$$\hat{\nu} = \frac{1}{M} \sum_{i=1}^{M} \hat{\nu}_i \tag{14}$$

where M is the number of paired nuclei in a given image. As Rosenfeld and colleagues demonstrate, this estimator for ν maximizes the likelihood of observing the O_{ij} values if the binomial distributions are approximated by Gaussian distributions having the same means and variances [RPA+06].

3.1.1 Estimating the siblings

The task of estimating the sibling pairs is a difficult one. Since the embryos are only ever imaged once, there is no prior positional information to help identify nuclear lineage, as was the case with Rosenfeld and colleagues[RPA+06]. The problem can be stated as follows: given a set of nuclei, each with a pair of coordinates that identify their position in the embryo, assign matches to these nuclei so that as many of the true sibling nuclei as possible are paired. Even after inspecting over 400 images in the data set, only a handful show a majority of clearly identifiable sibling pairs, and so even human experts have a difficult time pairing nuclei. I chose to use a heuristic that nuclei which are in close proximity to each other are more likely to be related. Since the data lies in a plane (the embryos are flattened during the preparation process), Euclidean distance was used as a metric. I employed a greedy algorithm to determine the nuclei pairs, described here in pseudo code:

- Generate a distance matrix M where $M[i,j]$ is the distance between the ith and jth nuclei. Only the lower triangular part is calculated, since M is a symmetric matrix.
- Add a large value, L, to the diagonal of the distance matrix so that no nucleus is paired with itself.
- Find the minimum value in the matrix, m_{ij}
- Add the pair of nuclei i,j to the list of paired nuclei
- Change the values of rows i,j and columns i,j to L.
- Repeat steps 3-5 until the minimum value in the matrix is L.

The distance matrix takes $O(n^2)$ to create, while each pass of the elimination loop is also $O(n^2)$, and so the overall complexity of the algorithm is $O(n^3)$. The algorithm runs sufficiently quickly for images with several hundred nuclei, on the order of a few minutes using my desktop machine. However, the algorithm is less appropriate for embryos with several thousand nuclei, such as those in cleavage cycle 14.

3.2 Poisson Estimator

The second method is applied to embryos that are significantly past their most recent nuclear division[1], and where the expression programs of the proteins involved are at a more steady state behavior. As with the binomial estimator, the central idea of the model is that variability in expression can somehow be related to copy numbers. However, it is far from clear what an appropriate model for the steady state distribution of protein copy number should be. In simpler, prokaryotic situations, stochastic chemical kinetic models have been quite successful at capturing and explaining stochasticity in gene expression (see [RvO08] for a recent review). Such models usually incorporate stochastic production and decay of mRNAs and proteins, and possibly other processes. A common finding is that variability in protein levels is often due more to variability in the mRNA levels than to the inherent stochasticity in protein production and decay. Bar-Even et al. ([BEPM+06]), in an empirical study using S. cerevisiae, found that under a variety of conditions and for a variety of genes, variance in protein expression across cells in a population was proportional to mean protein expression. By analyzing different possible sources of expression variability, they came to the conclusion that mRNA fluctuations were the main cause, just as in the bacterial models.

As in Bar-Even et al. ([BEPM+06]), I find that protein expression variance scales roughly linearly with mean protein expression. However, I consider it unlikely that mRNA fluctuations are as significant a source of noise as they are in that study or in the bacterial models. The Drosophila embryo is syntical[2] at this stage of segmentation. For most of the segmentation genes, mRNAs are transcribed and exported from the nuclei and accumulate in inter-nuclear space, apparently at much greater concentration than is typical for the models or experiments cited above. This is supported by

[1] this event is also referred to as a *cleavage cycle*

[2] there are no cell membranes formed that partition the nuclei into cells

mRNA stains in embryos at the same developmental stage ([JSR07]), although, like proteins, mRNA copy numbers have not be quantified in this setting. Proteins are translated from mRNAs in the extra-nuclear space, and are taken up quickly by nearby nuclei, where they may eventually decay. Assuming that the total amount of mRNA in the vicinity of a nucleus is relatively constant, the main factors influencing protein expression would thus be stochastic production and uptake along with stochastic decay.

Based on this reasoning, I assume that the number of proteins in a nucleus, N_i, is Poisson distributed, with a nucleus-specific Poisson parameter λ_i. As for the previous estimator. I assume that the observed fluorescent intensity of a nucleus, O_i, is proportional to the expression ($O_i = \nu N_i$). Under these conditions, I can relate the ratio of expression to ν:

$$\frac{Var(O_i)}{E(O_i)} = \frac{\nu^2 Var(N_i)}{\nu E(N_i)} = \nu \tag{15}$$

So the problem of estimating ν reduces to estimating the mean and variances for the O_i. The problem is that expression in each nucleus is being driven at a different rate λ_i, and one single observation is not sufficient to estimate the mean and variance of the intensity distribution.

R $\sim$ Poisson(λ) if P(R = k) = exp $(-\lambda) * \frac{\lambda^k}{k!} for \lambda, k > 0$ if $O_i = \nu * N_i$, where $N_i \sim Poisson(\lambda) =>$ $O_i \sim Poisson(\nu * \lambda)$

If the O_i are independent and identically distributed, the joint likelihood is as below

$$L(O_1, \ldots, O_n \mid \nu, \lambda) = \left(\exp(-\nu\lambda)\frac{\nu\lambda^{O_1}}{O_1!}\right) \ldots \left(\exp(-\nu\lambda)\frac{\nu\lambda^{O_n}}{O_n!}\right) \tag{16}$$

$$= \prod_{i=1}^{n} \exp(-\nu\lambda) * \frac{\nu\lambda^{O_i}}{O_i!} \tag{17}$$

$$= \exp(-n\nu\lambda) \prod_{i=1}^{n} \frac{\nu\lambda^{O_i}}{O_i!} \tag{18}$$

Then the log-likelihood is

$$l(O_1, \ldots, O_n \mid \nu, \lambda) = \log(\exp(-n\nu\lambda)) + \sum_{i=1}^{n} \log \frac{\nu\lambda^{O_i}}{O_i!}$$

$$= -n\nu\lambda + \sum_{i=1}^{n} O_i * (\log \nu\lambda - \log O_i!)$$

Take the partial derivative of l by $\nu\lambda$ to get

$$\frac{\partial l}{\partial \nu\lambda} = -n + \sum_{i=1}^{n} \frac{O_i}{\nu\lambda}$$

Set the partial derivative of l by $\nu\lambda$ equal to zero to see that the optimal value is

$$\nu\lambda = \frac{\sum_{i=1}^{n} O_i}{n}$$

But the O_i are not all independent and identically distributed, as inspection of the image plainly shows that the position of the nucleus within the embryo has an great effect on the protein expression.

So the λ values for the N_i are not identical. For the Poisson estimator to work, I need to group nuclei by similar expression levels so as to satisfy the assumption that the N_i are independent and identically distributed.

Since nearby nuclei usually show similar expression levels, because they are subject to similar regulatory signals, the proposed solution is to relax the assumption that each nucleus has an independent value of λ_i, and assume that nuclei with similar expression share a common λ parameter. Then for each nucleus i, other nuclei in the neighbourhood with highly similar expression levels are used to estimate $E(O_i)$. With estimates of the mean intensity for each nucleus in a neighbourhood, a per-nucleus estimate of ν is derived as:

$$\hat{\nu}_i = \frac{(O_i - \hat{E}(O_i))^2}{\hat{E}(O_i)} \tag{19}$$

These estimates are averaged across nuclei that share similar values for λ_i, to generate the neighbourhood estimate:

$$\hat{\nu} = \frac{1}{M} \sum_{i=1}^{M} \hat{\nu}_i \tag{20}$$

Assuming there are M nuclei in a neighbourhood. After partitioning the nuclei, the Poisson estimates for ν were obtained for each channel, and were averaged across partitions to obtain ν estimates for each channel in the embryo.

3.2.1 Partitioning

A number of different ways to partition the nuclei in each embryo were explored. The measure of validation for the partitioning methods was taken to be the degree of agreement across embryos for the estimates produced by the Poisson estimator for a given transcription factor, as well as the agreement with the estimate produced by the binomial estimator for the same transcription factor. First, I tried to classify the nuclei by establishing three classes of expression level: low, medium and high. The thresholds that defined the classes were those values that split the empirical distribution over intensity into thirds. This method frequently assigned matches to nuclei which were spatially distant from each other, and produced results with excessively large variance in estimates (data not shown). Next, the previous thresholding method was applied, followed by a nearest neighbour classification algorithm to try and identify regions of low, medium and high classes. This nearest neighbour post thresholding approach produced similarly poor results (data not shown). Finally, I chose a more fine grained partitioning approach. Each embryo was split into boxes along the anterior-posterior (AP) axis. The boxes spanned the length of the AP axis, each box being 5% of the total length of the embryo. Each box had a height which began at 40% of the dorsal-ventral (DV) axis, and ended at 60% of the DV axis. According to [MSK+01], the measured intensity values in the image are most reliable at the midpoint of the DV axis (50%), so nuclei in both extremes of the DV axis were discarded. The nuclei in each box were assumed to share a similar value for λ_i. This partitioning method achieved the most consistent results not just within the Poisson estimates, but across binomial and Poisson estimates.

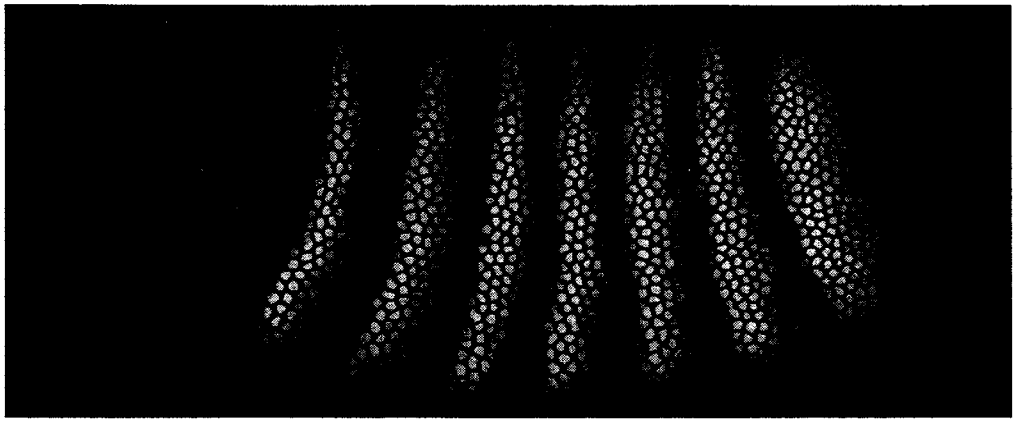

Figure 4: An image of embryo bd5. Notice the curved lines representing the Even-Skipped protein. Compared to Caudal and Bicoid, the even-skipped stripes are difficult to isolate in terms of a sub-interval of the AP axis.

3.2.2 Estimating mean intensity

A localized regression model is fitted to predict the expected intensity of each nucleus. For these, I generate $\hat{E}(O_i)$ in three steps:

1. identify all nuclei in the same neighbourhood box as nucleus i
2. fit their observed intensities by linear least squares regression, as a function of the AP or DV position values
3. evaluate the regressor at the coordinates of nucleus i to obtain the estimated intensity

The maternal and gap genes have relatively simple, spatially smooth expression patterns, and so linear regression works well enough to capture the expression given the neighbourhoods. For the pair-rule genes, such as even-skipped, I tried a full quadratic regression, which fits the observed intensities as a function of the linear and squared terms of AP and DV position, as well as the AP*DV cross term.

Chapter 4

Results

This chapter summarizes the results obtained from the experiments performed. Results from the binomial method and the poisson method are presented in separate sections. The chapter includes a description of two simple image noise models that were applied to try and increase the accuracy among estimates. Discussion of the results are postponed until the next chapter. Readers can reference the appendix to see the R scripts used to generate the results. All the data used may be downloaded from the FlyEx website, found at `http://flyex.ams.sunysb.edu/flyex/`.

I applied the binomial method to the 15 embryos, identified from a thorough search of the FlyEx database, which displayed signs of recent division. Also, I applied the Poisson method to 16 embryos, most from cleavage cycle 14A, time class 8. Results appear in the tables below for each embryo and gene. Results for each gene and each method are also reported, and are obtained by averaging across the per-embryo estimates for the same method and gene.

The binomial and Poisson methods also produced some dramatically large per-nucleus or per-nucleus- pair estimates. When traced back, these obvious outliers had a variety of causes. Some were due to problems with the underlying data, such as falsely detected nuclei in the images or nuclear segmentation errors. Some also resulted from poor expected intensity estimates in the Poisson method, particularly for complex parts of the pair-rule expression patterns.

4.1 Biomial Results

The results for the binomial estimators appear in the tables 8, 9, and 10. Each of the 15 embryos were classified as being from cleavage cycles 11 or 12 (see table 2).

Table 8: ν estimates for embryos stained for Even-Skipped, Caudal, and Bicoid

Binomial Results (Quantitative Data)	*Eve*	*Cad*	*Bcd*
ab18	0.17190	0.31617	0.07740
ad14	0.22614	0.38697	0.24118
ad33	0.13556	0.42078	0.08333
ac22	0.16454	0.33208	0.21805
ad22	0.23921	0.44745	0.13639
ad23	0.17811	0.39404	0.26862
ad24	0.11234	0.22964	0.17684
cb16	0.14463	0.30007	0.24257

Table 9: ν estimates for embryos stained for Even-Skipped, Hunchback, and Bicoid

Binomial Results (Quantitative Data)	*Eve*	*Hun*	*Bcd*
hx11	0.18843	0.11722	0.30020
hx16	0.11210	0.30868	0.18779
hz12	0.03769	0.11267	0.62909

Table 10: ν estimates for embryos stained for Even-Skipped, Kruppel, and Bicoid

Binomial Results (Quantitative Data)	*Eve*	*Kr*	*Bcd*
iz3	0.16675	0.03119	0.15768
ms21	0.13646	0.03031	0.18715
ms9	0.13155	0.02297	0.36884
ms3	0.16682	0.06814	0.16438

4.2 Poisson Results

The results for the Poisson estimator appear below in table 11. Each of the 16 embryos was classified as being in cleavage cycle 14 time classes 7 or 8 (see table 3).

Table 11: ν Estimates for the Poisson method

Poisson Results (Quantitative Data)	*Eve*	*Cad*	*Bcd*
dq2	1.53293	0.14535	0.18959
tu7	2.01023	0.16715	0.19152
bd5	3.83405	0.12084	0.16155
ab11	0.28054	0.29971	0.22424
cb15	0.27526	0.43451	0.57685
ac10	0.13858	0.46014	0.40930
	Eve	*Kr*	*Bcd*
ms14	1.02874	0.21848	0.07911
ms36	1.92512	0.24482	0.06328
	Eve	*Kn*	*Hun*
dm14	2.92935	2.11017	0.76879
tn2	2.55198	2.31921	0.58113
hne8	2.03786	0.53148	0.33134
fq4	4.19338	2.29609	0.64039
	Eve	*Kr*	*Hun*
kf9	2.21019	0.27503	0.38056
ba3	3.27101	0.29012	0.90235
rf11	2.75806	0.32786	0.82912
rf6	2.94637	0.28394	0.71134

4.3 Estimates for ν Across Methods

Table 12: Copy number estimates from the binomial and Poisson estimation methods, along with the mean estimates of $\hat{\nu}$ and the standard deviation

	Binomial Estimates			Poisson Estimates		
	$\hat{\nu}$	Std Dev	Peak Copy	$\hat{\nu}$	Std Dev	Peak Copy
Maternal Genes						
Bicoid	0.2293007	0.13494150	1112.077	0.2369300	0.17339670	1076.2670
Hunchback	0.1795233	0.11187610	1420.428	0.6431275	0.20432390	396.4999
Caudal	0.3534000	0.07187813	721.562	0.2712833	0.14996930	939.9767
Gap Genes						
Kruppel	0.0381525	0.02032847	6683.704	0.2733750	0.03792682	932.7846
Knirps	0.0000000	0.00000000	0.000	1.8142370	0.86027550	140.5549
Pair-rule Genes						
Even-skipped	0.1541487	0.04873279	1654.247	2.1202280	1.23305800	120.2701

The estimates for ν are shown for each gene, along with the predicted copy number under conditions of maximal expression in table 12. This prediction is obtained simply as $\frac{255}{\hat{\nu}}$, as 255 is the maximum possible intensity in the 8-bit images. The genes are segregated into three classes according to the standard categories of maternal, gap, and pair-rule genes. For the maternal and gap genes, the binomial and Poisson methods were in broad agreement. For all genes, the estimates of ν by each method were within a factor of four of each other. There was particularly good agreement for the caudal gene. The agreement between the two methods was worse for the even-skipped gene, with the binomial estimate for ν being much much smaller than the Poisson estimate.

4.4 Noise Models

The models above fail to account for many potential sources of noise. This includes noise in the biological system, such as mRNA fluctuations in space or time, diffusion of mRNA or protein, fluctuations in regulatory factors, or fluctuations in ribosomes. The models also do not account for imaging or image processing noise. The net effect of all these sources of variation is impossible to know. In this section, I begin by exploring the effect of simple noise models on the binomial and Poisson estimation methods, along with estimates for ν. I examined two common noise models, additive Gaussian noise and multiplicative Gaussian noise.

4.4.1 Binomial model

For the binomial model, additive Gaussian noise means that $O_{i1} = \nu N_{i1} + \epsilon_{i1}$ and $O_{i2} = \nu N_{i2} + \epsilon_{i2}$, where the ϵ_{ij} are independent, mean zero, standard deviation σ random variables for all i and j. In the presence of this noise, the expectation of the binomial estimator with respect to the noise and the randomness in N_{i1} and N_{i2} can be approximated as

$$\hat{\nu} = \frac{E[(O_{i1} - O_{i2})^2]}{E[O_{i1} + O_{i2}]} \tag{21}$$

$$= \frac{E[(\nu(N_{i1} - N_{i2}) + \epsilon_{i1} - \epsilon_{i2})^2]}{E[\nu * N_{i1} + \epsilon_{i1} + \nu * N_{i2} + \epsilon_{i2}]} \tag{22}$$

$$= \frac{\nu^2 * N_i + 2\sigma^2}{\nu * N_i} \tag{23}$$

$$= \nu + \frac{2\sigma^2}{\nu * N_i} \tag{24}$$

Now consider the binomial model in the presence of multiplicative Gaussian noise. The model now becomes $O_{i1} = \nu N_{i1}\epsilon_{i1}$ and $O_{i2} = \nu N_{i2}\epsilon_{i2}$ where ϵ_{ij} are independent, mean 1, standard deviation σ Gaussian random variables for all i and j. In the presence of this noise, the expectation of the binomial estimator can be approximated as

$$\hat{\nu} = \frac{E[(O_{i1} - O_{i2})^2]}{E[O_{i1} + O_{i2}]} \tag{25}$$

$$= \frac{E[(N_{i1} * \nu * \epsilon_{i1} - N_{i2} * \nu * \epsilon_{i2})^2]}{E[N_{i1} * \nu * \epsilon_{i1} + N_{i2} * \nu * \epsilon_{i2}]} \tag{26}$$

$$= \frac{\frac{\nu^2 * N_i}{2}(\sigma^2 + \sigma^2 * N_i + 2)}{\nu * N_i} \tag{27}$$

$$= \nu + \frac{\sigma^2 * \nu}{2} * (1 + N_i) \tag{28}$$

In either case, if $\sigma > 0$ the binomial estimator will be biased to larger vales than ν. Fittingly, the size of the bias shrinks as N_i gets large in the case of additive noise, while the opposite is true in the case of multiplicative noise. This makes intuitive sense.

4.4.2 Poisson model

Similar to the case with the binomial estimator in the presence of noise, additive noise for the Poisson estimator means that $O_i = \nu N_i + \epsilon_i$, where the ϵ_i are independent, mean 0 Gaussian random variables with standard deviation σ. If I assume that the local regression for predicting $E(O_i)$ is accurate, so

that $\hat{E}(O_i) = \nu\lambda_i$ then

$$E\left[\frac{(O_i - \hat{E}(O_i))^2}{\hat{E}(O_i)}\right] = E\left[\frac{(\nu N_i + \epsilon_i - \nu\lambda_i)^2}{\nu\lambda_i}\right] \tag{29}$$

$$= E\left[\frac{\nu^2 N_i^2 - 2\nu^2 N_i\lambda_i + \epsilon_i^2 + \nu^2\lambda_i^2}{\nu\lambda_i}\right] \tag{30}$$

$$= \frac{\nu^2 E[N_i] - 2\nu^2\lambda_i E[N_i] + E[\epsilon_i^2] + \nu^2\lambda_i^2}{\nu\lambda_i} \tag{31}$$

$$= \frac{\nu^2(\lambda_i^2 + \lambda i) - \nu^2\lambda_i^2 + \sigma^2)}{\nu\lambda_i} \tag{32}$$

$$= \nu(\lambda_i + 1) - \nu\lambda_i + \frac{\sigma^2}{\nu\lambda_i} \tag{33}$$

$$= \nu + \frac{\sigma^2}{\nu\lambda_i} \tag{34}$$

If I assume the poisson model is subject to multiplicative Gaussian noise, then $O_i = \nu N_i\epsilon_i$, where ϵ_i are independently distributed Gaussian random variables with mean 1 and standard deviation σ. So then

$$E\left[\frac{(O_i - \hat{E}(O_i))^2}{\hat{E}(O_i)}\right] = E\left[\frac{(\nu N_i\epsilon_i - \nu\lambda_i)^2}{\nu\lambda_i}\right] \tag{35}$$

$$= \frac{E[\nu N_i^2\epsilon_i^2 - 2\nu N_i\lambda_i\epsilon_i + \nu\lambda_i^2]}{\lambda_i} \tag{36}$$

$$= \frac{\nu(\lambda_i^2 + \lambda_i)(\sigma^2 + 1) - \nu\lambda_i^2}{\lambda_i} \tag{37}$$

$$= \nu(1 + \lambda_i)(\sigma^2 + 1) - \nu \tag{38}$$

$$= \nu(\lambda_i + \sigma^2(1 + \lambda_i)) \tag{39}$$

As with the binomial, either source of noise will bias the estimate of ν higher, but as λ_i grows large, the bias will decrease and increase respectively.

4.4.3 Interpretation and validation of noise models

Based on these results, there should be a positive trend between $\hat{\nu}$ and intensity for multiplicative noise, and a negative trend between $\hat{\nu}$ and intensity for additive noise. If additive noise were present in the expression data, there should be a slight but significant negative correlation between the per-nucleus (or nuclear pair) estimate for ν and the expression intensity. For multiplicative noise, there should be a positive correlation between the same. However, the expression data does not support these models. Figure 5 plots the per-nucleus (or nucleus pair) estimate $\hat{\nu}$ against expression for three embryos stained for *even-skipped*. Figures 6 and 7 report the same for the *bicoid* and *caudal* genes respectively. Similar results were observed for other embryos.

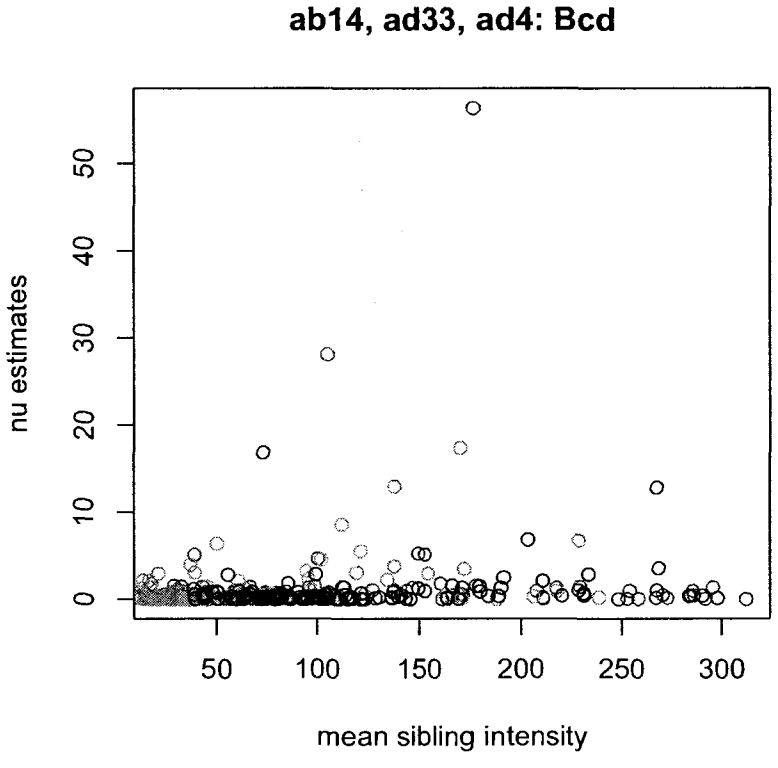

Figure 5: Even-skipped intensities versus ν estimates for ab18, ad33, ad4 respectively.

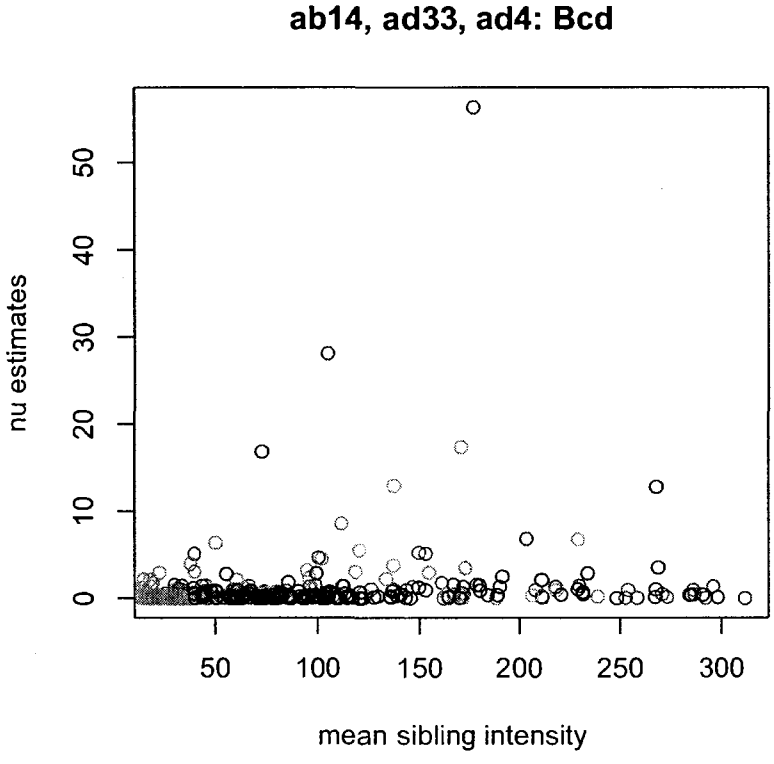

Figure 6: Bicoid intensities versus ν estimates for ab18, ad33, ad4 respectively.

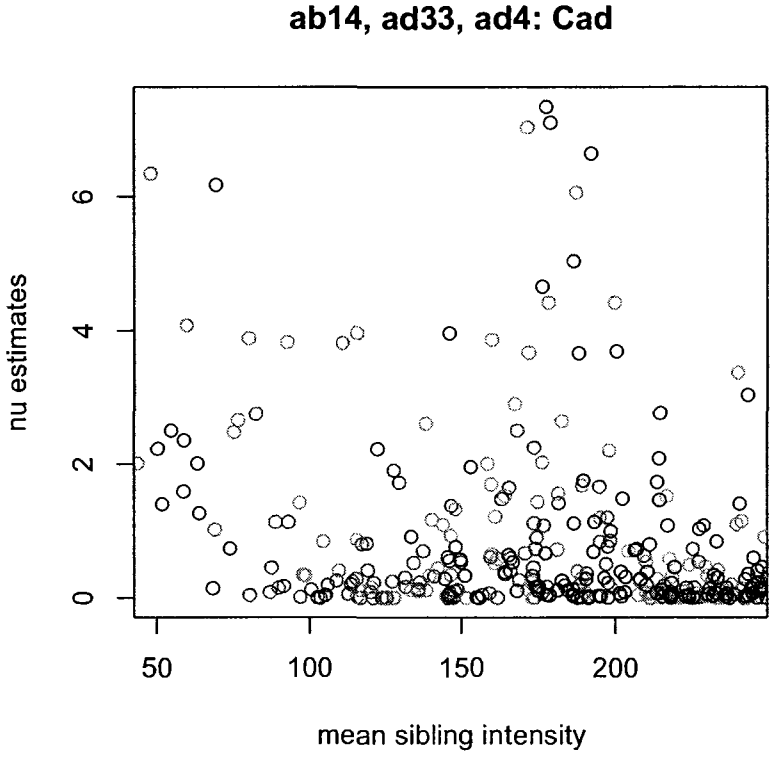

Figure 7: Caudal intensities versus ν estimates for ab18, ad33, ad4 respectively.

Chapter 5

Discussion and Conclusion

This chapter discusses the validity of the models in light of the results, and compares the results to other reported estimates of protein concentration. Several limitations of the methods and data are explored. Some potential avenues of future research are briefly outlined, followed by concluding remarks.

5.1 Discussion

The agreement of the binomial and Poisson estimators for the maternal and gap genes is encouraging, lending some measure of confidence to the methods. Another measure of confidence specific to the Poisson estimator is that it is the simplest model in the exponential family consistent with the observation that variance in expression seems to scale with mean expression. As previously suggested, both methods are expected to produce copy number estimates that are biased low, based on the other sources of variability that are all combined into one. Some of the other sources of variability include noise in the biological system, such as fluctuations in mRNA levels, diffusion of mRNA or protein, fluctuations in the concentration of unobserved transcription factors that regulate the observed factor, or fluctuations in the number of ribosomes near the nucleus. Further lab experiments would be required to characterize the extent to which each of these sources of variability plays a role in the observed fluorescence levels. All things considered, the agreement between the two methods is not sufficient to conclude that they produce correct estimates.

For the pair-rule gene *even-skipped*, the binomial estimator suggests protein copy numbers 13.7 times as large as the Poisson estimator suggests. It is unknown which is closer to the true value, but I suspect the binomial estimator is closer. A simple explanation is that since both methods are expected to underestimate true copy numbers, whichever estimate is higher may be closer to the true value. Another reason is that the Poisson estimator relies on the proper identification of similarly regulated nuclei by position. But in some parts of the embryos, expression does not vary smoothly as a function of position, which makes the task of partitioning the nuclei to into sets that share the same λ_i parameters very difficult. Consequently the assumptions of the Poisson method are not satisfied, and the confidence in the estimate is low for even-skipped.

5.1.1 Pairing sibling nuclei

The greedy approach to identifying nuclear pairs is clearly not optimal, since a simple mistake in the ordering of the pairs to be considered will result in a lot of propagated errors. Furthermore, in some cases, the correct partner for a nucleus is not the nearest one, but the second or third nearest. However, this only becomes clear in the context of the other nearby nuclei, making a greedy algorithm only an approximate solution. Pairing the nuclei by hand would produce more reliable results, but comes at a cost of too much time. In a situation with a data set of hundreds of embryos, each with hundreds of nuclei, pairing by hand is clearly not feasible.

There are algorithms from computational geometry that can solve this problem in less time (a simple computational geometry approach to finding closest pairs will reduce the complexity to $O(n^2 \log n)$ ([CLRS01]) but this is still not very satisfying, since it also employs a greedy method to pair points. Furthermore, there is a limit to how much information about regulatory inputs can be extracted from nuclear position alone ([GTWB07]). A method that could automatically identify reliable pairings would be a substantial achievement on its own, and is a possible avenue for future research.

5.1.2 Noise complexity

Nuclear position within the embryo acts as a surrogate for the more complex regulatory network that influences expression. It seems very likely that some for some of the genes, particularly the pair rule genes, the expression is clearly not a linear function of nuclear position along the AP and DV axes. Other measurements are required to obtain a fuller picture of that network.

5.1.3 Comparison to other estimates

Gregor et al. ([GTWB07]) performed experiments to calibrate observed fluorescence to absolute expression of the Bicoid protein. They estimated that in the highest expressing nuclei, Bicoid is present at a concentration of 55 $\pm$ 3 nM, or 33 $\pm$ 1.8 molecules/m^3. Assuming spherical nuclei that are an estimated 6.5 nm in diameter, or 144 μm^3 in volume, this corresponds to 4750 $\pm$ 260 molecules per nucleus. This value is just over 4.2 times as large as I obtained from the binomial method, and 4.4 times as large as from the Poisson method. However, like the estimates produced in this work, the Gregor et al. estimate may be subject to a number of potential biases. So the disagreement between their results and the results reported herein cannot be interpreted as either supportive or contradictory. To my knowledge, it is the only available experimental estimate of Bicoid protein copy number. Should their estimate be more accurate, it suggests that our estimates may be significantly low. This may also be the case for the estimates of the other genes.

5.2 Conclusion

I developed and presented two estimators of protein copy number that apply to fluorescence expression images containing clearly identifiable nuclei (or cells). The two methods both assume a

proportional relationship between observed intensity and protein concentration, but make different assumptions about the concentration distributions. The first method is applied to images where the nuclei (or cells) have just undergone division, and assumes an even binomial distribution over the proteins passed from the mother nucleus to the daughter nuclei. It tries to exploit this variability to estimate ν, the proportional relationship between fluorescence and concentration. The second method is applied to images where the main factors influencing protein expression are stochastic production and uptake along with stochastic decay. It assumes a Poisson distribution over protein concentration, and tries to pool nuclei that share the same poisson parameter together to produce an estimate of ν. Using these two approaches, I estimated copy numbers for 6 genes in the segmentation network of *Drosophila*. Estimates ranged from several hundreds of proteins per nucleus up to thousands. These estimates are considered to be lower bounds on the true values, since they assume all variability in expression is due to fundamental stochastic chemical processes. The extend to which this biases the estimates is unknown. Further calibration experiments would help to reveal the extend of this bias, and hopefully lead to an unbiased estimator for protein copy number.

Bibliography

[BEPM+06] Arren Bar-Even, Johan Paulsson, Narendra Maheshri, Miri Carmi, Erin O'Shea, yitzhak Pilpel, and Naama Barkai. Noise in protein expression scales with natural protein abundance. *Nature genetics*, 2006.

[BS03] K. S. Brown and J. P. Sethna. Statistical mechanical approaches to models with many poorly known parameters. *Physical Review*, 68(2):021904–1 – 021904–9, August 2003.

[CLRS01] Thomas T. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, USA, 2001.

[CMRG02] Yan Chen, Joachim D. Muller, QiaoQiao Ruan, and Enrico Gratton. Molecular Brightness Characterization of EGFP In Vivo by Fluorescence Fluctuation Spectroscopy. *Biophys. J.*, 82(1):133–144, 2002.

[GTWB07] Thomas Gregor, David W. Tank, Eric F. Wieschaus, and William Bialek. Probing the limits to positional information. *Cell*, 130(1):153 – 164, 2007.

[IGH01] Trey Ideker, Timothy Galitski, and Leroy Hood. A new approach to decoding life: Systems biology. *Annual Review of Genomics and Human Genetics*, 2(1):343–372, 2001. PMID: 11701654.

[JKVA+05] Hilde Janssens, Dave Kosman, Carlos E Vanario-Alonso, Johannes Jaeger, Maria Samsonova, and John Reinitz. A high-throughput method for quantifying gene expression data from early Drosophila embryos. *Dev Genes Evol*, 215(7):374–81, 2005.

[JSR07] Johannes Jaeger, David H. Sharp, and John Reinitz. Known maternal gradients are not sufficient for the establishment of gap domains in drosophila melanogaster. *Mechanisms of Development*, 124(2):108 – 128, 2007.

[Kit02] Hiroaki Kitano. Systems Biology: A Brief Overview. *Science*, 295(5560):1662–1664, 2002.

[KMP+02] Konstantin Kozlov, Ekaterina Myasnikova, Andrei Pisarev, Maria Samsonova, and John Reinitz. A method for two-dimensional registration and construction of the two-dimensional atlas of gene expression patterns in *situ*. *In Silico Biology*, 2(2):125–141, 2002.

[KPUG99] Peet Kask, Kaupo Palo, Dirk Ullmann, and Karsten Gall. Fluorescence-intensity distribution analysis and its application in biomolecular detection technology. *Proceedings of the National Academy of Sciences of the United States of America*, 96(24):13756–13761, 1999.

[Lev08] Mike Levine. A systems view of drosophila segmentation. *Genome Biology*, 9(2):p. 207, 2008.

[LPDA08] Vinzenz Lange, Paola Picotti, Bruno Domon, and Ruedi Aebersold. Selected reaction monitoring for quantitative proteomics: a tutorial. *Mol Syst Biol*, 4, 2008.

[LPS07] Eric Libby, Theodore J. Perkins, and Peter S. Swain. Noisy information processing through transcriptional regulation. *Proceedings of the National Academy of Sciences*, 104(17):7151–7156, 2007.

[MO02] Neil J. McKenna and Bert W. O'Malley. Combinatorial control of gene expression by nuclear receptors and coregulators. *Cell*, 108(4):465 – 474, 2002.

[MSK+01] Ekaterina Myasnikova, Anastassia Samsonova, Konstantin Kozlov, Maria Samsonova, and John Reinitz. Registration of the expression patterns of Drosophila segmentation genes by two independent methods . *Bioinformatics*, 17(1):3–12, 2001.

[MSKR05] Ekaterina Myasnikova, Maria Samsonova, David Kosman, and John Reinitz. Removal of background signal from in situ data on the expression of segmentation genes in Drosophila. *Dev Genes Evol*, 215(6):320–6, 2005.

[OR02] George Orphanides and Danny Reinberg. A unified theory of gene expression. *Cell*, 108:439–451, 2002.

[PAD07] Paola Picotti, Ruedi Aebersold, and Bruno Domon. The Implications of Proteolytic Background for Shotgun Proteomics. *Mol Cell Proteomics*, 6(9):1589–1598, 2007.

[PJRG06] Theodore J Perkins, Johannes Jaeger, John Reinitz, and Leon Glass. Reverse engineering the gap gene network of drosophila melanogaster. *PLoS Comput Biol*, 2(5):e51, 05 2006.

[PPB+04] Ekaterina Poustelnikova, Andrei Pisarev, Maxim Blagov, Maria Samsonova, and John Reinitz. A database for management of gene expression data in situ. *Bioinformatics*, 20(14):2212–2221, 2004.

[RO05] Jonathan M. Raser and Erin K. O'Shea. Noise in Gene Expression: Origins, Consequences, and Control. *Science*, 309(5743):2010–2013, 2005.

[RPA+06] Nitzan Rosenfeld, Theodore J Perkins, Uri Alon, Michael B Elowitz, and Peter S Swain. A fluctuation method to quantify in vivo fluorescence data. *Biophys. J.*, page biophysj.105.073098, 2006.

[RvO08] Arjun Raj and Alexander van Oudenaarden. Nature, nurture, or chance: Stochastic gene expression and its consequences. *Cell*, 135(2):216 – 226, 2008.

[RWA02] Christopher V. Rao, Denise M. Wolf, and Adam P. Arkin. Control, exploitation and tolerance of intracellular noise. *Nature*, 420:231–237, 2002.

[SC87] Matthew P. Scott and Sean B. Carroll. The segmentation and homeotic gene network in early drosophila development. *Cell*, 51(1):689–698, 1987.

[SES02] Peter S. Swain, Michael B. Elowitz, and Eric D. Siggia. Intrinsic and extrinsic contributions to stochasticity in gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, 99(20):12795–12800, 2002.

[SKK+08] Svetlana Surkova, David Kosman, Konstantin Kozlov, Manu, Ekaterina Myasnikova, Anastasia A. Samsonova, Alexander Spirov, Carlos E. Vanario-Alonso, Maria Samsonova, and John Reinitz. Characterization of the drosophila segment determination morphome. *Developmental Biology*, 313(2):844 – 862, 2008.

[SPF+04] Mark D Schroeder, Michael Pearce, John Fak, HongQing Fan, Ulrich Unnerstall, Eldon Emberly, Nikolaus Rajewsky, Eric D Siggia, and Ulrike Gaul. Transcriptional control in the segmentation gene network of drosophila. *PLoS Biol*, 2(9):e271, August 2004.

[Swa04] Peter S. Swain. Efficient attenuation of stochasticity in gene expression through post-transcriptional control. *Journal of Molecular Biology*, 344(4):965 – 976, 2004.

[TCB08] Gaper Tkaik, Curtis G. Callan, and William Bialek. Information flow and optimization in transcriptional regulation. *Proceedings of the National Academy of Sciences*, 105(34):12265–12270, 2008.

[TXGB07] Tianhai Tian, Songlin Xu, Junbin Gao, and Kevin Burrage. Simulated maximum likelihood method for estimating kinetic rates in gene expression. *Bioinformatics*, 23(1):84–91, 2007.

[ZR01] Yi Zhang and Danny Reinberg. Transcriptional regulation by histone methylation: interplay between different covalent modifications of the core histone tails. *Genes & Development*, 15(18):2343 – 2360, 2001.

Appendix A

Colour figures

As mentioned in this section, each embryo is observed only once, as the process of confocal scanning destroys the embryo.

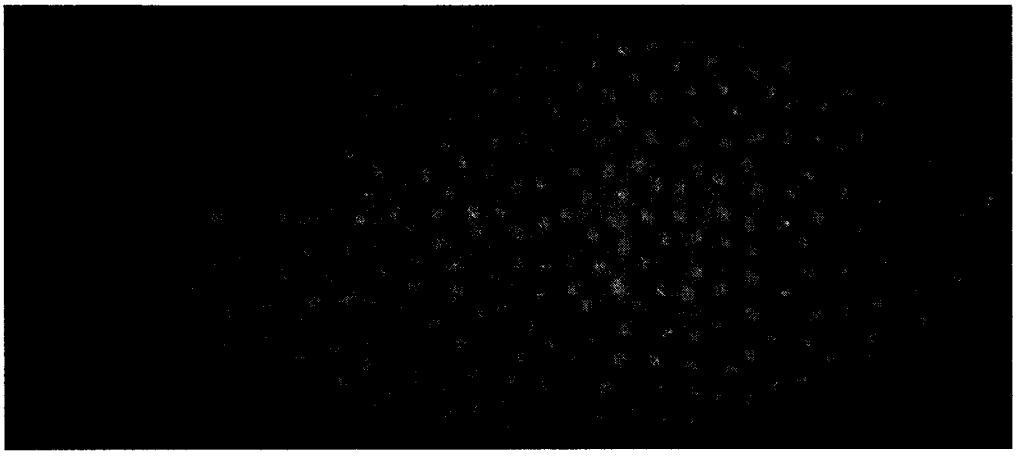

Figure 8: A colour sample of the images from the data set, embryo ad24 from cleavage cycle 11.

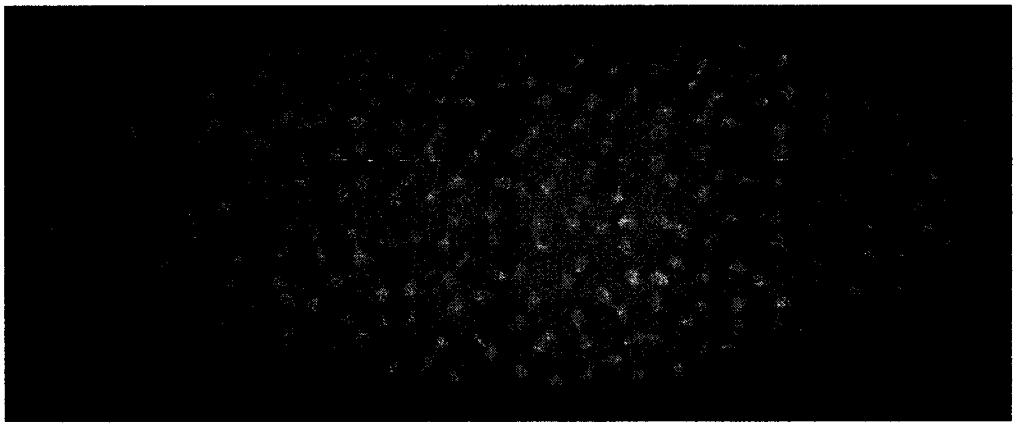

Figure 9: An embryo (in colour) that has just undergone a recent nuclear division. The developmental program in drosophila makes the nuclei divide roughly simultaneously. The recent division makes the task of estimating sibling pairs much easier.

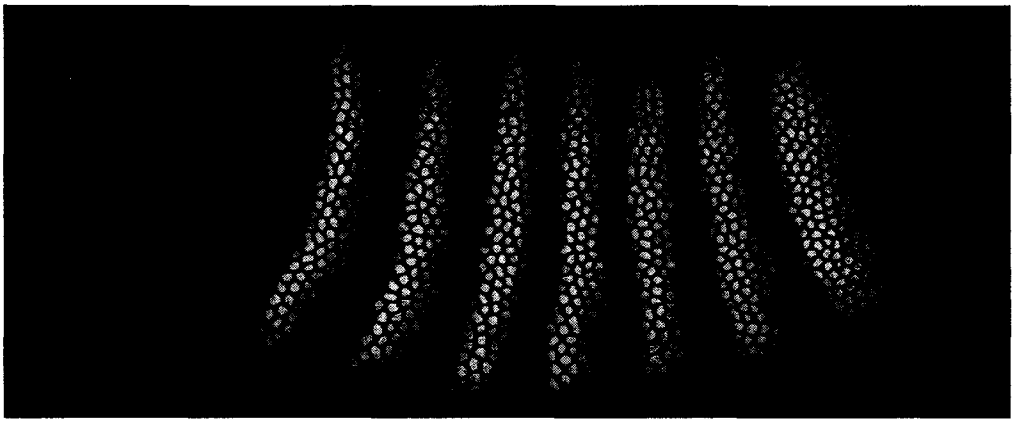

Figure 10: A colour image of embryo bd5. Notice the curvature of the green lines representing the Even-Skipped protein. Compared to the red and blue channels, Caudal and Bicoid, the even-skipped stripes are difficult to isolate in terms of a sub-interval of the AP axis.

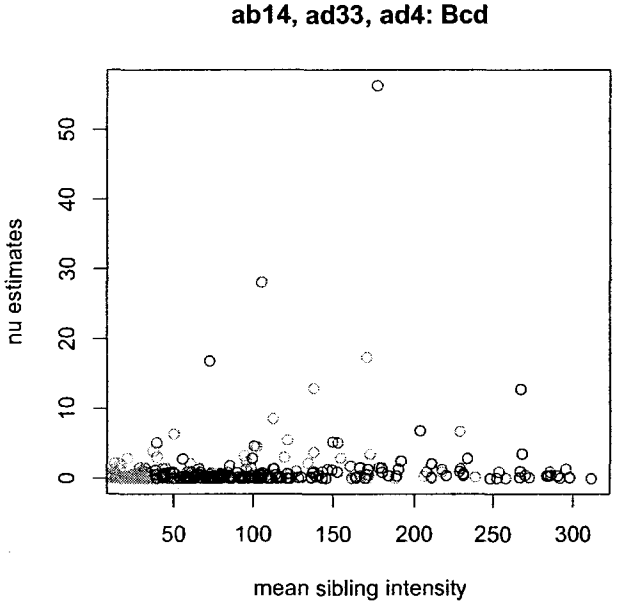

Figure 11: Even-skipped intensities versus ν estimates for ab18 (red), ad33 (green), ad4(blue) respectively.

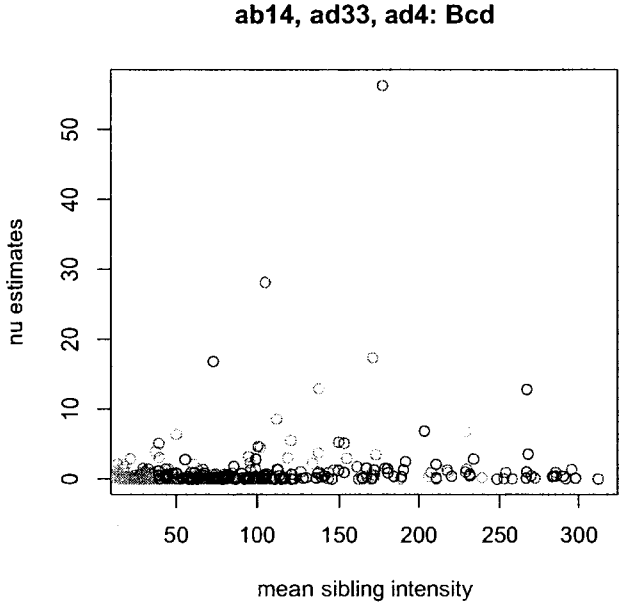

Figure 12: Bicoid intensities versus ν estimates for ab18 (red), ad33 (green), ad4(blue) respectively.

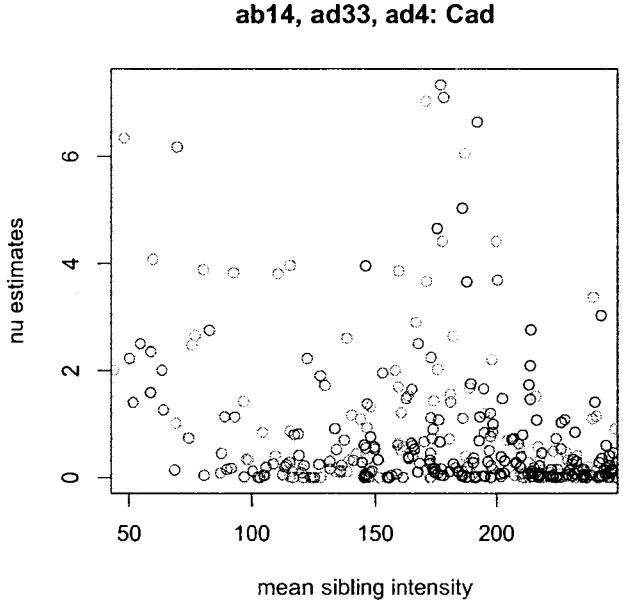

Figure 13: Caudal intensities versus ν estimates for ab18 (red), ad33 (green), ad4(blue) respectively.

Appendix B

Scripts for binomial method experiments

B.1 R script to load data

Listing B.1: R script to load the data

```
### Loads the data in ~/thesis/textdata/early
### early embryos that have undergone background correction. into the current R workspace
### Meant to be run in R from the working directory <whatever>/thesis/scripts/

#save the current directory
curDir <- getwd()
splitDir <- unlist(strsplit(curDir,'thesis'))
setwd(paste(splitDir[1],'thesis/textdata/early/',sep=''))

# Load the ECB data
ab18 <- read.table(file="ab18.txt",col.names=c("num","AP","DV","Eve","Cad","Bcd"),skip=5)
ac22 <- read.table(file="ac22.txt",col.names=c("num","AP","DV","Eve","Cad","Bcd"),skip=5)
ad22 <- read.table(file="ad22.txt",col.names=c("num","AP","DV","Eve","Cad","Bcd"),skip=5)
ad24 <- read.table(file="ad24.txt",col.names=c("num","AP","DV","Eve","Cad","Bcd"),skip=5)
ad14 <- read.table(file="ad14.txt",col.names=c("num","AP","DV","Eve","Cad","Bcd"),skip=5)
ad23 <- read.table(file="ad23.txt",col.names=c("num","AP","DV","Eve","Cad","Bcd"),skip=5)
ad33 <- read.table(file="ad33.txt",col.names=c("num","AP","DV","Eve","Cad","Bcd"),skip=5)
cb16 <- read.table(file="cb16.txt",col.names=c("num","AP","DV","Eve","Cad","Bcd"),skip=5)

# Load the EHB data
hx16 <- read.table(file="hx16.txt",col.names=c("num","AP","DV","Eve","Hun","Bcd"),skip=5)
hx11 <- read.table(file="hx11.txt",col.names=c("num","AP","DV","Eve","Hun","Bcd"),skip=5)
hz12 <- read.table(file="hz12.txt",col.names=c("num","AP","DV","Eve","Hun","Bcd"),skip=5)

# Load the EKrB data
iz3 <- read.table(file="iz3.txt",col.names=c("num","AP","DV","Eve","Kr","Bcd"),skip=5)
ms21 <- read.table(file="ms21.txt",col.names=c("num","AP","DV","Eve","Kr","Bcd"),skip=5)
ms9 <- read.table(file="ms9.txt",col.names=c("num","AP","DV","Eve","Kr","Bcd"),skip=5)
ms3 <- read.table(file="ms3.txt",col.names=c("num","AP","DV","Eve","Kr","Bcd"),skip=5)

# Load new hand-paired embryos: FESce05, HETae4, HETae10
FESce05 <- read.table(file="BECab17.txt",col.names=c("num","AP","DV","Fsh","Eve","Slp"),skip=5)
FESce05 <- FESce05[,c(1,2,3,5,4,6)]
```

```
HETae4 <- read.table(file="BECab17.txt",col.names=c("num","AP","DV","Hun","Eve","Tls"),skip=5)
HETae4 <- HETae4[,c(1,2,3.5,4,6)]
HETae10 <- read.table(file="BECab17.txt",col.names=c("num","AP","DV","Hun","Eve","Tls"),skip=5)
HETae10 <- HETae10[,c(1,2.3,5,4,6)]

#restore current working directory, clean up temporary vars.
setwd(curDir)
rm(curDir,splitDir)
```

B.2 R script to perform experiments

Listing B.2: R script to perform the binomial estimates

```
# Performs the binomial estimates for younger embryos

############ First method of estimates: binomial model
# This script assumes that the data have been loaded into data frames,
# each bearing the name of the embryo from which it has been extracted.


# Step one, find the sibling nuclei in each image
if (!exists("ab18EDM")){
        ab18EDM <- genEDM(ab18[,2:3])
        }
if (!exists("ac22EDM")){
        ac22EDM <- genEDM(ac22[,2:3])
        }
if (!exists("ad22EDM")){
        ad22EDM <- genEDM(ad22[,2:3])
        }
if (!exists("ad24EDM")){
        ad24EDM <- genEDM(ad24[,2:3])
        }
if (!exists("ad14EDM")){
        ad14EDM <- genEDM(ad14[,2:3])
        }
if (!exists("ad23EDM")){
        ad23EDM <- genEDM(ad23[,2:3])
        }
if (!exists("ad33EDM")){
        ad33EDM <- genEDM(ad33[,2:3])
        }
if (!exists("hx11EDM")){
        hx11EDM <- genEDM(hx11[,2:3])
        }
if (!exists("hx16EDM")){
        hx16EDM <- genEDM(hx16[,2:3])
        }
if (!exists("cb16EDM")){
        cb16EDM <- genEDM(cb16[,2:3])
        }
if (!exists("hz12EDM")){
        hz12EDM <- genEDM(hz12[,2:3])
        }
if (!exists("iz3EDM")){
        iz3EDM <- genEDM(iz3[,2:3])
        }
if (!exists("ms21EDM")){
        ms21EDM <- genEDM(ms21[,2:3])
```

```
        }
if (!exists("ms9EDM")){
        ms9EDM <- genEDM(ms9[,2:3])
        }
if (!exists("ms3EDM")){
        ms3EDM <- genEDM(ms3[,2:3])
        }
if (!exists("FESce05EDM")){
        FESce05EDM <- genEDM(FESce05[,2:3])
        }
if (!exists("HETae4EDM")){
        HETae4EDM <- genEDM(HETae4[,2:3])
        }
if (!exists("HETae10EDM")){
        HETae10EDM <- genEDM(HETae10[,2:3])
        }

# load the pairs data from file

curDir <- getwd()
splitDir <- unlist(strsplit(curDir,'thesis'))
setwd(paste(splitDir[1],'thesis/textdata/pairings/',sep=''))

if (!exists("ab18sibs")){
        ab18sibs <- read.table(file="ab18pairs.txt")
        }
if (!exists("ac22sibs")){
        ac22sibs <- findSibs(ac22EDM)
        }
if (!exists("ad22sibs")){
        ad22sibs <- findSibs(ad22EDM)
        }
if (!exists("ad24sibs")){
        ad24sibs <- findSibs(ad24EDM)
        }
if (!exists("ad14sibs")){
        ad14sibs <- findSibs(ad14EDM)
        }
if (!exists("ad23sibs")){
        ad23sibs <- findSibs(ad23EDM)
        }
if (!exists("ad33sibs")){
        ad33sibs <- read.table(file="ad33pairs.txt")
        }
if (!exists("cb16sibs")){
        cb16sibs <- findSibs(cb16EDM)
        }
if (!exists("hz12sibs")){
        hz12sibs <- read.table(file="hz12pairs.txt")
        }
if (!exists("hx11sibs")){
        hx11sibs <- findSibs(hx11EDM)
}
if (!exists("hx16sibs")){
        hx16sibs <- findSibs(hx16EDM)
}
if (!exists("iz3sibs")){
        iz3sibs <- read.table(file="iz3_pairs.txt")
        }
if (!exists("ms21sibs")){
```

```
        ms21sibs <- findSibs(ms21EDM)
        }
if (!exists("ms9sibs")){
        ms9sibs <- read.table(file="ms9_pairs.txt")
        }
if (!exists("ms3sibs")){
        ms3sibs <- findSibs(ms3EDM)
        }
if (!exists("FESce05sibs")){
        FESce05sibs <- read.table(file="FESce05_pairs.txt")
        }
if (!exists("HETae4sibs")){
        HETae4sibs <- findSibs(HETae4EDM)
        }
if (!exists("HETae10sibs")){
        HETae10sibs <- read.table(file="HETae10_pairs.txt")
        }

#restore current working directory, clean up temporary vars.
setwd(curDir)
rm(curDir,splitDir)

# Step two, calculate the binomial model estimates of \hat(nu)
# pairUpNew: used
# for embryos where the siblings are estimated via the greedy pairing method.
# Nuclei in these files are indexed from one.
# pairUpZero: used for embryos where
# the siblings are paired by hand.  Nuclei in these files are indexed from zero.

ab18ests <-
c(binEstimate(pairUpZero(ab18[,"Eve"],ab18sibs)),binEstimate(pairUpZero(ab18[,"
Cad"],ab18sibs)),binEstimate(pairUpZero(ab18[,"Bcd"],ab18sibs))) ad14ests <-
c(binEstimate(pairUpNew(ad14[,"Eve"],ad14sibs)),binEstimate(pairUpNew(ad14[,"Cad
"],ad14sibs)),binEstimate(pairUpNew(ad14[,"Bcd"],ad14sibs))) ad33ests <-
c(binEstimate(pairUpZero(ad33[,"Eve"],ad33sibs)),binEstimate(pairUpZero(ad33[,"
Cad"],ad33sibs)),binEstimate(pairUpZero(ad33[,"Bcd"],ad33sibs))) ac22ests <-
c(binEstimate(pairUpNew(ac22[,"Eve"],ac22sibs)),binEstimate(pairUpNew(ac22[,"Cad
"],ac22sibs)),binEstimate(pairUpNew(ac22[,"Bcd"],ac22sibs))) ad22ests <-
c(binEstimate(pairUpNew(ad22[,"Eve"],ad22sibs)),binEstimate(pairUpNew(ad22[,"Cad
"],ad22sibs)),binEstimate(pairUpNew(ad22[,"Bcd"],ad22sibs))) ad23ests <-
c(binEstimate(pairUpNew(ad23[,"Eve"],ad23sibs)),binEstimate(pairUpNew(ad23[,"Cad
"],ad23sibs)),binEstimate(pairUpNew(ad23[,"Bcd"],ad23sibs))) ad24ests <-
c(binEstimate(pairUpNew(ad24[,"Eve"],ad24sibs)),binEstimate(pairUpNew(ad24[,"Cad
"],ad24sibs)),binEstimate(pairUpNew(ad24[,"Bcd"],ad24sibs))) cb16ests <-
c(binEstimate(pairUpNew(cb16[,"Eve"],cb16sibs)),binEstimate(pairUpNew(cb16[,"Cad
"],cb16sibs)),binEstimate(pairUpNew(cb16[,"Bcd"],cb16sibs)))

hz12ests <-
c(binEstimate(pairUpZero(hz12[,"Eve"],hz12sibs)),binEstimate(pairUpZero(hz12[,"
Hun"],hz12sibs)),binEstimate(pairUpZero(hz12[,"Bcd"],hz12sibs))) hx11ests <-
c(binEstimate(pairUpNew(hx11[,"Eve"],hx11sibs)),binEstimate(pairUpNew(hx11[,"Hun
"],hx11sibs)),binEstimate(pairUpNew(hx11[,"Bcd"],hx11sibs))) hx16ests <-
c(binEstimate(pairUpNew(hx16[,"Eve"],hx16sibs)),binEstimate(pairUpNew(hx16[,"Hun
"],hx16sibs)),binEstimate(pairUpNew(hx16[,"Bcd"],hx16sibs)))

iz3ests <-
c(binEstimate(pairUpZero(iz3[,"Eve"],iz3sibs)),binEstimate(pairUpZero(iz3[,"Kr"]
,iz3sibs)),binEstimate(pairUpZero(iz3[,"Bcd"],iz3sibs))) ms21ests <-
c(binEstimate(pairUpNew(ms21[,"Eve"],ms21sibs)),binEstimate(pairUpNew(ms21[,"Kr"
],ms21sibs)),binEstimate(pairUpNew(ms21[,"Bcd"],ms21sibs))) ms9ests <-
```

```
c(binEstimate(pairUpZero(ms9[,"Eve"],ms9sibs)),binEstimate(pairUpZero(ms9[,"Kr"]
,ms9sibs)),binEstimate(pairUpZero(ms9[,"Bcd"],ms9sibs))) ms3ests <-
c(binEstimate(pairUpNew(ms3[,"Eve"],ms3sibs)),binEstimate(pairUpNew(ms3[,"Kr"],
ms3sibs)),binEstimate(pairUpNew(ms3[,"Bcd"],ms3sibs)))

FESce05ests <-
c(binEstimate(pairUpZero(FESce05[,"Eve"],FESce05sibs)),binEstimate(pairUpZero(
FESce05[,"Fsh"],FESce05sibs)),binEstimate(pairUpZero(FESce05[,"Slp"],FESce05sibs
))) HETae4ests <-
c(binEstimate(pairUpNew(HETae4[,"Eve"],HETae4sibs)),binEstimate(pairUpNew(HETae4
[,"Hun"],HETae4sibs)),binEstimate(pairUpNew(HETae4[,"Tls"],HETae4sibs)))
HETae10ests <-
c(binEstimate(pairUpZero(HETae10[,"Eve"],HETae10sibs)),binEstimate(pairUpZero(
HETae10[,"Hun"],HETae10sibs)),binEstimate(pairUpZero(HETae10[,"Tls"],HETae10sibs
)))

# collect the data in a matrix

ECBbin <- rbind(ab18ests, ad14ests, ad33ests, ac22ests, ad22ests, ad23ests, ad24ests, cb16ests)
rownames(ECBbin) <- c("ab18","ad14","ad33","ac22","ad22","ad23","ad24","cb16")
colnames(ECBbin) <- c("Eve", "Cad", "Bcd")

EHBbin <- rbind(hx11ests, hx16ests, hz12ests)
rownames(EHBbin) <- c("hx11","hx16", "hz12")
colnames(EHBbin) <- c("Eve","Hun","Bcd")

EKrBbin <- rbind(iz3ests, ms21ests, ms9ests, ms3ests)
rownames(EKrBbin) <- c("iz3", "ms21", "ms9", "ms3")
colnames(EKrBbin) <- c("Eve", "Kr", "Bcd")

# collect the hand-paired data in a matrix
HPbin <- rbind(ab18ests,ad33ests,hz12ests,iz3ests,ms9ests,FESce05ests,HETae10ests)
rownames(HPbin) <- c("ab18","ad33","hz12","iz3","ms9","FESce05","HETae10")
colnames(HPbin) <- c("Eve","Cad/Hun/Fsh/Kr","Bcd/Tls/Slp")

# remove temporary vars
rm(ab18ests, ad14ests, ad33ests, ac22ests, ad22ests, ad23ests, ad24ests)
rm(cb16ests, hz12ests, hx11ests, hx16ests, iz3ests, ms21ests, ms9ests, ms3ests, FESce05ests)
rm(HETae4ests, HETae10ests)
```

B.3 R functions

Listing B.3: R functions for the binomial estimator scripts

```
##################################################
#########   functions needed to generate binomial estimates


# Takes a euclidean distance matrix EDM, returns a matrix
# sibs where sibs[i,] is one nuclei pairing
# Elimination is greedy, which is NOT ideal.
findSibs <- function(EDM){
        # generate pairing vector
        size <- dim(EDM)[1]
        sibs <- matrix(0,nrow=1,ncol=2)

        # make diagonal large (currently zero vector)
        diag(EDM) <- rep(99,size)
```

```
        # Now. find minimum value in EDM at (rowInd,colInd),
        # pair elements in sibs, eliminate row,col 'rowInd' row,col 'colInd'
        # Keep track of the number of pairings, stop when
        # we've hit floor(rows(EDM)/2)
        stop <- floor(nrow(EDM)/2)
        pairs <- 0
        while(pairs < stop){
                rowInd <- which.min(apply(EDM,1,min))
                colInd <- which.min(apply(EDM,2,min))
                sibs <- rbind(sibs,c(rowInd,colInd))
                EDM[rowInd,] <- rep(99,size)
                EDM[,colInd] <- rep(99,size)
                EDM[,rowInd] <- rep(99,size)
                EDM[colInd,] <- rep(99,size)
                pairs <- pairs + 1
        }
        # Remove zero vector in row 1
        sibs[-1,]
}

# Generate what should be a lower triangular distance matrix
genEDM <- function(M){
        EDM <- matrix(99,nrow=dim(M)[1],ncol=dim(M)[1])
        for(i in 1:dim(EDM)[1]){
                for(j in 1:i){
                        EDM[i,j] <- el2norm(M[i,],M[j,])
                }
        }
        EDM
}


# Calculates the Euclidean distance between two vectors, x and y
el2norm <- function(x,y){
        l <- sqrt((x[1] - y[1])^2 + (x[2] - y[2])^2)
        l
}

# Performs the same function as pairUpNew, but is intended to take
# hand-paired sibling matrices (the sibs argument) which count nuclei from 0
# as opposed to sibling matrices which count nuclei from 1, and therefore cause
# no problems
pairUpZero <- function(ints,sibs){
        pairs <- matrix(0,nrow=1,ncol=2)
        bv <- rep(0,length(ints))
        for(i in 1:nrow(sibs)){
                if (sibs[i,1] < length(ints) & sibs[i,2] < length(ints)){
                                if(bv[sibs[i,1] + 1] == 0){
                                # add ints[i+1], ints[sibs[i]+1] to pairs matrix
                                # flag that both i and sibs[i] are paired in bv
                                pairs <- rbind(pairs, c(ints[sibs[i,1] +1],ints[sibs[i,2] +1]))
                                bv[sibs[i,1] + 1] <- 1
                                bv[sibs[i,2] + 1] <- 1
                                }
                        }
        }
        # get rid of the first row zero vector
        pairs <- pairs[-1,]
        pairs
}
```

```
# Takes a vector of intensity values (M) and a matrix of siblings
# (either M/2 square or (M-1)/2 square) to generate a matrix (in [~M/2,2]) where each row
# represents one of the pairs of sibling intensities
# bv is a bit vector that keeps track of which pairs of siblings are in the
# pairs matrix already.
pairUpNew <- function(ints,sibs){
        pairs <- matrix(0,nrow=1,ncol=2)
        bv <- rep(0,length(ints))
        for(i in 1:nrow(sibs)){
                if (sibs[i,1] < length(ints) & sibs[i,2] < length(ints)){
                                if(bv[sibs[i,1]] == 0){
                                # add ints[i], ints[sibs[i]] to pairs matrix
                                # flag that both i and sibs[i] are paired in bv
                                pairs <- rbind(pairs, c(ints[sibs[i,1]],ints[sibs[i,2]]))
                                bv[sibs[i,1]] <- 1
                                bv[sibs[i,2]] <- 1
                                }
                        }
        }
        # get rid of the first row zero vector
        pairs <- pairs[-1,]
        pairs
}

# Takes a row vector. Return true if the vector is the zero vector
# The vectors are strictly positive, so sum == 0 iff vec is the
# zero vector
isZero <- function(vec){
        sum(vec) == 0
        }

# Takes a matrix M where each row represents the intensity values of a pair of
# sibling nuclei, and calculates nu, the binomial assumption intensity to
# concentration estimator
binEstimate <- function(M){

        # remove all zero vector rows
        M <- M[!apply(M,1,isZero),]
        # first, generate a vector of differences squared
        diffs <- M[,1] - M[,2]
        diffsSq <- diffs^2
        # next, generate a vector of sums
        sums <- M[,1] + M[,2]
        # finally, divide diffsSq by sums and sum the resulting vector to get \hat(k)
        khat <- median(diffsSq / sums)
}

# Using Hmisc's latex() function, create a latex file containing a nicely formatted table
# for the results.
# Requires: all results wrapped up in a nice matrix.
# Params:
# tableName - Title for the table
# fileName - name of the resulting .tex file
# rowGroups - vector of lables for each group (of resMatrix)
# rGparts - vector indicating how resMatrix is to be partitioned by rows
# resMatrix - matrix with rows labled for embryo, columns for TF
makeTable <- function(tableName,fileName,rowGroups,rGparts,resMatrix,aval=FALSE){
```

```
# get current working dir
curWD <- getwd()

# set current working dir to ~/thesis/tex
splitDir <- unlist(strsplit(curWD,'thesis'))
setwd(paste(splitDir[1],'thesis/tex/',sep=''))

# write the matrix to a file
# need to remove wraps (add explicit wraps)
cell.format <- matrix(rep("",nrow(resMatrix) * ncol(resMatrix)), ncol = 3)
latex(resMatrix, title = tableName, file = fileName, append=aval, colnamesTexCmd = "itshape",
rownamesTexCmd = "bfseries", cgroupTexCmd = "color{green}", rgroupTexCmd = "palatino",
cellTexCmds = cell.format, numeric.dollar = FALSE, rgroup = rowGroups, n.rgroup = rGparts,
ctable = TRUE, label = "k_estimates", caption = "K_Estimates")

# return wd to the former working directory
setwd(curWD)
}
```

Appendix C

Scripts for poisson method experiments

C.1 R script to load data

Listing C.4: R script to load the data

```
### Loads the data in ~/thesis/textdata/late
### into the current R workspace
### Meant to be run in R from the working directory <whatever>/thesis/scripts/

#save the current directory
curDir <- getwd()
splitDir <- unlist(strsplit(curDir,'thesis'))
setwd(paste(splitDir[1],'thesis/textdata/late/',sep=''))

# Load the ECB data
ab11 <- read.table(file="ab11.txt",col.names=c("num","AP","DV","Eve","Cad","Bcd"),skip=5)
cb15 <- read.table(file="cb15.txt",col.names=c("num","AP","DV","Eve","Cad","Bcd"),skip=5)
ac10 <- read.table(file="ac10.txt",col.names=c("num","AP","DV","Eve","Cad","Bcd"),skip=5)

# Transform CEB data into ECB data
CEBbd5 <- read.table(file="bd5.txt",col.names=c("num","AP","DV","Cad","Eve","Bcd"),skip=5)
bd5 <- CEBbd5[,c(1,2,3,5,4,6)]
rm(CEBbd5)


# Load the EKrH data
kf9 <- read.table(file="kf9.txt",col.names=c("num","AP","DV","Eve","Kr","Hun"),skip=5)
ba3 <- read.table(file="ba3.txt",col.names=c("num","AP","DV","Eve","Kr","Hun"),skip=5)
rf11 <- read.table(file="rf11.txt",col.names=c("num","AP","DV","Eve","Kr","Hun"),skip=5)
rf6 <- read.table(file="rf6.txt",col.names=c("num","AP","DV","Eve","Kr","Hun"),skip=5)

# Load the EKnH data
dm14 <- read.table(file="dm14.txt",col.names=c("num","AP","DV","Eve","Kn","Hun"),skip=5)
tn2 <- read.table(file="tn2.txt",col.names=c("num","AP","DV","Eve","Kn","Hun"),skip=5)
hne8 <- read.table(file="hne8.txt",col.names=c("num","AP","DV","Eve","Kn","Hun"),skip=5)
fq4 <- read.table(file="fq4.txt",col.names=c("num","AP","DV","Eve","Kn","Hun"),skip=5)

# Load the new ECB/EKrB data
```

```
dq2 <- read.table(file="dq2.txt",col.names=c("num","AP","DV","Eve","Cad","Bcd"),skip=5)
ms14 <- read.table(file="ms14.txt",col.names=c("num","AP","DV","Eve","Kr","Bcd"),skip=5)
ms36 <- read.table(file="ms36.txt",col.names=c("num","AP","DV","Eve","Kr","Bcd"),skip=5)
tu7 <- read.table(file="tu7.txt",col.names=c("num","AP","DV","Cad","Eve","Bcd"),skip=5)
tu7 <- tu7[,c(1,2,3,5,4,6)]

#restore current working directory. clean up temporary vars.
setwd(curDir)
rm(curDir,splitDir)
```

C.2 R script to perform the experiments

Listing C.5: R script to perform the Poisson experiments

```
## Experiment script file to generate the nu poisson estimates for all embryos. The results are
#  returned in a matrix called 'results'.
############ Boxes for a given embryo are constructed, statistics are computed and tabled
#
# Embryos currently used:
#
# ab11, hne8, rf6, ac10, dm14, kf9, tn2, ba3, dq2, ms14, tu7, bd5, ms36, fq4, rf11
#
# Embryos should be organized into groups based on like channels: ECB, EKrB, EKrH, etc.
#
# For each embryo group, loop through the embryos
#       for each channel within each embryo
#               - compute the window estimates
#               - compute the statistics
#               - record the estimate in the results matrix
#
# Embryo is indicated by 'embryo', cycling through the embryos listed above
# Channel is indicated by 'ch'
# Embryo name is indicated by 'embname'

# Compute two sets of statistics for each box
#       1) var(Oi) / mean(Oi)
# Each box is to be (4*(i-1) to i*5) by (40 to 60)


# Define temporary vars

hatpstats <- rep(NA,20)
results <- matrix(0,nrow=1,ncol=3)
colnames(results) <- c("Eve","Cad,Kr,Kn","Bcd,Hun")
channels <- c()

# generate results matrix
eCb <- c("dq2","tu7","bd5","ab11","cb15","ac10")
eKrB <- c("ms14","ms36")
eKnH <- c("dm14","tn2","hne8","fq4")
eKrH <- c("kf9","ba3","rf11","rf6")

emblist <- c(eCb,eKrB,eKnH,eKrH)
for (j in 1:length(emblist)){
        for (k in 4:6){
                results[j,k-3] <- tabulate(emblist[j],colnames(get(emblist[j]))[k])
                channels <- c(channels,colnames(get(emblist[j]))[k])
        }
        # add next row here unless this is the last embryo.
```

```
        if (j < length(emblist)) {
                results <- rbind(results,c(0,0,0))
        }
}

# Set up column names, row names for results
rownames(results) <- emblist

# remove temporary vars
rm(hatpstats,emblist,channels,j,k,eCb.eKrB,eKnH,eKrH)
```

C.3 R script to generate results

Listing C.6: R script to generate the Poisson results table

```
### Generate a nice looking results matrix for the poisson
### estimates.

# Get results matrix by sourcing 'expLate.R' first!
if(!exists("results")){stop("No results matrix.  Run one of the late experiment scripts first.")}

if(!exists("rowGroups")){stop("rowGroups is not defined (embryos by channel)")}

if(!exists("rGparts")){stop("rGparts is not defined (correspond with rowGroups)")}


# Merge results into a data frame
df <- data.frame(results,stringsAsFactors=FALSE)

# Format data frame to 5 significant digits, call latex() on the data frame
### Fix so that first column doesn't appear!

library(Hmisc)
df <- format.df(df,dec=4)

# get current working dir
curWD <- getwd()

# set current working dir to ~/thesis/tex
splitDir <- unlist(strsplit(curWD,'thesis'))
setwd(paste(splitDir[1],'thesis/tex/',sep=''))

# make the table
# rowGroups - vector of lables for each group (of resMatrix)
# rGparts - vector indicating how resMatrix is to be partitioned by rows
# set rgroup to rowGroups, n.rgroup to rGparts when I figure out the ordering
#rowGroups <- c("Eve - Cad - Bcd","Eve - Kr - Bcd","Eve - Kn - Hun","Eve - Kr - Hun")
#rGparts <- c(length(eCb), length(eKrB), length(eKnH), length(eKrH))

latex(df, title = "Embryos", file = "poissonEsts.tex", colnamesTexCmd = "itshape",
rownamesTexCmd = "bfseries", cgroupTexCmd = "color{green}", rgroupTexCmd = "palatino",
numeric.dollar = FALSE, ctable = TRUE, label = "Nu_Estimates", caption = "Nu_Estimates",
rgroup = rowGroups, n.rgroup = rGparts)

# return wd to the former working directory
setwd(curWD)

# remove temporary vars
rm(emblist,df,channels,j,k,curWD,splitDir,eCb,eKrB,eKnH,eKrH,rowGroups,rGparts)
```

C.4 R functions

Listing C.7: R functions for the Poisson estimator scripts

```
##########################################
########   functions needed to generate poisson estimates

# Takes a row vector. Return true if the vector is the zero vector
# The vectors are strictly positive, so sum == 0 iff vec is the
# zero vector
isZero <- function(vec){
        sum(vec) == 0
        }

# Using Hmisc's latex() function, create a latex file containing a nicely formatted table
# for the results.
# Requires: all results wrapped up in a nice matrix.
# Params:
# tableName - Title for the table
# fileName - name of the resulting .tex file
# rowGroups - vector of lables for each group (of resMatrix)
# rGparts - vector indicating how resMatrix is to be partitioned by rows
# resMatrix - matrix with rows labled for embryo, columns for TF
makeTable <- function(tableName,fileName,rowGroups,rGparts,resMatrix,aval=FALSE){

# get current working dir
curWD <- getwd()

# set current working dir to ~/thesis/tex
splitDir <- unlist(strsplit(curWD,'thesis'))
setwd(paste(splitDir[1],'thesis/tex/',sep=''))

# write the matrix to a file
cell.format <- matrix(rep("",nrow(resMatrix) * ncol(resMatrix)), ncol = 3)
latex(resMatrix, title = tableName, file = fileName, append=aval, colnamesTexCmd = "itshape",
rownamesTexCmd = "bfseries", cgroupTexCmd = "color{green}", rgroupTexCmd = "palatino",
cellTexCmds = cell.format, numeric.dollar = FALSE, rgroup = rowGroups, n.rgroup = rGparts,
ctable = TRUE, label = "k_estimates", caption = "$\nu$_Estimates")

# return wd to the former working directory
setwd(curWD)
}

# Generate hat(O_i) values, with intercept term.
obsHats <- function(data, coefs){
        hats <- rep(0,dim(data)[1])
        for (i in 1:length(hats)){
                hats[i] <- coefs["AP"]*data[i,"AP"] + coefs["DV"]*data[i,"DV"] + coefs["(Intercept)"]
        }
        hats
}

# Generate hat(O_i) values, with intercept term for EVE.
obsEveHats <- function(data,coefs){
        hats <- rep(0,dim(data)[1])
        for (i in 1:length(hats)){
                hats[i] <- coefs["AP"]*data[i,"AP"] + coefs["DV"]*data[i,"DV"] + coefs["(Intercept)"]
                coefs["AP:DV"]*data[i,"AP"]*data[i,"DV"] + coefs["I(AP^2)"]*(data[i,"AP"])^2 +
                coefs["I(DV^2)"]*(data[i,"DV"])^2
                }
```

```
        hats
        }

# For a given embryo / channel, compute the table statistics
##### Add code to handle missing / empty subsets in the boxes (if dim(box)[1] > 0)
tabulate <- function(embname, ch){
        for (i in 1:length(hatpstats)){
                box <- subset(get(embname), AP > 5*(i-1) & AP < i*5 & DV > 40 & DV < 60,
                select = c("AP","DV",ch))
                if(dim(box)[1] > 0){     # for all non-empty boxes
                        if(tolower(toupper(ch)) == "eve"){
                                form <- as.formula(paste(ch," ~ ","I(AP^2) + I(DV^2) + AP:DV + AP + DV
                                coefs <- coefficients(lm(form,box))
                                hats <- obsEveHats(box,coefs)
                        }
                        else {
                                form <- as.formula(paste(ch," ~ ","AP + DV"))
                                coefs <- coefficients(lm(form,box))
                                hats <- obsHats(box,coefs)
                        }
                        newvar <- (1/length(hats)) * sum((box[,ch] - hats)^2)
                        hatpstats[i] <- newvar / mean(hats)
                        # hatpstats[i] <- mean( (box[,ch] - hats)^2 / hats)
                }
        }
        # Return the mean of the corrected estimates
        mean(hatpstats,na.rm = TRUE)
}


# Takes an embryo, partition size
# Returns a matrix containing the conditional poisson estimates
# one row per partition. All partitions hopefully contain some nucleii!
cpEstimate <- function(embryo, size){
        min <- min(embryo[,2])
        max <- max(embryo[,2])
        parts <- seq(min,max,(max - min)/size)
        estMat <- matrix(0,nrow=size,ncol=3)
        colnames(estMat) <- c(colnames(embryo)[4:6])
        for (i in 1:size){
                slice <- embryo[,2] > parts[i] & embryo[,2] < parts[i+1]
                vals <- embryo[slice,4:6]
                estMat[i,] <- apply(vals,2,var) / apply(vals,2,mean)
        }
        estMat
 }

# Takes an embryo, number of neighbours
# Returns a matrix containing the class conditional poisson estimates
# one row per class.
clpEstimate <- function(embryo,k,C=c(NA)){
        if (is.na(C)){
                C <- intensityClass(embryo,k,2)
        }
        size <- dim(C)[2]
        estMat <- matrix(0,nrow=size,ncol=3)
        colnames(estMat) <- c(colnames(embryo)[4:6])
        rownames(estMat) <- c("low","med","high")
        for (i in 1:size){
                # isolate values by class, remove NA values
```

```
                chOneVals <- embryo[,4][C[,1] == i]
                chOneVals <- chOneVals[!is.na(chOneVals)]
                chTwoVals <- embryo[,5][C[,2] == i]
                chTwoVals <- chTwoVals[!is.na(chTwoVals)]
                chThrVals <- embryo[,6][C[,3] == i]
                chThrVals <- chThrVals[!is.na(chThrVals)]
                if (length(chOneVals) > 10){
                        chOneEst <- var(chOneVals) / mean(chOneVals)
                } else {
                        chTwoEst <- -1
                }
                if (length(chTwoVals) > 10){
                        chTwoEst <- var(chTwoVals) / mean(chTwoVals)
                } else {
                        chTwoEst <- -1
                }
                if (length(chThrVals) > 10){
                        chThrEst <- var(chThrVals) / mean(chThrVals)
                } else {
                        chThrEst <- -1
                }
                estMat[i,] <- c(chOneEst,chTwoEst,chThrEst)
        }
        estMat
}

# Return the kNN classification of the values for an embryo
# Takes one embryo matrix, number of neighbours k, and number of breaks br
# Returns a 3xN matrix of classes for each nucleus in the embryo
# Currently hardcoded for br = 2 (high, medium, low regions)
intensityClass <- function(emb, kval, br){
        library(class)          # need knn.cv from classification library
        classMat <- matrix(0,nrow=nrow(emb),3)
        for (i in 4:6){
                classMat[,i-3] <- threshold(emb[,i],br)
                # perform knn classification on this channel
                classMat[,i-3] <- as.vector(knn(emb[,2:3],emb[,2:3],classMat[,i-3],k=kval, l=kval-2))
        }
        classMat
}

# Takes an embryo, number of neighbours
# Returns a matrix (|classes| x 9) where each row is one class of values
# and there are three sets of three columns.  Within each set, the
# first, second and third column represent E[log(O_i)], Var[log(O_i)],
# and E[(log(O_i))^3] for each channel.
momentMatrix <- function(embryo,C){
        size <- dim(C)[2]
        momentMat <- matrix(-1,nrow=size,ncol=9)
        colnames(momentMat) <- rep(c("E[log(O_i)]","Var[log(O_i)]","E[(log(O_i))^3]"),3)
        rownames(momentMat) <- c("low","med","high")
        for (i in 1:size){
                # isolate values by class, remove NA values
                chOneVals <- embryo[,4][C[,1] == i]
                chOneVals <- chOneVals[!is.na(chOneVals)]
                chTwoVals <- embryo[,5][C[,2] == i]
                chTwoVals <- chTwoVals[!is.na(chTwoVals)]
                chThrVals <- embryo[,6][C[,3] == i]
                chThrVals <- chThrVals[!is.na(chThrVals)]
```

```
                # perform log transformation
                logChOne <- log(chOneVals)
                logChTwo <- log(chTwoVals)
                logChThr <- log(chThrVals)

                # fill in all 9 columns, one set of 3 at a time
                momentMat[i,c(1,2,3)] <- c(mean(logChOne),var(logChOne),mean(logChOne^3))
                momentMat[i,c(4,5,6)] <- c(mean(logChTwo),var(logChTwo),mean(logChTwo^3))
                momentMat[i,c(7,8,9)] <- c(mean(logChThr),var(logChThr),mean(logChThr^3))
        }
        momentMat
}

# Classification of points that does not include a call to knn
# Takes an embryo, a number of neighbours to consider (k)
# and calculates the regression of each nucleus' class based on the
# discretization of the k-NN predicted value.
# Outputs the class matrix.
initIntensityClass <- function(emb, kval, br, Nmat=c(NA)){
        library(class)
        N <- nrow(emb)
        classMat <- matrix(-1,nrow=N,3)
        if(is.na(Nmat)) { Nmat <- getNmat(emb,kval) }
        Wmat <- getWmat(emb,kval,Nmat)

        # Calculate y_hat, y and assign value for class matrix
        for (i in 1:3){
                chVals <- emb[,i+3]
                B <- matrix(0,nrow=N,ncol=kval)
                # generate the values matrix
                for (j in 1:N) {
                        B[j,] <- chVals[Nmat[j,]]
                }

                if(any(B == 0)) { stop("Zeros in the B matrix") }
                # apply thresholding to the weighted expression estimate vector
                yhat <- threshold(apply(Wmat * B,1,sum),br)
                y <- threshold(chVals,br)
                for (j in 1:N){
                        if (y[j] == yhat[j]) {
                                classMat[j,i] <- yhat[j]
                        } else {
                                classMat[j,i] <- NA
                        }
                }
        }
        classMat
}


# Calculate a nearest neighbour index matrix
getNmat <- function(emb,kval){
        N <- nrow(emb)
        Nmat <- matrix(0,nrow=N,ncol=kval)
        # Calculate neighbour matrix, distance matrix
        for (i in 1:N){
                Nmat[i,] <- nearest(emb[,2:3],i,kval)
        }
        Nmat
```

```
}

# Calculate a nearest neighbour weight matrix
getWmat <- function(emb,kval,Nmat){
        Dmat <- matrix(0,nrow=nrow(emb),ncol=kval)
        # Calculate neighbour matrix, distance matrix
        for (i in 1:nrow(emb)){
                Dmat[i,] <- apply(emb[Nmat[i,],2:3],1,el2norm,y=emb[i,2:3])
        }
        # Calculate the weight matrix
        Wmat <- Dmat^(-2) / apply(Dmat^(-2),1,sum)
        Wmat
}

# Takes a vector of levels and a smaller vector of breaks
# Returns a vector of classes after apply a threshold
threshold <- function(levels,br){
        breaks <- hist(levels, breaks = br, plot = FALSE)$breaks
        classVec <- rep(0,length(levels))
        chLow <- levels > breaks[1] & levels < breaks[2]
        chMed <- levels > breaks[2] & levels < breaks[3]
        chHigh <- levels > breaks[3] & levels < breaks[4]
        # write the class number to the class matrix
        classVec[chLow] <- 1
        classVec[chMed] <- 2
        classVec[chHigh] <- 3
        classVec
}

## Find k nearest neighbors of X[n, ] in the data frame
## or matrix X, utilizing function knn1 k-times.
nearest <- function (X, n, k){
    N <- nrow(X)
    # inds contains the indices of nearest neighbors
    inds <- c(n); i <- 0
    while (i < k) {
        # use knn1 for one index...
        j <- as.integer(knn1(X[-inds, ], X[n, ], 1:(N-length(inds))))
        # ...and change to true index of neighbor
        inds <- c(inds, setdiff(1:N, inds)[j])
        i <- i+1
    }
    # return nearest neighbor indices (without n, of course)
    return(inds[-1])
}

# Plot the class values for one channel of an embryo
# X is the Nx3 position and class of each nucleus within the embryo
# title is the title of the scatter plot
plotIntensityClass <- function(X,title){
        # set up axes
        plot(X[,1] , X[,2] , xlab="AP position",ylab="DV position",main=title,type = "n")

        # collect low, medium, high point sets. Is using cbind over rbind messing
        # things up?
        lowpts <- cbind(X[,1][X[,3] == 1],X[,2][X[,3] == 1])
        medpts <- cbind(X[,1][X[,3] == 2],X[,2][X[,3] == 2])
        highpts <- cbind(X[,1][X[,3] == 3],X[,2][X[,3] == 3])

        # plot them on the graph
```

```
        points(lowpts, pch=".")
        points(medpts, pch=2)
        points(highpts, pch=3)
}
```