Small Blob Detection in Medical Images

by

Min Zhang

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2015 by the
Graduate Supervisory Committee:

Teresa Wu, Chair
Jing Li
William Pavlicek
Ronald Askin

ARIZONA STATE UNIVERSITY

May 2015

ABSTRACT

Recent advances in medical imaging technology have greatly enhanced imaging based diagnosis which requires computational effective and accurate algorithms to process the images (e.g., measure the objects) for quantitative assessment. In this dissertation, one type of imaging objects is of interest: small blobs. Example small blob objects are cells in histopathology images, small breast lesions in ultrasound images, glomeruli in kidney MR images etc. This problem is particularly challenging because the small blobs often have inhomogeneous intensity distribution and indistinct boundary against the background.

This research develops a generalized four-phased system for small blob detections. The system includes (1) raw image transformation, (2) Hessian pre-segmentation, (3) feature extraction and (4) unsupervised clustering for post-pruning. First, detecting blobs from 2D images is studied where a Hessian-based Laplacian of Gaussian (HLoG) detector is proposed. Using the scale space theory as foundation, the image is smoothed via LoG. Hessian analysis is then launched to identify the single optimal scale based on which a pre-segmentation is conducted. Novel Regional features are extracted from pre-segmented blob candidates and fed to Variational Bayesian Gaussian Mixture Models (VBGMM) for post pruning. Sixteen cell histology images and two hundred cell fluorescent images are tested to demonstrate the performances of HLoG. Next, as an extension, Hessian-based Difference of Gaussians (HDoG) is proposed which is capable to identify the small blobs from 3D images. Specifically, kidney glomeruli segmentation from 3D MRI (6 rats, 3 humans) is investigated. The experimental results show that HDoG has the potential to automatically detect glomeruli, enabling new measurements of

renal microstructures and pathology in preclinical and clinical studies. Realizing the computation time is a key factor impacting the clinical adoption, the last phase of this research is to investigate the data reduction technique for VBGMM in HDoG to handle large-scale datasets. A new coreset algorithm is developed for variational Bayesian mixture models. Using the same MRI dataset, it is observed that the four-phased system with coreset-VBGMM has similar performance as using the full dataset but about 20 times faster.

DEDICATION

To my parents, my wife, and my lovely daughter.

ACKNOWLEDGMENTS

First I would like to gratefully and sincerely thank my advisor - Dr. Teresa Wu, for whose recognition and appreciation I owe a debt of. Dr. Wu's mentoring and guidance has made my Ph.D. study a very rich and rewarding journey.

Second, I want to thank Dr. Pavlicek for being a very nice, encouraging and supportive supervisor in Mayo Clinic throughout the years. His supporting for his colleagues, hard-working spirit will keep encouraging me in my future career development.

Third, I would like to thank Dr. Jing Li and Dr. Bennett, who offered me a great help and valuable suggestions for my Ph.D. research and job searching. And I also want to thank Dr. Ronald Askin, who gave me valuable advices during my comprehensive exam to complete my study.

In addition, I would also like to thank my lab mates and Mayo colleagues. I am grateful to have met so many wonderful people during my Ph.D. study.

Finally, and most importantly, I would like to thank my wife Jiayi for her endless support, encouragement and love.

TABLE OF CONTENTS

Page

LIST OF TABLES vii

LIST OF FIGURES x

CHAPTER

1 INTRODUCTION 1

1.1 Background 1

1.2 Research Objective and Contributions 2

1.3 Dissertation Organization 3

2 SMALL BLOBS DETECTION IN 2D MEDICAL IMAGES 5

2.1 Introduction 6

2.2 Hessian-based Laplacian of Gaussian Detector 11

2.3 Comparison Experiments 31

2.4 Conclusion 40

3 DETECTING A LARGE-SCALE OF SMALL BLOBS IN KIDNEY 3D MR IMAGES 42

3.1 Introduction 43

3.2 Hessian-based Difference of Gaussian Detector 48

3.3 Experiments on 3D MR Images 61

3.4 Conclusion 68

4 COMPUTATIONAL EFFICIENT VARIATIONAL BAYESIAN GAUSSIAN MIXTURE MODELS VIA CORESET 70

CHAPTER Page

4.1 Introduction 70

4.2 Coreset and Construction 73

4.3 Weighted Variational Bayesian Mixture Models via Coresets 78

4.4 Experimental Validation of Weighted VBGMM via Coresets 84

4.5 Discussion 94

4.6 Conclusion 96

5 CONCLUSIONS AND FUTURE WORK 97

5.1 Summary 97

5.2 Discussion and Future Work 98

REFERENCES 100

APPENDIX

A HESSIAN PRESEGMENTATION ON GAUSSIAN BLOBS 104

B EXPERIMENTAL RESULTS OF CORESETS ON RAT AND HUMAN DATASETS 112

LIST OF TABLES

Table Page

1 Comparison Results of HLoG, gLoG, Radial-symmetry and LoG on 15 Pathologic Images without Post-pruning Process 19

2 ANOVA on Detectors without Post-pruning using Tukey's HSD Pairwise Test on 15 Pathologic Images with 0.05 Significance Level 20

3 Summary of Features over Blob Candidate T 26

4 Evaluation of Different Feature Sets on 15 Pathologic Images (Mean and Standard Deviation of the Measures on Precision, Recall, F-score) 27

5 Statistical Groups based on ANOVA Tukey's HSD Pairwise Test 28

6 Detail Steps of HLoG 30

7 Comparison Results of Complete Version of HLoG, gLoG, Radial-Symmetry and LoG on 15 Pathologic Images. F-score Metric is Highlighted in Gray since It Provides a Comprehensive Measurement to Evaluate the Performance 32

8 ANOVA using Tukey's HSD Pairwise Test on 15 Pathologic Images with 0.05 Significance Level 33

9 ANOVA using Tukey's HSD Pairwise Test on 200 Fluorescent Images with 0.05 Significance Level 36

10 Comparing HLoG, gLoG, and LoG Performance on Dataset 1: 15 2D Pathologic Images (the Normalizing Factor $\boldsymbol{\gamma}$ =2 based on Prior Experiments. Detailed of the Parameter Setting Please Refer to (Zhang et al. 2014)). Paired T-tests were Performed at a Significance Level of 0.05. Symbols in Cell Means: + Statistically Outperformed; = Statistically Comparable; - Statistically Underperformed 58

Table Page

11 Comparing HLoG, gLoG, and LoG on Dataset 2: 200 2D Fluorescent Images (the Normalizing Factor γ =1 based on Prior Experiments. Details of the Parameter Settings are Described in (Zhang et al. 2014)). Paired T-tests were Performed at a Significance Level of 0.05. + Statistically Outperformed; = Statistically Comparable; - Statistically Underperformed 60

12 Glomerular Counts for Six Rat Kidneys using the HDoG, Acid Maceration, and Stereology Methods (γ =2, Intel Xeon 2.0 GHz CPU and 32 GB of Memory). .. 62

13 Glomerular Counts for Three Human Kidneys using the HDoG and Stereology Techniques (γ =2, Intel Xeon 2.0 GHz CPU and 32 GB of Memory). 64

14 Coreset Construction via Adaptive Sampling (Feldman et al. 2011) 75

15 Proposed Coreset Construction for Variational Bayesian Framework 77

16 Variational Inference for Gaussian Mixture Models (Bishop 2006) 81

17 Weighted VBGMM via Coresets Clustering 83

18 Comparison of VBGMM on Full Dataset, Coreset and Uniform Sampling (based on 5000 Sample Size) on Rat Kidney Data 85

19 Comparison of VBGMM on Full Dataset, Coreset and Uniform Sampling (based on 5000 Sample Size) on Human Kidney Data 90

20 Comparison Results of Feldman Coreset and Our Coreset on Rat 1 Dataset 95

21 Full Comparison of VBGMM on Full Dataset, Coreset and Uniform Sampling on Rat 1 Kidney Data 113

22 Full Comparison of VBGMM on Full Dataset, Coreset and Uniform Sampling on Rat 2 Kidney Data 114

Table Page

23 Full Comparison of VBGMM on Full Dataset, Coreset and Uniform Sampling on Rat 3 Kidney Data 116

24 Full Comparison of VBGMM on Full Dataset, Coreset and Uniform Sampling on Rat 4 Kidney Data 118

25 Full Comparison of VBGMM on Full Dataset, Coreset and Uniform Sampling on Rat 5 Kidney Data 120

26 Full Comparison of VBGMM on Full Dataset, Coreset and Uniform Sampling on Rat 6 Kidney Data 122

27 Full Comparison of VBGMM on Full Dataset, Coreset and Uniform Sampling on Human 1 Kidney Data 124

28 Full Comparison of VBGMM on Full Dataset, Coreset and Uniform Sampling on Human 2 Kidney Data 126

29 Full Comparison of VBGMM on Full Dataset, Coreset and Uniform Sampling on Human 3 Kidney Data 128

LIST OF FIGURES

Figure Page

1 Transformed-bright Blob Has a Negative Definite Hessian in Normalized LoG Space. (A) Dark Blob in the Raw Image. (B) Transformed- bright Blob after Normalized LoG Transformation from (A). (C) Region of Pixels having Negative Definite Hessian in (B) is Contoured in Green over (B). (D) Region of Pixels having Negative Definite Hessian in (B) is Contoured in Green over Original Image (A). As Seen in (D), Negative Definite Hessian Outlines the Blob over (A). The Irregular Shape of the Blob is Clearly Delineated. 15

2 Hessian Pre-segmentation Results on Selected Pathologic Cell Image. (A) Raw Image. (B) Auto Scale Selection: Section $t = t_{best} = 7$ is Selected. (C) LoG Transformed Image. (D) Hessian Pre-segmentation Result: Purple Mask Shows the Shapes Retrieved by Hessian Pre-segmentation Method. 21

3 Blob Identification Result from Part of Figure 2 (A). (A) Pre-segmentation Result. (B) Final Identification Result. The Purple Regions are Blob Candidates and Their Centroids are Marked as Green Circle, While the Centroids of Ground Truth Data are Marked as Yellow Cross. 29

4 HLOG for Blob Detection 30

5 Comparison Results of Full Version of HLoG, gLoG, Radial-Symmetry and LoG on 200 Fluoro Images. The Error Bar Indicates the Standard Deviation of the Corresponding Measure across 200 Images. 35

Figure Page

6 Detection Results on Selected Fluorescent Image. (A) Detection Result by HLoG. (B) Detection Result by gLoG Algorithm. (C) Detection Result by Radial-Symmetry Algorithm. (D) Detection Result by LoG..... 37

7 F-Score of HLoG, gLoG, Radial-Symmetry and LoG on 15 Pathological Images at Different Parameter d..... 38

8 F-Score of HLoG, gLoG, Radial-Symmetry and LoG on 200 Pathological Images at Different Parameter d..... 38

9 Flowchart of the Hessian-based Difference of Gaussians (HDoG) Analysis..... 48

10 Glomerular Segmentation Results from 3D MR Images of Rat Kidneys (Selected Slices Presented). (A–C) Slice 100 for Rats CF1, CF2, and CF3. (D–F) Slice 150 for Rats CF4, CF5, and CF6. (G–I) Identification Results for (A–C), Respectively. Identified Glomeruli are Contoured in Green. (J–L) Identification Results for (D–F), Respectively, where Identified Glomeruli are Contoured in Green..... 63

11 Glomerular Segmentation Results for 3D MR Images of Human Kidneys (Selected Slices): (A–C) Original Slice 100 for Human CF1, CF2, and CF3 Kidneys. (D–F) Slice 500 for Human CF1, CF2, CF3 Kidneys. (G–I) Identification Results for (A–C), Respectively, where Identified Glomeruli are Contoured in Green. (J–L) Identification Results for (D–F), respectively, where Identified Glomeruli are Contoured in Green..... 65

12 Intensity Frequency Histograms of Glomeruli against Whole Kidney Image from: (A) Human CF1 (B) Human CF2 (C) Human CF3. Frequency Range is $[0, 0.6]$

and the Intensity Range is $[0,1]$ in the Figure. Vertical Lines Indicate the Modes of the Intensity Distribution. 66

13 Frequency Histograms of Average Intensity, Regional Blobness and Regional Flatness for Human CF1, CF2 and CF3 Kidney 3D MR Images. (A)-(C) True Glomerular Cluster Frequency Histograms for Human CF1, CF2 and CF3 respectively. (D)-(F) Non-glomerular Cluster Frequency Histograms for Human CF1, CF2 and CF3 respectively. Frequency Range was $[0, 0.5]$ and the *x*-axis Range was $[0,1]$ in the Figure. 67

14 Glomerular Segmentation Results for 3D MR Images of Rat CF1 Kidney and Human CF1 Kidney (Part of the Slice on Figure 11): (A) Part of Slice for Rat CF1. (B) Identification Results for (A), where Identified Glomeruli Centers are Marked in Red-cross. (C) Part of Slice for Human CF1. (D) Identification Results for (C), where Identified Glomeruli Centers are Marked in Red-cross. Circles Show the Error of Missed Detection while the Rectangles Show the Error of False Positive Detection 68

15 Coreset Construction Comparison. (A) Full Dataset (B) Coreset Construction using (Feldman et al. 2011) (C) Coreset Construction using Proposed Algorithm. Radius of Point Reflects the Probability of the Data Point to be Sampled..... 77

16 Directed Acyclic Graph (DAG) for VBGMM..... 79

17 Varional Lower Bound Comparison on Weighted VBGMM via Coresets, VBGMM via Uniform Samping and VBGMM via Full Data..... 85

Figure Page

18 Glomerular Identification Results for 3D MR Images of Rat Kidneys using Coresets (5000 Samples): (A–C) Segmentation Results from Slice 100 for Rat CF1, CF2, and CF3 Kidneys using Full Dataset. (D–F) Segmentation Results from Slice 100 for Rat CF1, CF2, and CF3 Kidneys using Coreset. (G–I) Segmentation Results from Slice 150 for Rat CF4, CF5, and CF6 Kidneys using Full Dataset. (J–L) Segmentation Results from Slice 150 for Rat CF4, CF5, and CF6 Kidneys using Coreset. 88

19 Varional Lower Bound Comparison on Weighted VBGMM via Coresets, VBGMM via Uniform Samping and VBGMM via Full Data on Human Kidney Dataset..... 90

20 Glomerular Identification Results for 3D MR Images of Human Kidneys Using Coresets (5000 Samples): (A–C) Segmentation Results from Slice 100 for Human CF1, CF2, and CF3 Kidneys using Full Dataset. (D–F) Segmentation Results from Slice 100 for Human CF1, CF2, and CF3 Kidneys using Coreset. (G–I) Segmentation Results from Slice 500 for Human CF1, CF2, and CF3 Kidneys using Full Dataset. (J–L) Segmentation Results from Slice 500 for Human CF1, CF2, and CF3 Kidneys using Coreset. 93

21 Data Samples from Rat 1 Dataset. (A) Full Dataset (B) Coreset using (Feldman et al. 2011) (C) Coreset using Our Method. 94

22 Framework for Small Blob Detection..... 98

23 Three Roots for *F(a)* in Case 1 107

24 One Root for *F(a)* in Case 2 107

Figure Page

25 Blob Transformation. Left to Right: Raw blob, LoG Transformed Blob, Purple Area of Positive Definite Hessian..... 109

CHAPTER 1

INTRODUCTION

1.1 Background

Structural objects exist everywhere in medical image analysis, like cells or nuclei in histopathology images or fluoroscopic images, renal stones in computed tomography (CT) images, cerebral blood vessels in magnetic resonance (MR) images, breast lesions in ultrasound images, glomeruli in kidney MR Images etc. Such structures have various shapes across the images: some are blob-like shape, some are curvilinear, and some are irregular. To identify those structures is of importance in medical image research fields for disease diagnosis and is very challenging due to their variant shapes as well as their inhomogeneous image intensity distributions. In addition, the computational cost problem may arise and become another main concern when the size and/or the dimension of images grow.

In this research, the small structure identification problem, as a sub problem of structural object segmentation is of interest. Some example applications are cell or nuclei segmentation problem (Al-Kofahi et al. 2010, Bergeest et al. 2012, Zhang et al. 2014), glomerular segmentation problem (Beeman et al. 2014, Zhang et al. 2015), just to name a few. Here the specific challenges are (1) the medical images may often have the poor imaging quality, which makes the small structures harder to be labelled; (2) there exists large intensity variation for the structures and (3) boundaries of the structures are often fuzzy. Please note this type of small structure identification is also known as blob detection in the field of medical image analysis and computing vision.

Extensive studies have been done on blob detection, mainly in 2D medical images. However, there is currently a lack of unsupervised computationally efficient techniques to perform fast, reliable and accurate detection on large-scale small blobs in 3D images (e.g., MRI) due to the challenges of acquisition noise, partial volume effect (the mixture of several tissue signals in a voxel) and bias field (spatial intensity inhomogeneity). Such challenges are becoming even severe for those blobs (like glomerulus in 3D MR Kidney Images) are very small and the number of blobs is very large. To fill this gap, we have developed an efficient framework to identify the large-volume small blobs on both 2D and 3D images.

1.2 Research Objective and Contributions

The objective of this research is to develop computational efficient system to detect and segment small blobs in both 2D and 3D medical images. The contributions in this dissertation are:

1. Hessian based pre-segmentation algorithm is proposed which can theoretically segment all the small blobs in images.
2. Novel efficient regional features that can characterize the local geometry properties of blobs are derived for both 2D and 3D images.
3. A novel detector termed Hessian based Laplacian of Gaussian (HLoG) is proposed for 2D blob detection.
4. An extension of HLoG, an efficient detector termed Hessian based Difference of Gaussian (HDoG) is proposed which is capable for both 2D and 3D blob detections.

5. A new coreset construction algorithm is introduced for the post-pruning algorithm in HLoG and HDoG resulting a much improved computational efficient pipeline for small blob detection.

1.3 Dissertation Organization

In Chapter 2, we focus on 2D blob detection, specifically, cell detection in 2D histopathology images and/or fluoroscopic images. We propose a novel detector termed Hessian-based Laplacian of Gaussian (HLoG) using the scale space theory as the foundation. Like most imaging detectors, an image is first smoothed via Laplacian of Gaussian (LoG). Since small blobs (e.g., cell) may in general have similar sizes, Hessian analysis is launched to identify the single optimal scale based on which a pre-segmentation is conducted. The second advantage of the Hessian process is it is capable to delineate the blobs. As a result, regional features can be retrieved. These features enable the unsupervised clustering algorithm for post pruning which shall be more robust than the traditional threshold-based post pruning commonly used in the most imaging detectors. To test the performance of the proposed HLoG, two sets of 2D grey medical images are studied. HLoG is compared against three state-of-the-art detectors: gLoG (Kong et al. 2013), Radial-Symmetry (Loy et al. 2003) and LoG (Lindeberg 1998) using precision, recall and F-score metrics. We observe HLoG has statistically outperformance over the compared detectors.

In Chapter 3, HLoG is extended to 3D blob detection, specifically 3D glomeruli detection in MRI. A modified version of HLoG, termed efficient Hessian based Difference of Gaussians (HDoG) detector is proposed. The regional features proposed in

the first phase are extended to 3D to preserve the detectability on 3D blobs. Dataset consisting of six rat 3D kidney MR images, and three human 3D MR images are studied to test the applicability of HDoG for segmenting renal glomeruli. The experimental result shows that, Hessian based DoG can identify similar number of glomeruli as our maceration counts and stereology counts which have been taken as golden standard in medical research.

In Chapter 4, in order to improve the computational speed of the post-pruning algorithm, Variational Bayesian Gaussian Mixture Models (VBGMM) for HLoG and HDoG, a data reduction technique called coreset (Agarwal et al. 2005) is introduced in VBGMM. A new coreset algorithm is proposed which derives the importance of the representative data in the forms of weights to be fed into VBGMM. As a result, a modified version of VBGMM, weighted VBGMM is developed. To test the performance of this new weighted VBGMM, same two sets of 3D MR images discussed in Chapter 3: six rat kidney MR images and three human kidney MR images are evaluated. The results show while preserving the similar clustering performance, weighted VBGMM greatly reduces the computation time by about 20 times.

Chapter 5 concludes the research studies in this dissertation and future works are discussed in this chapter.

CHAPTER 2

SMALL BLOBS DETECTION IN 2D MEDICAL IMAGES

Recent advances in medical imaging technology have greatly enhanced imaging based diagnosis which requires computational effective and accurate algorithms to process the images (e.g., measure the objects) for quantitative assessment. In this research, we are interested in one type of imaging object: small blobs. Examples of small blob objects are cells in histopathology images, glomeruli in MR images, etc. This problem is particularly challenging because the small blobs often have inhomogeneous intensity distribution and an indistinct boundary against the background. Yet, in general, these blobs have similar sizes. Motived by this finding, we propose a novel detector termed Hessian-based Laplacian of Gaussian (HLoG) using the scale space theory as the foundation. Like most imaging detectors, an image is first smoothed via LoG. Hessian analysis is then launched to identify the single optimal scale based on which a pre-segmentation is conducted. The advantage of the Hessian process is it is capable of delineating the blobs. As a result, regional features can be retrieved. These features enable the unsupervised clustering algorithm for post-pruning which shall be more robust and sensitive than the traditional threshold-based post-pruning commonly used in most imaging detectors. To test the performance of the proposed HLoG, two sets of 2D grey medical images are studied. HLoG is compared against three state-of-the-art detectors: gLoG, Radial-Symmetry and LoG using precision, recall and F-score metrics. We observe that HLoG statistically outperforms the compared detectors.

2.1 Introduction

The rapid development of medical imaging technology has dramatically increased the spatial and temporal resolution, and therefore size, of clinical imaging data. Typically, image segmentation methods are used to delineate specific objects and boundaries. The derived features, such as the number, size, and shape of the objects, are then used for clinical analysis. Some examples of objects in images for clinical studies include cell nuclei in histopathology images or fluoroscopic images to quantify cytology or histology (Al-Kofahi et al. 2010, Bergeest et al. 2012, Zhang et al. 2014), cerebral blood vessels in magnetic resonance (MR) images to diagnose vascular disease (Frangi et al. 1998), brain tumors in MR images to assess treatment (Prastawa et al. 2004), breast lesions in ultrasound images to stage breast cancer lesions (Moon et al. 2013), and glomeruli in kidney MR images to characterize renal disease (Beeman et al. 2014, Beeman et al. 2011, Zhang et al. 2015). Of particular interest in this research is a common type of object which is small in structure and convex elliptic in shape. In the field of computer vision, the problem of detecting such objects is known as blob detection. Using mathematical methods, blobs with different properties such as brightness or shape can be identified against the image background.

Extensive research has proposed various blob detectors, among which one popular approach is to use local extrema in a transformed space in conjunction with a vector of derived features, (e.g., local intensity histogram or orientation histogram), to identify the blobs. In general, this type of blob detector can be categorized as an interest point detector or interest region detector. As interest point detectors, the Radial-Symmetry (Loy et al. 2003) and the Radial Gradient Transform detectors (Takacs et al.

2013) use radial symmetric space as the transformation base. One advantage is that the Radial-Symmetry and Radial Gradient Transform detectors are fast to compute and are rotationally invariant, which would be beneficial to detect radial symmetric blobs. However, for radial asymmetric blobs, these two detectors may lose their sensitivities (Kong et al. 2013). Other types of interest point detectors like SIFT (Lowe 2004), SURF (Bay et al. 2008) and BRISK (Leutenegger et al. 2011) are rooted in a scale invariant space transformation. Scale invariant features are extracted, associated with affine invariant features such as the orientation histogram to characterize the local properties in the affine space. These detectors are claimed to be affine invariant such that local structures with similar affine properties within or across images can be identified.

Note interest point detectors are developed for each pixel/voxel, ignoring the contributions of neighboring pixels/voxels to the object. Consequently, these detectors tend to be less tolerant of local noise. Motivated by the affine invariant interest point detector, interest region detectors are introduced to derive aggregated measures of a number of pixels/voxels within regions of local extrema. Some examples of interest region detectors are the Harris-Affine detector, Hessian-affine detector and Hessian-Laplace detector (Mikolajczyk et al. 2004). While these detectors are shown to be more robust to noise, they are computationally expensive to adapt the shape estimation. In addition, the massive number of local extrema necessitates careful pruning, which adds to the computational burden. It is also challenging to identify the appropriate pruning parameter, which tends to be subjective for both interest point and interest region detectors.

Although both interest point and region detectors well describe affine invariant properties of local image structures, it is observed in the study by (Kong et al. 2013) that the performance of these detectors on some blob identification problems in pathological and fluorescent images are unsatisfactory. Instead, another type of detector, Laplacian of Gaussian (LoG) detector (Lindeberg 1998), under the scale space theory (Lindeberg 1993b), has attracted great attention for the blob detection problem. Scale space theory is a formal theory that considers one image as a stack of images controlled by one parameter (scale parameter *t*), in the so-called scale space representation. The Gaussian scale space representation of an image is defined by the convolution of the image function with the Gaussian kernel, preserving important spatial properties of the imaged structures (Lindeberg 1993b). Specifically, as the scale parameter increases, the number of local minima in a dark blob does not increase, and the number of local maxima in a bright blob does not decrease. This means that neighboring blob objects will diffuse and eventually combine to be identifiable blobs at a specific scale. If a multi-scaled representation with regards to scale parameter t is constructed, there thus exists one "optimal" scale for blobs with similar sizes. Individual blobs can then be detected with corresponding scale parameters. Previous research has shown that the detector generated by applying the LoG kernels can successfully detect blobs (Lindeberg 1993a, 1998). However, the symmetric nature of the LoG detector limits its performance in rotational asymmetric blob detection and computational cost for scale adaption under multi-scale representation is expensive. Therefore, a number of LoG extensions have been proposed. For example, the Difference of Gaussian (DoG) (Lowe 2004, Mikolajczyk et al. 2004) is developed to approximate the LoG and improve computational performance. The

Generalized Laplacian of Gaussian (gLoG) (Kong et al. 2013) is proposed to extend the detection to rotational asymmetric structures by using different Gaussian kernels. The gLoG is thus able to detect general elliptical structures such as rotationally symmetric and asymmetric blobs.

All detectors reviewed above use the same workflow, pinpointing the centroid of the blob and carefully pruning to remove the overlapping local extrema. A regular ellipse with a proper radius that is associated with its scale is then superimposed on the center to approximate the blob shapes. Consequently, the derived features (e.g., the volumes of the blobs) are only estimates rather than accurate measurements defined by true boundaries. Furthermore, while scale space theory provides the foundation for complex detectors, it is based on a multi-scale representation, which may waste computing effort in optimally selecting scales when the blobs are approximately uniform in size. Uniformly sized blobs are common in a number of clinical applications, such as in detecting cells and nuclei in pathologic or fluorescent microscopic images (Al-Kofahi et al. 2010, Bergeest et al. 2012, Zhang et al. 2014), and segmenting kidney glomerulus in 3D MR images (Beeman et al. 2014, Beeman et al. 2011, Zhang et al. 2015). These images generally contain very large number of small blobs, each with a convex elliptical shape. Though "uniform size" may relax the blob recognition problem, there exist some unique challenges: (1) the blob's small size corresponds to a high spatial frequency close to that of image noise. As a result, small blob detection is sensitive to local noise; (2) the heterogeneous distribution of intensities and heterogeneous boundaries make it difficult to threshold the small blobs from the image background. We contend features derived from small imaging

objects/regions may help address the challenges. This will require the detailed delineations of the blobs instead of estimating the blob boundaries.

In this study, the convexity and elliptic shape of the blobs of interest motivates us to explore the Hessian analysis. Here we propose a novel approach, named Hessian-based Laplacian of Gaussian detector (HLoG) for small blob detection. Specifically, similar to the aforementioned detectors, a multi-scale representation is first derived using LoG for each image. Since the blobs are approximately homogeneous in size, an "optimal" scale can be identified from the Hessian analysis thus a single-scale representation can be obtained from the images. Hessian analysis is then applied to pre-segment the blob candidates. It is known that the theoretical foundation of Hessian guarantees the pre-segmentation will recognize all the true blobs, as well as some non-blob objects. This leads to the need to fine-tune the pre-segmentation results. We want to note the detectors reviewed above such as LoG, gLoG, Radial-Symmetry all employ threshold-based fine prune procedure which may be less tolerant to image noises. Fortunately, the Hessian pre-segmentation has greatly reduced data size (by filtering out most non-blob regions) and delineated the boundary of blob candidates. As a result, we can afford to extract multiple features from the small blob candidates. In addition to the eight features commonly used in the literature, we introduce three new features to measure the "blobness." After comprehensive assessment, three out of eleven significant features are selected and used in Variational Bayesian Gaussian mixture model (VBGMM) (Bishop 2006) to finalize blob detection. VBGMM is chosen here since it is an unsupervised clustering algorithm which is also tuning-free. During the LoG transformation, a dark blob is converted to a bright blob and vice versa. To avoid confusion, we define the blob

after the LoG operation as transformed-blob hereafter. This paper focuses on the dark blob (transformed-bright blob) identification. Same process applies for the bright blob (transformed-dark blob) identification.

The main contributions of this proposed HLoG detector for small blobs with similar sizes are three-fold. First, the proposed novel Hessian pre-segmentation is capable of generating blob candidate regions that theoretically enclose all the true blobs and accurately delineate the shape of imaging objects so multiple regional features can be extracted for post pruning. Second, the use of Hessian pre-segmentation enables the identification of a single optimum scale as the smoothing parameter for normalized LoG filter to greatly reduce the computational burden. Third, together with average intensity information (known from literature), two out of three new regional features, after evaluation, are introduced to characterize the local blobs to prune out the non-blob objects for improved final segmentation.

The following sections are organized as follows: Chapter 2.2 describes our method in details followed by the comparison experiments in Chapter 2.3. Conclusions are presented in Chapter 2.4. All code and results in this paper can be found at our website (http://swag.engineering.asu.edu/HLoG.htm).

2.2 Hessian-based Laplacian of Gaussian Detector

2.2.1 Normalized Laplacian of Gaussian Transformation

Given 2D image $f : R^2 \to R$, the scale space representation $L(x, y; t)$ at point (x, y) with scale parameter t is the convolution of image $f(x, y)$ with Gaussian kernel $G(x, y; t)$

$$L(x, y; t) = f(x, y) * G(x, y; t) \tag{2.1}$$

Where $*$ is the convolution operator and $G(x, y; t) = \frac{1}{(2\pi t^2)} e^{-\frac{(x^2+y^2)}{t^2}}$. The Laplacian of $L(x, y; t)$ is:

$$\nabla^2 L(x, y; t) = \nabla^2 f(x, y) * G(x, y; t) \tag{2.2}$$

Since differentiation commutes with convolution, we have:

$$\nabla^2 L(x, y; t) = f(x, y) * \nabla^2 G(x, y; t) \tag{2.3}$$

A seminal paper by (Lindeberg 1998) explains that the LoG response always decreases when t increases resulting the maximum LoG is at the stage without the convolution. A normalizing factor γ is introduced as the power of scale to obtain the maximum LoG invariance over scale by (Lindeberg 1998):

$$LoG(x, y; t) = f(x, y) * t^{\gamma} \nabla^2 G(x, y; t) \tag{2.4}$$

By using γ, the maximum of LoG responses are the same regardless of the scales. Ideally, let the intensity distribution of blob be a perfect Gaussian (without noise and distortion), it is proved that when $\gamma = 2$, the scale invariance is achieved (Lindeberg 1998). As a result, the size of blobs can be determined at the scale when the normalized LoG reaches the maximum. Yet, in practice, the normalizing factor γ needs to be tuned to make the LoG maxima invariant to the blob sizes (note we have adopted different γ

values in our experiments, see Chapter 2.3). Let γ be set, Eq. (2.4) transforms the raw image into LoG space, a *t*-controlled, smoothed space in which transformed-blobs are highlighted with enhanced boundaries.

In scale space theory, different structures can be highlighted with regards to the scale parameter t in the multi-scale space representation. Every transformed-blob can be graphically represented by a circle centered at the local spatial maximum over LoG space with the radius r proportional to the scale at which the maximum over scales is obtained. Since the blobs studied in this research have approximately uniform sizes, one optimal *t* may be identified for all the blobs in the image. When t is small, unnecessary details can inadvertently be highlighted, leading to over-segmentation. When *t* is large, several small structures could be identified as one object, leading to under-segmentation. Therefore, determining the optimum t to extract most of small blobs of interest is critical. Fortunately, Hessian analysis can assist in determining the optimum *t*. In addition, Hessian analysis can be used to highlight blob candidates and preserve the true geometric shapes of the candidate regions as pre-segmentation.

2.2.2 Hessian Pre-segmentation

2.2.2.1 Hessian Analysis

It is known that the eigenvalues of the Hessian matrix of a blob-like structure can be used to describe the structure's geometry. Let the image be smoothed via LoG first, for any pixel (x, y) in the LoG image $LoG(x, y; t)$ at scale *t*, the Hessian Matrix for this pixel is:

$$H(x,y;t)=\begin{pmatrix} \dfrac{\partial^2 LoG(x,y;t)}{\partial x^2} & \dfrac{\partial^2 LoG(x,y;t)}{\partial x \partial y} \\ \dfrac{\partial^2 LoG(x,y;t)}{\partial x \partial y} & \dfrac{\partial^2 LoG(x,y;t)}{\partial y^2} \end{pmatrix} \tag{2.5}$$

Given geometric classification as a pixel (Salden et al. 1991) and specific orientation patterns (Frangi et al. 1998), if pixel (x, y) is concave elliptic, both of the eigenvalues λ_1,λ_2 of $H(x, y;t)$ are negative, meaning $\lambda_1 < 0, \lambda_2 < 0$. This motivates us to identify the transformed-bright blobs by the following proposition.

Proposition 1. In a transformed 2D LoG image, every pixel of a transformed-bright blob has a negative definite Hessian.

Proof. Since every pixel in the transformed-bright blob is concave elliptic, the eigenvalues of its Hessian are negative, requiring that the Hessian matrix is negative definite.■

Proposition 1 provides one necessary but not sufficient property that a pixel in a transformed-bright blob must satisfy. In other words, if a pixel resides in a transformed-bright blob, the Hessian matrix of the pixel is negative definite. But the pixel having negative definite Hessian may not be from a transformed-bright blob. This proposition provides us the theoretical foundation to identify the blob candidates by using Hessian matrix to ensure all true blobs are recognized.

Hessian Pre-segmentation Algorithm: A blob candidate *T* in LoG space can be segmented as a 4-connected component of set $U = \{(x, y) \mid (x, y) \in LoG(x, y;t), I(x, y;t) = 1\}$, where $I(x, y;t)$ is the binary indicator

such that if the pixel (x, y) has a negative definite Hessian then $I(x, y; t) = 1$, otherwise $I(x, y; t) = 0$.

Note the definiteness of the Hessian can be assessed by the leading principal minors instead of calculating its eigenvalues of the matrix. Taking a 2x2 matrix A as an example, if A_{11} is negative and $|A| > 0$, then A is negative definite. Following the proposition and the definition, the Hessian matrix can populate the pool of blob candidates that theoretically is the superset of all the blobs with their geometric shapes. In this dissertation, we have proved that when blobs are symmetric Gaussians, Hessian pre-segmentation algorithm is able to segment blobs and split touched blobs mixtures naturally (see Appendix B). Figure 1 is an illustrative example showing an identified blob candidate using the Hessian matrix and the true blob for a given scale. In the next section, we will explain how to use the Hessian analysis to obtain the optimal scale.

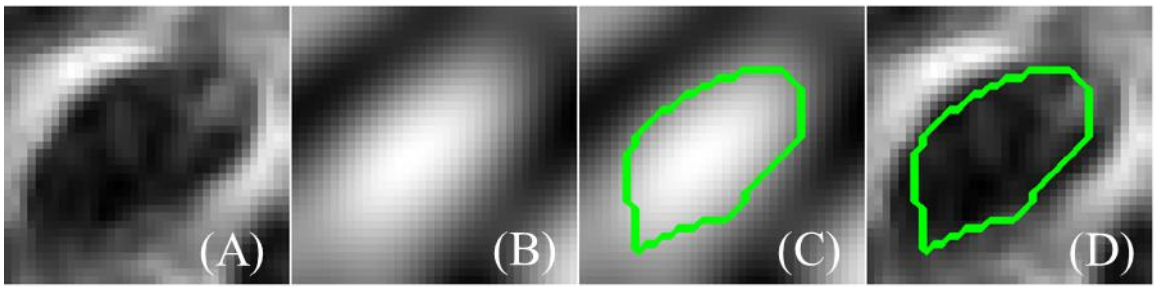

Figure 1 Transformed-bright Blob Has a Negative Definite Hessian in Normalized LoG Space. (A) Dark Blob in the Raw Image. (B) Transformed- bright Blob after Normalized LoG Transformation from (A). (C) Region of Pixels having Negative Definite Hessian in (B) is Contoured in Green over (B). (D) Region of Pixels having Negative Definite Hessian in (B) is Contoured in Green over Original Image (A). As Seen in (D), Negative Definite Hessian Outlines the Blob over (A). The Irregular Shape of the Blob is Clearly Delineated.

2.2.2.2 Hessian based Optimal Scale Identification

In addition to identifying blob candidates, Hessian analysis can be used to determine the optimal scale parameter *t*. For blobs in different scales, (Lindeberg 1998) uses the maximum normalized LoG (trace of Hessian) to select the optimum scales across the scale-space for each individual blob. Specifically, each blob achieves the most saliency at the scale at which its average of LoG reaches the maxima. Since only blobs of similar sizes are discussed in this paper, a single scale can be selected to approximate the size of all blobs. The maximum value of averaged normalized LoG is used here to auto-determine the single optimum scale for small blobs. Let the image be transformed to a normalized multi-scale LoG space representation. To determine the optimum scale of the blob candidates, let the average LoG value per candidate pixel measure C_r be:

$$C_r(t) = \frac{\sum_{(x,y)} |LoG(x,y;t)| I(x,y;t)}{\sum_{(x,y)} I(x,y;t)} \tag{2.6}$$

Where $I(x,y;t)$ is the binary indicator defined previous in ***Hessian Pre-segmentation Algorithm***. As C_r increases, the blob candidates are more salient against their background. Therefore, the optimum scale t_{best} can be calculated as:

$$t_{best} = argmax_t C_r(t) \tag{2.7}$$

Using t_{best}, the raw image is transformed into a single-scale LoG space from which the blob candidates are populated. It is proved that when blobs are symmetric Gaussians and identical in size, the selected optimum scale is equal to the optimum scale selected from (Lindeberg 1998) for each individual blob (see Appendix B).

2.2.2.3 Validation of Hessian Pre-segmentation

To evaluate the performance of Hessian pre-segmentation, the precision, recall and F-score metrics are used. Precision measures the fraction of retrieved candidates that are relevant to the ground-truth. Recall measures the fraction of ground-truth data retrieved. F-score measures overall performance. Since ground truth data are provided in the form of dots (the coordinates of the blob centers), the same as in the literature (Bernardis et al. 2011), A blob candidate *i* is considered as a true positive if and only if it is in a detection pair (i, j) where the corresponding (nearest) true dot *j* that has not been paired, and their Euclidean distance D_{ij} is within the threshold *d*. Therefore the number of true positives (*TP*) is calculated by Eq.(2.8). Precision, recall, and F-score are calculated by Eqs.(2.9),(2.10) and (2.11), respectively:

$$TP = Min\left\{ \#\left\{(i, j) : Min_{j=1}^{m} D_{ij} \le d\right\}, \#\left\{(i, j) : Min_{i=1}^{n} D_{ij} \le d\right\}\right\} \tag{2.8}$$

$$precision = \frac{TP}{n} \tag{2.9}$$

$$recall = \frac{TP}{m} \tag{2.10}$$

$$F - score = 2 \times \frac{precision \times recall}{(precision + recall)} \tag{2.11}$$

Where *m* is the number of ground-truth and n is the number of blob candidates; *d* is a thresholding parameter for evaluation purpose and can be set to a positive value $(0, +\infty)$. If *d* is small, fewer blob candidates will be counted since the distance between blob candidate centroid and ground-truth should be small in order to be counted. If *d* is set to large, more blob candidates will be counted due to the relaxation of the distance. In this paper, since the local intensity extreme could be located anywhere within the small blob

region with an irregular shape, we set evaluation threshold parameter *d* to the average diameter of blobs $d = 2 \times \sqrt{\frac{\sum_{(x,y)} I(x, y; t)}{\pi}}$. The comprehensive experimental results with different values of *d* will be discussed in Chapter 2.3.3.

For validation purpose, three commonly used detectors are chosen for comparison. One is an interest point detector, Radial-symmetry detector, which is fast and has good performance on rational symmetric image structure detection (Loy et al. 2003). The other two are LoG detectors which are specifically designed for blob detection: the original LoG method (Lindeberg 1998) and its variant - gLoG method (Kong et al. 2013). Note that all three detectors employ threshold-based post pruning to finalize the segmentation. To validate the Hessian analysis as a pre-segmentation procedure, we compare the Hessian pre-segmentation with the three detectors without the post pruning. The comprehensive comparison of HLoG with the three detectors with post pruning is provided in Chapter 2.3.

In this experiment, a dataset of fifteen 600×800 pathological cell images (Kong et al. 2013) is studied. The source code for the LoG and gLoG algorithms are implemented from (Kong et al. 2013) and source code of Radial-Symmetry is implemented from (Loy et al. 2003) (all those source codes are available online). Since the Radial-Symmetry Matlab package only provides the transformation from raw image to Radial-Symmetry space, we use the radial-symmetric centers (local extrema) with maximum intensity values as the centroids of the small blobs.

The parameter settings for the proposed method are the following: the normalizing factor γ is set to 2 based on our rough tuning experiments. The optimum

scale t_{best} of small blobs is automatically selected for each images with Eqs. (2.6) and (2.7) from the range of 2 to 10 and step size 0.5. For each image, all the algorithms adopt the same value d ($d = 2 \times \sqrt{\frac{\sum_{(x,y)} I(x,y;t)}{\pi}}$ as discussed earlier in this section).

Table 1 Comparison Results of HLoG, gLoG, Radial-symmetry and LoG on 15 Pathologic Images without Post-pruning Process

IMG	d	Hessian			gLoG (no thresholding)			Radial-Symmetry (no thresholding)			LoG (no thresholding)		
		Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
1	13.85	0.915	0.946	0.931	0.929	0.924	0.927	0.339	0.997	0.506	0.038	0.997	0.073
2	13.54	0.804	0.957	0.873	0.844	0.927	0.883	0.285	0.995	0.444	0.032	0.998	0.061
3	14.30	0.696	0.985	0.815	0.677	0.974	0.799	0.246	0.991	0.394	0.018	1.000	0.036
4	11.75	0.846	0.966	0.902	0.933	0.931	0.932	0.314	0.998	0.478	0.040	0.981	0.077
5	13.88	0.884	0.942	0.912	0.907	0.925	0.916	0.339	0.995	0.506	0.040	0.969	0.076
6	12.03	0.826	0.968	0.891	0.920	0.898	0.909	0.398	0.994	0.568	0.034	0.998	0.066
7	14.01	0.781	0.970	0.865	0.822	0.959	0.885	0.304	0.993	0.466	0.030	1.000	0.059
8	14.00	0.846	0.971	0.904	0.881	0.951	0.915	0.405	0.991	0.575	0.034	0.994	0.065
9	14.06	0.836	0.972	0.899	0.875	0.943	0.908	0.354	0.987	0.522	0.029	0.997	0.057
10	11.99	0.840	0.958	0.895	0.922	0.879	0.900	0.297	0.995	0.457	0.041	0.973	0.079
11	14.00	0.808	0.971	0.882	0.850	0.957	0.900	0.296	1.000	0.456	0.031	0.991	0.061
12	14.58	0.790	0.960	0.867	0.805	0.960	0.876	0.300	1.000	0.461	0.025	0.998	0.048
13	13.87	0.774	0.969	0.860	0.812	0.950	0.876	0.281	0.998	0.438	0.028	0.993	0.055
14	12.96	0.829	0.972	0.895	0.875	0.930	0.902	0.361	0.999	0.530	0.030	0.995	0.058
15	13.14	0.809	0.968	0.881	0.862	0.952	0.905	0.335	1.000	0.502	0.027	0.993	0.053
Avg	13.46	0.819	0.965	0.885	0.861	0.937	0.895	0.324	0.996	0.487	0.032	0.992	0.061
Std	0.89	0.051	0.011	0.027	0.066	0.025	0.032	0.044	0.004	0.050	0.006	0.010	0.012

The performance of Hessian pre-segmentation on the three metrics: precision, recall and F-score is summarized in Table 1 with regards to t_{best} and d. It is interesting to note that though both gLoG and Radial-Symmetry detectors claim no post-pruning is necessary, the results indicate the room for much improvement. Indeed, in the literature, both detectors (Kong et al. 2013, Loy et al. 2003) have employed the post pruning for experiments. As for LoG, it is designed to have post pruning as a must. Therefore, it is not surprising the performance of LoG in this validation experiment is far inferior.

In looking at the average performance (Table 1), gLoG performs the best (0.861) in precision and F-score (0.895), Radial-Symmetry performs the best in recall (0.996). We further conduct Analysis of Variance (ANOVA) for statistical conclusion. As seen in Table 2, there is no significant difference in precision and F-score in comparing our Hessian pre-segmentation with gLoG. Same conclusion is drawn in comparing our Hessian pre-segmentation with Radial-Symmetry in recall. However, Hessian pre-segmentation significantly outperforms gLoG in recall, and Radial-Symmetry in both precision and F-score.

Table 2 ANOVA on Detectors without Post-pruning using Tukey's HSD Pairwise Test on 15 Pathologic Images with 0.05 Significance Level

Contrast (Hessian v.s)	Precision p (Significant)	Recall p	F-Score p
gLoG	0.081(No)	< 0.0001 (**Yes**)	0.813 (No)
Radial-Symmetry	< 0.0001 (**Yes**)	0.893 (No)	< 0.0001 (**Yes**)

Theoretically, all the true blobs with elliptic convex shape assumption can be retrieved by Hessian pre-segmentation which is proved by Proposition 1. However, in practice, as expected, the Hessian pre-segmentation in this experiment retrieves most true blobs (96.5% close to 100%). The discrepancy is due to the modeling error that some true blobs may not comply with the elliptic convex assumption.

As discussed earlier, the results from Table 1 indicate that both gLoG and Radial-Symmetry need post-pruning for improved performance. All three comparison detectors employ thresholding-based post-pruning process. We argue that the advantage of our proposed Hessian pre-segmentation enables the extraction of blob specific features to be

used in the post-pruning. Such advantage can be shown in the full comparison (see Chapter 2.3).

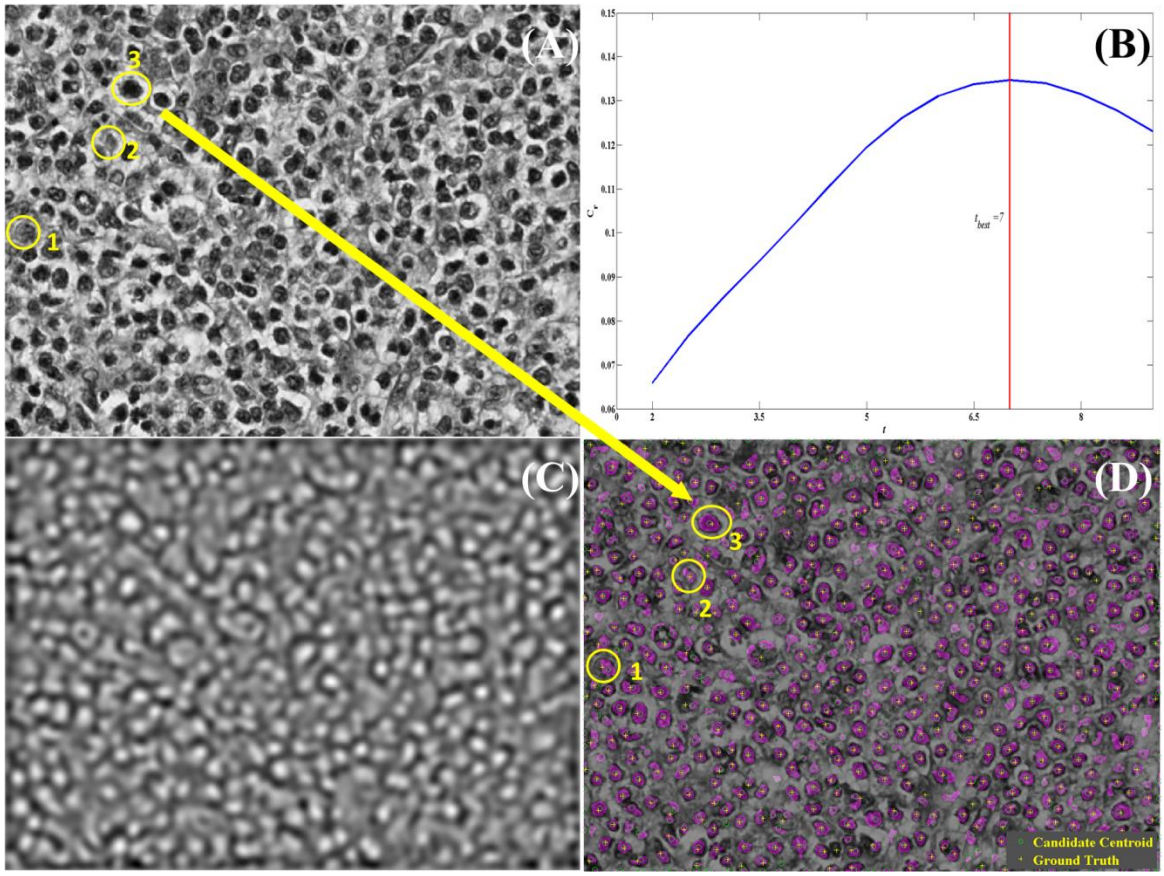

Figure 2 Hessian Pre-segmentation Results on Selected Pathologic Cell Image. (A) Raw Image. (B) Auto Scale Selection: Section $t = t_{best} = 7$ is Selected. (C) LoG Transformed Image. (D) Hessian Pre-segmentation Result: Purple Mask Shows the Shapes Retrieved by Hessian Pre-segmentation Method.

For illustration purposes, Figure 2 shows the process of our proposed Hessian pre-segmentation on a selected pathologic image. For this specific image, $t_{best} = 7$ (Figure 2(B)). Figure 2 (C) shows the LoG-transformed image, in which small blob structures are enhanced. Based on the transformed image, Hessian pre-segmentation method generates

blob candidates, (purple in Figure 2 (D)), and the LoG intensity extrema are marked as green circles to represent the centroids of these candidates compared to the ground truth dots marked as yellow cross (see Figure 2 (D)). For a better view, we circle three representative blobs in Figure 2 (A). As seen, these blobs have inhomogeneous intensity distributions: blob 3 is much darker than blob 1 and blob 2. Blobs 1 and 2 also have ambiguous boundaries against the background. Figure 2 (D) shows that Hessian pre-segmentation is able to recognize these blobs and we conclude the Hessian analysis is robust for identifying blobs with inhomogeneous intensity distributions and blurred boundaries. In the next section, the regional features extracted from the blobs are discussed.

2.2.3 Features Extraction and Evaluation

2.2.3.1 Regional-based Features Extraction

There are two common geometric measures in blob detection: *R*- the likelihood of "blobness" measure, and *S*- the second order structureness used by (Frangi et al. 1998):

$$R = \frac{|\lambda_1|}{|\lambda_2|} \tag{2.12}$$

$$S = \sqrt{\left(\lambda_1^2 + \lambda_2^2\right)} \tag{2.13}$$

Where λ_1 and λ_2 are eigenvalues of the Hessian and $|\lambda_1| \leq |\lambda_2|$. *R* is the ratio of the two principal curvatures and falls in (0,1]. (The negative definite Hessian guarantees $\lambda_1, \lambda_2 \neq 0$). If $R \to 1$, and the curvatures along two principal directions are similar, the pixel most likely resides in a blob-like structure; If $R \to 0$, i.e. the curvatures along two

principal directions are fairly different and the pixel is most likely on a line. Measure *S* indicates the contrast of the pixel against the background. With larger *S*, the pixel within the object is more salient against the background. Given *R* and *S*, the single measure to describe blobness is given by (Moon et al. 2013) as:

$$B(R,S)=\left(1-\exp\left(-\frac{R^2}{2\beta^2}\right)\right)\left(1-\exp\left(-\frac{S^2}{2c^2}\right)\right) \tag{2.14}$$

Where β and c are constant parameters that control the sensitivity of *R* and *S* respectively, they are usually set to 0.5. Although these measures quantify the geometric information of blobs, the calculations are relatively computationally expensive. Computing the likelihood of blobness R is particularly expensive because it requires the calculation and sorting of all eigenvalues in every pixel. To efficiently calculate the likelihood of blobness, we propose a modified likelihood blobness measure *R′* for each pixel as:

$$R'(x,y;t)=\frac{2}{\frac{|\lambda_1|}{|\lambda_2|}+\frac{|\lambda_2|}{|\lambda_1|}}=\frac{2\times\left|det\left(H\left(LoG(x,y;t)\right)\right)\right|}{tr\left(H\left(LoG(x,y;t)\right)\right)^2-2\times\left|det\left(H\left(LoG(x,y;t)\right)\right)\right|} \tag{2.15}$$

Where $H(\bullet)$ is the Hessian. The advantage of this modification is that instead of calculating the eigenvalues λ_1,λ_2, only trace and determinant are calculated which is more computationally efficient. Moreover, *R′* functions the same as *R* for the measurement of blobness as replacement. Next we will prove that *R′* is a monotonic increasing function of *R*, thus preserving the ability to measure the blobness.

Proposition 2. *R′* is a monotonic increasing function of *R*.

Proof. When $|\lambda_1| \le |\lambda_2|, R = \frac{|\lambda_1|}{|\lambda_2|}, R' = \frac{2}{R + \frac{1}{R}}$; When $|\lambda_1| \ge |\lambda_2|, R = \frac{|\lambda_2|}{|\lambda_1|}, R' = \frac{2}{R + \frac{1}{R}}$,

so $R' = \frac{2}{\frac{1}{R} + R}$. Since we have $\frac{\partial R'}{\partial R} = \frac{\frac{1}{R^2} - 1}{\left(\frac{1}{R} + R\right)^2} > 0$, when $R \in (0,1)$. R' is a monotonic increasing function of R. ■

This proposition proves that R' is a valid and efficient replacement of R that preserves the measurement of blobness. The Eq.(2.15) shows that $R' \in (0,1]$. For any pixel, if it is within a blob-like structure, R is close to 1, the modified R' is also close to 1; otherwise, R is close to 0, the modified R' is also close to 0. We conclude R' is equivalent to R in measuring blobness, and R' is much efficient to compute compared to R since it only requires the computation of trace and determinant. Keeping second-order structureness the same, Eq.(2.13) can be rewritten using the trace and determinant of the Hessian:

$$S(x, y; t) = \sqrt{tr\left(H\left(LoG(x, y; t)\right)\right)^2 - 2 \times det\left(H\left(LoG(x, y; t)\right)\right)} \qquad (2.16)$$

The modified blobness measure based on R' and S is $B' = B(R', S)$. Since we are interested in the features within the regions, i.e. blob candidates instead of individual pixels, aggregated measures of the pixels within each region are required. One approach is to average the measures, giving us $R_{T,mean}$, $S_{T,mean}$ and $B_{T,mean}$. Alternatively, the maximum values of the measures within a region can be calculated as $R_{T,\max}$, $S_{T,\max}$ and $B_{T,\max}$. In this research, given the true shape information available from the Hessian pre-segmentation and inspired by the design of Harris regional detectors (Mikolajczyk et al.

2004), we introduce three new measures: R_T - regional likelihood of blobness, S_T - regional structureness, and B_T for each blob candidate T (a function of R_T and S_T, as Eq.(2.14)) based on the matrix constructed by the sum of second-order derivatives over the candidate T:

$$H_T = \sum_{(x,y)\in T} H(x,y;t) = \begin{bmatrix} \sum_{(x,y)\in T} \frac{\partial^2 \left(LoG(x,y;t)\right)}{\partial x^2} & \sum_{(x,y)\in T} \frac{\partial^2 LoG(x,y;t)}{\partial x \partial y} \\ \sum_{(x,y)\in T} \frac{\partial^2 LoG(x,y;t)}{\partial x \partial y} & \sum_{(x,y)\in T} \frac{\partial^2 \left(LoG(x,y;t)\right)}{\partial y^2} \end{bmatrix} \quad (2.17)$$

Based on this matrix, we get the regional likelihood of blobness R_T:

$$R_T = \frac{2 \times \left| det(H_T) \right|}{S_T^2} \quad (2.18)$$

Where

$$S_T = \sqrt{(tr(H_T))^2 - 2 \times det(H_T)} \quad (2.19)$$

The summation of Hessian matrix in Eq. (2.17) describes the second-order derivative distribution within the region of blob candidate. The derivatives are equally weighted averaged (sum over the region T) at the centroid of T over the region. The eigenvalues of this matrix represent the two principal curvatures of the centroid over blob candidate. Thus this can be utilized to measure the likelihood of blobness over the region. Together we have three groups of features for blobness measures: mean-based, max-based and blob-candidate based. In addition, the common features Average Intensity of candidate T (M_T) and the Area of candidate T (A_T) are added into the features list. Table 3 summarizes the features and their calculations.

Table 3 Summary of Features over Blob Candidate T

Group	Features	Formulation	Description
Common Features	M_T	$\frac{\sum_{(x,y)\in T} f(x,y)}{A_T}$	Average Intensity
	A_T	$\sum_{(x,y)\in T} I(x,y;t_{best})$	Area
Using regional max information	$R_{T,max}$	$\max\{R'(x,y;t_{best}),(x,y)\in T\}$	Max likelihood of blobness
	$S_{T,max}$	$\max\{S(x,y;t_{best}),(x,y)\in T\}$	Max Structureness
	$B_{T,max}$	$\max\{B(R',S),(x,y)\in T\}$	Max blobness
Using regional mean information	$R_{T,mean}$	$\text{mean}\{R'(x,y;t_{best}),(x,y)\in T\}$	Mean likelihood of blobness
	$S_{T,mean}$	$\text{mean}\{S(x,y;t_{best}),(x,y)\in T\}$	Mean Structureness of
	$B_{T,mean}$	$\text{mean}\{B(R',S),(x,y)\in T\}$	Mean blobness
Proposed Features	R_T	$\frac{2\times\lvert det(H_T)\rvert}{S_T^2}$	Regional likelihood of blobness
	S_T	$\sqrt{(tr(H_T))^2-2\times\lvert det(H_T)\rvert}$	Regional structureness
	B_T	$B(R_T,S_T)$	Regional blobness

2.2.3.2 Regional-based Features Selection

With the features extracted above, a clustering algorithm based on the variational Bayesian Gaussian Mixture Model (VBGMM) (Bishop 2006) is used to evaluate the contributions from the features in identifying the blobs. Unlike the supervised learning algorithm, where every feature contributes with adjusted weights after training, the VBGMM is an unsupervised learning algorithm with equal weights for all features without training. Since this paper studies blobs with similar size, to verify the trivialness of blob size feature, we add additional tests with and without area features. Also, because blobness is the nonlinear combination of likelihood of blobness and structureness, these features need to have separate experiments to test their performances. Therefore, to evaluate the regional features based on different types of measures, the features listed in Table 3 are categorized into the eighteen features sets (2×3×3) listed in Table 4. The experiments are performed on the same dataset as for Hessian pre-segmentation. The

parameter settings and evaluation metrics are the same as in Chapter 2.2.2 for model setup and performance evaluations. The results are shown in Table 4.

Table 4 Evaluation of Different Feature Sets on 15 Pathologic Images (Mean and Standard Deviation of the Measures on Precision, Recall, F-score)

Feature Set	Precision Mean±Std	Recall Mean±Std	F-Score Mean±Std
$M_T, A_T, R_{T,max}, S_{T,max}, B_{T,max}$	0.856±0.046	0.953±0.014	0.901±0.021
$M_T, A_T, R_{T,max}, S_{T,max}$	0.865±0.046	0.950±0.014	0.904±0.021
$M_T, A_T, B_{T,max}$	0.832±0.054	0.964±0.011	0.892±0.029
$M_T, A_T, R_{T,mean}, S_{T,mean}, B_{T,mean}$	0.821±0.051	0.965±0.011	0.886±0.027
$M_T, A_T, R_{T,mean}, S_{T,mean}$	0.828±0.053	0.964±0.011	0.890±0.028
$M_T, A_T, B_{T,mean}$	0.832±0.053	0.964±0.011	0.892±0.029
M_T, A_T, R_T, S_T, B_T	0.969±0.017	0.767±0.051	0.855±0.030
M_T, A_T, R_T, S_T	0.945±0.027	0.880±0.034	0.911±0.016
M_T, A_T, B_T	0.963±0.022	0.822±0.029	0.887±0.014
$M_T, R_{T,max}, S_{T,max}, B_{T,max}$	0.854±0.047	0.953±0.014	0.900±0.022
$M_T, R_{T,max}, S_{T,max}$	0.862±0.047	0.950±0.014	0.903±0.022
$M_T, B_{T,max}$	0.828±0.057	0.964±0.011	0.890±0.030
$M_T, R_{T,mean}, S_{T,mean}, B_{T,mean}$	0.822±0.053	0.965±0.011	0.887±0.028
$M_T, R_{T,mean}, S_{T,mean}$	0.828±0.057	0.964±0.011	0.890±0.030
$M_T, B_{T,mean}$	0.828±0.057	0.964±0.011	0.890±0.030
M_T, R_T, S_T, B_T	0.972±0.015	0.775±0.049	0.861±0.029
M_T, R_T, S_T	**0.924±0.036**	**0.925±0.025**	**0.924±0.013**
M_T, B_T	0.959±0.028	0.829±0.043	0.888±0.018

Table 4 shows that the feature set (M_T, R_T, S_T) provides the best performance on the testing data over other feature sets. This feature set balances the precision and recall measures leading to the highest F-score. Moreover, the feature set (M_T, R_T, S_T) is relatively more stable than other feature sets since it has minimal standard deviation across all the combinations. In addition, we conduct the ANOVA to test the statistical performance of the selected feature set against other features. As shown in Table 5, the test groups the feature sets into four groups (A, B, C, D). The selected feature set

(M_T, R_T, S_T) statistically outperforms the 12 feature sets from groups B, C and D. Though there is no significant difference between the feature set (M_T, R_T, S_T) with the other five feature sets from group A, the other five feature sets have also been assigned to group B (inferior to group A). Therefore, we consider (M_T, R_T, S_T) as a reasonable feature set describing the characteristics of blobs.

Table 5 Statistical Groups based on ANOVA Tukey's HSD Pairwise Test

Category	F-score means	Statistical Groups*			
$\boldsymbol{M_T, R_T, S_T}$	0.924	**A**			
M_T, A_T, R_T, S_T	0.911	A	B		
$M_T, A_T, , R_{T,max}, S_{T,max}$	0.904	A	B		
$M_T, , R_{T,max}, S_{T,max}$	0.903	A	B		
$M_T, A_T, R_{T,max}, S_{T,max}, B_{T,max}$	0.901	A	B		
$M_T, R_{T,max}, S_{T,max}, B_{T,max}$	0.900	A	B		
$M_T, A_T, B_{T,max}$	0.892		B	C	
$M_T, A_T, S_{T,mean}$	0.892		B	C	
$M_T, R_{T,mean}, S_{T,mean}$	0.890		B	C	
$M_T, B_{T,max}$	0.890		B	C	
$M_T, B_{T,mean}$	0.890		B	C	
$M_T, A_T, R_{T,mean}, S_{T,mean}$	0.890		B	C	
M_T, B_T	0.888		B	C	
M_T, A_T, B_T	0.887		B	C	D
$M_T, R_{T,mean}, S_{T,mean}, B_{T,mean}$	0.887		B	C	D
$M_T, A_T, R_{T,mean}, S_{T,mean}, B_{T,mean}$	0.886		B	C	D
M_T, R_T, S_T, B_T	0.861			C	D
M_T, A_T, R_T, S_T, B_T	0.855				D

*Note: there is no statistically significant difference (with 0.05 significance level) between Feature sets within a statistical group.

Figure 3 shows a comparison between Hessian pre-segmentation (from Figure 2) and final identification using the three regional features. False positive blobs are removed, as shown in the circles in Figure 3. We conclude that using the three features in VBGMM can refine the pre-segmentation by removing false positive blobs in the image.

Please note the pruning may also remove a few true blobs. This is seen by comparing Table 1 (Precision: 0.819, Recall: 0.965, F-Score: 0.885), and Table 3 (Precision: 0.924, Recall: 0.925, F-Score: 0.924). The recall measure decreases by 0.04, the precision increases from 0.819 to 0.924. The overall performance of F-score increases from 0.885 to 0.924. We conclude regional features based post-pruning is promising to improve the blob detection.

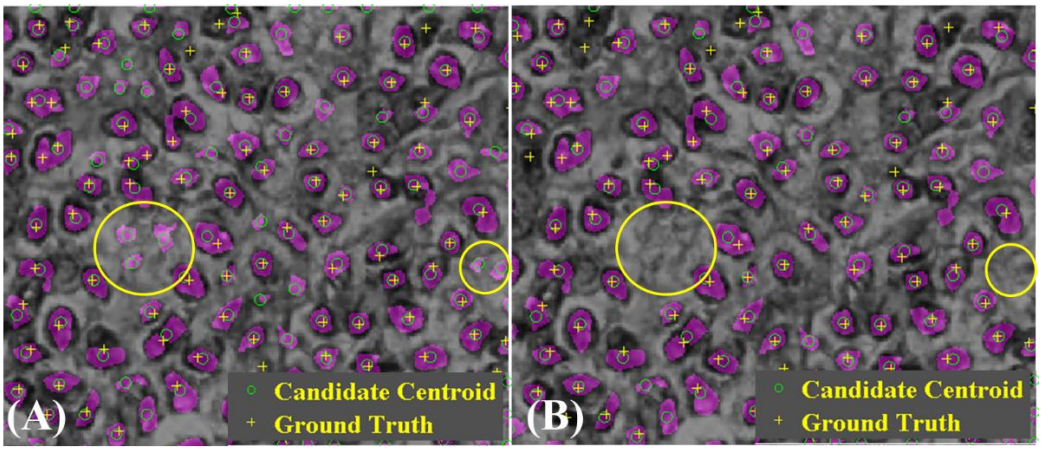

Figure 3 Blob Identification Result from Part of Figure 2 (A). (A) Pre-segmentation Result. (B) Final Identification Result. The Purple Regions are Blob Candidates and Their Centroids are Marked as Green Circle, while the Centroids of Ground Truth Data are Marked as Yellow Cross.

In the next section, we discuss the integration of Hessian analysis, regional feature extraction and clustering based final pruning for the blob identification problem.

2.2.4 HLoG for Blob Identification

We propose a three-phased HLoG workflow for blob detection, integrating raw image transformation, Hessian Pre-segmentation, feature extraction, and evaluation (Figure 4). First, the raw image is transformed into Normalized LoG space. Next, the

Hessian Pre-segmentation method is conducted for initial segmentation to generate the blob candidates. Finally the average intensity feature M_T, regional likelihood of blobness R_T, and regional structureness S_T, are used in VBGMM clustering algorithm for identification. The VBGMM is more robust than maximum likelihood Gaussian Mixture Models because it treats parameters i.e. mean vector and variance-covariance matrix in Gaussian Mixtures Models as distributions instead of deterministic values and uses hyper parameters to control them. This helps avoid the singularity issues faced by the maximum likelihood Gaussian mixture models. In addition, unlike other pruning algorithms like thresholding, the VBGMM requires no parameter tuning. The detailed steps are listed in Table 6.

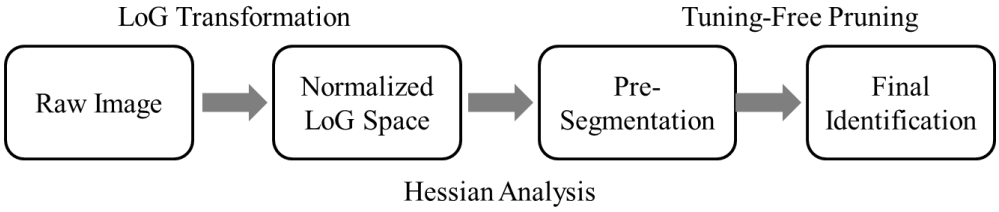

Figure 4 HLOG for Blob Detection

Table 6 Detail Steps of HLoG

1. Initialize the normalize factor γ, range and step-size of parameter t to transform the raw image into normalized LoG space
2. Binarize each section of normalized LoG space with the negative definite Hessian (for dark small blob in raw image).
3. Calculate average LoG intensity $C_r(t) = \frac{\sum_{(x,y)} \lvert LoG(x,y;t) \rvert I(x,y;t)}{\sum_{(x,y)} I(x,y;t)}$ and find optimum scale section by $t_{best} = \arg\max C_r(t)$
4. Choose the optimum scale section $t = t_{best}$ and extract the regional features M_T in raw image space and R_T, S_T in LoG space

5. Input those three features to variational Bayesian Mixture Models with 2 clusters setting
6. Choose the cluster with highest value of R_T as final segmentation

2.3 Comparison Experiments

In this section, three sets of experiments are conducted to validate of the performance of our proposed HLoG detector. In the first set of experiments, the complete version of the three blob detectors above are compared on 15 pathological images and a new supplemental data consisting of 200 fluorescence microscopy cell images (Lempitsky et al. 2010a) is tested in the second set of experiments. The 200-cell image dataset is of interest because the blobs are small, and each image can be used to test the performance of the algorithm in tolerating the noise from the background. The first two sets of experiments are to evaluate the performance given the estimated diameter d, of blob candidates. As explained in Chapter 2.2, the choice of d may impact the evaluation outcomes. Here we further conduct a separate experiment to validate the performance of HLoG with different values of d compared to the other detectors on both image datasets.

2.3.1 Experiments on Pathologic Images

Since the results of detection by the complete version of gLoG method, Radial-Symmetry method and LoG methods on 15 pathological images are available online (Rasmussen 1999), the results are directly used in this paper to avoid any of the parameters tunings. The parameter settings for HLoG are the same as presented in Section II: the normalizing factor γ is set to 2, and the range of parameter t is set from 2 to 10 with step-size 0.5, resulting in 17 sections of normalized LoG space in total. For

each image, all the algorithms adopt the same value d, which is the estimated average diameter of all small blobs, to calculate the precision, recall and F-score. The results are shown in Table 7.

As shown in Table 7, HLoG outperforms the three algorithms on the F-score for 11 out of 15 images. HLoG underperforms the gLoG method for 4 images where the blob candidates are either under-pruned (image 4 and image 13) or over-pruned (image 5 and image 8) resulting in a lower F-Score. This is due to the fact that the parameter (γ) is tuned to the group of images instead of each individual image. We argue the under-pruned and over-pruned issue may be overcome by tuning the parameter (γ) for each image, yet this will require manual processing. In looking into overall performance, our approach outperforms the three algorithms on average F-score with lower variation. Radial-symmetry and LoG do not perform as well as gLoG and HLoG, evidenced by lower detection of rotationally asymmetric blobs. HLoG balances the recall and precision metrics leading to overall better F-score. Although the average recall (0.925) of HLoG is marginally lower than that of the gLoG (0.928) and radial-symmetry (0.927), the precision performance is significantly better. The pruning algorithm sacrifices marginal recall to improve F-score and precision.

Table 7 Comparison Results of Complete Version of HLoG, gLoG, Radial-Symmetry and LoG on 15 Pathologic Images. F-score Metric is Highlighted in Gray since it Provides a Comprehensive Measurement to Evaluate the Performance

IMG	d	HLoG			gLoG			Radial-Symmetry			LoG		
		Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
1	13.85	0.980	0.886	**0.931**	0.950	0.903	0.926	0.932	0.903	0.918	0.832	0.897	0.863
2	13.54	0.969	0.884	**0.925**	0.906	0.915	0.911	0.858	0.927	0.891	0.825	0.889	0.855
3	14.30	0.858	0.930	**0.892**	0.769	0.971	0.858	0.704	0.958	0.812	0.691	0.932	0.793

4	11.75	0.900	0.949	0.924	0.943	0.916	**0.929**	0.939	0.863	0.899	0.825	0.769	0.796
5	13.88	0.962	0.897	0.928	0.951	0.920	**0.935**	0.911	0.921	0.916	0.825	0.734	0.777
6	12.03	0.902	0.943	**0.922**	0.941	0.899	0.919	0.944	0.871	0.906	0.810	0.885	0.846
7	14.01	0.948	0.955	**0.952**	0.900	0.950	0.924	0.816	0.952	0.878	0.844	0.889	0.866
8	14.00	0.957	0.910	0.933	0.939	0.947	**0.943**	0.897	0.951	0.923	0.811	0.855	0.833
9	14.06	0.936	0.917	**0.926**	0.915	0.937	0.926	0.869	0.924	0.896	0.829	0.880	0.854
10	11.99	0.898	0.946	**0.922**	0.939	0.865	0.901	0.943	0.857	0.898	0.822	0.803	0.812
11	14.00	0.905	0.954	**0.929**	0.877	0.946	0.910	0.865	0.964	0.912	0.704	0.869	0.778
12	14.58	0.897	0.943	**0.919**	0.856	0.947	0.899	0.810	0.970	0.882	0.802	0.854	0.827
13	13.87	0.878	0.937	0.906	0.885	0.942	**0.913**	0.818	0.965	0.886	0.767	0.726	0.746
14	12.96	0.937	0.935	**0.936**	0.904	0.926	0.915	0.905	0.942	0.923	0.810	0.858	0.833
15	13.14	0.941	0.894	**0.917**	0.888	0.932	0.910	0.876	0.941	0.907	0.803	0.890	0.844
Avg	13.46	0.924	0.925	**0.924**	0.904	0.928	0.915	0.873	0.927	0.897	0.800	0.849	0.821
Std	0.89	0.036	0.025	**0.013**	0.048	0.026	0.020	0.065	0.038	0.027	0.045	0.062	0.036

We further conduct ANOVA Tukey's HSD test to draw statistical conclusions. Table 8 indicates, on F-score metric, HLoG statistically outperforms the Radial-Symmetry and LoG methods ($p<0.05$) while comparable to the gLoG.

Table 8 ANOVA using Tukey's HSD Pairwise Test on 15 Pathologic Images with 0.05 Significance Level

Contrast (HLoG v.s)	Precision p (Significant)	Recall p	F-Score p
gLoG	0.679 (No)	0.998 (No)	0.741 (No)
Radial Symmetry	0.029 (**Yes**)	0.999 (No)	0.023 (**Yes**)
LoG	< 0.0001(**Yes**)	< 0.0001 (**Yes**)	< 0.0001 (**Yes**)

2.3.2 Experiments on Fluorescence Images

A new supplemental dataset is added in this experiment consisting of 200 256×256 fluorescence-light microscopy cell images. Unlike the data used above, these images contain bright blobs rather than dark blobs. Therefore the data is converted into

image with dark small structures by $1 - f(x, y)$. (We assume $f(x, y)$ varies from 0 to 1, otherwise we need to standardize $f(x, y)$ into $[0,1]$ range).

For the LoG algorithm, the range t is set as $\log(t) \in [0.5, 3]$ with step-size 0.2, as suggested in (Kong et al. 2013), and the extrema intensity value is set to 0.005 based on our experiments after tuning. For the gLoG algorithm, $\alpha = 1, \sigma_{step} = -1, \theta_{step} = \frac{\pi}{9}$ and the post-pruning threshold is set to 100. (This algorithm uses the intensity range $[0, 255]$ rather than $[0,1]$). The other parameter settings are the same as presented in (Kong et al. 2013). For Radial-Symmetry detector, based on our tuning after experiments, the local intensity threshold is set to 0.0003, and the post pruning threshold is set to 1 for refinement. In HLoG, the normalizing factor γ is set to 1 to avoid over smoothing since many blobs are clustered in this set of images by observation. The scale-space representation is the same as that of the first experiment. Similarly, the parameter d is the same for all the algorithms.

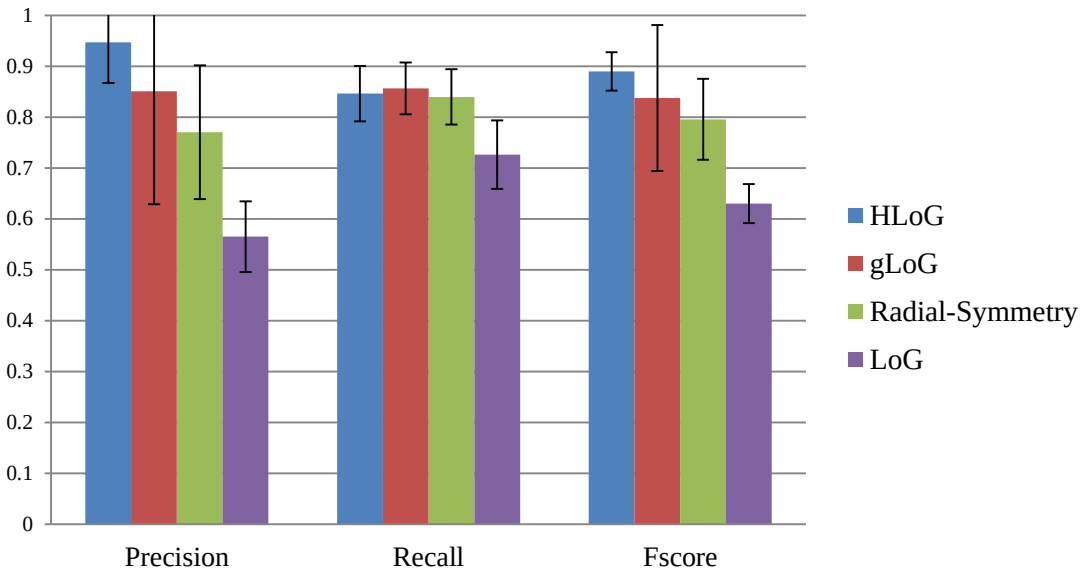

Figure 5 Comparison Results of Full Version of HLoG, gLoG, Radial-Symmetry and LoG on 200 Fluoro Images. The Error Bar Indicates the Standard Deviation of the Corresponding Measure across 200 Images.

Figure 5 compares HLoG to the gLoG, LoG and Radial-Symmetry algorithms. The result shows that though HLoG is comparable to gLoG and Radial Symmetry algorithms on recall, it outperforms the three algorithms in both precision and F-score. The variation of our results is also lower than others (The standard deviation of F-score in HLoG is 0.0377, compared to 0.1436 with the gLoG method, 0.0795 with the Radial-Symmetry method, and 0.0385 with the LoG method). We conclude that HLoG provides more accurate and stable detection of blobs in this dataset. Again, statistical analysis is performed with the results summarized in Table 9. It is observed that while comparable to the three algorithms on the recall metric, our approach statistically outperforms the comparison algorithms on precision and F-score.

Table 9 ANOVA using Tukey's HSD Pairwise Test on 200 Fluorescent Images with 0.05 Significance Level

Contrast (HLoG v.s)	Precision p (Significant)	Recall p	F-Score p
gLoG	< 0.0001(**Yes**)	0.269 (No)	< 0.0001(**Yes**)
Radial Symmetry	< 0.0001(**Yes**)	0.654 (No)	< 0.0001(**Yes**)
LoG	< 0.0001(**Yes**)	< 0.0001(**Yes**)	< 0.0001(**Yes**)

Figure 6 shows detection of cells in a single fluorescence image. In Figure 6(B), false positive blobs using gLoG algorithm (example as yellow circle 1 in Figure 6(B)) are caused by the added noise, since gLoG uses aggregated LoG map which may be sensitive to local noise. Some false positive blobs using Radial-Symmetry detector are due to the symmetric structures near true blobs as shown circle 2 in Figure 6 (C), because Radial-Symmetry detector only detects symmetric structures and lacks the ability to distinct the differences between symmetric structures. There are false blobs using LoG around the edge of the image in Figure 6 (D) (circle 3 as an example). This is because many local extrema occurring around the boundary of the image in multi-scale space which are difficult to remove only by thresholding. However, those types of false positive blobs are not shown in HLoG as seen in circles 1, 2, 3 in Figure 6 (A). This is because rather than utilizing symmetric properties and using thresholding, HLoG uses the three regional geometric features for pruning and is therefore more robust in the presence of removing false positive blobs against noisy background.

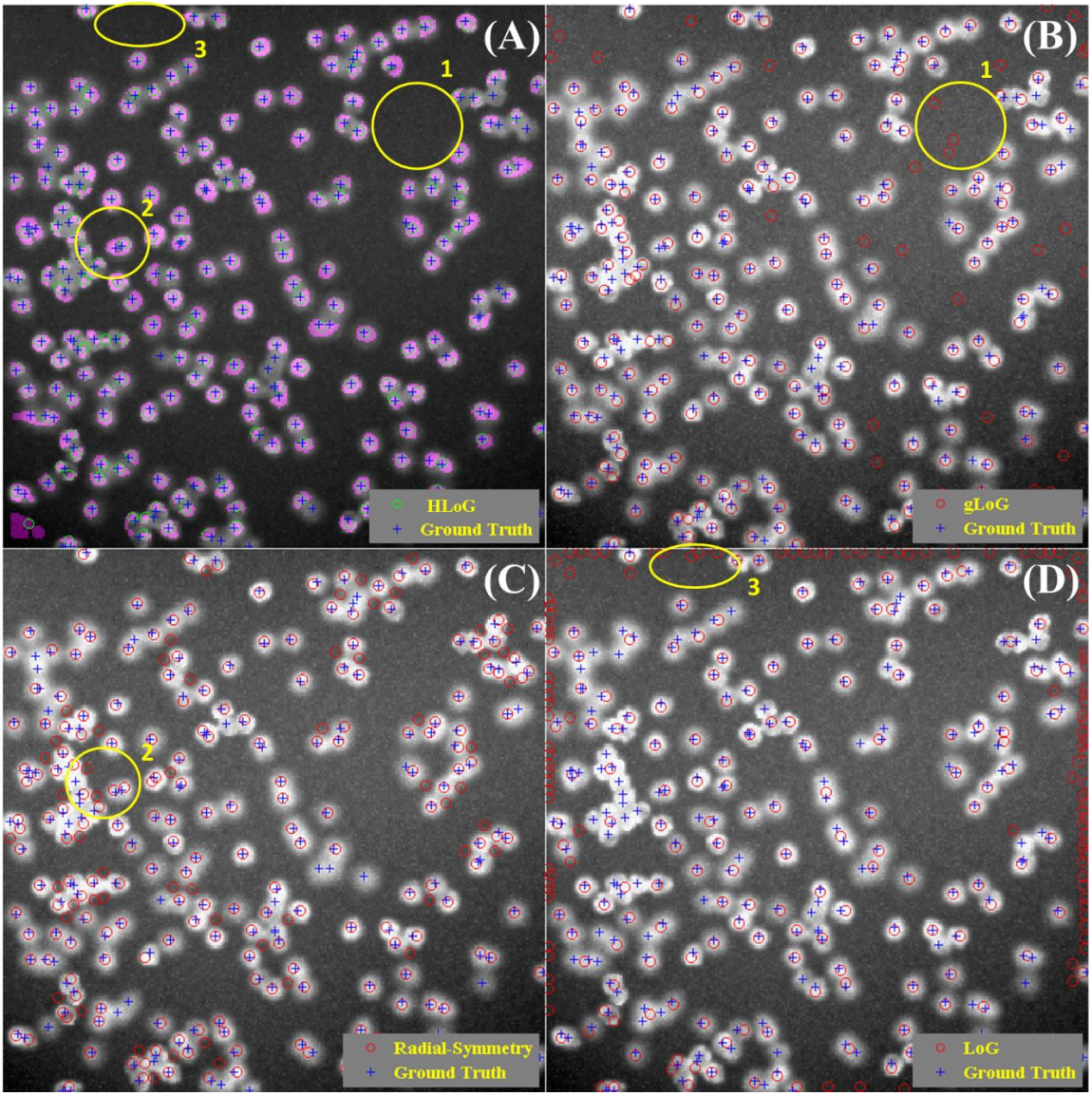

Figure 6 Detection Results on Selected Fluorescent Image. (A) Detection Result by HLoG. (B) Detection Result by gLoG Algorithm. (C) Detection Result by Radial-Symmetry Algorithm. (D) Detection Result by LoG.

2.3.3 Evaluation of HLoG at Different d

The performance metrics (precision, recall, F-score) calculated by Eqs. (2.8) (2.9) (2.10) (2.11) could be highly affected by the value of parameter *d*. In the previous experiments, *d* is set to be the estimated diameter of blobs generated by Hessian pre-

segmentation. To explore the effects of the change of d on the performance of HLoG detector, additional experiments are conducted.

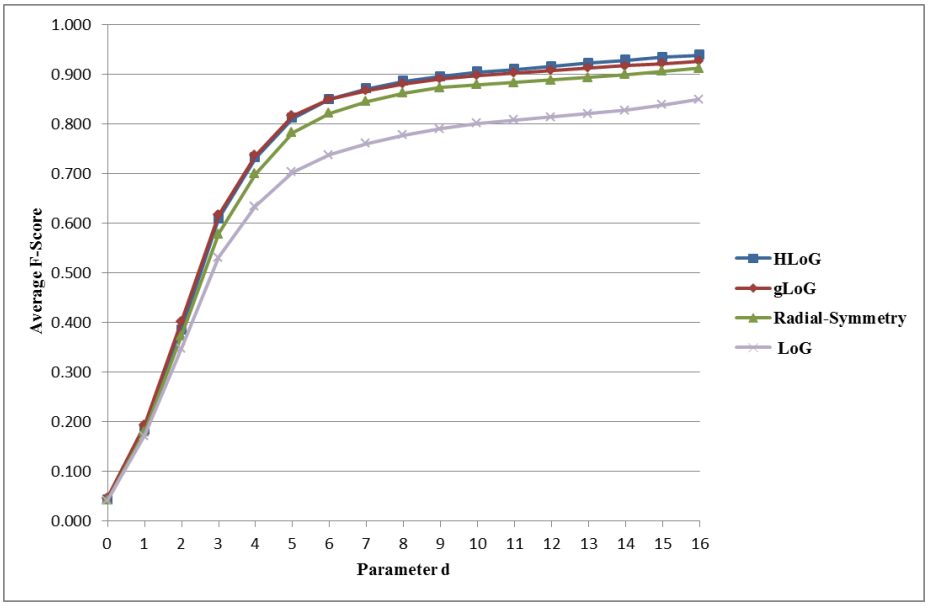

Figure 7 F-Score of HLoG, gLoG, Radial-Symmetry and LoG on 15 Pathological Images at Different Parameter *d*

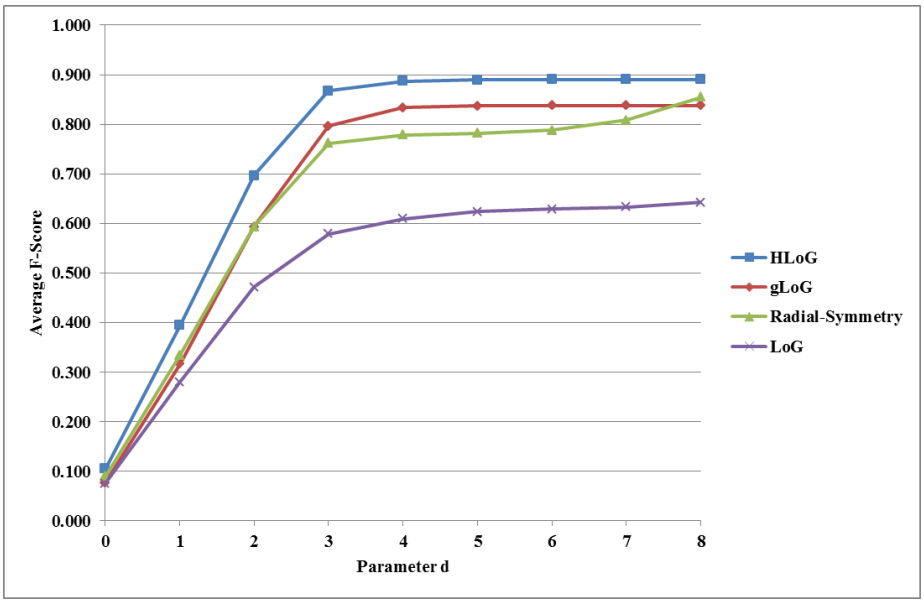

Figure 8 F-Score of HLoG, gLoG, Radial-Symmetry and LoG on 200 Pathological Images at Different Parameter *d*

Figure 7 and Figure 8 show the comparison results of HLoG, gLoG, Radial-Symmetry and LoG at different *d* on 15 pathological images and 200 fluorescence images respectively. Since F-Score is the geometric average of precision and recall, only F-score is plotted in the Figure. As discussed in Chapter 2.2.2.3, when *d* is increasing, the average F-Score will increase. More and more blob candidates will be treated as true positives because more and more blob candidates have their distance to the ground truth within the range of d. Since the blob size and image size (256×256) of fluorescence data are smaller than the pathological image (600×800), the range of d is set to be small ($[0,8]$ compared to $[0,16]$) to avoid it being greater than the distance of two neighboring blobs.

From both Figures, it is evident that HLoG outperforms other detectors on F-Score across the change of *d*. On the first set of images, HLoG is comparable to gLoG and outperforms the other two detectors on F-Score across the change of *d*. On the second set of images, HLoG outperforms the other three detectors on F-Score across the change of d. We conclude HLoG in general outperforms the three detectors regardless of the *d* value.

2.3.4 Discussion on Computational Cost

The proposed method was programmed in Matlab 2012b on a Windows (Microsoft, Inc) platform, and the experiments are done on a Windows PC with Intel Xeon 2.0 GHz CPU and 32GB of memory. For the 600×800 pathologic images, the average time cost is about 10.0 s/image compared to 30s/image for gLoG algorithm (Kong et al. 2013). The time cost spent on 200 256×256 fluorescence-light microscopy images of cells is 1.2s/image on average, compared to 10.0s/image for the gLoG

algorithm. This shows that our algorithm is efficient for 2D images and may be extended to 3D grey images. Also, the LoG transformation could be replaced by Difference of Gaussian approximation to improve the computational time .

2.4 Conclusion

In this Chapter, we propose a novel imaging detector, termed HLoG to identify small blobs in medical images. After the raw image is transformed into normalized LoG space, an optimum scale is automatically determined based on the Hessian analysis. The blob candidates are also populated with their geometric shapes as the result of Hessian pre-segmentation. This process allows us to extract multiple regional features to characterize the accurate regional properties of small blobs. Three regional features: the average intensity feature, the regional likelihood of blobness, and the regional structureness, are extracted and used in a tuning-free, Variational Bayesian Gaussian Mixture Model to prune the pre-segmentation results. One set of pathologic images (15) and one set of fluorescence-light microscopy images (200) are used to compare HLoG with gLoG, Radial-Symmetry and LoG using recall, precision and F-score metrics. In the experiments when d is estimated based on the size of the blobs, we observe HLoG outperforms Radial-Symmetry, LoG on both datasets, outperforms gLoG on the second dataset but with comparable performance for the first dataset for the precision metric. On the recall metric, HLoG only outperforms LoG on both datasets, but comparable to gLoG and Radial-Symmetry. In exploring the impact of d on the performance evaluation (F-Score), we observe HLoG outperforms Radial-Symmetry, LoG on both datasets. In addition, HLoG is computational efficient and provides a tuning-free pruning process

with only one parameter, the normalizing factor γ, in the LoG transformation which needs to be specified.

CHAPTER 3

DETECTING A LARGE-SCALE OF SMALL BLOBS IN KIDNEY 3D MR IMAGES

In Chapter 2, a 2D blob detector-HLoG is proposed to detect 2D small blobs in medical images (e.g., cell images). In this chapter, our objective is to extend HLoG to detect a large volume of 3D small blobs from the images. Specifically, the 3D kidney glomerulus (as an instance of 3D small blob) detection problem is studied. To achieve this goal, a first computational efficient 3D detector, termed Hessian-based Difference of Gaussians (HDoG), is proposed to segment glomeruli on 3D MR images. As with other similar detectors (from 2D), we first smooth the image with difference of Gaussians (DoG) kernel to identify all potential glomeruli. The Hessian analysis is applied to pre-segment and outline the candidate glomeruli. Next, we identify novel regional features associated with each candidate and conduct post-pruning using an unsupervised clustering algorithm, eliminating any false glomeruli for the final segmentation. Since extensive literature proposes various detectors for 2D images only, we compare the performance of HDoG with that of three other detectors (Laplacian of Gaussian, generalized Laplacian of Gaussian and Hessian based Laplacian of Gaussian) using 15 2D pathological images and 200 2D fluorescence microscopy images. HDoG outperforms the three detectors in identifying structures of interest and is more computationally efficient. We then test the HDoG to detect labeled renal glomeruli in both rat and human kidneys from 3D nanoparticle-enhanced MR images. The results indicate HDoG is a robust and efficient technique for unsupervised segmentation of blobs in 3D MRI.

3.1 Introduction

Nephrons are the primary functional units of the kidney. They balance the body's water, electrolyte, and pH composition, and they maintain blood volume. Each nephron contains a glomerulus, which is a tuft of fenestrated capillaries that is responsible for filtering the blood that enters from the afferent arteriole. In humans, people with fewer functioning nephrons and glomeruli tend to be more susceptible to chronic kidney and cardiovascular disease (Hoy et al. 2008). Because nephrogenesis ends shortly before birth, any deficit in nephron number at birth is likely permanent. The number of functioning nephrons can also indicate kidney disease progression. As such, being able to count functional nephrons/glomeruli is critical for assessing disease risk and progression. Glomeruli have traditionally been counted using stereological or morphometric techniques applied to histological sections. One robust approach is the disector/fractionator stereological technique (Bertram et al. 1992, Cullen-McEwen et al. 2012, Bertram 1995). While this technique is able to assess the total glomerular number, it is limited to kidneys obtained at autopsy. Furthermore, this method involves sampling just a fraction of the kidney tissue, offering a statistical estimate of the total glomerular number and volume.

Recently, the advancement in Magnetic Resonance Imaging (MRI) has made it an important tool for investigating tissue microstructure (e.g., glomeruli). The relatively new field of molecular MRI is becoming important, for example, in studies of cardiovascular disease, therapy for diabetes, cellular therapy, liver disease, and detection of cancer. Nanoparticle contrast agents, meanwhile, have emerged as contrast agents that deliver a large payload to a specific site within the tissue, changing the image where they

accumulate and allowing detection of the labeled molecules or cells. One example is superparamagnetic cationic ferritin (CF) nanoparticles for kidney glomerular assessment. After intravenous injection, CF binds to anionic proteoglycans in the glomerular basement membrane, and the accumulation of CF is detected with T2*-weighted MRI in 3D. In most cases, this local darkening is roughly spherical in shape. This technique can be used to detect, measure, and count every glomerulus in the whole kidney *ex vivo* and *in vivo* (Beeman et al. 2011, Bennett et al. 2008). However, this technique cannot be extensively used in preclinical and clinical studies because efficient image processing tools that reliably and accurately segment glomeruli from MR images do not yet exist. Initial investigations have only explored the use of local intensity thresholds to segment the glomeruli, which may not be accurate.

A CF-labeled glomerulus appears as small, convex ellipse in 3D, CF-enhanced MR images - an object termed a blob in the field of computer vision. Detecting blobs in 3D MR images is difficult because MR imaging produces acquisition noise, partial volume effects (several tissue signals mix in a voxel), and bias fields (inhomogeneity in spatial intensity). Furthermore, blob detection may require considerable computational effort; only highly efficient detectors are suitable for high-throughput *in vivo* studies. Another issue is that glomeruli are small; they have a high spatial frequency close to that of the image noise. Taking the rat kidney MRI as an example, the average volume of a rat glomerulus is ~6–7×10^5 μm^3 (Bertram et al. 1992), which is fewer than 10 voxels with a resolution of 62 x 62 x 62 μm^3 in a 3D MR image. Therefore, a robust detector can segment glomeruli from the noisy images is needed.

Some studies have attempted to segment small structures from 3D images such as breast lesions detection in ultrasound images (Moon et al. 2013) , identification of coronary calcifications in CT scans (Sanchez et al. 2012), just to name a few. However, to the best of our knowledge, most of them are focused on supervised methods, which require prior knowledge of labeled training blobs. If this information is unavailable, as in glomerular segmentation, this kind of approach does not work anymore. In addition, to manually create training labels for many small 3D blob is quite labor intensive. Indeed, it is almost impossible in glomeruli segmentation cases since the number of glomeruli is very large (~30k for rat (Beeman et al. 2011) and over 1 million for human (Beeman et al. 2014)). Therefore we focused on unsupervised blob detectors in this work.

Various unsupervised blob detectors have already been developed but most of them are restricted in 2D medical images. One classic blob detectors is Laplacian of Gaussian (LoG) (Lindeberg 1998) based on the scale-space theory. In scale-space theory, a 2D image or a slice of a 3D image is treated as part of a stack of images controlled by the scale parameter *t*. A multi-scale Gaussian scale-space representation of the image is derived as the convolution of the raw image over the Gaussian kernel with respect to the scale *t*, preserving the key spatial properties of the imaged structures (Lindeberg 1993b). When *t* increases, the number of local minima in a dark blob does not increase and the number of local maxima in a bright blob does not decrease, so a diffusion process can identify the blobs. For similarly sized blobs, one "optimal" scale exists. As such, detectors generated via LoG kernels have been successfully used to detect some blobs (Lindeberg 1998). But, the symmetric nature of the LoG detector means it cannot identify rotationally asymmetric blobs. To address this, the generalized Laplacian of Gaussian

(gLoG) (Kong et al. 2013) algorithm is proposed to detect rotational asymmetric structures by using different Gaussian kernels. One critical concern of these detectors is robustness. The way they work is to identify the centroid of the blob first, a regular ellipse with an estimated radius over the centroid is superimposed on the images to derive necessary measurements. This approach may be insufficient for MR images, which have considerable local noise. To improve the robustness of the detectors, Hessian-based Laplacian of Gaussian (HLoG) detector (Zhang et al. 2014) is proposed to accurately detect and delineate blob shapes upon which more accurate regional blob features can be extracted. As a result, HLoG detector is robust to local noise and can provide accurate detection. However, all the detectors reviewed above are designed for 2D applications, which is probably due to the computational burden of the LoG transformation.

To address these challenges, we propose a Hessian-based Difference of Gaussians (HDoG) detector for blob detection in 3D MR images. HDoG is motivated by DoG (Lowe 2004, Mikolajczyk et al. 2004, Tuytelaars et al. 2008), which has great computational advantages over LoG because it uses the approximation of LoG in detection. In HDoG, we first obtain the DoG transformation of the raw image, smoothing local noises and producing enhanced blob structures. Hessian analysis is then applied to the transformed imaging matrix to pre-segment and delineate the blob (glomerulus) candidates. Since the theory behind the Hessian analysis guarantees that pre-segmentation will recognize all true glomeruli and some "false" glomerular objects, post-pruning is necessary to remove the false glomeruli. To do this, we introduce two novel features with fast computation termed regional blobness and regional flatness. With the average intensity (commonly used in literature), the three features are derived from each

Hessian-identified glomerulus candidate and fed to a tuning-free unsupervised clustering algorithm - the variational Bayesian Gaussian mixture model (VBGMM) (Bishop 2006), to remove the false detection. Since the detectors from existing literature are for 2D images, we first compare our HDoG with LoG, gLoG and HLoG using 15 pathological images and 200 fluorescence microscopy images of cells to determine how well HDoG detector identifies small blobs. Three metrics: precision, recall, and F-scores are used for comparison. We observe, while comparable to HLoG and gLoG (most cases), HDoG outperforms LoG. In addition, it requires the least computing time. This motivates us to apply the HDoG detector on six rat and three human kidney MR images, and compare the resulting glomerular counts with corresponding stereological glomerular counts. We conclude that HDoG detector is able to automatically and accurately segment glomeruli in 3D MR images.

In summary, the main contribution of this research lies in the development of a novel detector, HDoG, a first 3D blob detector for kidney glomeruli from enhanced MR images. To alleviate the computational burden, HDoG employs DoG, the approximation of the LoG to speed up the detection. In addition, two new 3D features with fast calculation termed regional blobness and regional flatness are developed. To improve robustness, HDoG applies Hessian pre-segmentation to precisely delineate the blob structures based on which these regional features are to be extracted for post processing. The remaining of the paper is organized as follows: Chapter 3.2 describes our method in details followed by 2D validation experiments. Chapter 3.3 describes experiments using 3D MRI followed by the discussion. The conclusions are drawn in Chapter 3.4. All code

and results in the paper can be found at the website (http://swag.engineering.asu.edu/HDoG.htm).

3.2 Hessian-based Difference of Gaussian Detector

The proposed HDoG analysis is a four-phase process that involves (1) 3D raw image transformation, (2) Hessian pre-segmentation, (3) 3D feature extraction, and (4) post-pruning for final identification. Each phase is discussed below.

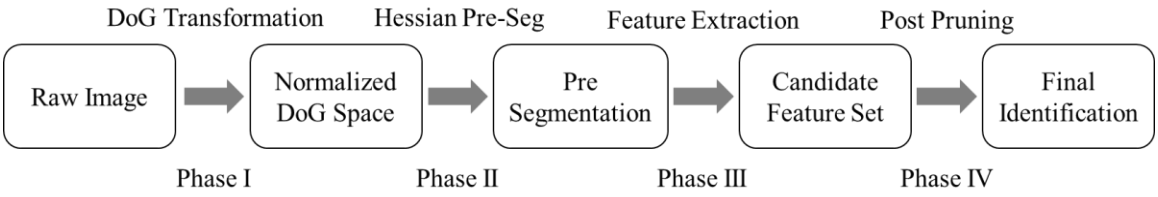

Figure 9 Flowchart of the Hessian-based Difference of Gaussians (HDoG) Analysis

3.2.1 Phase I: DoG approximation of LoG transformation

In this study, we treat each glomerulus as a blob - a region that is darker than its surroundings where the convexity of the intensity function within a blob is taken to be consistent. In reality, the convexity of the intensity function within a blob may have discontinuities because of image noise, so we must smooth out the noise to give the blob an asymptotic convex (or concave) shape. Here the DoG is chosen to serve the purpose because (1) it can smooth the image noise by enhancing the objects at the selected scale (Lindeberg 1993b), (2) it is a fast approximation of the LoG filter highlighting the blob structure (Lindeberg 1998), and (3) compared to LoG, DoG is computationally efficient

and preserves the detection accuracy (Lowe 2004). These properties are crucial for detecting/segmenting glomeruli on 3D MR images.

Let a 3D image be $f: R^3 \rightarrow R$, the scale-space representation $L(x, y, z; t)$ at point (x, y, z) with scale parameter t is the convolution of image $f(x, y, z)$ with the Gaussian kernel $(x, y, z; t)$:

$$L(x, y, z; t) = G(x, y, z; t) * f(x, y, z) \tag{3.1}$$

Where $*$ is the convolution operator and $G(x, y, z; t) = \frac{1}{(2\pi t^2)^{\frac{3}{2}}} \exp(-\frac{x^2+y^2+z^2}{2t^2})$. The Laplacian of $L(x, y, z; t)$ is:

$$\nabla^2 L(x, y, z; t) = L_{xx} + L_{yy} + L_{zz} \tag{3.2}$$

Since $\nabla^2 L(x, y, z; t) = \partial_t L(x, y, z; t)/t$, we have:

$$\nabla^2 L(x, y, z; t) \approx \frac{L(x, y, z; t+\delta t) - L(x, y, z; t)}{t\delta t} \tag{3.3}$$

That is,

$$\nabla^2 L(x, y, z; t) \approx f(x, y, z) * \frac{G(x, y, z; t+\delta t) - G(x, y, z; t)}{t\delta t} \tag{3.4}$$

To locate the optimal scale for the blobs, we follow what is done in (Lindeberg 1998) and add γ-normalization to the LoG detector as the normalized LoG detector $t^{\gamma}\nabla^2 L(x, y, z; t)$. Thus, the approximation of normalized LoG is:

$$DoG_{nor}(x, y, z; t) = t^{\gamma-1} f(x, y, z) * \frac{\left(G(x, y, z; t+\delta t) - G(x, y, z; t)\right)}{\delta t} \tag{3.5}$$

During the normalized DoG transformation, a dark glomerular blob is converted to a bright glomerular blob and vice versa. To avoid confusion, we subsequently refer to the normalized DoG blob as the "transformed blob". We want to note that the following

discussion focuses on identifying the dark glomerulus blobs (which are transformed into bright blobs, called "transformed bright blobs"), but the same process applies to identifying bright blobs (transformed into dark blobs) for other applications as shown in the experimental validation in Chapter 3.2.5.2. This normalized DoG transformation underlies the Hessian analysis we use for pre-segmentation described next.

3.2.2 Phase II: Hessian pre-segmentation

If the image is smoothed via normalized DoG, for any voxel (x, y, z) in the normalized DoG image $DoG_{nor}(x, y, z; t)$ at scale t, the Hessian matrix for this voxel is:

$$H\left(DoG_{nor}(x,y,z;t)\right)=\begin{bmatrix} \frac{\partial^2 DoG_{nor}(x,y,z;t)}{\partial x^2} & \frac{\partial^2 DoG_{nor}(x,y,z;t)}{\partial x \partial y} & \frac{\partial^2 DoG_{nor}(x,y,z;t)}{\partial x \partial z} \\ \frac{\partial^2 DoG_{nor}(x,y,z;t)}{\partial x \partial y} & \frac{\partial^2 DoG_{nor}(x,y,z;t)}{\partial y^2} & \frac{\partial^2 DoG_{nor}(x,y,z;t)}{\partial y \partial z} \\ \frac{\partial^2 DoG_{nor}(x,y,z;t)}{\partial x \partial z} & \frac{\partial^2 DoG_{nor}(x,y,z;t)}{\partial y \partial z} & \frac{\partial^2 DoG_{nor}(x,y,z;t)}{\partial z^2} \end{bmatrix} \quad (3.6)$$

Since the transformed bright blob is shaped as a concave ellipse, (where brightness fades isotropically), every voxel within the blob is a concave ellipse. Therefore, we identify the transformed bright blobs using the following proposition.

Proposition 1. In a transformed, 3D, normalized DoG image, every voxel of a transformed bright blob has a negative definite Hessian matrix.

Proof. Given the geometric classification of a voxel (Salden et al. 1991) and specific orientation patterns (Frangi et al. 1998), if voxel (x, y, z) is concave elliptical, all the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of $H(DoG_{nor}(x, y, z; t))$ are negative, meaning $\lambda_1 < 0,\ \lambda_2 < 0$, and $\lambda_3 < 0$. Since each voxel in the transformed bright blob is concave elliptical, its eigenvalues are all negative, the Hessian matrix of the voxel is negative definite. ■

Proposition 1 provides one necessary but not sufficient property that a voxel in a transformed bright blob must satisfy: if a voxel resides in a transformed bright blob, the Hessian matrix of the voxel is negative definite. However, not every voxel that has a negative definite Hessian matrix must be within a transformed bright blob. This proposition only ensures that all the true blobs are identified in the group of blob candidates (see Hessian Pre-segmentation Algorithm).

Hessian Pre-segmentation Algorithm: A blob candidate T in the normalized DoG space can be calculated as a 6-connected component of set $U = \{(x, y, z) | (x, y, z) \in DoG_{nor}(x, y, z; t), I(x, y, z; t) = 1\}$, where $I(x, y, z; t)$ is the binary indicator such that if the voxel (x, y, z) has a negative definite Hessian matrix, then $I(x, y, z; t) = 1$; otherwise, $I(x, y, z; t) = 0$.

Fast Hessian analysis: Instead of calculating the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of $H(DoG_{nor}(x, y, z; t))$, the definiteness of the Hessian matrix can be assessed by the three leading principal minors. Specifically, let D_1, D_2 and D_3 be the first, second and third leading principal minors, it is known the Hessian matrix is negative definite if and only if $D_1<0$, $D_2>0$ and $D_3<0$. As a result, from ***Proposition 1*** and ***Hessian Pre-segmentation Algorithm***, we are able to highlight the voxels belonging to transformed bright blobs using the three leading principal minors which are much more computational efficient than computing the three eigenvalues. Theoretically, the set of the transformed bright blobs with the boundaries clearly delineated is the superset of all true blobs with some false identifications. To tackle this issue, we derived regional features (Phase III) from the superset and conduct post pruning (Phase IV) to remove the false blobs.

3.2.3 Phase III: Extracting 3D Regional Features

Geometrically, the Hessian describes the second order ellipsoid of the blob structure and the absolute eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of the Hessian denote the semi-axis lengths of the ellipsoid, (Frangi et al. 1998) introduce two classic geometric features in blob detection: R_B, the likelihood of blobness, and S_B, flatness (the second-order structureness). Based on the assumption that$|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$, they are defined as:

$$R_B = \frac{\left|\lambda_1\lambda_2\lambda_3\right|}{\max\left(\left|\lambda_1\lambda_2\right|,\left|\lambda_2\lambda_3\right|,\left|\lambda_1\lambda_3\right|\right)^{\frac{3}{2}}} \tag{3.7}$$

$$S_B = \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2} \tag{3.8}$$

Where $0 < R_B \leq 1$, for an idealized blob, that is, $|\lambda_1| = |\lambda_2| = |\lambda_3|$, R_B=1. $0 < S_B \leq \infty$, the higher the S_B value is, the more salient the blob is against its background.

Fast Regional Blobness Extraction: To calculate R_B, eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of the Hessian matrix $H(DoG_{nor}(x, y, z; t))$ must be solved at each voxel which requires intensive computations. To address this concern, we propose regional blobness feature motivated by the Hessian-affine detector in (Mikolajczyk et al. 2004).

Given the Hessian matrix of each voxel defined in Eq.(3.6), the regional Hessian matrix after the DoG transformation (smoothed image) is defined as:

$$H_T\left(DoG_{nor}\left(x,y,z;t\right)\right) = \sum_{(x,y,z)\in T}\begin{bmatrix} \frac{\partial^2 DoG_{nor}\left(x,y,z;t\right)}{\partial x^2} & \frac{\partial^2 DoG_{nor}\left(x,y,z;t\right)}{\partial x\partial y} & \frac{\partial^2 DoG_{nor}\left(x,y,z;t\right)}{\partial x\partial z} \\ \frac{\partial^2 DoG_{nor}\left(x,y,z;t\right)}{\partial x\partial y} & \frac{\partial^2 DoG_{nor}\left(x,y,z;t\right)}{\partial y^2} & \frac{\partial^2 DoG_{nor}\left(x,y,z;t\right)}{\partial y\partial z} \\ \frac{\partial^2 DoG_{nor}\left(x,y,z;t\right)}{\partial x\partial z} & \frac{\partial^2 DoG_{nor}\left(x,y,z;t\right)}{\partial y\partial z} & \frac{\partial^2 DoG_{nor}\left(x,y,z;t\right)}{\partial z^2} \end{bmatrix} \tag{3.9}$$

Eq.(3.9) is the summation of the Hessian matrices of the voxels within the candidate region T. This regional Hessian matrix describes the second-order derivative distribution of the blob candidate region. Now let $\lambda'_1,\ \lambda'_2,\ \lambda'_3$ be the eigenvalues of the regional Hessian matrix, Eq.(3.7) is rewritten as:

$$R_{B_T'} = \frac{3^{\frac{3}{2}} \times \left|\lambda'_1\lambda'_2\lambda'_3\right|}{\left(\left|\lambda'_1\lambda'_2\right| + \left|\lambda'_1\lambda'_3\right| + \left|\lambda'_2\lambda'_3\right|\right)^{\frac{3}{2}}} \tag{3.10}$$

Substitute $\max(|\lambda'_1\lambda'_2|, |\lambda'_1\lambda'_3|, |\lambda'_2\lambda'_3|)$ with $\frac{|\lambda'_1\lambda'_2|+|\lambda'_1\lambda'_3|+|\lambda'_2\lambda'_3|}{3}$, we get

$$R_{B_T'} = \frac{3^{\frac{3}{2}} \times \left|\lambda'_1\lambda'_2\lambda'_3\right|}{\left(\left|\lambda'_1\lambda'_2\right| + \left|\lambda'_1\lambda'_3\right| + \left|\lambda'_2\lambda'_3\right|\right)^{\frac{3}{2}}} \tag{3.11}$$

Then R_T can be derived as

$$R_T = \left(R_{B_T'}\right)^{\frac{2}{3}} = \frac{3 \times \left|\lambda'_1\lambda'_2\lambda'_3\right|^{\frac{2}{3}}}{\left|\lambda'_1\lambda'_2\right| + \left|\lambda'_1\lambda'_3\right| + \left|\lambda'_2\lambda'_3\right|} \tag{3.12}$$

Note that R_T still maintains the same property, that is, $0 < R_T \leq 1$. Since

$$R_T = \frac{3 \times \left|\lambda'_1\lambda'_2\lambda'_3\right|^{\frac{2}{3}}}{\left|\lambda'_1\lambda'_2\right| + \left|\lambda'_1\lambda'_3\right| + \left|\lambda'_2\lambda'_3\right|} = \frac{3}{\frac{\left|\lambda'_1\lambda'_2\right|}{\left|\lambda'_1\lambda'_2\lambda'_3\right|^{\frac{2}{3}}} + \frac{\left|\lambda'_1\lambda'_3\right|}{\left|\lambda'_1\lambda'_2\lambda'_3\right|^{\frac{2}{3}}} + \frac{\left|\lambda'_2\lambda'_3\right|}{\left|\lambda'_1\lambda'_2\lambda'_3\right|^{\frac{2}{3}}}}$$

Based on Arithmetic-Geometric Mean Inequality, we have:

$$\frac{\left|\lambda'_1\lambda'_2\right|}{\left|\lambda'_1\lambda'_2\lambda'_3\right|^{\frac{2}{3}}} + \frac{\left|\lambda'_1\lambda'_3\right|}{\left|\lambda'_1\lambda'_2\lambda'_3\right|^{\frac{2}{3}}} + \frac{\left|\lambda'_2\lambda'_3\right|}{\left|\lambda'_1\lambda'_2\lambda'_3\right|^{\frac{2}{3}}} \geq 3\sqrt[3]{\frac{\left|\lambda'_1\lambda'_2\lambda'_3\right|^2}{\left|\lambda'_1\lambda'_2\lambda'_3\right|^2}}$$

Therefore, $R_T \leq \frac{3}{3\sqrt[3]{\frac{|\lambda'_1\lambda'_2\lambda'_3|^2}{|\lambda'_1\lambda'_2\lambda'_3|^2}}} = 1$, we conclude as an estimate of R_{B_T}, R_T maintains the same property, $0 < R_T \leq 1$.

By this substitution the advantage is, instead of calculating the eigenvalues, we can use the principal minors to get the blobness feature:R_T. Specifically, since the Hessian matrix is negative definite at every voxel within a blob candidate, the regional Hessian matrix is negative definite, meaning, $\lambda_1', \lambda_2', \lambda_3' < 0$. Thus,

$$R_T = \frac{3 \times \left| det\left(H_T\right) \right|^{\frac{2}{3}}}{pm\left(H_T\right)} \tag{3.13}$$

Where $pm(H_T) = \lambda'_1\lambda'_2 + \lambda'_2\lambda'_3 + \lambda'_1\lambda'_3$. By (Meyer 2000), $pm(H_T)$ can be obtained by calculating three 2×2 principal minors of H_T:

$$\lambda_1'\lambda_2' + \lambda_2'\lambda_3' + \lambda_1'\lambda_3' = \det\begin{pmatrix} H_T^{1,1} & H_T^{1,2} \\ H_T^{2,1} & H_T^{2,2} \end{pmatrix} + \det\begin{pmatrix} H_T^{2,2} & H_T^{2,3} \\ H_T^{3,2} & H_T^{3,3} \end{pmatrix} + \det\begin{pmatrix} H_T^{1,1} & H_T^{1,3} \\ H_T^{3,1} & H_T^{3,3} \end{pmatrix} \tag{3.14}$$

Fast Regional Flatness Extraction: Similar to regional blobness, we introduce regional flatness as:

$$S_T = \sqrt{\lambda'^2_1 + \lambda'^2_2 + \lambda'^2_3} \tag{3.15}$$

It can be rewritten as:

$$S_T = \sqrt{\left(\lambda_1' + \lambda_2' + \lambda_3'\right)^2 - 2\left(\lambda_1'\lambda_2' + \lambda_2'\lambda_3' + \lambda_1'\lambda_3'\right)} = \sqrt{tr\left(H_T\right)^2 - 2 \times pm\left(H_T\right)} \tag{3.16}$$

These two modified features greatly reduce the computational cost of Eqs. (3.7) and (3.8), because first R_T and S_T are based on the regional Hessian matrix evaluated at each blob region instead of at every voxel. Secondly, R_T and S_T only require us to calculate the trace and determinant values rather than the roots (eigenvalues) of the Hessian matrix.

In addition to R_T and S_T, a third feature A_T the average intensity of region T (commonly used in both literature and empirical studies), is derived. We then input these

three novel features into a clustering algorithm, the variational Bayesian Gaussian Mixture Model (VBGMM), to remove the false identifications from the glomeruli candidate pool.

3.2.4 Phase IV: Variational Bayesian Gaussian Mixture Model

Mixture models are computationally convenient ways to model complex probability distributions, and they are based on a linear combination of some number (M) component distributions. We employ the VBGMM because (1) compared to the maximum likelihood Gaussian mixture model, the variational model cannot be trapped in a singularity solution, and (2) the variational model can automatically identify the number of clusters needed for optimum performance without requiring initialization and subjective parameter settings.

In the VBGMM, given a 3D MR image, we assume that several multivariate Gaussian distribution components form the entire image. One of the components is the group of glomeruli, and the others make up background and image noise. If $X = \{X^1, \dots, X^N\}$ is the observation, $N(X^i|\mu, \Lambda)$ is the multivariate Gaussian distribution with mean μ and inverse covariance Λ that X^i follows. The mixture distribution for M components is

$$P\left(X^i|\pi, \mu, \Lambda\right) = \sum_{j=1}^{M} \pi_j N(X^i \mid \mu, \Lambda) \tag{3.17}$$

Where π_j is the weight for component j.

Assuming elements in X are independent of each other, we introduce the binary latent variable $Z = \{Z^1, \ldots, Z^{NM}\}$, where $z_{im} = 1$ indicates that X^i belongs to class m, and $\sum_{j=1}^{M} Z^{ij} = 1$. The conditional probability of the image dataset is

$$P(X|Z, \mu, \Lambda) = \prod_{i=1}^{N}\prod_{j=1}^{M} N\left(X^i|\mu, \Lambda\right)^{Z^{ij}} \tag{3.18}$$

Thus, the VBGMM approximates the posterior $P(\theta|X)$, given any distribution $P(X)$ and unknown parameters θ, by a simpler distribution $Q(\theta)$ that marginalizes the unknown parameter θ (Bishop 2006). Here, observation X^i is a vector of the three features A_T, R_T, S_T for a glomerular candidate region. The observations (glomerular candidate regions) form a multivariate Gaussian mixture and therefore are clustered into glomerular regions and non-glomerular regions using a Bayesian inference method.

3.2.5 Validation tests on 2D images

To examine HDoG performance, we carry out initial tests on 2D images using three evaluation metrics: precision, recall, and F-score. Three state-of-art detectors, LoG, gLoG and HLoG, are studied for comparison.

In the literature of 2D detectors, the ground truth data are usually provided in the form of dots (the coordinates of the blob centers). A blob candidate i is considered as a true positive if and only if it is in a detection pair (i, j) where the corresponding (nearest) true dot j that has not been paired, and their Euclidean distance D_{ij} is within the threshold d. Therefore the number of true positives (TP) is calculated by Eq.(3.19). Precision, recall, and F-score are calculated by Eqs.(3.20), (3.21) and (3.22), respectively:

$$TP = Min\left\{ \#\left\{(i,j): Min_{j=1}^{m} D_{ij} \le d\right\}, \#\left\{(i,j): Min_{i=1}^{n} D_{ij} \le d\right\}\right\} \tag{3.19}$$

$$precision = \frac{TP}{n} \tag{3.20}$$

$$recall = \frac{TP}{m} \tag{3.21}$$

$$F - score = 2 \times \frac{precision \times recall}{(precision + recall)} \tag{3.22}$$

Where m is the number of ground truth points and n is the number of blob candidates; d is a thresholding parameter and can be set to a positive value $(0, +\infty)$. In the following experiments, we set d >1 to tolerate the potential impact from the image noises. If d is small, fewer blob candidates will be counted because the distance between the blob candidate centroid and the ground truth point must be small. If d is large, more blob candidates will be counted because the threshold distance is relaxed.

For 2D images, the 2D versions of the regional Hessian matrix H_T, the regional blobness R_T, and the regional flatness S_T are used in (Zhang et al. 2014):

$$H_{T,2D} = \sum_{(x,y)\in T} \begin{bmatrix} \frac{\partial^2 DoG_{nor}(x,y;t)}{\partial x^2} & \frac{\partial^2 DoG_{nor}(x,y;t)}{\partial x \partial y} \\ \frac{\partial^2 DoG_{nor}(x,y;t)}{\partial x \partial y} & \frac{\partial^2 DoG_{nor}(x,y;t)}{\partial y^2} \end{bmatrix} \tag{3.23}$$

$$R_{T,2D} = \frac{2}{\frac{\lambda'_{1,2D}}{\lambda'_{2,2D}} + \frac{\lambda'_{2,2D}}{\lambda'_{1,2D}}} = \frac{2?\lambda'_{1,2D}\lambda'_{2,2D}}{\left(\lambda'_{1,2D} + \lambda'_{2,2D}\right)^2 - 2?\lambda'_{1,2D}\lambda'_{2,2D}} \tag{3.24}$$

$$S_{T,2D} = \sqrt{\left(\lambda'_{1,2D} + \lambda'_{2,2D}\right)^2 - 2\lambda'_{1,2D}\lambda'_{2,2D}} \tag{3.25}$$

Where $\lambda'_{1,2D}, \lambda'_{2,2D}$ are eigenvalues of $H_{T,2D}$.

3.2.5.1 Dataset 1: 2D pathological images

15 pathological images were used to compare HDoG with the LoG, gLoG and HLoG methods (images available online (Kong et al. 2013)). Also, to thoroughly evaluate HDoG performance, the threshold parameter d was varied from 2 to 16. The results are shown in Table 10.

Table 10 Comparing HLoG, gLoG, and LoG Performance on Dataset 1: 15 2D Pathologic Images (the Normalizing Factor γ =2 based on Prior Experiments. Detailed of the Parameter Setting Please Refer to (Zhang et al. 2014)). Paired T-tests were Performed at a Significance Level of 0.05. Symbols in Cell Means: + Statistically Outperformed; = Statistically Comparable; - Statistically Underperformed

d	HDoG (Avg)			HDoG v.s. LoG			HDoG v.s. gLoG			HDoG v.s. HLoG		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
2	0.411	0.400	0.405	+	+	+	=	=	=	=	=	=
3	0.632	0.615	0.623	+	+	+	=	=	=	=	=	=
4	0.749	0.728	0.738	+	+	+	=	=	=	=	=	=
5	0.822	0.800	0.810	+	+	+	=	-	=	=	=	=
6	0.860	0.837	0.847	+	+	+	=	-	=	=	=	=
7	0.881	0.857	0.868	+	+	+	=	-	=	=	=	=
8	0.897	0.874	0.885	+	+	+	=	-	=	=	=	=
9	0.906	0.883	0.894	+	+	+	=	-	=	=	=	=
10	0.913	0.890	0.901	+	+	+	=	-	=	=	=	=
11	0.919	0.895	0.906	+	+	+	=	-	=	=	=	=
12	0.924	0.900	0.911	+	+	+	=	-	=	=	=	=
13	0.931	0.907	0.918	+	+	+	=	=	=	=	=	=
14	0.936	0.911	0.923	+	+	+	=	=	=	=	=	=
15	0.940	0.916	0.927	+	+	+	=	=	=	=	=	=
16	0.944	0.920	0.931	+	+	+	=	=	=	=	=	=

As shown in Table 10, in term of F-score, HDoG statistically outperformed LoG but was comparable to gLoG and HLoG. As for precision measure, we observed similar results, that is, HDoG statistically outperformed LoG and was comparable to gLoG and HLoG. While for recall measure, HDoG outperformed LoG, was comparable to HLoG,

and gLoG in most cases. However, for *d* in the range of 5 to 12, HDoG underperformed gLoG. We contented this was because gLoG is a generalized detector which performs well robustly on the images where the blob candidates are within the distance (roughly the size of the blobs) to the ground truth. For cases where the blobs are very small objects in the images, (requiring tighter *d*), or for cases there are massive number of blobs, (requiring larger *d* to relax the threshold), its performance is equivalent to our HDoG. Please note that even for the 2D images, gLoG needed 30 seconds per image and HDoG was 5x faster than that of gLoG (6 seconds per image) on the same computational environment (Windows PC with Intel Xeon 2.0 GHz CPU and 32 GB of memory). The computation of HLoG is 10 seconds per image.

3.2.5.2 Dataset 2: 2D fluoroscopic images

In this second validation experiment, 200 256×256 fluorescence light microscopy images of cells were studied. The images were added with additional noise to test the robustness of the detectors (Lempitsky et al. 2010b). Moreover, unlike Dataset 1, these images contained bright blobs rather than dark blobs. Therefore, the data were converted into images containing small dark structures using $1 - f(x, y)$. (We assume $f(x, y)$ varied from 0 to 1, or we would have to standardize $f(x, y)$ to the $[0,1]$ range). The parameter settings for LoG and gLoG were as suggested in (Kong et al. 2013). Since the cell sizes in Dataset 2 were much smaller than those in Dataset 1, for comprehensive comparison, we varied the threshold parameter d from 2 to 12. The results are summarized in Table 11.

Table 11 Comparing HLoG, gLoG, and LoG on Dataset 2: 200 2D Fluorescent Images (the Normalizing Factor $\boldsymbol{\gamma}$ =1 based on Prior Experiments. Details of the Parameter Settings are Described in (Zhang et al. 2014)). Paired T-tests were Performed at a Significance Level of 0.05. + Statistically Outperformed; = Statistically Comparable; - Statistically Underperformed

d	HDoG (Avg)			HDoG v.s. LoG			HDoG v.s. gLoG			HDoG v.s. HLoG		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
2	0.623	0.559	0.586	+	=	+	=	-	=	=	=	=
3	0.881	0.793	0.829	+	+	+	+	-	+	=	=	=
4	0.944	0.851	0.890	+	+	+	+	=	+	=	=	=
5	0.952	0.858	0.897	+	+	+	+	=	+	=	=	=
6	0.953	0.859	0.898	+	+	+	+	=	+	=	=	=
7	0.953	0.859	0.898	+	+	+	+	=	+	=	=	=
8	0.953	0.860	0.898	+	+	+	+	=	+	=	=	=
9	0.953	0.860	0.899	+	+	+	+	=	+	=	=	=
10	0.954	0.860	0.899	+	+	+	+	=	+	=	=	=
11	0.954	0.861	0.899	+	+	+	+	=	+	=	=	=
12	0.955	0.862	0.900	+	+	+	+	=	+	=	=	=

For smaller blobs in this dataset, HDoG outperformed LoG and gLoG on the F-score measure, and was comparable to HLoG. On precision measure, HDoG outperformed LoG and gLoG, and was comparable to HLoG. As for recall, it is interesting to note that HDoG outperformed LoG in most cases (except *d*=2) and was comparable to HLoG and gLoG (except $d = 2,3$). This may be explained by the outperformance of gLoG on images with the blob candidates tightly within the distance to the ground truth. Yet for the images with large number of blobs which may require a larger *d* to identify the blobs, it had a statistically similar performance as HDoG. In term of computing time, the average processing time for HDoG was 0.9 second/per image, HLoG was 1.2 second/ per image and gLoG was 10 second/per image (Windows PC with Intel Xeon 2.0 GHz CPU and 32 GB of memory).

In summary, initial tests with both 2D datasets showed that HDoG has the potential for blob detection in 3D images. It had a statistically similar performance (or even outperformance) compared to the 2D detectors from the literature, yet more computationally efficient. This advantage will be more obvious when the size of image increases and when it is 3D image.

3.3 Experiments on 3D MR Images

Because our initial tests on the 2D images were promising, we explored how well the HDoG segments renal glomeruli in 3D MR images. To test this, we performed two experiments. In the first (section 3.1), we evaluated preclinical data: six 3D kidney MR images from rats. We compared the resulting glomerular counts with those obtained through acid maceration and stereology. In the second experiment (section 3.2), we studied clinical data: three 3D kidney MR images from humans. We compared these glomerular counts with those obtained using stereology.

3.3.1 Segmenting glomeruli from rat kidney MR images

Six CF-labeled 3D MR images of rat kidneys were studied. After obtaining glomerular counts using HDoG, we compared our counts with counts obtained using a manual acid maceration method (Bonvalet et al. 1972), which uses acid to extract glomeruli from kidney tissue, as well as to the disector-fractionator stereological method (Bertram et al. 1992), whereby we estimated the number of glomeruli by analyzing pairs of histological sections. These methods are established histological techniques for estimating glomerular number, so we consider them to be “ground truth” data.

All rat studies were approved by the institutional animal care and use committee, consistent with the NIH Guide to the Care and Use of Laboratory Animals. A 19T NMR with a DOTY 3-axis imaging probe was used to scan rat kidneys. The total scan time was 6 h per kidney with a 3D GRE sequence with TE/TR = 7/40 ms and a resolution of $62\times62\times78$ μm^3 and a re-sliced matrix size of $256\times256\times256$. Table 12 shows the glomerular counts obtained with each method (HDoG, acid maceration, and disector-fractionator stereology) for the six kidneys.

Table 12 Glomerular Counts for Six Rat Kidneys using the HDoG, Acid Maceration, and Stereology[1] Methods ($\boldsymbol{\gamma}$ =2, Intel Xeon 2.0 GHz CPU and 32 GB of Memory).

Rat	Acid Maceration	Stereology	HDoG	Time (seconds)
CF1	27,504	34,504	29,484	268
CF2	31,190	35,421	34,460	294
CF3	28,944	24,156	27,051	242
CF4	31,075	-	35,296	243
CF5	33,321	-	31,196	237
CF6	31,478	-	35,248	242
Avg	30,585	31,360	32,122.5	255
Std	2,053	6,256	3131.6	20

From Table 12, we observed that HDoG consistently identified glomeruli in all six kidneys with reasonable computing times (<5 min). Although there were some differences between the glomerular counts, the three methods generally agreed well. Figure 10 shows the segmentation results on representative axial view slices of the six rats, where slice 100 (of 256) is shown for rats CF1, CF2, and CF3, and slice 150 (of 256) is shown for rats CF4, CF5, and CF6. From the figure, we see that most glomeruli were identified and contoured in green. After comparing count numbers and visually checking

[1] Only 3 CF rats were counted by stereology.

the accuracy, we concluded that HDoG can automatically identify glomeruli in rat MR images.

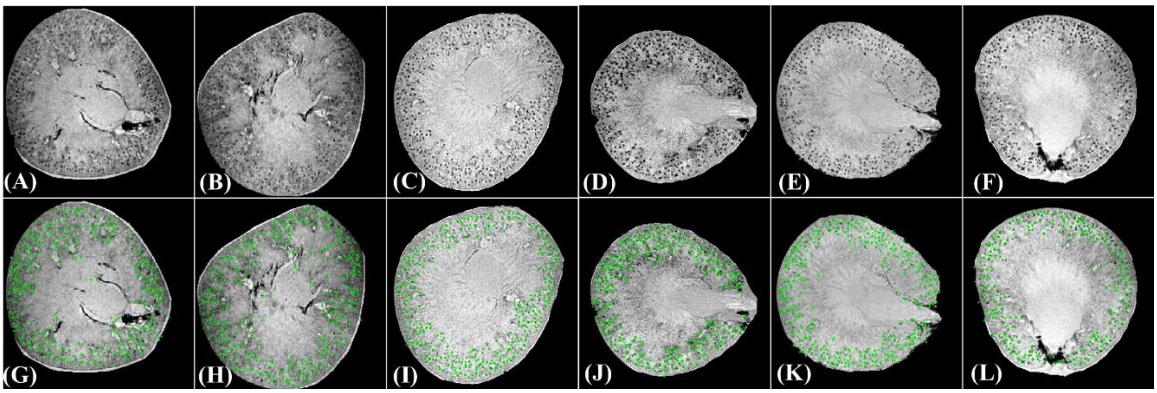

Figure 10 Glomerular Segmentation Results from 3D MR Images of Rat Kidneys (Selected Slices Presented). (A–C) Slice 100 for Rats CF1, CF2, and CF3. (D–F) Slice 150 for Rats CF4, CF5, and CF6. (G–I) Identification Results for (A–C), respectively. Identified Glomeruli are Contoured in Green. (J–L) Identification Results for (D–F), respectively, where Identified Glomeruli are Contoured in Green.

3.3.2 Segmenting glomeruli from human kidney MR images

With Institutional Review Board approval and informed consent, we obtained three post-autopsy human kidneys through a donor network. The kidneys were perfused with cationized horse spleen ferritin (CF) and imaged with a Bruker 7T/35 MRI scanner using a T2*-weighted protocol. 3D MR images were acquired at 117×117×117 μm^3 resolution with an image matrix size of 512×512×896. On average, the total scan time was10 h, 39 min per kidney. The kidneys ranged from healthy (CF1) to untreated hypertensive (CF2) and treated hypertensive (CF3) (Beeman et al. 2014). This range of diseases allowed us to assess the robustness of automated segmentation to specific pathologies. We validated the MRI-based glomerular counts using the physical

disector/fractionator stereological method described by Cullen-McEwen et al. (Cullen-McEwen et al. 2012, Cullen-McEwen et al. 2003).

We applied the HDoG algorithm to each human 3D MR image using the same platform as for the rats. The results are shown in Table 13.

Table 13 Glomerular Counts for Three Human Kidneys using the HDoG and Stereology Techniques ($\boldsymbol{\gamma}$ =2, Intel Xeon 2.0 GHz CPU and 32 GB of Memory).

Human	Stereology	HDoG	Time (seconds)		
			DoG & Feature Extraction	VBGMM	Total
Human CF1	1,130,000	1,242,008	942	12,375	**13,317**
Human CF2	740,000	711,397	935	2,366	3,301
Human CF3	1,460,000	1,370,095	1,072	2,088	3,160
Avg	1,110,000	1,107,833	983	5610	6593
Std	360,416	349,246	77	5861	5824

From Table 13, HDoG consistently counted glomeruli in all three kidneys with reasonable computing times (<1 h for CF2 and CF3, <4h for CF1). We again observed discrepancies between the glomerular counts, but our histology experts confirmed that the two methods generally agreed well. Figure 11 shows the segmentation results on selected axial-view slices of the human kidneys. As seen on the figure, almost all glomeruli were identified by HDoG, (contoured in green), and the image intensity distribution was inhomogeneous across slices. To illustrate the ability of HDoG to segment glomeruli in different regions, a top slice (slice 100/896) and a middle slice (slice 500/896) are shown. After comparing count numbers and visually checking the accuracy, we concluded that the HDoG algorithm can automatically identify glomeruli in human MR images.

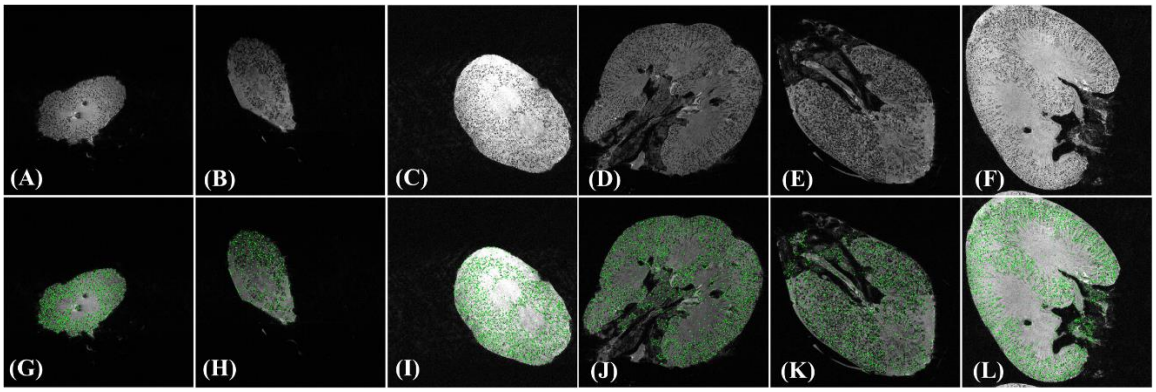

Figure 11 Glomerular Segmentation Results for 3D MR Images of Human Kidneys (Selected Slices): (A–C) Original Slice 100 for Human CF1, CF2, and CF3 Kidneys. (D–F) Slice 500 for Human CF1, CF2, CF3 Kidneys. (G–I) Identification Results for (A–C), respectively, where Identified Glomeruli are Contoured in Green. (J–L) Identification Results for (D–F), Respectively, where Identified Glomeruli are Contoured in Green.

3.3.3 Discussion of Computation Time

Since the computing time on the rat MRI is satisfactory (less than 5 minutes/image), we focus the discussion on the human MRI in this section. We observe that the glomerular segmentation of Human CF1 kidney took a much longer time than the other two. The time is mainly contributed by the VBGMM clustering process. We contend the low contrast of CF1 vs. CF2 and CF3 may be the main factor leading to the longer time. To explore the variations of the contrasts among the three human kidney, the intensity distribution of both glomeruli vs. the whole image is generated and shown in Figure 12. As seen, the difference of the distribution mode between the true glomeruli and the image (an indicator of contrast), highlighted in the figure, shows that CF1 has the lowest contrast while CF3 has the highest contrast. As a result, CF1 requires longer time for VBGMM for converged solution (12,375 seconds). While CF2 has medium contrast, it has comparable computing time (2366 seconds) to that of CF3 (2088 seconds). This can be explained that CF2 is from the hypertensive patient, having fewer perfused

glomeruli and regions of vascular and glomerular sclerosis, coupled with glomerular hypertrophy in the perfused portions of the kidney (Beeman et al. 2014). We suspect that variation in image contrast, either through normal physiological variation or image acquisition, will impact computation time.

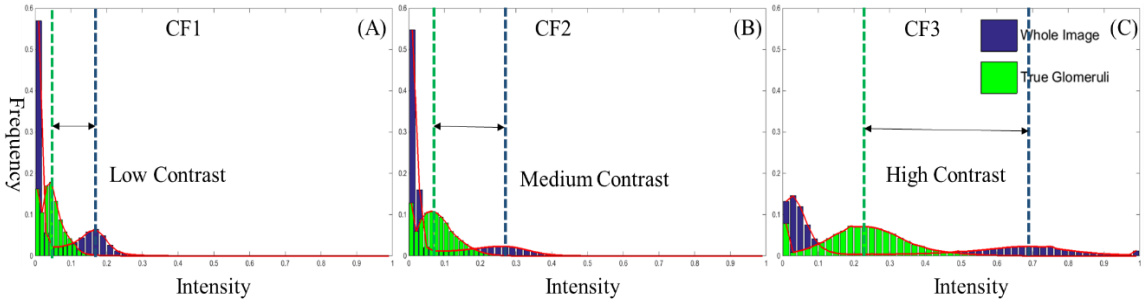

Figure 12 Intensity Frequency Histograms of Glomeruli against Whole Kidney Image from: (A) Human CF1 (B) Human CF2 (C) Human CF3. Frequency Range is $[0, 0.6]$ and the Intensity Range is $[0,1]$ in the Figure. Vertical Lines Indicate the Modes of the Intensity Distribution.

3.3.4 Discussion of HDoG Performance

In HDoG, a significant contributor to its performance is the regional features which used in the post-pruning process. The distributions of the three features are shown in Figure 13. Figure 13 (A), Figure 13 (B) and Figure 13 (C) are the distributions for the true glomeruli, while the Figure 13 (D), Figure 13 (E) and Figure 13 (F) are the distributions for the non-glomeruli from CF1, CF2 and CF3 respectively. It is evident that the true glomerular cluster had distinct pattern on the distributions of average intensity (in blue), regional blobness (in red) and regional flatness (in green). We note that the true-glomerular cluster has a higher regional blobness, a higher regional flatness and a lower average intensity compared to the non-glomerular cluster (e.g., background) as expected.

This can be explained that in the nanoparticle-enhanced MR imaging, a glomeruli is shown as a dark ellipse, which has higher blobness, higher flatness and lower intensity measures.

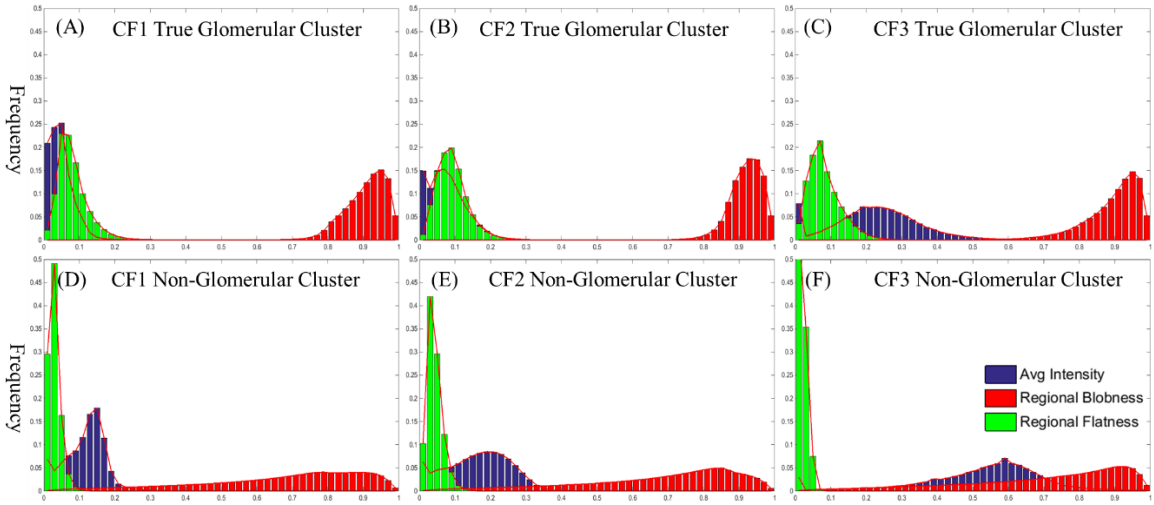

Figure 13 Frequency Histograms of Average Intensity, Regional Blobness and Regional Flatness for Human CF1, CF2 and CF3 Kidney 3D MR Images. (A)-(C) True Glomerular Cluster Frequency Histograms for Human CF1, CF2 and CF3 respectively. (D)-(F) Non-glomerular Cluster Frequency Histograms for Human CF1, CF2 and CF3 respectively. Frequency Range was $[0, 0.5]$ and the x-axis Range was $[0,1]$ in the Figure.

While promising, we do observe some limitations of HDoG in two ways: missed detection and false positive detection (Figure 14). For example, some glomeruli in the images may have shown discontinuity due to the imaging acquisition artifact. As in our **Definition 1**, a blob is a connected region, this discontinuity will lead to the missed detection even after applying the DoG transformation (smooth process). A second possible reason of missed detection may be the results of post-pruning process where some true glomeruli may be clustered to the non-glomeruli group (False negative, also known as Type II error). Similar issue from the post-pruning unsupervised algorithm is

the false positive (also known as Type I error). Both types of errors are acknowledged to be a universal issue with clustering algorithms (as compared to supervised learning algorithms). A possible solution is to explore a semi-supervised algorithm requiring partial labeling as training. Moreover, additional prior knowledge inputs, (for example, glomeruli mostly lie on the area of kidney cortex), may help improve detection accuracy.

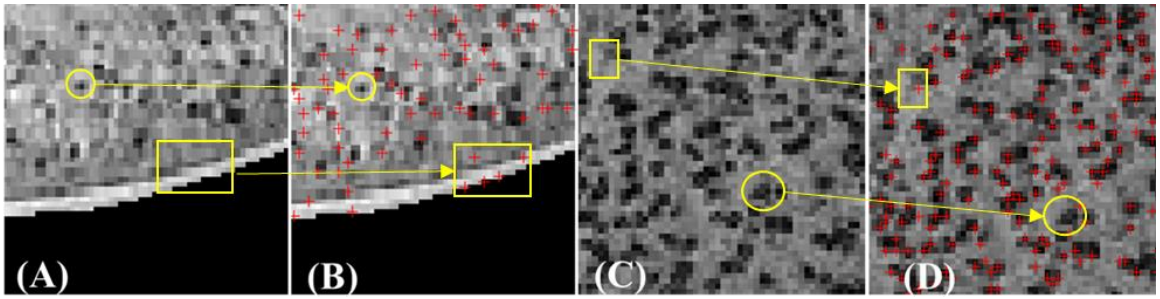

Figure 14 Glomerular Segmentation Results for 3D MR Images of Rat CF1 Kidney and Human CF1 Kidney (Part of the Slice on Figure 11): (A) Part of Slice for Rat CF1. (B) Identification Results for (A), where Identified Glomeruli Centers are Marked in Red-cross. (C) Part of Slice for Human CF1. (D) Identification Results for (C), where Identified Glomeruli Centers are Marked in Red-cross. Circles Show the Error of Missed Detection while the Rectangles Show the Error of False Positive Detection

3.4 Conclusion

In this study, we propose a computationally efficient blob detector, called the Hessian-based difference of Gaussians (HDoG) detector, for labeling small objects, such as renal glomeruli, on 3D nanoparticle-enhanced MR images. First, images are smoothed via DoG approximation. The Hessian analysis is then conducted, which (1) theoretically, ensures that all true glomeruli are contained within the glomerulus candidate pool and (2) precisely generates the region of each glomerular candidate. In the following step, we derive three regional features including: a novel regional blobness, a novel regional flatness, and a commonly used regional average intensity. The three features are fed into

the variational Bayesian Gaussian mixture model, to remove false glomeruli from the candidate pool. To initially test the performance of HDoG, we compare the HDoG results with three known detectors, LoG, gLoG and HLoG using 15 pathological images and 200 fluorescence light microscopy images, both in 2D. We observe that HDoG outperforms LoG, while is comparable to gLoG and HLoG with the least computing time. We then assess how well HDoG identifies kidney glomeruli on 3D MR images of six rat kidneys and three human kidneys. The glomerular counts obtained with HDoG are compared with counts obtained by design-based stereology and acid maceration. Overall, we conclude that HDoG automatically and accurately labels glomeruli in a reasonable time frame (<5 minutes per rat MRI; <4 hours per human MRI). To our knowledge, this is the first report of a robust, unsupervised technique to detect a magnetically labeled structure in 3D from MR images. Thus, HDoG may be a powerful preclinical or clinical tool to noninvasively detect labeled molecular structures with MRI in tissue.

CHAPTER 4

COMPUTATIONAL EFFICIENT VARIATIONAL BAYESIAN GAUSSIAN MIXTURE MODELS VIA CORESET

In Chapter 2, we have successfully developed HLoG for small blob detection using 2D images. HDoG is then proposed for kidney glomeruli detection in 3D MR images in Chapter 3. It is noted in both HLoG and HDoG, Hessian pre-segmentation is able to identify all the true positive blobs (2D and 3D) with false positives. To remove the false positive detections, a reliable and efficient post pruning process is needed. In HLoG and HDoG detectors, we have proposed the application of Variational Bayesian Gaussian mixture model (VBGMM) to serve the purpose (Zhang et al. 2014). However, it is noted that the training of VBGMM model takes a long time, especially when dealing large scale of images. To address this issue and to detect large numbers of small blobs, an efficient post-pruning algorithm-weighted VBGMM via coresets is proposed in this chapter. Specifically, a new coreset construction algorithm is first proposed. The weights derived from the coreset are applied to VBGMM, resulting weighted VBGMM as the post-pruning algorithm. The experiments tested on the same dataset as in Chapter 3, the six rat and three human kidney images show that with similar detecting performance, weighted VBGMM via coresets is about 20 times faster than the classic VBGMM.

4.1 Introduction

Gaussian mixture models (GMM), an unsupervised learning algorithm in data mining and pattern recognition, has been commonly used in medical imaging research. One standard method to estimate the GMM parameters is to employ Expectation-

Maximization (EM) algorithm to maximize the likelihood function, known as MLGMM. Two criticisms of MLGMM exist: (1) it prefers to choose complex models to fit the dataset thus may lead to overfitting problem (Attias 2000); (2) its log likelihood function may lead to singularities when the Gaussian component 'collapses' into a single point (Bishop 2006). To address these issues, many researchers have adopted Bayesian approach (Attias 2000, Jordan et al. 1999, Bishop 2006, Rasmussen 1999) in the GMM estimation. In the Bayesian framework, the whole class of models rather than a single model is considered to evaluate the posterior distributions of the GMM parameters. The predictions from all the models, weighted averaging by their posterior are calculated as the overall prediction. As a result, the over-fitting and singularity problems are resolved. Apparently, how to derive the posterior probability is the key here. However, the large number of model parameters may prohibit numerical integration. In addition, the high complexity of the posterior distribution may have no closed-form analytical solutions. Therefore, it is often impossible to compute the exact posterior distribution. Instead, approximation approaches are taken to evaluate the posterior distribution.

In general, the approximation approaches are categorized in two classes: sampling-based methods and variational inference methods. The sampling based approximation, such as Markov chain Monte Carlo (MCMC), requires intensive computational resources which limits it applications to large scale problems. The second limitation is it requires the pre-assumptions of the sampling distribution which in practice may not be easily available, and even if the distribution can be derived, it may not be accurate(Attias 2000, Bishop 2006). The variational inference methods, on the other hand, can provide analytical approximations to the posterior distributions over the parameters

with less computational burden. On the purpose of minimizing the Kullback-Leibler (KL) divergence, the posterior distributions can be estimated by maximizing the variational free energy (the variational lower bound). This procedure can be computed by an iterative approach which is very similar to Expectation-Maximization (EM) whose convergence is guaranteed. One example is VBGMM. While promising, for big dataset, the convergent time for model fitting is still of concern. For example, it requires 1 hour ~ 4 hours for human kidney glomeruli segmentation (see Chapter 3.3). To speed up the computation of VBGMM, a promising strategy is to reduce training data size for posteriors estimation.

Lately, an emerging data reduction technique is coreset which has attracted great attentions (Agarwal et al. 2005, Feldman et al. 2011, Chen 2009). A coreset is a small subset of the original data, on which the solution has guaranteed approximation to that from the full dataset. Since coreset is initially introduced in the computational geometry field, its first applications have been focused on geometric approximations where the distances between/among the data points are studied. Example algorithms are K-means, K-median (Har-Peled et al. 2004). (Feldman et al. 2011) explores the application of coreset for maximum likelihood Gaussian mixture model (MLGMM), a generative modeling approach rooted on distribution of the dataset. It is worth noting that given the deterministic Gaussian parameters, the Gaussian probability density function can be represented as Euclidean distances. Since this will allow the expression of the likelihood function of a data point in a pure geometric manner, MLGMM indeed adapts the coresets construction in a similar manner as that of K-means.

However, in variational inference setting, i.e. when the Gaussian parameters become stochastic variables, the application of coreset construction from MLGMM

would flatten the dataset distribution and increase the variability of the stochastic variables. Therefore, to directly implement the coreset construction algorithm from MLGMM in VBGMM is questionable (please refer to Chapter 4.5 Discussion for expanded discussions). To adopt the coreset in VBGMM, a new coreset construction algorithm is proposed in this chapter, which is trying to sharpen the data distribution while preserving the component modes in the mixture model. The posterior distributions are estimated via the new coresets with guaranteed performance on the predictions compared to the use of the whole dataset. To test the performance of VBGMM via the proposed new coresets, same two sets of 3D MR images discussed in Chapter 3: six rat kidney MR images and three human kidney MR images are evaluated. The results are compared with (1) HDoG using the VBGMM with uniform sampling dataset and (2) HDoG using the VBGMM on the full dataset. It is observed that the HDoG using the VBGMM with the coreset is improved greatly with much less computational cost which makes it possible to adopt HDoG for the clinical *in vivo* studies.

The rest of the chapter is organized as following: Chapter 4.2 first introduces the basics of coresets followed by our proposed coreset construction algorithm. Chapter 4.3 discusses the enhanced VBGMM with the new coresets. The comparison experiments are detailed in Chapter 4.4, with the discussion in Chapter 4.5 and conclusions drawn in Chapter 4.6.

4.2 Coreset and Construction

4.2.1 Coreset Basics

Coreset is a weighted subset of original data which guarantees that models fitting using the coreset will provide similar performance on fitting using the original dataset. Given a full dataset D, a cost function f, the subset $C \subset D$ is an ε-coreset of D if for every test data x, we have:

$$(1-\varepsilon) f(D \mid x) \leq f(C \mid x) \leq (1+\varepsilon) f(D \mid x) \tag{4.1}$$

Where ε is small, $0 \leq \varepsilon \leq 1$.

This equation means that for a test point x, the function f evaluated at the coreset C has $(1 \pm \varepsilon)$ approximation to the function f evaluated at the whole dataset D. There are many ways to construct coreset C. The key is to partition the whole dataset space into smaller disjoint regions and the representative data point(s) from each region is (are) chosen forming the coreset. Sariel Har-Peled et. al (2004) propose an exponential grid method to construct coreset for K-median and K-means algorithms. Specifically, the whole dataset is partitioned into hierarchical grids and one arbitrary point is picked from each non-empty grid cell to construct the coreset, with the weight assigned based on the number of points in each cell. Instead of exponential grid partition, Chen (2006) partitions the whole dataset into disjoined ring sets and random sampling was applied to each ring set, with weights assigned based on the number of points in each cell for the multiple points selected. Please note earlier work on coresets has focused on forming the coreset using the geometric distances between the data points. It would be interesting to explore the application of coreset to generative models (e.g., GMM) which are built upon the data distributions, and the distribution functions are more complicated for cost function evaluations. One notable research effort is from Feldman et al. (Feldman et al. 2011) which first proved the MLGMM adopts the same coreset constructs as the way as

that in the K-means. They showed that with the coreset constructed via adaptive sampling for *k* clusters (as shown in Table 14), given the testing Gaussian parameters θ, the log-likelihood function $L(C|\theta)$ on coreset C has $(1 \pm \varepsilon)$ approximation on full data D (i.e. $L(D|\theta)$), with probability $1 - \delta$ (δ is a small number), that is:

$$(1-\varepsilon)L(D|\theta) \le L(C|\theta) \le (1+\varepsilon)L(D|\theta) \tag{4.2}$$

Where θ is the set of parameters for GMM.

Table 14 Coreset Construction via Adaptive Sampling (Feldman et al. 2011)

Input: Data set $D, \varepsilon, \delta, k$

Output: Coreset $C = \{(\gamma(x_1), x_1), \dots, (\gamma(x_{|C|}), x_{|C|})\}$

1. $D' \leftarrow D; B \leftarrow \emptyset; \beta = 10dkln\left(\frac{1}{\delta}\right)$;
2. **While** $|D'| > \beta$ **do**
3. Sample set S of β points uniformly from D';
4. Remove $\lceil |D'|/2 \rceil$ points $x \in D'$ closest to S (the minimum distance) from D';
5. Set $B \leftarrow B \cup S$;
6. **Set** $B \leftarrow B \cup D'$;
7. **For** each $b \in B$ **do** $D_b \leftarrow$ the points in D whose closest point in *B* is *b*. Ties broken arbitrary;
8. **For** each $b \in B$ and $x \in D_b$ **do**
9. $m(x) \leftarrow \left\lceil \frac{5}{|D_b|} + \frac{dist(x,B)^2}{\sum_{x' \in D} dist(x',B)^2} \right\rceil$
10. Pick a non-uniform random sample C of $10\left\lceil dk|B|^2 ln(\frac{1}{\delta})/\varepsilon^2 \right\rceil$ points from D with probability $p(x) \leftarrow m(x)/\sum_{x' \in D} m(x')$ for point $x \in D$
11. **For** each $x' \in C$ **do** $\gamma(x') \leftarrow \frac{1}{|C|p(x')}$;

Table 14 shows the algorithm to construct coreset for MLGMM (Feldman et al. 2011). At first, it selects the representative data points and adds them into set *B* iteratively by uniformly sampling data from the original dataset, and removing its neighborhood (line2-line6). In practice, the β can be set accordingly based on the data. Then based on

the set B as seeds, the whole dataset is partitioned into Voronoi diagram (see Figure 15 (C) for example), which is a partitioning of a set into subsets based on distance to points in a specific subset (line 7). Each data point is then assigned with the probability to be selected in coreset (line 8). The probability is proportional to the distance between the point and the seed (point in B) within the region and penalized by the inverse of region density, as shown in Figure 15 (B). To make the expectation of the weights over whole dataset is equal to 1, i.e. $\sum_{x\in D} p(x)\gamma(x) = 1$, the weight of the point in coreset is then assigned as the inverse of the probability normalized by the number of points in coreset to compensate the non-uniform probabilities (line 11). In practice, the number of points sampled for coreset can be set based on the problems as referred in (Feldman et al. 2011).

It is noted Feldman's approach attempts to remove denser data points but keep sparser data points with high probabilities. As a result, the distribution of the coreset varies from the full dataset and becomes flatter. This approach may perform well when dealing with discriminative clustering algorithms, since the boundary (extreme) points of clusters are of the main interests of the study. For generative model, given the deterministic parameters, MLGMM can be treated as soft K-means clustering (Hastie et al. 2009) and adapts the coresets construct in a similar manner as that of K-means (Feldman et al. 2011). However, when constructing coreset in variational inference setting, those parameters are stochastic variables, and the cost function is the variational lower bound that makes the application of coreset construction from Table 14 questionable (see the discussion part). Therefore, this research proposes a new coreset construct for GMM with variational inference which requires preserving the data distribution from the full dataset in the coreset.

4.2.2 Proposed Coreset Construction

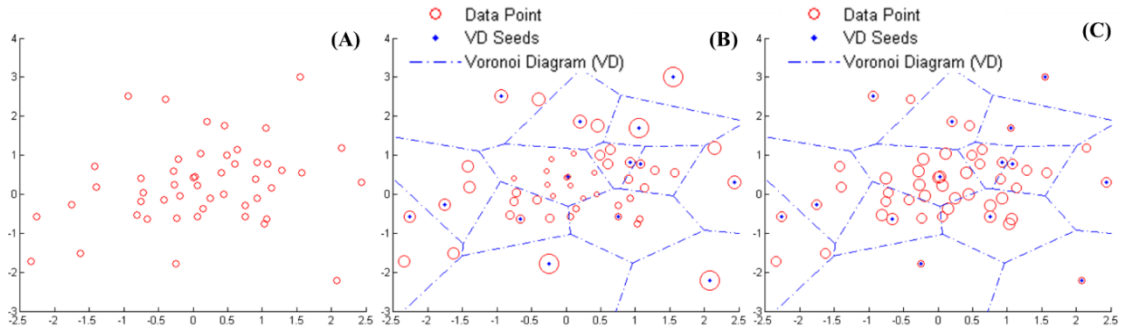

Figure 15 Coreset Construction Comparison. (A) Full Dataset (B) Coreset Construction using (Feldman et al. 2011) (C) Coreset Construction using Proposed Algorithm. Radius of Point Reflects the Probability of the Data Point to be Sampled

As discussed in Chapter 4.2.1, to make the coreset applicable to GMM in variational inference setting, the modes of data distributions need to be maintained instead of flattened. Here we introduce a new coreset construction algorithm that inverse the probability assigned to each point (highlighted in line 9, Table 15) as shown in Figure 15 (C). This inverse operation will sharpen the data distributions to reduce the variability of Gaussian variables and enhance the discernibility among the modes for clustering. Indeed, the expectation of means remains the same as the previous since the weights are the normalized inverse of probabilities leading the weighted averages the same.

Table 15 Proposed Coreset Construction for Variational Bayesian Framework

Input: Data set $D, \varepsilon, \delta, k$

Output: Coreset $C = \{(\gamma(x_1), x_1), \ldots, (\gamma(x_{|C|}), x_{|C|})\}$

1. $D' \leftarrow D; B \leftarrow \emptyset; \beta \leftarrow 10dkln\left(\frac{1}{\delta}\right);$

2. **While** $|D'| > \beta$ **do**
3. Sample set S of β points uniformly from D';
4. Remove $\lceil |D'|/2 \rceil$ points $x \in D'$ closest to S (the minimum distance) from D';
5. Set $B \leftarrow B \cup S$;
6. **Set** $B \leftarrow B \cup D'$;
7. **For** each $b \in B$ **do** $D_b \leftarrow$ the points in D whose closest point in B is b. Ties broken arbitrary;
8. **For** each $b \in B$ and $x \in D_b$ **do**
9. $\boldsymbol{m(x) \leftarrow 1.5 - \left[\frac{1}{|D_b|} + \frac{dist(x,B)^2}{\sum_{x' \in D} dist(x',B)^2}\right];}$
10. Pick a non-uniform random sample C of $10\left\lceil dk|B|^2 ln(\frac{1}{\delta})/\varepsilon^2 \right\rceil$ points from D with probability $p(x) \leftarrow m(x)/\sum_{x' \in D} m(x')$ for point $x \in D$
11. **For** each $x' \in C$ **do** $\gamma(x') \leftarrow \frac{1}{|C|p(x')}$;

With the coreset constructed by the algorithm, weighted VBGMM is built in next section and the experimental evaluations are discussed later in this chapter.

4.3 Weighted Variational Bayesian Mixture Models via Coresets

4.3.1 VBGMM Clustering

Consider GMM with *K* components, let $X = \{x_1, \dots, x_N\}$ be the set of *N* i.i.d observations with dimension d, and the latent variable, $Z = \{Z_1, \dots, Z_{NK}\}$ is introduced where $z_{nk} = 1$ indicates that X_n belongs to component k and $\sum_{i=k}^{K} Z_{nk} = 1$. Let the variables π_k, μ_k, Λ_k be the weight, mean, and precision matrix (inverse of covariance matrix) for component *K* of the mixture model respectively. Then the graphic model for GMM in Bayesian framework is shown in Figure 16.

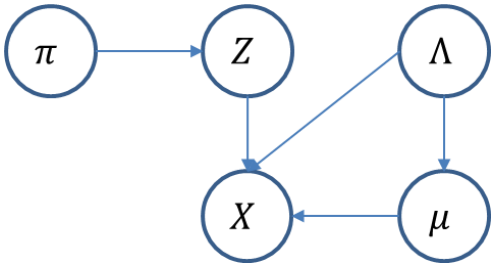

Figure 16 Directed Acyclic Graph (DAG) for VBGMM

Based on the directed acyclic graph, we have

$$p(X,Z,\pi,\Lambda,\mu)=p(X|Z,\mu,\Lambda)\,p(Z|\pi)\,p(\pi)\,p(\mu|\Lambda)\,p(\Lambda) \tag{4.3}$$

Where

$$p(X|Z,\mu,\Lambda)=\prod_{n=1}^{N}\prod_{k=1}^{K}N\left(x_n|\mu_k,\Lambda_k^{-1}\right)^{Z_{nk}} \tag{4.4}$$

$$p(Z|\pi)=\prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{Z_{nk}} \tag{4.5}$$

$$p(\pi)=A(\alpha_0)\prod_{k=1}^{K}\pi_k^{\alpha_0-1} \tag{4.6}$$

$$p(\mu|\Lambda)=\prod_{k=1}^{K}N(\mu_k\,|\,m_0,(\beta_0\Lambda_k)^{-1}) \tag{4.7}$$

$$p(\Lambda)=\prod_{k=1}^{K}W(\Lambda_k\,|\,W_0,\nu_0) \tag{4.8}$$

$A(\alpha_0)$ is the normalized constant for the Dirichlet distribution, $p(\Lambda)$ is Wishart distribution governing the precision matrix. Those prior distributions are selected either naturally or at the convenience of computation, details can be found in (Bishop 2006).

The key challenge of variational Bayesian approach is to compute the posterior distribution of the model variables i.e. $p(Z,\pi,\Lambda,\mu|X)$. In most cases, especially in GMM, it is computational prohibitive to calculate posterior distributions directly. Therefore, the

variational inference is employed to approximate the posterior distribution $p(Z,\pi,\Lambda,\mu|X)$ with other computable distribution $q(Z,\pi,\Lambda,\mu)$ by minimizing their Kullback-Leibler (KL) divergence, which is:

$$KL(q||p) = -\int q(Z,\pi,\Lambda,\mu) ln \frac{p(Z,\pi,\Lambda,\mu|X)}{q(Z,\pi,\Lambda,\mu)} d(Z,\pi,\Lambda,\mu) \tag{4.9}$$

The log marginal probability of *X* can also be decomposed as:

$$lnp(X) = L(q) + KL(q||p) \tag{4.10}$$

Where $L(q) = \int q(Z,\pi,\Lambda,\mu) ln \frac{p(X,Z,\pi,\Lambda,\mu)}{q(Z,\pi,\Lambda,\mu)} d(Z,\pi,\Lambda,\mu)$

Since given the observations, the model evidence $p(X)$ is pre-determined. The minimization of $KL(q||p)$ leads to the maximization of variational lower bound $L(q)$. Assuming $q(\pi,Z,\Lambda,\mu)$ follows the *mean fields theory* that is $q(Z,\pi,\Lambda,\mu) = q(Z)q(\pi,\Lambda,\mu)$, considering the part $q(Z)$, we can rewrite $L(q)$ as:

$$L(q) = \int q(Z) ln\tilde{p}(X,Z) dZ - \int q(Z) lnq(Z) dZ + const \tag{4.11}$$

Where $ln\tilde{p}(X,Z) = E_{(\pi,\Lambda,\mu)}[lnp(X,Z,\pi,\Lambda,\mu)]$, and *const* consists of items that are independent of $q(Z)$.

To maximize $L(q)$ with regards to $q(Z)$, we have:

$$lnq^*(Z) = E_{(\pi,\Lambda,\mu)}[lnp(X,Z,\pi,\Lambda,\mu)] + const \tag{4.12}$$

$$q^*(Z) = \frac{\exp\left(E_{(\pi,\Lambda,\mu)}[lnp(X,Z,\pi,\Lambda,\mu)]\right)}{\int \exp\left(E_{(\pi,\Lambda,\mu)}[lnp(X,Z,\pi,\Lambda,\mu)]\right) dZ} \tag{4.13}$$

Similarly, to maximize $L(q)$ with regards to $q(\pi,\Lambda,\mu)$, we have

$$q^*(\pi,\Lambda,\mu)=\frac{\exp\left(E_Z\left[lnp(X,Z,\pi,\Lambda,\mu)\right]\right)}{\int\exp\left(E_Z\left[lnp(X,Z,\pi,\Lambda,\mu)\right]\right)d(\pi,\Lambda,\mu)} \tag{4.14}$$

From the equations, it is learned $q(Z)$ and $q(\pi,\Lambda,\mu)$ depend on each other. To maximize $L(q)$, we can fix one term to compute the other. Therefore, the maximization of $L(q)$ leads to an iterative procedure which is similar to EM algorithm, the variational inference treatment for GMM from (Bishop 2006) is listed in Table 16.

Table 16 Variational Inference for Gaussian Mixture Models (Bishop 2006)

Inputs: $\alpha_0,\beta_0,v_0,m_0,W_0^{-1}$, full dataset X and initial labels Z_{nk}
Outputs: labels $Z_n = argmax_k(r_{nk})$, hyper parameters: $\{\alpha_k,\beta_k,v_k,m_k,W_k^{-1}\}$

1. **Set** $r_{nk}=Z_{nk}$
2. **While** $L(q)$ not converged **do**
3. **M-Setp:** calculate the posterior $q^*(\pi)$ and $q^*(\mu,\Lambda)$
4. First calculate:

$$N_k=\sum_1^N r_{nk}\,,\bar{x}_k=\frac{1}{N_k}\sum_1^N r_{nk}\,x_n,S_k=\frac{1}{N_k}\sum_1^N r_{nk}\,(x_n-\bar{x}_k)^T(x_n-\bar{x}_k)$$

5. Then we can calculate
6. $q^*(\pi_k)=Dir(\pi_k|\alpha_k)$
7. $q^*(\mu_k,\Lambda_k)=N(\mu_k|m_k,(\beta_k\Lambda_k)^{-1})W(\Lambda_k|W_k,v_k)$
8. by
9. $\alpha_k=\alpha_0+N_k$
10. $\beta_k=\beta_0+N_k$
11. $m_k=\frac{1}{\beta_k}(\beta_0 m_0+N_k\bar{x}_k)$
12. $W_k^{-1}=W_0^{-1}+N_kS_k+\frac{\beta_0 N_k}{\beta_0+N_k}(\bar{x}_k-m_0)(\bar{x}_k-m_0)^T$
13. $v_k=v_0+N_k$
14. **E-Step:** calculate the posterior $q^*(Z_n)=\prod_{k=1}^K r_{nk}^{Z_{nk}}$, where
15. $r_{nk}=\frac{\rho_{nk}}{\sum_{j=1}^K\rho_{nk}}$
16. $\rho_{nk}=\exp\left(E\left[\ln\pi_k\right]+E\left[\ln\left|\Lambda_k\right|\right]-\frac{d}{2}\ln(2\pi)-\frac{1}{2}E_{\mu_k,\Lambda_k}\left[\left(x_n-\mu_k\right)^T\Lambda_k\left(x_n-\mu_k\right)\right]\right)$
17. **Re-evaluate** $L(q)$

From the table above, we know that the quantity r_{nk} plays the key role in VBGMM cluster. In discrete setting, $E\left[Z_{nk}\right]=r_{nk}$. Since Z_{nk} is a binary variable, r_{nk} actually is the probability of data point x_n belonging to the component k.

4.3.2 Weighted VBGMM via Coresets

To use the coreset in VBGMM, the coreset weight needs to be considered into weighted averaging. In the MLGMM setting, Feldman et al. (Feldman et al. 2011) assign the coreset weight $\gamma(x_n)$ to the probability r_{nk}^{ML} of data point x_n belonging to the component *k*, that is:

$$r_{nk}^{ML}=\gamma\left(x_n\right)r_{nk}^{ML}=\gamma\left(x_n\right)\frac{\pi_k N\left(x_n|\mu_k,\Lambda_k^{-1}\right)}{\sum_{k=1}^{K}\pi_k N\left(x_n|\mu_k,\Lambda_k^{-1}\right)} \tag{4.15}$$

Taking a further step in comparing r_{nk}^{ML} in MLGMM and r_{nk} in VBGMM, we have:

$$r_{nk}^{ML}\propto\pi_k\left|\Lambda_k\right|^{\frac{1}{2}}\exp\left(-\left(x_n-\mu_k\right)^T\Lambda_k\left(x_n-\mu_k\right)\right)$$

$$r_{nk}\propto\tilde{\pi}_k\tilde{\Lambda}_k\exp\left(-\frac{d}{2\beta_k}-\frac{v_k}{2}\left(x_n-m_k\right)^T\mathrm{W}_k\left(x_n-m_k\right)\right)$$

Where $\tilde{\pi}_k=E\left[\ln\pi_k\right],\tilde{\Lambda}_k=E\left[\ln\left|\Lambda_k\right|\right]$. Accordingly, in VBGMM setting, the coreset weight $\gamma(x_n)$ naturally takes probability r_{nk} into account, that is $r_{nk}=\gamma(x_n)r_{nk}=\gamma(x_n)\frac{\rho_{nk}}{\sum_{j=1}^{K}\rho_{nk}}$. This modification makes the coreset weight takes effect on all the hyper parameter estimations. The details of weighted VBGMM via coresets algorithm are listed in Table 17.

Table 17 Weighted VBGMM via Coresets Clustering

Training Process:

Inputs: $\{\alpha_0, \beta_0, v_0, m_0, W_0^{-1}\}$, coreset $C = \{(\gamma(x_1), x_1), \dots, (\gamma(x_{|C|}), x_{|C|})\}$, initial labels Z_{nk}

Outputs: trained hyper parameters: $\{\alpha_k, \beta_k, v_k, m_k, W_k^{-1}\}$

1. **Set** $r_{nk} = Z_{nk}$
2. **While** $L(q)$ not converged **do**
3. **M-Setp:** calculate the posterior $q^*(\pi)$ and $q^*(\mu, \Lambda)$
4. First calculate:

$$N_k = \sum_{1}^{|C|} r_{nk}\,, \bar{x}_k = \frac{1}{N_k}\sum_{1}^{|C|} r_{nk}\, x_n, S_k = \frac{1}{N_k}\sum_{1}^{|C|} r_{nk}\,(x_n - \bar{x}_k)^T(x_n - \bar{x}_k)$$

5. Then we can calculate
6. $q^*(\pi_k) = Dir(\pi_k|\alpha_k)$
7. $q^*(\mu_k, \Lambda_k) = N(\mu_k|m_k, (\beta_k\Lambda_k)^{-1})W(\Lambda_k|W_0, v_0)$
8. by
9. $\alpha_k = \alpha_0 + N_k$
10. $\beta_k = \beta_0 + N_k$
11. $m_k = \frac{1}{\beta_k}(\beta_0 m_0 + N_k\bar{x}_k)$
12. $W_k^{-1} = W_0^{-1} + N_k S_k + \frac{\beta_0 N_k}{\beta_0 + N_k}(\bar{x}_k - m_0)(\bar{x}_k - m_0)^T$
13. $v_k = v_0 + N_k$
14. **E-Step:** calculate the posterior $q^*(Z_n) = \prod_{k=1}^{K} r_{nk}^{Z_{nk}}$, where
15. $\boldsymbol{r_{nk} = \gamma(x_n)\frac{\rho_{nk}}{\sum_{j=1}^{K}\rho_{nk}}}$
16. $\rho_{nk} = \exp\left(E\left[\ln \pi_k\right] + E\left[\ln|\Lambda_k|\right] - \frac{d}{2}\ln(2\pi) - \frac{1}{2}E_{\mu_k,\Lambda_k}\left[(x_n - \mu_k)^T\,\Lambda_k\,(x_n - \mu_k)\right]\right)$
17. **Re-evaluate** $L(q)$

Testing Process:

Inputs: $\{\alpha_0, \beta_0, v_0, m_0, W_0^{-1}\}$, full dataset X and $\{\alpha_k, \beta_k, v_k, m_k, W_k^{-1}\}$

Outputs: labels $Z_n = argmax_k(r_{nk})$

18. **Calculate** the posterior $q^*(Z_n) = \prod_{k=1}^{K} r_{nk}^{Z_{nk}}$, where
19. $r_{nk} = \frac{\rho_{nk}}{\sum_{j=1}^{K}\rho_{nk}}$
20. $\rho_{nk} = \exp\left(E\left[\ln \pi_k\right] + E\left[\ln|\Lambda_k|\right] - \frac{d}{2}\ln(2\pi) - \frac{1}{2}E_{\mu_k,\Lambda_k}\left[(x_n - \mu_k)^T\,\Lambda_k\,(x_n - \mu_k)\right]\right)$

To implement weighted VBGMM in clustering data, two processes are needed as shown in Table 17. The training process trains the hyper parameters of VBGMM via coreset, and the testing process is to cluster the whole dataset with fitted VBGMM. With much less data to train the VBGMM, VBGMM converges very fast, thus the higher

efficiency can be achieved. The experimental evidences showing the efficiency and reliable performance of weight VBGMM clustering are discussed in next section.

4.4 Experimental Validation of Weighted VBGMM via Coresets

To validate the performance of weighted VBGMM clustering, two experiments of clustering the glomeruli were performed. In the first experiment, weighted VBGMM via coresets was implemented to cluster the glomeruli candidates and identify the true glomeruli group on six 3D rat kidney MR images (matrix size: 256×256×256). The experiments were done in Matlab 2014b on a Windows platform with Intel Xeon 2.0 GHz CPU and 32GB of memory. The results were compared with VBGMM via full dataset and VBGMM via uniform sampling. In the second experiment, three much larger datasets (matrix size: 512×512×896) from three 3D human kidney MR images were studied. The results of weighted VBGMM were also compared with other two VBGMMs.

4.4.1 Evaluation on Rat Kidney MR Images

Six 3D MR images of rat kidneys were studied in this experiment. The image data were first processed by HDoG without post pruning. Therefore, glomeruli candidates were generated with three features average including intensity, regional blobness and regional flatness were extracted. The glomeruli candidates were clustered using weighted VBGMM via coresets, VBGMM via uniform sampling and full data VBGMM. The training sample size was chosen from 500, 1000, 2000, 5000, 8000 and 10000 for weighted VBGMM via coresets and VBGMM via uniform sampling, then the full datasets were tested. Each data sample was run 10 times. The results from full dataset

trained VBGMM were chosen as benchmark. The variational lower bounds *L(q)* for weighted VBGMM and VBGMM via uniform sampling are shown in Figure 17.

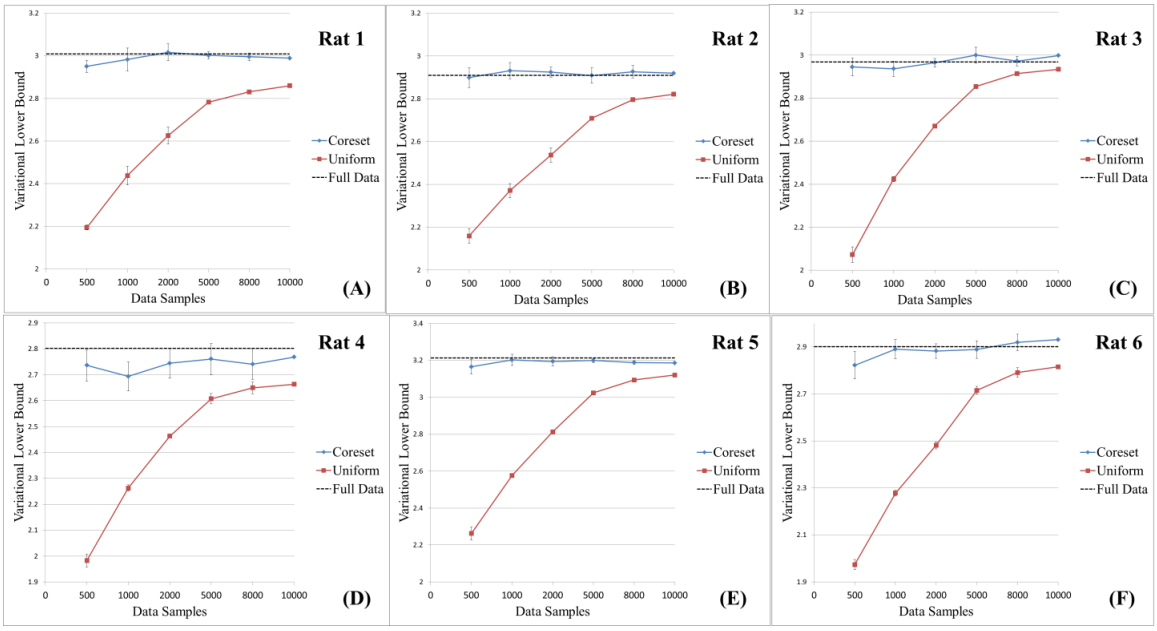

Figure 17 Variational Lower Bound Comparison on Weighted VBGMM via Coresets, VBGMM via Uniform Sampling and VBGMM via Full Data

Table 18 Comparison of VBGMM on Full dataset, Coreset and Uniform Sampling (based on 5000 Sample Size) on Rat Kidney Data

Evaluation	Method	Rat 1 (86111)*	Rat 2 (94084)*	Rat 3 (94186)*	Rat 4 (81908)*	Rat 5 (81950)*	Rat 6 (92338)*
Time (seconds)	Full Dataset	20.907	18.684	10.647	10.711	13.095	11.083
	Coreset	1.678±0.236	1.951±0.427	1.86±0.791	1.664±0.294	1.399±0.155	1.798±0.431
	Uniform	1.205±0.437	1.354±0.373	0.728±0.237	0.74±0.217	0.596±0.069	0.583±0.108
Variational Lower Bound	Full Dataset	3.008	2.910	2.969	2.802	3.213	2.901
	Coreset	3.002±0.039	2.909±0.023	2.972±0.037	2.749±0.061	3.208±0.03	2.904±0.027
	**p-value	0.300	0.442	0.127	0.012	0.310	0.347
	Uniform	2.783±0.04	2.709±0.033	2.914±0.005	2.61±0.01	3.023±0.007	2.709±0.016
	***p-value	0.000	0.000	0.000	0.000	0.000	0.000
Glomeruli Counts	Full Dataset	29484	34460	27051	35295	31196	35248
	Coreset	30723±2949	34894±4675	30553±3150	33006±3354	32127±1358	34620±7265
	**p-value	0.108	0.388	0.003	0.030	0.029	0.395
	Uniform	42473±11641	38888±7970	31241±365	30843±2530	31523±223	31696±288

***p-value	0.003	0.056	0.000	0.000	0.001	0.000

**number of all glomeruli candidates (data instances)*
***t-test on coreset with null hypothesis that coreset mean value is equal to full dataset value*
****t-test on uniform sampling with null hypothesis that uniform mean value is equal to full dataset value*

The results show that the variational lower bound on weighted VBGMM via Coreset is very close to the full dataset VBGMM and has no significant difference ($\alpha = 0.01$), which is empirically proved the inequalities $(1-\varepsilon)L(q|X) \leq L(q|C) \leq (1+\varepsilon)L(q|X)$ still hold in variational Bayesian framework. When the sample size changes, weighted VBGMM via coreset remains stable, which indicates its independency on sample size. In contrast, the VBGMM via uniform sampling highly depends on the sample size. When the sample size increases, it is more close to the population variational lower bound.

Besides the variational lower bound, to evaluation the efficiency and detectability of weighted VBGMM via coreset on glomeruli identification, glomeruli counts and computational time were also examined to check how close that coreset and uniform sampling can approximate the performance of VBGMM on full dataset. The results are shown in Table 18.

Due to the limit of page size, only the results from 5000 (~ 6% of the full dataset) sample size are listed here. From Table 18, we can see that for 5000 sample size, both coresets and uniform sampling can highly reduce the computation time dramatically (coreset construction time included). Regarding to the approximation to full dataset, coreset approach has better approximation than uniform sampling on all rats when checking the statistical *p-values* (the smaller, the more significant different compared to

full dataset). Readers can also check the full comparison in Appendix B. For the sample size less than 5000, coreset performed much better than uniform sampling. For other sample sizes, in most cases, coreset achieved better approximation than uniform sampling.

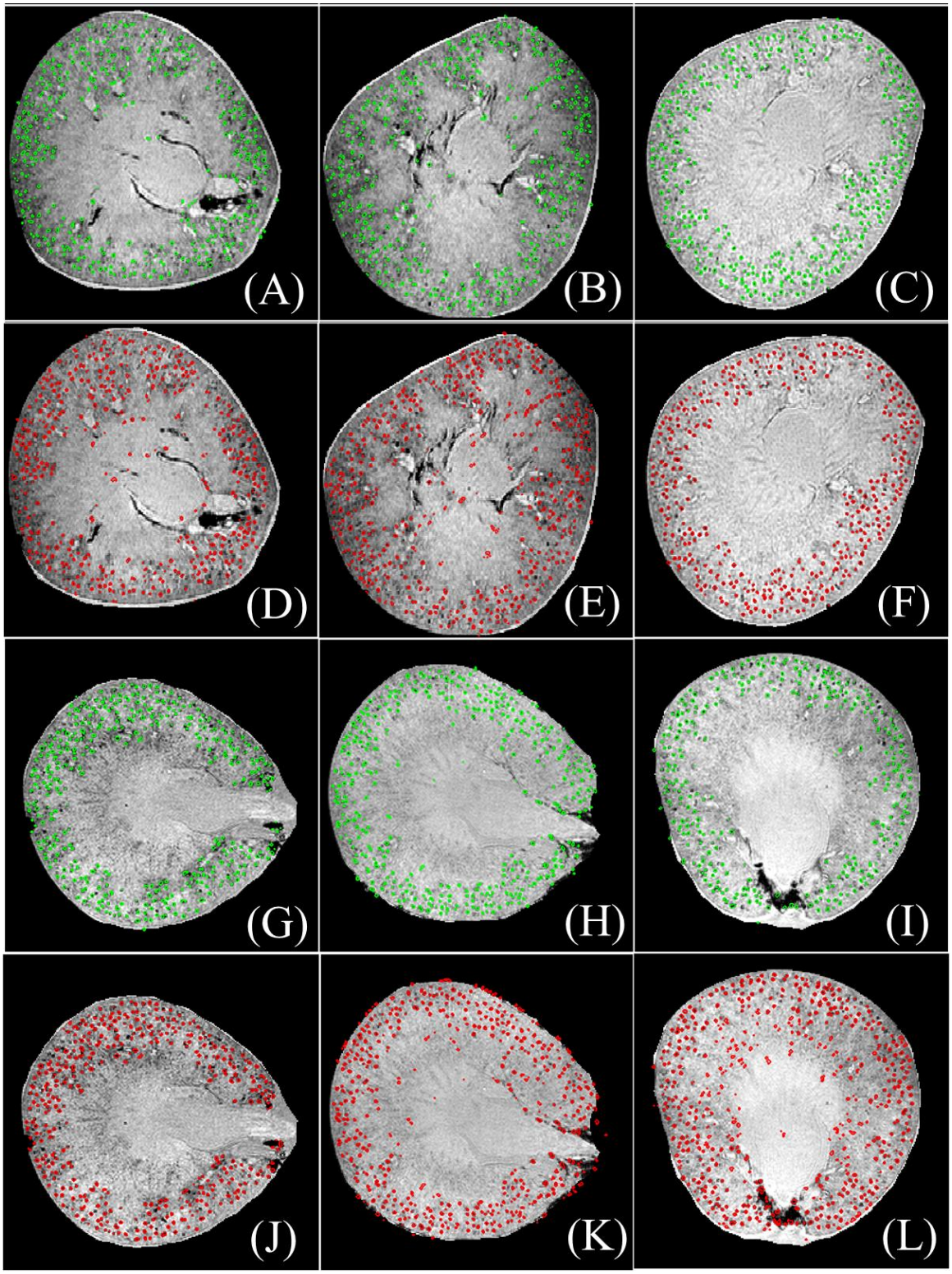

Figure 18 Glomerular Identification Results for 3D MR Images of Rat Kidneys using Coresets (5000 Samples): (A–C) Segmentation Results from Slice 100 for Rat CF1, CF2, and CF3 Kidneys using Full Dataset. (D–F) Segmentation Results from Slice 100 for Rat

CF1, CF2, and CF3 Kidneys using Coreset. (G–I) Segmentation Results from Slice 150 for Rat CF4, CF5, and CF6 Kidneys using Full Dataset. (J–L) Segmentation Results from Slice 150 for Rat CF4, CF5, and CF6 Kidneys using Coreset.

Figure 18 shows the identification results from rat kidneys using weighted VBGMM via coreset compared to the VBGMM using full dataset. Glomeruli colored in green are the results using full dataset while the glomeruli colored in red are from the coreset using 5000 samples. By visually checking those results, there are no big differences between the results using full dataset and using coreset for rat datasets. This shows the result that the weighted VBGMM via coreset is an accurate approximation that can be used to detect the glomeruli in a much shorter time.

4.4.2 Evaluation on Human Kidney MR Images

To further investigate the computation efficiency and approximation performance of weighted VBGMM via coreset, three much larger data were examined from human kidney 3D MR images. The image data were also processed by HDoG in Chapter 3 without post pruning. Glomeruli candidates were therefore generated with features average intensity, regional blobness and regional flatness. The glomeruli candidates were post-pruned using weighted VBGMM via coresets, VBGMM via uniform sampling and full data VBGMM. The training sample size was chosen from 500, 1000, 2000, 5000, 8000 and 10000 for weighted VBGMM and VBGMM via uniform sampling, and predictions were made on full dataset. Each data sample was run 10 times in this experiment. The clustering results from full dataset trained VBGMM were consider as

golden standard for performance evaluation. The variational lower bounds *L(q)* for weighted VBGMM and VBGMM via uniform sampling are shown in Figure 19.

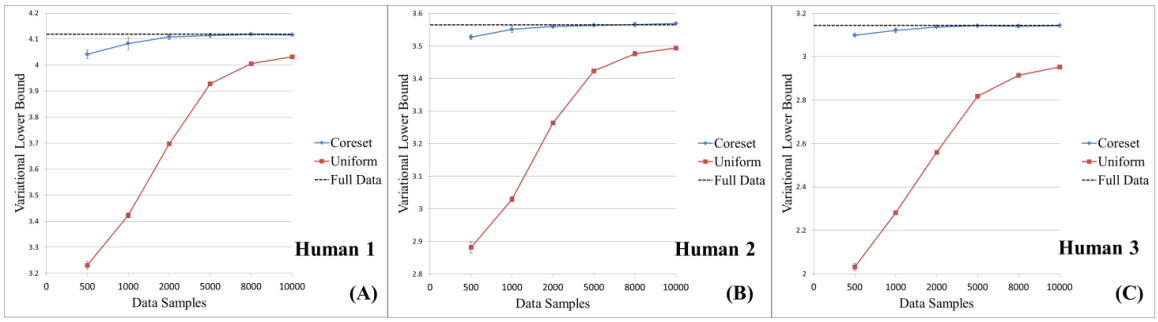

Figure 19 Varional Lower Bound Comparison on Weighted VBGMM via Coresets, VBGMM via Uniform Sampling and VBGMM via Full Data on Human Kidney Dataset

Figure 19 also confirmed that the inequalities $(1-\varepsilon)L(q|X) \leq L(q|C) \leq (1+\varepsilon)L(q|X)$ still hold in variational Bayesian framework. In addition, by showing the independency on sample size, weighted VBGMM via Coreset can provide good approximation even on a small portion (~ 0.01% - 0.2%) of full dataset compared to uniform sampling.

Computation time, glomeruli counts, features of identified glomeruli were also examined to evaluate the performance on weighted VBGMM via coresets comparing to VBGMM via uniform sampling and full dataset. The results are shown in Table 19.

Table 19 Comparison of VBGMM on Full Dataset, Coreset and Uniform Sampling (based on 5000 Sample Size) on Human Kidney Data

Evaluation	Method	Human 1 (4886637)*	Human 2 (3912414)*	Human 3 (4816653)*
Time (seconds)	Full Dataset	1744.600	891.232	648.299
	Coreset	41.274 ±0.597	34.035 ±1.413	39.868 ±0.455
	Uniform	*3.663 ±0.098*	*3.281 ±0.083*	*5.05 ±1.439*

Variational Lower Bound	Full Dataset	4.119	3.565	3.145
	Coreset	4.114 ±0.011	3.563 ±0.004	3.143 ±0.007
	**p-value	0.120	0.071	0.248
	Uniform	3.928 ±0.005	3.423 ±0.005	2.818 ±0.005
	***p-value	0.000	0.000	0.000
Glomeruli Identified	Full Dataset	1241144	711397	1370193
	Coreset	1287396 ±225512	787960 ±187001	1276104 ±163616
	**p-value	0.266	0.114	0.051
	Uniform	1893917 ±194489	1805765 ±84993	1082084 ±24045
	***p-value	0.000	0.000	0.000

** number of all glomeruli candidates (data instances)*
***t-test on coreset with null hypothesis that coreset mean value is equal to full dataset value*
****t-test on uniform sampling with null hypothesis that uniform mean value is equal to full dataset value*

The same as previous section, only the results from 5000 sample size are listed. Table 19 shows that for 5000 sample size, both weighted VBGMM via coresets and VBGMM via uniform sampling are very computational efficient. Unlike the first set of experiment, sample size of 5000 instances is an extremely small portion (0.1%) of the full dataset. At this level, weighted VBGMM can still prove a very good approximation and has no significant difference ($\alpha = 0.01$) compared to the full dataset in terms of both variational lower bound and glomeruli counts. However, as seen in the table, the VBGMM via uniform sampling lost its detectability on glomeruli identification. On other sample size, readers can check with the Appendix B, which also shows weighted VBGMM via coreset has much better approximation than VBGMM via uniform sampling.

Figure 20 shows the identification results using weighted VBGMM via coresets compared to the VBGMM using full dataset. Glomeruli colored in green are the results using full dataset while the glomeruli colored in red are from the coreset using 5000

samples. By visually checking those results, there are no big differences between the results using full dataset and using coreset. This also confirms that the weighted VBGMM via coreset is an accurate approximation that can be used to detect the glomeruli in a much shorter time.

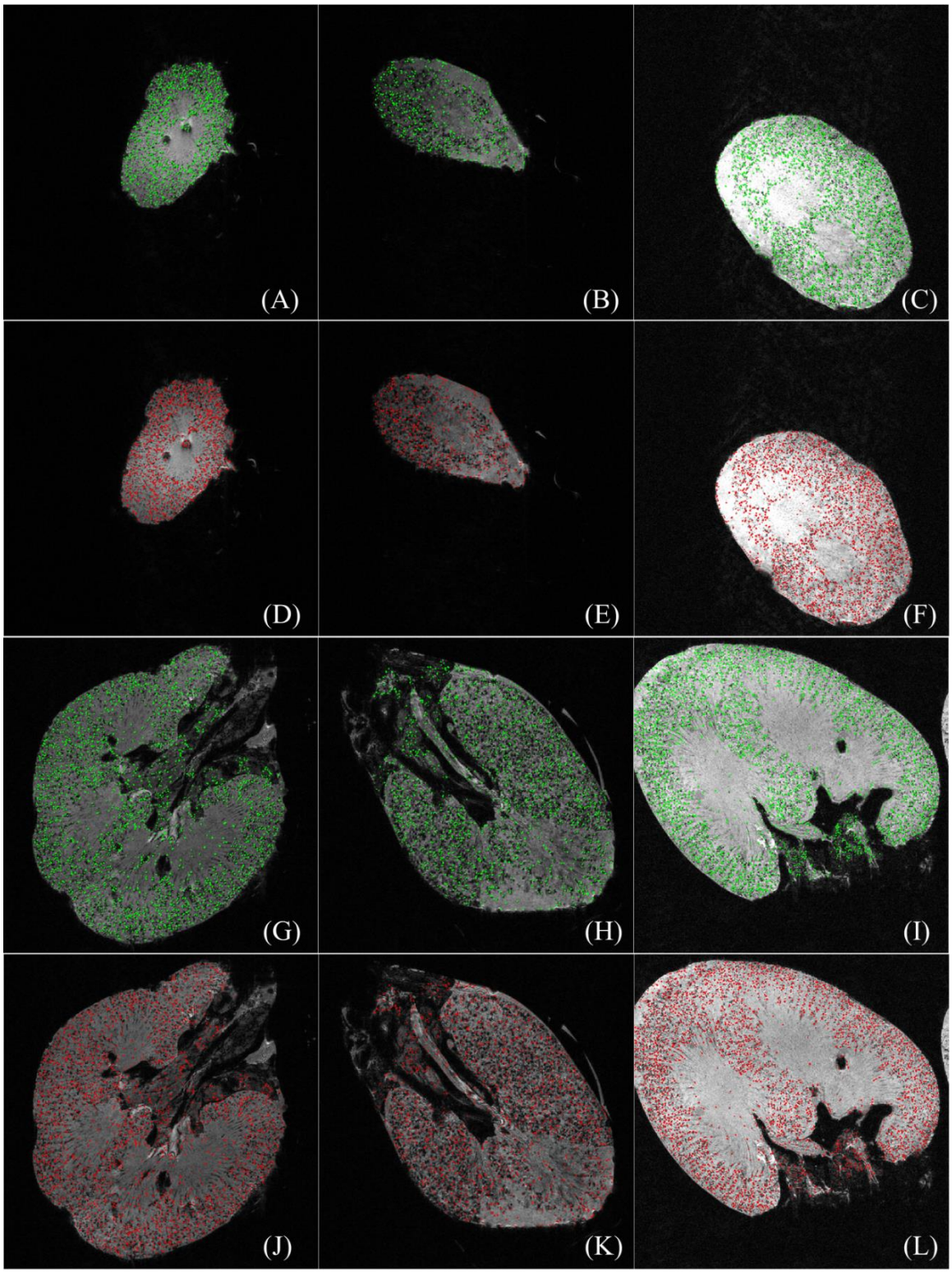

Figure 20 Glomerular Identification Results for 3D MR Images of Human Kidneys using Coresets (5000 Samples): (A–C) Segmentation Results from Slice 100 for Human CF1,

CF2, and CF3 Kidneys using Full Dataset. (D–F) Segmentation Results from Slice 100 for Human CF1, CF2, and CF3 Kidneys using Coreset. (G–I) Segmentation Results from Slice 500 for Human CF1, CF2, and CF3 Kidneys using Full Dataset. (J–L) Segmentation Results from Slice 500 for Human CF1, CF2, and CF3 Kidneys using Coreset.

4.5 Discussion

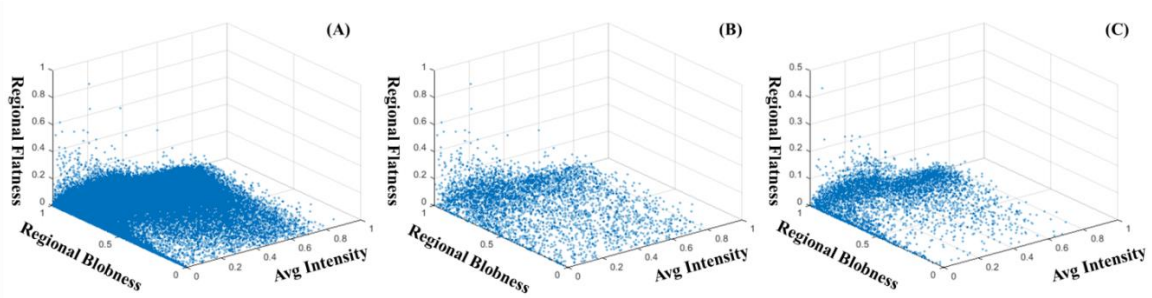

Figure 21 Data Samples from Rat 1 Dataset. (A) Full Dataset (B) Coreset using (Feldman et al. 2011) (C) Coreset using Our Method.

As mentioned before, Feldman's coreset construction algorithm tends to remove the points from dense regions and keep the points from sparse regions. This algorithm first partitions the whole dataset into regions and then selects the boundary points with high probability for sparse region. This choosing "sparse of sparse" points strategy results in flattening the data distribution as seen in Figure 21 (B). Therefore, when the Gaussian parameters become random variables, the increased variability caused by the flattened coreset would lead to inaccurate predictions on full dataset. Instead, after dataset partitioned into disjoint regions, our coreset construction algorithm tries to select the dense points compared to sparse points within the region. By preserving the distribution modes, this choosing "dense of dense" points strategy will sharpen the data distribution (Figure 21 (C)). Thus the distribution modes are well modeled in the variational

inference setting, resulting in better approximation on predictions on full dataset. The Rat 1 dataset is checked to compare the two coreset construction algorithms, and the results are shown in Table 20.

Table 20 Comparison Results of Feldman Coreset and Our Coreset on Rat 1 Dataset

Sample Size	Method	Converge Steps	Time (s)	Variational Lower Bound	Glomeruli Counts
86111	Full Dataset	230	20.907	3.008	29484
500	Feldman Coreset	118±103	1.091±0.29	2.935±0.079	36876±9566
	Our Coreset	88±33	1.056±0.266	2.95±0.125	35655±5847
1000	Feldman Coreset	100±50	1.103±0.172	3.013±0.063	42509±5937
	Our Coreset	100±53	1.153±0.192	2.983±0.028	31120±2982
2000	Feldman Coreset	73±18	1.128±0.09	3.056±0.008	43969±2450
	Our Coreset	126±97	1.404±0.444	3.016±0.054	33724±4826
5000	Feldman Coreset	70±7	1.318±0.065	3.067±0.002	44029±1704
	Our Coreset	103±30	1.678±0.236	3.002±0.039	30723±2949
8000	Feldman Coreset	90±19	1.794±0.224	3.07±0.002	44191±1949
	Our Coreset	127±35	2.262±0.31	2.995±0.017	30564±1907
10000	Feldman Coreset	80±7	1.827±0.094	3.071±0.001	44539±897
	Our Coreset	136±43	2.582±0.457	2.989±0.018	30929±2403

Table 20 states though the variational lower bounds of weighted VBGMM using Feldman's coreset are close to those on full dataset, the glomeruli counts are very different, showing that VBGMM using Feldman's fails to cluster the full dataset. This result also indicates that even if the cost function of coreset has guaranteed approximation on full dataset, the model trained by coreset does not guarantee to maintain the similar clustering ability. It needs to be cautious when trying to employ coreset in model building. Further steps need to be done to evaluation the clustering ability rather than just taking advantages of its fast computing. In this chapter, though the empirical study showed that the weighted VBGMM trained by our proposed coreset maintains the similar clustering ability as the VBGMM trained using full dataset, and

their variational lower bounds (cost functions) are very close, more theoretic work needs to be done for solid proof. The theoretic analysis will be conducted as our future work.

4.6 Conclusion

In this chapter, to speed up the model fitting of VBGMM, which is the post-pruning algorithm of HDoG, a new schema – weighted VBGMM via coresets is proposed. While preserving the similar glomeruli detectability, the weighted VBGMM via coresets takes the advantage of coresets for their efficient computations. To employ the coreset in variational Bayesian framework, a new coreset construction algorithm is also proposed aiming to preserve the distribution modes and to reduce the variations by sharpening the distribution. Two experiments on glomeruli detections were performed to evaluate the performance of weighted VBGMM via coresets, by comparing the VBGMM via uniform sampling. The results show the weighted VBGMM via coresets can much reduce the computation time comparing to the full dataset but with similar performance in glomeruli detections. In addition, unlike the VBGMM via uniform sampling, the weighted VBGMM via coresets is independent of sample size and can provide much better approximation in glomeruli detections.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Summary

In this dissertation, we first propose a novel blob detector, termed Hessian-based Laplacian of Gaussian (HLoG) detector to identify 2D small blobs in medical images. As an implementation, cell images are studied, which showed the good performance of HLoG in 2D images. However, for 3D large images, HLoG is computational expensive when using LoG kernel to convolve with higher dimension images. As an extension of HLoG in 3D space, Hessian-based Difference of Gaussians (HDoG) is proposed to approximate LoG kernel by using fast DoG kernel for 3D image convolution. In the HDoG, two novel efficient 3D regional features are derived that enable the HDoG to detect large-scale of small blobs in 3D medical images. To further improve the computation speed of HDoG, the post-pruning algorithm, i.e. variational Bayesian Gaussian mixture models (VBGMM) is examined, and an efficient post-pruning algorithm- weighted VBGMM via coresets is proposed which can greatly reduce the computation cost during the model training process. Figure 22 summarizes the models and methods proposed in this dissertation. All those models aim to solve the small blob detection problem but in different scales of data size. These models naturally form a framework for small blob detection in medical images for different data size as seen in Figure 22.

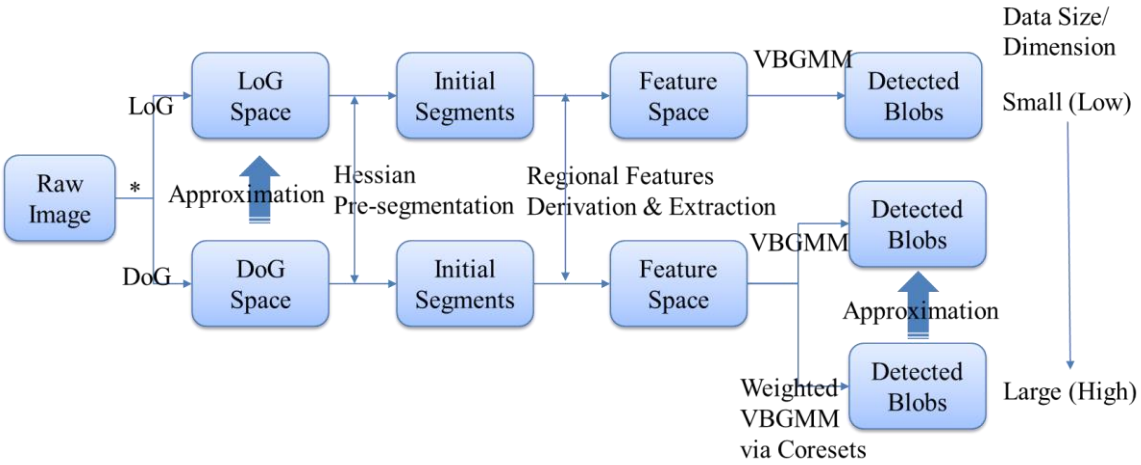

Figure 22 Framework for Small Blob Detection

Through comprehensive comparison experiments using datasets available from literature as well as our clinical partners, we conclude HLoG in general performs well for small blob detection in 2D images. Yet, the expensive computation prohibits its application to 3D images. Its extension, HDoG is demonstrated to be promising for small blob detection (e.g., glomeruli) from 3D images. Using intelligent sampling, coreset, the enhanced HDoG is capable to identify the small blobs, in this study, glomeruli, within 40 second, making it a potential tool used in preclinical and clinical study.

5.2 Discussion and Future Work

In this research, we conclude that the Hessian pre-segmentation is an important process in blob detection, it can naturally segment the blobs and split the touched separable blob mixtures. When handling the ideal blobs (e.g., intensity distributions are Gaussians), theoretically, it can identify all the blobs. However, in practice, most blobs in the images may have discontinuities, which cause the frequent changes of intensity

convexity within a blob. Therefore, simply applying the Hessian pre-segmentation algorithm leads to unsatisfactory results. A smoothing procedure shall be employed prior to the Hessian implementation. In this dissertation, two smoothing techniques are explored: Laplacian of Gaussian (LoG) and its fast approximation Difference of Gaussians (DoG). The smoothing parameter of those two techniques is associated with the blob scales. When all the blobs are small and similar in size, which is the basic assumption in this dissertation, one optimum scale (smoothing parameters) can be selected for all the blobs. In this case, HLoG and HDoG algorithms can provide good performance on blob detection. However, when blobs are not identical in size, the selected single smoothing parameter will cause over-smoothing for some small blobs and under-smoothing for some large blobs. For over-smoothing blobs, some blobs will merge together and are not separable. For under-smoothing blobs, the discontinuities will cause the intensity convexity variations within a blob. In this case, HLoG and HDoG algorithms may not perform well as expected. This situation drives the needs to develop a dynamic smoothing algorithm as our future work, that is, the smoothing parameters shall be adjusted according to the varied sizes from the blobs. This generalization to HLoG and HDoG will enable the detectors to be applicable to identify blobs with different sizes.

Additionally, we only studied the application of coreset to VBGMM empirically. It is our intention to investigate the theoretical aspects of the coreset and propose its application to generative models in general.

REFERENCES

Agarwal, Pankaj K, Sariel Har-Peled, and Kasturi R Varadarajan. 2005. "Geometric approximation via coresets." *Combinatorial and computational geometry* no. 52:1-30.

Al-Kofahi, Y., W. Lassoued, W. Lee, and B. Roysam. 2010. "Improved Automatic Detection and Segmentation of Cell Nuclei in Histopathology Images." *Biomedical Engineering, IEEE Transactions on* no. 57 (4):841-852. doi: 10.1109/TBME.2009.2035102.

Attias, Hagai. 2000. "A variational Bayesian framework for graphical models." *Advances in neural information processing systems* no. 12 (1-2):209-215.

Bay, Herbert, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. 2008. "Speeded-up robust features (SURF)." *Computer vision and image understanding* no. 110 (3):346-359.

Beeman, Scott C, Luise A Cullen-McEwen, Victor G Puelles, Min Zhang, Teresa Wu, Edwin J Baldelomar, John Dowling, Jennifer R Charlton, Michael S Forbes, and Amanda Ng. 2014. "MRI-based glomerular morphology and pathology in whole human kidneys." *American Journal of Physiology-Renal Physiology* no. 306 (11):F1381-F1390.

Beeman, Scott C., Min Zhang, Lina Gubhaju, Teresa Wu, John F. Bertram, David H. Frakes, Brian R. Cherry, and Kevin M. Bennett. 2011. "Measuring glomerular number and size in perfused kidneys using MRI." *American Journal of Physiology - Renal Physiology* no. 300 (6):F1454-F1457. doi: 10.1152/ajprenal.00044.2011.

Bennett, K. M., H. Zhou, J. P. Sumner, S. J. Dodd, N. Bouraoud, K. Doi, R. A. Star, and A. P. Koretsky. 2008. "MRI of the basement membrane using charged nanoparticles as contrast agents." *Magnetic Resonance in Medicine* no. 60 (3):564-74. doi: 10.1002/mrm.21684.

Bergeest, J. P., and K. Rohr. 2012. "Efficient globally optimal segmentation of cells in fluorescence microscopy images using level sets and convex energy functionals." *Medical Image Analysis* no. 16 (7):1436-44. doi: 10.1016/j.media.2012.05.012.

Bernardis, Elena, and Stella X Yu. 2011. "Pop out many small structures from a very large microscopic image." *Medical Image Analysis* no. 15 (5):690-707.

Bertram, John F, Mary C Soosaipillai, Sharon D Ricardo, and Graeme B Ryan. 1992. "Total numbers of glomeruli and individual glomerular cell types in the normal rat kidney." *Cell and tissue research* no. 270 (1):37-45.

Bertram, John F. 1995. "Analyzing Renal Glomeruli with the New Stereology." In *International Review of Cytology*, edited by W. Jeon Kwang and Jarvik Jonathan, 111-172. Academic Press.

Bishop, Christopher M. 2006. *Pattern recognition and machine learning*. Vol. 1: springer New York.

Bonvalet, Jean-Pierre, Monique Champion, Frida Wanstok, and Guy Berjal. 1972. "Compensatory renal hypertrophy in young rats: Increase in the number of nephrons." *Kidney Int* no. 1 (6):391-396.

Carreira-Perpiñán, Miguel Á, and Christopher KI Williams. 2003. On the number of modes of a Gaussian mixture. Paper read at Scale Space Methods in Computer Vision.

Chen, Ke. 2009. "On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications." *SIAM Journal on Computing* no. 39 (3):923-947.

Cullen-McEwen, L. A., R. N. Douglas-Denton, and J. F. Bertram. 2012. "Estimating total nephron number in the adult kidney using the physical disector/fractionator combination." *Methods in Molecular Biology* no. 886:333-50. doi: 10.1007/978-1-61779-851-1_30.

Cullen-McEwen, L. A., M. M. Kett, J. Dowling, W. P. Anderson, and J. F. Bertram. 2003. "Nephron number, renal function, and arterial pressure in aged GDNF heterozygous mice." *Hypertension* no. 41 (2):335-40.

Feldman, Dan, Matthew Faulkner, and Andreas Krause. 2011. Scalable training of mixture models via coresets. Paper read at Advances in Neural Information Processing Systems, 2011.

Feldman, Dan, and Michael Langberg. 2011. A unified framework for approximating and clustering data. Paper read at Proceedings of the forty-third annual ACM symposium on Theory of computing.

Frangi, AlejandroF, WiroJ Niessen, KoenL Vincken, and MaxA Viergever. 1998. "Multiscale vessel enhancement filtering." In *Medical Image Computing and Computer-Assisted Interventation — MICCAI' 98*, edited by WilliamM Wells, Alan Colchester and Scott Delp, 130-137. Springer Berlin Heidelberg.

Har-Peled, Sariel, and Soham Mazumdar. 2004. On coresets for k-means and k-median clustering. Paper read at Proceedings of the thirty-sixth annual ACM symposium on Theory of computing.

Hastie, Trevor, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. 2009. *The elements of statistical learning*. Vol. 2: Springer.

Hoy, Wendy E, John F Bertram, Rebecca Douglas Denton, Monika Zimanyi, Terence Samuel, and Michael D Hughson. 2008. "Nephron number, glomerular volume, renal disease and hypertension." *Current opinion in nephrology and hypertension* no. 17 (3):258-265.

Jordan, MichaelI, Zoubin Ghahramani, TommiS Jaakkola, and LawrenceK Saul. 1999. "An Introduction to Variational Methods for Graphical Models." *Machine Learning* no. 37 (2):183-233. doi: 10.1023/A:1007665907178.

Kong, H., H. C. Akakin, and S. E. Sarma. 2013. "A Generalized Laplacian of Gaussian Filter for Blob Detection and Its Applications." *Cybernetics, IEEE Transactions on* no. PP (99):1-15. doi: 10.1109/TSMCB.2012.2228639.

Lempitsky, Victor, and Andrew Zisserman. 2010a. "Learning to count objects in images."

Lempitsky, Victor, and Andrew Zisserman. 2010b. Learning to count objects in images. Paper read at Advances in Neural Information Processing Systems.

Leutenegger, Stefan, Margarita Chli, and Roland Yves Siegwart. 2011. BRISK: Binary robust invariant scalable keypoints. Paper read at Computer Vision (ICCV), 2011 IEEE International Conference on.

Lindeberg, Tony. 1993a. "Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention." *International Journal of Computer Vision* no. 11 (3):283-318. doi: 10.1007/BF01469346.

Lindeberg, Tony. 1993b. *Scale-space theory in computer vision*: Springer.

Lindeberg, Tony. 1998. "Feature detection with automatic scale selection." *International journal of computer vision* no. 30 (2):79-116.

Lowe, DavidG. 2004. "Distinctive Image Features from Scale-Invariant Keypoints." *International Journal of Computer Vision* no. 60 (2):91-110. doi: 10.1023/B:VISI.0000029664.99615.94.

Loy, Gareth, and Alexander Zelinsky. 2003. "Fast radial symmetry for detecting points of interest." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* no. 25 (8):959-973.

Meyer, Carl D. 2000. *Matrix analysis and applied linear algebra*: Siam.

Mikolajczyk, Krystian, and Cordelia Schmid. 2004. "Scale & Affine Invariant Interest Point Detectors." *International Journal of Computer Vision* no. 60 (1):63-86. doi: 10.1023/B:VISI.0000027790.02288.f2.

Moon, Woo Kyung, Yi-Wei Shen, Min Sun Bae, Chiun-Sheng Huang, Jeon-Hor Chen, and Ruey-Feng Chang. 2013. "Computer-aided tumor detection based on multi-

scale blob detection algorithm in automated breast ultrasound images." *Medical Imaging, IEEE Transactions on* no. 32 (7):1191-1200.

Prastawa, Marcel, Elizabeth Bullitt, Sean Ho, and Guido Gerig. 2004. "A brain tumor segmentation framework based on outlier detection." *Medical Image Analysis* no. 8 (3):275-283. doi: http://dx.doi.org/10.1016/j.media.2004.06.007.

Rasmussen, Carl Edward. 1999. The infinite Gaussian mixture model. Paper read at NIPS.

Salden, Alfons H., Bart M. ter Haar Romeny, Max A. Viergever, Luc M. J. Florack, and Jan J. Koenderink. 1991. "Differential Geometric Description of 3D Scalar Images." *Internal Report 3DCV* no. 91-05.

Sanchez, C. I., M. Niemeijer, I. Isgum, A. Dumitrescu, M. S. Suttorp-Schulten, M. D. Abramoff, and B. van Ginneken. 2012. "Contextual computer-aided detection: improving bright lesion detection in retinal images and coronary calcification identification in CT scans." *Med Image Anal* no. 16 (1):50-62. doi: 10.1016/j.media.2011.05.004.

Takacs, G., V. Chandrasekhar, S. S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod. 2013. "Fast Computation of Rotation-Invariant Image Features by an Approximate Radial Gradient Transform." *Image Processing, IEEE Transactions on* no. 22 (8):2970-2982. doi: 10.1109/TIP.2012.2230011.

Tuytelaars, Tinne, and Krystian Mikolajczyk. 2008. "Local invariant feature detectors: a survey." *Found. Trends. Comput. Graph. Vis.* no. 3 (3):177-280. doi: 10.1561/0600000017.

Zhang, M., T. Wu, and K. Bennett. 2014. "Small Blob Identification in Medical Images Using Regional Features from Optimum Scale." *IEEE Trans Biomed Eng*. doi: 10.1109/TBME.2014.2360154.

Zhang, Min, Teresa Wu, and Kevin M. Bennett. 2015. A novel Hessian based algorithm for kidney glomeruli detection in 3D MRI. Paper read at Proc. SPIE 9413, Medical Imaging 2015, at Orlando, FL.

APPENDIX A

HESSIAN PRESEGMENTATION ON GAUSSIAN BLOBS

Assumption 1: the intensity function of blob is rotationally symmetric Gaussian.

Assumption 2: All blobs are identical in size and intensity function.

Consider two-blob mixture:

$$L\left(x, y; \sigma^2\right) = \frac{1}{2} f_1 + \frac{1}{2} f_2$$
$$= \frac{1}{2}\frac{1}{2\pi\sigma^2}\exp\left(-\frac{\left(x-\mu_{1x}\right)^2+\left(y-\mu_{1y}\right)^2}{2\sigma^2}\right)+\frac{1}{2}\frac{1}{2\pi\sigma^2}\exp\left(-\frac{\left(x-\mu_{2x}\right)^2+\left(y-\mu_{2y}\right)^2}{2\sigma^2}\right)$$

Where f_1 is the intensity function of blob 1 with centroid (μ_{1x}, μ_{1y}) and bandwidth related to size σ; f_2 is the intensity function of blob 2 with centroid (μ_{2x}, μ_{2y}) and bandwidth related to size σ.

Let $E^2 = \left(\mu_{2x} - \mu_{1x}\right)^2 + \left(\mu_{2y} - \mu_{1y}\right)^2$, we have ***Lemma 1*** showing the condition that blobs *i*,*j* are seperatable.

Lemma 1: In order to be able to split, the Euclidean distance between two blobs (Gaussian modes) satisfies $E^2 > 4\sigma^2$.

Proof sketch.

To find the stationary points of $L(x, y; \sigma^2)$, we have

$$\begin{aligned}\frac{\partial L\left(x, y; \sigma^2\right)}{\partial x} &= 0 \\ \frac{\partial L\left(x, y; \sigma^2\right)}{\partial y} &= 0\end{aligned} \tag{5.1}$$

However, those equations are the transcendental equations that do not have analytical solution in general cases. In (Carreira-Perpiñán et al. 2003), it is proved that the number of modes (maxima) of the Gaussian mixture cannot exceed the number of

components having same covariance matrix and the modes lie in the convex hull of the components' modes. Thus we assume that the modes of the mixtures are in the format of $(x^*, y^*) = (a\mu_{1x} + (1-a)\mu_{2x}, a\mu_{1y} + (1-a)\mu_{2y})$, where $0 < a < 1$.

$$\begin{aligned} &\frac{\partial L\left(x^*, y^*; \sigma^2\right)}{\partial x} = 0 \\ &\frac{\partial L\left(x^*, y^*; \sigma^2\right)}{\partial y} = 0 \end{aligned} \Rightarrow \frac{a}{1-a}\exp\left(-Da\right) = \exp\left(-\frac{D}{2}\right) \tag{5.2}$$

Where $D = \frac{E^2}{\sigma^2}$.

Obviously, $a = \frac{1}{2}$ is one of the solutions. Let $F(a) = \frac{a}{1-a} exp(-Da) - exp\left(-\frac{D}{2}\right)$, to consider the solutions of $F(a) = 0$, then we have:

$$\frac{\partial F\left(a\right)}{\partial a} = \frac{\left(a^2 D - aD + 1\right)e^{-aD}}{\left(a-1\right)^2} \tag{5.3}$$

Case 1: If $D^2 - 4D > 0 \Rightarrow D > 4$, $\frac{\partial}{\partial a} F(a) = 0$ has two roots, that $F(a) = 0$ has three real roots, therefore, there are two modes of the mixtures lie between (μ_{1x}, μ_{1y}) and $(\frac{1}{2}(\mu_{1x} + \mu_{2x}), \frac{1}{2}(\mu_{1y} + \mu_{2y}))$, (μ_{2x}, μ_{2y}) and $(\frac{1}{2}(\mu_{1x} + \mu_{2x}), \frac{1}{2}(\mu_{1y} + \mu_{2y}))$. Moreover, $(\frac{1}{2}(\mu_{1x} + \mu_{2x}), \frac{1}{2}(\mu_{1y} + \mu_{2y}))$ is the saddle point of $L(x, y; \sigma^2)$ since the Hessian of $L(x, y; \sigma^2)$ is indefinite ($L_{xx} > 0,\ L_{xx}L_{yy} - L_{xy}L_{xy} < 0$).

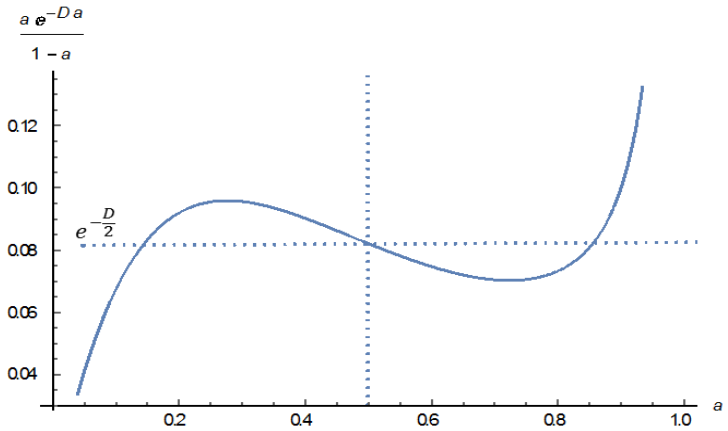

Figure 23 Three Roots for *F(a)* in Case 1

Case 2: If $D^2 - 4D \leq 0 \Rightarrow D \leq 4$, $\frac{\partial}{\partial a} F(a) \geq 0$, which means that $F(a)$ is a non-decreasing function of a, therefore, $a = \frac{1}{2}$ is the only solution to $F(a) = 0$. And the semi-Negative definite Hessian of $L(x, y; \sigma^2)$ indicates that $(\frac{1}{2}(\mu_{1x} + \mu_{2x}), \frac{1}{2}(\mu_{1y} + \mu_{2y}))$ is the global maximum (the only mode) of the mixture.

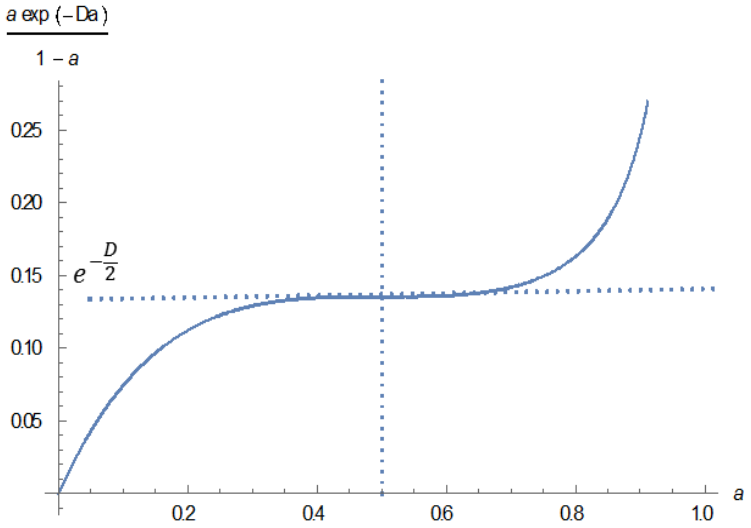

Figure 24 One Root for *F(a)* in Case 2

Thus, by following the discussion, in order to be spreadable (two Gaussian modes, ***Case 1***) we need to have $D = \frac{E^2}{\sigma^2} > 4$, which means the Euclidean distance between two blobs should satisfy $E^2 > 4\sigma^2$.■

***Lemma 2*:** the Gaussian distribution $g(x, y; t_1^2)$ convolutes a Gaussian kernel $g(x, y; t_2^2)$ is Gaussian, i.e. $g(x, y; t_1^2) * g(x, y; t_2^2) = g(x, y; t_1^2 + t_2^2)$.

***Theorem 1*:** In order to avoid merging any *blobs*, the Gaussian smoothing kernel parameter t should satisfy**:** $4t^2 < E^2 - 4\sigma^2$

Proof sketch. By ***Lemma 2***, we have $L(x, y; \sigma^2) * g(x, y; t^2) = L(x, y; \sigma^2 + t^2)$, to be spreadable, we have $E^2 > 4(\sigma^2 + t^2) \Rightarrow 4t^2 < E^2 - 4\sigma^2$. ■

***Theorem 2*:** Hessian pre-segmentation algorithm is able to split the separable blobs.

Proof sketch. From ***Lemma 1***, it is learned that the saddle point which has indefinite Hessian is between the two Gaussian modes which have Negative definite Hessian, therefore using the Hessian pre-segmentation algorithm can split separable blobs naturally in different regions that are of negative definite Hessian. ■

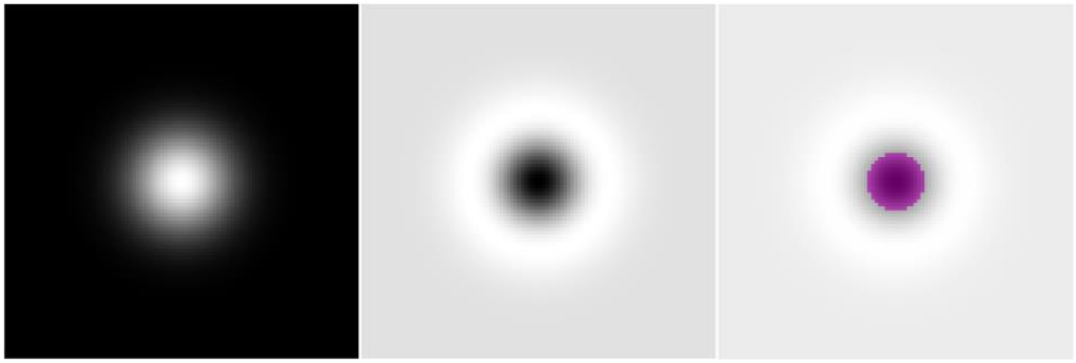

Figure 25. Blob Transformation. Left to Right: Raw blob, LoG Transformed Blob, Purple Area of Positive Definite Hessian

Lemma 3: The convex region of LoG kernel in 2D is $(x^2 + y^2) < \frac{1}{2}\left(7\sigma^2 - \sqrt{33}\sigma^2\right)$

Proof sketch.

Since we have

$$LoG\left(x, y; \sigma^2\right) = \frac{e^{-\frac{x^2+y^2}{2\sigma^2}}\left(-2\sigma^2 + x^2 + y^2\right)}{2\pi\sigma^6} \tag{5.4}$$

The convex regions of $LoG(x, y; \sigma^2)$ which is of positive definite Hessian:

$$\frac{2e^{\frac{-x^2-y^2}{2\sigma^2}}}{\pi\sigma^6} - \frac{7e^{\frac{-x^2-y^2}{2\sigma^2}}x^2}{2\pi\sigma^8} + \frac{e^{\frac{-x^2-y^2}{2\sigma^2}}x^4}{2\pi\sigma^{10}} - \frac{e^{\frac{-x^2-y^2}{2\sigma^2}}y^2}{2\pi\sigma^8} + \frac{e^{\frac{-x^2-y^2}{2\sigma^2}}x^2y^2}{2\pi\sigma^{10}} > 0 \tag{5.5}$$
$$\det\left(Hess\right) > 0$$

$$\Rightarrow \left(x^2 + y^2\right) < \frac{1}{2}\left(7\sigma^2 - \sqrt{33}\sigma^2\right) \tag{5.6}$$

Proved.■

Lemma 4: *The Gaussian convolved with LoG kernel is still LoG, i.e.* $G(x, y; \sigma^2) * LoG(x, y; t^2) = LoG(x, y; \sigma^2 + t^2)$

Proof sketch.

$$G\left(x,y;\sigma^2\right)*LoG\left(x,y;t^2\right)=G\left(x,y;\sigma^2\right)*\Delta G\left(x,y;t^2\right)$$
$$=\Delta G\left(x,y;\sigma^2\right)*G\left(x,y;t^2\right)=\Delta G\left(x,y;\sigma^2+t^2\right)$$
$$=LoG\left(x,y;\sigma^2+t^2\right).$$

Let $t_k^2=\sigma^2+t^2$, then we have

***Theorem 3*:** For a single blob, the optimum scale selected from normalized LoG

$t^*=argmax_t\dfrac{\iint_{\left(x^2+y^2\right)<\frac{1}{2}\left(7t_k^2-\sqrt{33}t_k^2\right)}t^\gamma\mid LoG\left(x,y;t_k^2\right)\mid dxdy}{\iint_{\left(x^2+y^2\right)<\frac{1}{2}\left(7t_k^2-\sqrt{33}t_k^2\right)}dxdy}$ is the same as the optimum scale

selected from $t^+=argmax_t\mid t^\gamma LoG(x,y;t_k^2)\mid$ in (Lindeberg 1998)

Sketch proof.

Since $Max\mid t^\gamma LoG(x,y;t_k^2)\mid=Max\dfrac{t^\gamma}{\pi t_k^4},\left(x=0,y=0\right)$

$$t^+=argmax_t\left(\frac{t^\gamma}{\pi t_k^4}\right)=argmax_t\frac{t^\gamma}{\pi\left(\sigma^2+t^2\right)^2}=argmax_t log\frac{t^\gamma}{\pi\left(\sigma^2+t^2\right)^2}$$

$$\Rightarrow t^{+2}=t^2=\frac{\gamma}{4-\gamma}\sigma^2$$

For the proposed part, we have

$$\iint\limits_{\left(x^2+y^2\right)<\frac{1}{2}\left(7t_k^2-\sqrt{33}t_k^2\right)}t^\gamma\mid LoG\left(x,y;t_k^2\right)\mid dxdy=t^\gamma\iint\limits_{\left(x^2+y^2\right)<\frac{1}{2}\left(7t_k^2-\sqrt{33}t_k^2\right)}\mid LoG\left(x,y;t_k^2\right)\mid dxdy$$
$$=t^\gamma\frac{\left(7-\sqrt{33}\right)e^{\frac{1}{4}\left(-7+\sqrt{33}\right)}}{2t_k^2}$$

Then we have

$$t^* = argmax_t \frac{t^\gamma \frac{\left(7-\sqrt{33}\right)e^{\frac{1}{4}\left(-7+\sqrt{33}\right)}}{2t_k^2}}{\frac{\pi}{2}\left(7-\sqrt{33}\right)t_k{}^2} = argmax_t \frac{e^{\frac{1}{4}\left(-7+\sqrt{33}\right)}t^\gamma}{\pi t_k^4}$$

$$\Rightarrow t^{*2} = t^2 = \frac{\gamma}{4-\gamma}\sigma^2$$

Therefore $t^* = t^+$. ■

Note that for individual blob, $\frac{\iint_{\left(x^2+y^2\right)<\frac{1}{2}\left(7t_k^2-\sqrt{33}t_k^2\right)} t^\gamma \mid LoG\left(x,y;t_k^2\right)\mid dxdy}{\iint_{\left(x^2+y^2\right)<\frac{1}{2}\left(7t_k^2-\sqrt{33}t_k^2\right)} dxdy}$ is

continuous case of $t^\gamma \frac{\sum_{(x,y)} \mid LoG\left(x,y;t_k^2\right)\mid I\left(x,y;t_k^2\right)}{\sum_{(x,y)} I\left(x,y;t_k^2\right)}$, where $I(x,y;t)$ is the binary

indicator such that if the pixel (x,y) of the blob has a positive definite Hessian matrix, then $I(x,y,z;t)=1$, otherwise, $I(x,y,z;t)=0$. If all the blobs have the identical size and intensity function, $t^* = \arg\max t^\gamma \frac{\sum_{(x,y)} \mid LoG\left(x,y;t_k^2\right)\mid I\left(x,y;t_k^2\right)}{\sum_{(x,y)} I\left(x,y;t_k^2\right)}$ capture the optimum size of all the blobs.

Note that the lemmas and theorems can be extended to 3D following the same way.

APPENDIX B

EXPERIMENTAL RESULTS OF CORESETS ON RAT AND HUMAN DATASETS

Table 21 Full Comparison of VBGMM on Full dataset, Coreset and Uniform Sampling on Rat 1 Kidney Data

Run	VBGMM Steps		Time		Vbound		Count		Features						Samples
									Coreset			Uniform			
	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Int	Blob	Flat	Int	Blob	Flat	
1	161	43	1.792	0.259	2.925	2.197	36476	78921	0.171	0.905	0.035	0.323	0.857	0.028	
2	72	51	0.958	0.282	3.056	2.189	44606	78008	0.368	0.931	0.027	0.324	0.861	0.028	
3	114	100	1.086	0.422	2.973	2.177	39032	80791	0.238	0.922	0.030	0.317	0.848	0.028	
4	60	147	0.901	0.516	2.650	2.206	33627	77044	0.185	0.925	0.047	0.325	0.865	0.028	
5	92	102	1.012	0.410	2.997	2.238	42177	72752	0.364	0.933	0.023	0.333	0.880	0.028	
6	94	72	1.026	0.330	2.839	2.213	25777	75176	0.554	0.908	0.014	0.329	0.872	0.028	
7	55	77	0.907	0.375	3.063	2.185	35119	78873	0.225	0.908	0.048	0.324	0.857	0.028	500
8	64	52	0.950	0.268	3.041	2.240	33461	70962	0.226	0.911	0.050	0.342	0.885	0.028	
9	59	65	0.909	0.303	2.998	2.199	38306	76564	0.199	0.910	0.040	0.324	0.867	0.029	
10	109	159	1.018	0.548	2.962	2.099	27965	84898	0.178	0.935	0.041	0.317	0.824	0.028	
Avg	88	86.8	1.056	0.371	2.950	2.194	35654.6	77398.9	0.271	0.919	0.035	0.326	0.862	0.028	
Std	33.407	40.061	0.266	0.102	0.125	0.039	5847.008	3963.658	0.122	0.012	0.012	0.007	0.017	0.000	
1	69	82	1.211	0.534	2.989	2.427	26293	71267	0.186	0.940	0.044	0.341	0.884	0.028	
2	185	122	1.463	0.605	2.996	2.414	30579	72876	0.191	0.933	0.041	0.340	0.878	0.027	
3	44	77	0.924	0.444	2.944	2.444	27856	69156	0.183	0.936	0.042	0.345	0.890	0.028	
4	202	538	1.492	1.977	2.997	2.439	31042	73108	0.187	0.931	0.042	0.333	0.878	0.027	
5	82	153	1.059	0.721	3.030	2.436	29502	72034	0.198	0.935	0.041	0.334	0.882	0.028	
6	90	68	1.040	0.396	2.991	2.455	29568	69547	0.182	0.932	0.042	0.343	0.889	0.028	
7	60	99	1.008	0.522	2.957	2.443	33665	70236	0.182	0.922	0.037	0.339	0.887	0.028	1000
8	77	104	1.054	0.560	3.011	2.450	35980	69633	0.218	0.926	0.036	0.340	0.889	0.028	
9	114	87	1.212	0.457	2.948	2.434	33934	70900	0.188	0.924	0.037	0.338	0.885	0.028	
10	79	60	1.063	0.369	2.964	2.437	32779	71915	0.176	0.922	0.038	0.338	0.882	0.028	
Avg	100.2	139	1.153	0.658	2.983	2.438	31119.8	71067.2	0.189	0.930	0.040	0.339	0.885	0.028	
Std	52.616	142.825	0.192	0.475	0.028	0.012	2982.291	1409.077	0.012	0.006	0.003	0.004	0.004	0.000	
1	90	77	1.353	0.522	3.089	2.569	35732	38633	0.218	0.914	0.047	0.181	0.913	0.039	
2	72	266	1.097	1.362	2.953	2.638	30974	65648	0.174	0.926	0.037	0.345	0.899	0.027	
3	174	278	1.631	1.444	2.996	2.573	28316	38733	0.189	0.937	0.043	0.173	0.908	0.038	
4	48	92	1.086	0.608	3.094	2.631	33529	66420	0.230	0.914	0.048	0.353	0.895	0.027	
5	146	97	1.519	0.651	2.966	2.649	35864	66639	0.190	0.917	0.037	0.346	0.897	0.027	
6	51	83	1.062	0.522	2.963	2.555	31928	39157	0.178	0.926	0.037	0.174	0.908	0.038	
7	69	327	1.071	1.694	2.965	2.650	34075	65849	0.171	0.916	0.038	0.347	0.899	0.028	2000
8	177	127	1.658	0.747	3.055	2.651	45614	66116	0.358	0.932	0.028	0.347	0.898	0.027	
9	73	94	1.100	0.601	3.040	2.667	30547	63033	0.195	0.934	0.039	0.346	0.906	0.028	
10	364	89	2.466	0.574	3.036	2.671	30665	63286	0.196	0.935	0.039	0.348	0.905	0.028	
Avg	126.4	153	1.404	0.872	3.016	2.625	33724.4	57351.4	0.210	0.925	0.039	0.296	0.903	0.031	
Std	96.567	96.871	0.444	0.445	0.054	0.043	4826.206	12831.9	0.055	0.009	0.006	0.083	0.006	0.005	
1	134	84	2.001	0.946	3.003	2.848	28149	58521	0.187	0.937	0.043	0.375	0.906	0.027	
2	95	108	1.632	1.106	3.004	2.759	27538	34594	0.185	0.938	0.043	0.159	0.912	0.039	5000

3	95	132	1.599	1.246	3.003	2.757	30177	34824	0.187	0.932	0.043	0.169	0.917	0.039	
4	73	169	1.389	1.499	2.970	2.843	33703	59409	0.180	0.921	0.038	0.372	0.905	0.027	
5	64	109	1.464	1.126	3.000	2.753	28261	35962	0.183	0.936	0.042	0.163	0.910	0.039	
6	85	86	1.565	0.891	2.968	2.756	33161	35860	0.179	0.922	0.037	0.159	0.908	0.039	
7	92	96	1.474	0.977	2.993	2.764	27019	33746	0.188	0.939	0.042	0.165	0.919	0.040	
8	109	298	1.713	2.315	2.969	2.827	33627	59941	0.177	0.920	0.037	0.377	0.902	0.027	
9	166	74	2.096	0.805	3.003	2.762	30609	36869	0.193	0.933	0.041	0.175	0.914	0.038	
10	120	123	1.848	1.139	3.104	2.756	34983	35001	0.228	0.913	0.048	0.157	0.909	0.039	
Avg	103.3	127.9	1.678	1.205	3.002	2.783	30722.7	42472.7	0.189	0.929	0.041	0.227	0.910	0.035	
Std	30.273	65.847	0.236	0.437	0.039	0.040	2948.953	11640.94	0.015	0.009	0.003	0.102	0.005	0.006	
1	111	96	2.127	1.360	3.006	2.829	29587	34142	0.192	0.934	0.043	0.164	0.916	0.039	
2	106	120	2.081	1.485	3.007	2.826	30231	33654	0.189	0.932	0.042	0.159	0.915	0.039	
3	98	84	1.953	1.263	2.970	2.834	33358	34052	0.181	0.923	0.038	0.163	0.916	0.039	
4	119	100	2.183	1.430	2.970	2.824	32997	34340	0.182	0.924	0.037	0.165	0.916	0.039	
5	171	72	2.684	1.071	3.007	2.828	29297	34998	0.186	0.934	0.042	0.168	0.915	0.039	
6	192	72	2.820	1.071	3.007	2.835	30028	34203	0.192	0.933	0.042	0.172	0.920	0.039	
7	118	82	2.235	1.275	3.002	2.834	28449	33890	0.185	0.935	0.043	0.163	0.916	0.039	8000
8	88	80	1.912	1.206	2.972	2.833	33285	34709	0.177	0.920	0.037	0.170	0.918	0.039	
9	162	93	2.532	1.199	3.006	2.835	28728	33441	0.186	0.935	0.042	0.162	0.917	0.039	
10	109	84	2.093	1.177	3.007	2.831	29677	35197	0.188	0.934	0.042	0.167	0.914	0.038	
Avg	127.4	88.3	2.262	1.254	2.995	2.831	30563.7	34262.6	0.186	0.930	0.041	0.165	0.916	0.039	
Std	34.821	14.530	0.310	0.139	0.017	0.004	1906.674	564.297	0.005	0.006	0.002	0.004	0.002	0.000	
1	137	89	2.717	1.501	3.006	2.857	28889	32943	0.186	0.935	0.043	0.160	0.917	0.039	
2	114	80	2.319	1.428	2.973	2.859	33032	32438	0.176	0.922	0.038	0.163	0.920	0.040	
3	153	91	2.777	1.367	3.007	2.859	28458	33649	0.187	0.936	0.043	0.164	0.917	0.039	
4	126	73	2.458	1.441	3.007	2.860	28790	34260	0.185	0.934	0.042	0.166	0.916	0.039	
5	90	62	2.095	1.370	2.973	2.854	33506	34369	0.178	0.921	0.038	0.169	0.918	0.039	
6	216	88	3.386	1.462	3.007	2.861	29307	32736	0.191	0.935	0.042	0.166	0.921	0.039	
7	76	87	1.871	1.416	2.972	2.860	32922	33827	0.179	0.923	0.038	0.164	0.916	0.039	10000
8	180	75	3.018	1.305	3.004	2.860	28055	33264	0.185	0.937	0.043	0.165	0.919	0.039	
9	105	103	2.301	1.563	2.972	2.865	33941	32490	0.181	0.921	0.037	0.168	0.923	0.039	
10	164	77	2.874	1.316	2.968	2.860	32385	33929	0.183	0.926	0.038	0.167	0.918	0.039	
Avg	136.1	82.5	2.582	1.417	2.989	2.860	30928.5	33390.5	0.183	0.929	0.040	0.165	0.918	0.039	
Std	43.039	11.492	0.457	0.081	0.018	0.003	2402.804	716.7531	0.005	0.007	0.003	0.003	0.002	0.000	

Table 22 Full Comparison of VBGMM on Full dataset, Coreset and Uniform Sampling on Rat 2 Kidney Data

	VBGMM Steps		Time		Vbound		Count		Features						
Run									Coreset			Uniform			samples
	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Int	Blob	Flat	Int	Blob	Flat	
1	64	94	1.457	0.374	3.001	2.174	42978	86231	0.256	0.922	0.055	0.293	0.861	0.034	
2	115	51	1.115	0.272	2.868	2.127	33009	88377	0.179	0.944	0.050	0.290	0.851	0.034	500

3	60	48	0.963	0.267	2.843	2.153	27403	84268	0.168	0.950	0.055	0.295	0.869	0.035	
4	77	61	1.039	0.298	2.926	2.147	44543	7	0.325	0.936	0.030	0.098	0.874	0.747	
5	177	87	1.256	0.381	2.904	2.131	46765	88415	0.332	0.934	0.042	0.291	0.851	0.034	
6	52	63	0.919	0.292	2.772	2.178	39854	81843	0.137	0.901	0.039	0.297	0.877	0.035	
7	70	80	0.962	0.328	2.895	2.207	37526	80693	0.201	0.938	0.049	0.301	0.881	0.035	
8	190	86	1.388	0.412	2.918	2.192	34224	82025	0.173	0.939	0.048	0.296	0.876	0.035	
9	129	299	1.262	1.071	2.922	2.187	33190	83448	0.196	0.946	0.050	0.296	0.871	0.035	
10	120	129	1.174	0.477	2.937	2.102	43835	90748	0.249	0.936	0.040	0.289	0.839	0.034	
Avg	105.4	99.8	1.154	0.417	2.899	2.160	38332.7	76605.5	0.222	0.935	0.046	0.274	0.865	0.106	
Std	49.253	73.965	0.186	0.239	0.061	0.033	6285.96	27115.12	0.067	0.014	0.008	0.062	0.014	0.225	
1	197	180	1.485	0.768	2.967	2.384	46670	76888	0.321	0.936	0.038	0.304	0.891	0.034	
2	56	111	0.987	0.618	2.996	2.353	40047	78851	0.231	0.922	0.056	0.310	0.884	0.034	
3	74	131	1.067	0.625	2.875	2.362	34227	80364	0.185	0.942	0.051	0.303	0.880	0.034	
4	112	101	1.176	0.536	2.865	2.413	29807	73762	0.162	0.946	0.051	0.313	0.898	0.035	
5	64	85	1.043	0.456	2.892	2.385	32025	75713	0.175	0.944	0.050	0.307	0.894	0.034	
6	90	115	1.107	0.549	2.939	2.362	31609	80520	0.165	0.944	0.048	0.301	0.880	0.034	
7	90	132	1.124	0.589	2.965	2.380	47162	79211	0.302	0.937	0.033	0.303	0.883	0.034	1000
8	244	140	1.671	0.651	2.962	2.413	44053	73105	0.317	0.941	0.038	0.308	0.901	0.035	
9	62	116	1.015	0.538	2.889	2.369	34161	77705	0.172	0.940	0.051	0.310	0.887	0.034	
10	322	183	1.931	0.790	2.959	2.295	47192	53852	0.263	0.933	0.041	0.161	0.886	0.043	
Avg	131.1	129.4	1.261	0.612	2.931	2.372	38695.3	74997.1	0.229	0.938	0.046	0.292	0.888	0.035	
Std	91.562	31.725	0.323	0.104	0.046	0.034	7088.667	7856.753	0.066	0.007	0.008	0.046	0.007	0.003	
1	170	223	1.723	1.125	2.977	2.535	46537	73993	0.320	0.937	0.037	0.313	0.896	0.034	
2	69	202	1.134	1.095	2.903	2.521	32016	44127	0.184	0.945	0.052	0.157	0.915	0.046	
3	75	121	1.147	0.797	2.884	2.541	31800	74145	0.176	0.945	0.049	0.319	0.894	0.034	
4	81	67	1.229	0.526	2.898	2.579	33411	71053	0.166	0.940	0.050	0.319	0.903	0.034	
5	116	84	1.374	0.537	2.967	2.497	48396	45670	0.285	0.934	0.037	0.157	0.909	0.045	
6	153	76	1.504	0.497	2.967	2.520	42883	42583	0.255	0.940	0.042	0.159	0.920	0.046	
7	89	136	1.223	0.732	2.901	2.517	33121	43965	0.175	0.942	0.050	0.155	0.913	0.046	2000
8	86	69	1.180	0.505	2.889	2.588	29091	68847	0.173	0.949	0.053	0.320	0.908	0.035	
9	96	139	1.264	0.797	2.962	2.504	47186	46813	0.323	0.936	0.035	0.168	0.913	0.044	
10	84	150	1.171	0.873	2.899	2.576	38726	69847	0.205	0.936	0.051	0.315	0.907	0.035	
Avg	101.9	126.7	1.295	0.748	2.925	2.538	38316.7	58104.3	0.226	0.940	0.046	0.238	0.908	0.040	
Std	34.099	54.817	0.188	0.235	0.038	0.033	7357.458	14333.41	0.063	0.005	0.007	0.083	0.008	0.006	
1	80	151	1.496	1.307	2.906	2.762	32523	61145	0.176	0.943	0.051	0.349	0.910	0.035	
2	195	169	2.211	1.381	2.905	2.679	32600	38150	0.177	0.943	0.051	0.140	0.920	0.048	
3	178	154	2.089	1.356	2.895	2.674	31891	37871	0.176	0.945	0.052	0.141	0.921	0.047	
4	92	96	1.657	0.950	2.900	2.734	32011	34243	0.180	0.945	0.051	0.158	0.938	0.053	
5	97	134	1.571	1.096	2.906	2.740	33813	35373	0.171	0.940	0.049	0.162	0.937	0.051	5000
6	239	132	2.477	1.163	2.906	2.685	35023	37296	0.189	0.940	0.051	0.146	0.926	0.048	
7	245	229	2.586	1.751	2.895	2.739	37162	34318	0.186	0.936	0.048	0.157	0.938	0.052	
8	200	294	2.297	2.217	2.973	2.677	47411	37013	0.325	0.936	0.037	0.134	0.919	0.048	

9	82	97	1.484	1.093	2.900	2.723	33026	35225	0.172	0.941	0.051	0.165	0.939	0.052	
10	103	119	1.642	1.229	2.900	2.681	33481	38247	0.176	0.942	0.051	0.144	0.922	0.048	
Avg	151.1	157.5	1.951	1.354	2.909	2.709	34894.1	38888.1	0.193	0.941	0.049	0.170	0.927	0.048	
Std	66.791	61.516	0.427	0.373	0.023	0.033	4675.041	7969.718	0.047	0.003	0.005	0.064	0.010	0.005	
1	74	77	1.698	1.140	2.906	2.800	32219	34268	0.174	0.944	0.051	0.165	0.940	0.052	
2	158	92	2.488	1.184	2.977	2.792	48227	34734	0.316	0.936	0.036	0.163	0.938	0.052	
3	177	95	2.640	1.159	2.907	2.791	33910	34196	0.181	0.942	0.050	0.163	0.939	0.052	
4	107	177	1.996	2.063	2.908	2.796	33666	34085	0.181	0.942	0.051	0.163	0.939	0.052	
5	125	87	2.132	1.206	2.978	2.795	46568	33981	0.289	0.937	0.038	0.161	0.939	0.052	
6	78	100	1.890	1.226	2.895	2.795	32761	33921	0.178	0.943	0.051	0.164	0.940	0.053	
7	124	88	2.215	1.188	2.907	2.801	32431	33787	0.174	0.943	0.051	0.161	0.940	0.052	8000
8	95	137	1.979	1.574	2.908	2.793	33629	34586	0.180	0.942	0.051	0.163	0.939	0.052	
9	172	95	2.534	1.159	2.978	2.799	46327	33398	0.300	0.938	0.038	0.162	0.941	0.053	
10	90	82	1.778	1.096	2.903	2.798	31846	34190	0.172	0.944	0.051	0.163	0.939	0.052	
Avg	120	103	2.135	1.299	2.927	2.796	37158.4	34114.6	0.215	0.941	0.047	0.163	0.939	0.052	
Std	38.023	30.710	0.328	0.299	0.035	0.003	6868.835	382.791	0.061	0.003	0.007	0.001	0.001	0.000	
1	210	77	3.453	1.219	2.976	2.820	46329	33923	0.298	0.939	0.037	0.164	0.940	0.053	
2	100	94	2.131	1.495	2.906	2.824	33238	33551	0.175	0.942	0.051	0.160	0.939	0.052	
3	108	69	2.165	1.274	2.903	2.821	33372	33368	0.179	0.942	0.050	0.164	0.941	0.052	
4	72	82	1.822	1.299	2.908	2.819	32803	33859	0.174	0.943	0.051	0.163	0.940	0.052	
5	76	78	1.920	1.266	2.904	2.819	32202	32969	0.177	0.944	0.051	0.161	0.941	0.053	
6	201	79	3.082	1.366	2.975	2.822	45575	33968	0.282	0.938	0.038	0.163	0.939	0.052	
7	67	94	1.880	1.445	2.906	2.822	32498	33596	0.175	0.943	0.051	0.165	0.940	0.053	10000
8	94	70	2.194	1.209	2.905	2.818	33178	33579	0.180	0.943	0.051	0.164	0.941	0.052	
9	72	85	1.938	1.248	2.902	2.821	32764	33832	0.171	0.942	0.050	0.164	0.940	0.052	
10	104	87	2.284	1.342	2.906	2.825	33926	33367	0.178	0.941	0.052	0.162	0.940	0.052	
Avg	110.4	81.5	2.287	1.316	2.919	2.821	35588.5	33601.2	0.199	0.942	0.048	0.163	0.940	0.052	
Std	52.248	8.708	0.545	0.095	0.030	0.002	5485.912	310.8725	0.048	0.002	0.006	0.001	0.001	0.000	

Table 23 Full Comparison of VBGMM on Full dataset, Coreset and Uniform Sampling on Rat 3 Kidney Data

Run	VBGMM Steps		Time		Vbound		Count		Features						samples
									Coreset			Uniform			
	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Int	Blob	Flat	Int	Blob	Flat	
1	40	37	1.387	0.215	2.943	2.094	38642	35661	0.286	0.935	0.084	0.228	0.937	0.084	
2	64	54	0.966	0.252	2.971	2.091	29312	42317	0.228	0.956	0.094	0.175	0.864	0.080	
3	49	34	0.896	0.225	2.937	2.101	25298	39854	0.227	0.963	0.082	0.190	0.898	0.082	
4	37	54	0.884	0.291	2.967	2.065	34483	42633	0.250	0.941	0.089	0.177	0.865	0.080	500
5	132	144	1.194	0.504	2.969	2.031	29348	42774	0.228	0.956	0.093	0.176	0.862	0.080	
6	43	134	0.893	0.476	2.938	2.073	29957	41687	0.225	0.955	0.093	0.177	0.874	0.081	

7	50	42	0.899	0.254	2.962	2.101	29872	41441	0.226	0.955	0.094	0.180	0.878	0.081	
8	57	36	0.938	0.225	2.970	2.046	29801	43776	0.230	0.956	0.093	0.181	0.861	0.079	
9	48	36	0.927	0.212	2.973	2.071	28933	40994	0.227	0.958	0.091	0.188	0.891	0.081	
10	42	64	0.870	0.283	2.819	2.053	32499	42513	0.641	0.909	0.025	0.179	0.869	0.080	
Avg	56.2	63.5	0.985	0.294	2.945	2.073	30814.5	41365	0.277	0.948	0.084	0.185	0.880	0.081	
Std	27.832	41.067	0.170	0.107	0.046	0.024	3630.059	2277.817	0.129	0.016	0.021	0.016	0.024	0.002	
1	54	45	0.956	0.298	2.960	2.361	29145	39895	0.227	0.957	0.094	0.181	0.889	0.082	
2	43	44	0.931	0.271	2.967	2.357	28896	40427	0.228	0.957	0.094	0.178	0.883	0.082	
3	34	21	0.906	0.210	2.928	2.442	30944	32992	0.247	0.952	0.085	0.229	0.945	0.088	
4	123	26	1.204	0.232	2.940	2.431	26519	33704	0.231	0.962	0.084	0.234	0.945	0.087	
5	50	25	0.964	0.236	2.959	2.428	30271	33433	0.226	0.953	0.094	0.230	0.945	0.087	
6	45	22	0.933	0.194	2.863	2.433	34934	33703	0.659	0.900	0.022	0.230	0.944	0.087	
7	43	66	0.943	0.375	2.935	2.432	28468	33818	0.227	0.959	0.093	0.224	0.942	0.087	1000
8	42	31	0.939	0.254	2.971	2.446	34948	33793	0.258	0.942	0.089	0.218	0.941	0.087	
9	67	166	1.004	0.701	2.864	2.455	35638	33612	0.657	0.900	0.022	0.223	0.943	0.087	
10	32	44	1.008	0.278	2.977	2.457	29765	32796	0.228	0.956	0.092	0.224	0.946	0.089	
Avg	53.3	49	0.979	0.305	2.937	2.424	30952.8	34817.3	0.319	0.944	0.077	0.217	0.932	0.086	
Std	26.441	43.443	0.085	0.148	0.042	0.036	3144.128	2839.544	0.179	0.024	0.029	0.020	0.024	0.002	
1	49	21	1.033	0.278	2.999	2.680	34573	32660	0.253	0.942	0.089	0.232	0.947	0.089	
2	38	33	0.956	0.349	2.924	2.686	29026	32499	0.236	0.959	0.091	0.229	0.948	0.089	
3	76	26	1.136	0.294	2.886	2.660	34773	32616	0.201	0.927	0.092	0.231	0.947	0.089	
4	63	25	1.069	0.266	2.984	2.673	29449	32325	0.227	0.956	0.094	0.232	0.948	0.089	
5	50	30	1.052	0.262	2.986	2.680	29222	32093	0.228	0.957	0.092	0.230	0.948	0.089	
6	41	54	1.015	0.379	2.968	2.655	29583	32091	0.228	0.956	0.094	0.228	0.949	0.090	
7	47	43	1.100	0.363	2.979	2.679	28528	32512	0.228	0.958	0.093	0.226	0.947	0.089	2000
8	20	59	0.873	0.386	2.988	2.674	31101	32251	0.229	0.950	0.093	0.227	0.948	0.089	
9	98	25	1.213	0.252	2.948	2.652	29189	32142	0.226	0.957	0.093	0.230	0.949	0.089	
10	50	27	1.063	0.259	2.983	2.666	29636	32005	0.227	0.956	0.093	0.230	0.949	0.089	
Avg	53.2	34.3	1.051	0.309	2.965	2.671	30508	32319.4	0.228	0.952	0.092	0.229	0.948	0.089	
Std	21.555	13.191	0.093	0.054	0.035	0.012	2292.836	238.4632	0.013	0.010	0.001	0.002	0.001	0.001	
1	54	41	2.360	0.563	2.979	2.859	28750	31097	0.228	0.958	0.088	0.227	0.952	0.091	
2	185	21	2.087	0.358	2.980	2.846	28249	31258	0.228	0.959	0.087	0.230	0.950	0.091	
3	78	49	1.489	0.545	3.012	2.852	32099	31580	0.235	0.947	0.092	0.231	0.949	0.091	
4	84	33	1.663	0.407	2.987	2.864	29367	31323	0.228	0.956	0.093	0.231	0.950	0.091	
5	68	24	1.356	0.425	2.927	2.841	30494	31170	0.236	0.954	0.092	0.230	0.951	0.090	
6	66	42	1.346	0.462	2.965	2.866	26492	31654	0.227	0.962	0.085	0.231	0.949	0.091	
7	139	43	1.753	0.471	2.982	2.858	29317	31096	0.228	0.956	0.094	0.228	0.952	0.091	5000
8	31	27	1.104	0.366	2.963	2.853	26447	31090	0.227	0.962	0.083	0.228	0.952	0.092	
9	26	41	2.755	0.484	2.989	2.845	29430	31207	0.228	0.957	0.093	0.228	0.952	0.090	
10	54	31	1.317	0.588	2.988	2.844	29638	31312	0.227	0.956	0.093	0.230	0.950	0.091	
Avg	78.5	35.2	1.723	0.467	2.977	2.853	29028.3	31278.7	0.229	0.957	0.090	0.229	0.951	0.091	
Std	48.890	9.319	0.526	0.080	0.022	0.009	1702.59	198.3516	0.003	0.004	0.004	0.001	0.001	0.000	
1	45	27	1.697	0.738	2.989	2.918	29442	31563	0.228	0.956	0.093	0.231	0.949	0.091	8000

2	93	82	1.742	1.193	2.957	2.914	28414	31365	0.225	0.959	0.092	0.231	0.950	0.091	
3	50	26	1.493	1.008	2.985	2.914	29428	31508	0.228	0.956	0.093	0.231	0.949	0.091	
4	338	47	3.990	0.694	2.957	2.904	28968	30728	0.227	0.958	0.093	0.227	0.953	0.091	
5	121	45	2.129	0.750	3.006	2.916	31877	30765	0.234	0.948	0.092	0.228	0.953	0.092	
6	83	52	1.852	0.786	2.951	2.911	27589	30728	0.235	0.959	0.083	0.228	0.953	0.092	
7	30	26	1.186	0.455	3.011	2.916	31868	31461	0.234	0.948	0.092	0.231	0.949	0.092	
8	51	45	1.446	0.711	2.989	2.910	29499	31685	0.228	0.956	0.092	0.231	0.949	0.091	
9	50	26	1.512	0.479	2.990	2.916	29785	31307	0.228	0.956	0.092	0.231	0.950	0.091	
10	69	24	1.552	0.462	2.885	2.924	38663	31296	0.191	0.900	0.089	0.231	0.950	0.092	
Avg	93	40	1.860	0.728	2.972	2.914	30553.3	31240.6	0.226	0.950	0.091	0.230	0.951	0.091	
Std	90.197	18.318	0.791	0.237	0.037	0.005	3150.447	364.5665	0.013	0.018	0.003	0.002	0.002	0.000	
1	82	26	1.869	0.686	3.009	2.923	33110	31281	0.241	0.944	0.091	0.230	0.950	0.091	
2	32	46	1.699	0.851	3.009	2.933	32105	30630	0.235	0.947	0.092	0.228	0.954	0.091	
3	109	31	2.217	0.706	2.967	2.932	27504	31299	0.227	0.960	0.086	0.230	0.950	0.092	
4	65	35	1.819	0.633	3.012	2.930	32503	30439	0.238	0.946	0.091	0.228	0.954	0.092	
5	35	24	1.440	0.552	3.012	2.941	32270	31568	0.237	0.947	0.092	0.232	0.949	0.092	
6	33	21	1.289	0.505	3.014	2.938	32642	31553	0.238	0.946	0.091	0.232	0.949	0.091	
7	77	48	1.802	1.005	3.010	2.931	33936	30466	0.247	0.943	0.090	0.227	0.954	0.092	10000
8	232	41	3.460	0.754	2.966	2.932	25895	30540	0.227	0.962	0.083	0.228	0.954	0.092	
9	46	30	1.428	0.701	3.014	2.939	32823	31711	0.239	0.945	0.091	0.232	0.948	0.091	
10	90	43	1.920	0.803	2.966	2.938	26704	31549	0.227	0.962	0.085	0.232	0.949	0.092	
Avg	80.1	34.5	1.894	0.720	2.998	2.934	30949.2	31103.6	0.236	0.950	0.089	0.230	0.951	0.092	
Std	59.607	9.583	0.615	0.145	0.022	0.005	2998.101	521.1914	0.007	0.008	0.003	0.002	0.002	0.000	

Table 24 Full Comparison of VBGMM on Full dataset, Coreset and Uniform Sampling on Rat 4 Kidney Data

	VBGMM Steps		Time		Vbound		Count		Features						
Run									Coreset			Uniform			sample
	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Int	Blob	Flat	Int	Blob	Flat	
1	69	38	1.403	0.257	2.731	1.999	26205	38035	0.485	0.925	0.035	0.078	0.902	0.094	
2	113	121	0.983	0.423	2.706	1.975	25294	35334	0.439	0.940	0.051	0.074	0.912	0.101	
3	35	315	0.781	0.999	2.643	1.921	27497	69934	0.074	0.942	0.115	0.282	0.896	0.059	
4	222	75	1.296	0.320	2.807	2.013	29110	38876	0.168	0.948	0.115	0.063	0.871	0.093	
5	66	249	0.879	0.819	2.752	1.919	34688	70689	0.361	0.938	0.063	0.288	0.890	0.056	
6	57	52	0.839	0.265	2.743	2.024	34673	39439	0.342	0.938	0.065	0.065	0.871	0.093	
7	68	90	0.871	0.358	2.802	1.964	31289	67918	0.180	0.945	0.111	0.295	0.897	0.055	500
8	70	89	0.883	0.424	2.759	1.991	38273	38555	0.307	0.946	0.058	0.484	0.858	0.026	
9	65	92	0.900	0.440	2.731	2.009	31637	39199	0.371	0.944	0.051	0.065	0.872	0.092	
10	37	43	0.818	0.268	2.692	2.017	28886	40279	0.488	0.916	0.032	0.072	0.879	0.092	
Avg	80.2	116.4	0.965	0.457	2.736	1.983	30755.2	47825.8	0.322	0.938	0.070	0.177	0.885	0.076	
Std	54.203	92.186	0.211	0.251	0.049	0.038	4159.88	15036.48	0.141	0.010	0.032	0.150	0.017	0.025	

1	72	85	1.012	0.413	2.668	2.276	27103	36783	0.085	0.950	0.117	0.066	0.892	0.097	
2	94	95	1.036	0.473	2.769	2.238	33850	38208	0.299	0.946	0.059	0.475	0.874	0.028	
3	55	128	0.896	0.590	2.778	2.244	34475	35927	0.379	0.937	0.051	0.067	0.900	0.098	
4	72	62	0.975	0.367	2.662	2.251	32163	36973	0.069	0.920	0.108	0.062	0.883	0.096	
5	66	69	0.943	0.430	2.644	2.277	32374	36087	0.068	0.917	0.108	0.061	0.888	0.097	
6	42	152	0.913	0.721	2.645	2.265	29153	37698	0.085	0.944	0.113	0.066	0.886	0.096	
7	74	92	1.029	0.495	2.782	2.289	30897	34160	0.432	0.930	0.046	0.073	0.917	0.101	1000
8	50	111	0.988	0.557	2.661	2.284	33478	34936	0.069	0.913	0.106	0.072	0.912	0.099	
9	52	40	0.966	0.289	2.707	2.288	29482	34531	0.488	0.913	0.030	0.075	0.917	0.101	
10	43	123	0.901	0.587	2.622	2.215	27688	60518	0.075	0.944	0.113	0.305	0.914	0.054	
Avg	62	95.7	0.966	0.492	2.694	2.263	31066.3	38582.1	0.205	0.931	0.085	0.132	0.898	0.087	
Std	16.459	33.784	0.052	0.126	0.061	0.025	2612.308	7820.676	0.174	0.015	0.034	0.142	0.016	0.025	
1	104	53	1.231	0.459	2.785	2.478	33450	30900	0.378	0.940	0.055	0.078	0.935	0.107	
2	46	35	0.972	0.281	2.677	2.455	27663	31854	0.080	0.946	0.114	0.078	0.932	0.105	
3	99	50	1.136	0.404	2.786	2.479	31785	30879	0.422	0.931	0.045	0.076	0.934	0.108	
4	166	46	1.492	0.356	2.778	2.475	30956	31471	0.263	0.945	0.084	0.072	0.928	0.106	
5	69	124	1.114	0.708	2.791	2.452	33416	32636	0.387	0.938	0.052	0.073	0.924	0.103	
6	89	221	1.128	1.094	2.673	2.449	26527	46860	0.083	0.950	0.120	0.364	0.920	0.047	
7	167	71	1.511	0.504	2.795	2.467	33436	31523	0.375	0.937	0.055	0.074	0.930	0.106	2000
8	219	48	1.722	0.359	2.790	2.471	33565	31302	0.357	0.938	0.062	0.079	0.935	0.107	
9	89	316	1.205	1.550	2.686	2.454	26562	32889	0.084	0.951	0.119	0.071	0.921	0.103	
10	61	92	1.004	0.585	2.682	2.454	27729	31411	0.081	0.948	0.115	0.077	0.933	0.106	
Avg	110.9	105.6	1.252	0.630	2.744	2.464	30508.9	33172.5	0.251	0.942	0.082	0.104	0.929	0.100	
Std	55.252	92.507	0.244	0.399	0.056	0.012	3056.829	4854.509	0.151	0.006	0.032	0.091	0.006	0.019	
1	87	62	1.466	0.628	2.683	2.609	28014	30249	0.082	0.947	0.114	0.075	0.934	0.111	
2	128	83	1.695	0.771	2.675	2.603	31646	30877	0.074	0.927	0.110	0.072	0.929	0.109	
3	97	72	1.484	0.725	2.679	2.611	29557	29725	0.080	0.940	0.113	0.080	0.940	0.111	
4	233	169	2.409	1.299	2.798	2.635	34082	37965	0.366	0.939	0.057	0.400	0.927	0.046	
5	107	56	1.656	0.628	2.794	2.611	33142	30364	0.378	0.939	0.052	0.078	0.937	0.110	
6	110	46	1.546	0.575	2.798	2.603	36749	29649	0.348	0.940	0.056	0.080	0.941	0.110	
7	110	53	1.552	0.637	2.796	2.602	36272	30077	0.319	0.943	0.062	0.079	0.938	0.111	5000
8	145	49	1.799	0.541	2.679	2.614	28962	29938	0.079	0.941	0.114	0.076	0.937	0.111	
9	127	81	1.692	0.796	2.798	2.604	34311	29764	0.340	0.943	0.060	0.078	0.939	0.111	
10	76	67	1.342	0.801	2.794	2.609	37326	29819	0.340	0.942	0.057	0.079	0.939	0.111	
Avg	122	73.8	1.664	0.740	2.749	2.610	33006.1	30842.7	0.241	0.940	0.079	0.110	0.936	0.104	
Std	43.957	35.786	0.294	0.217	0.061	0.010	3353.898	2529.833	0.140	0.005	0.029	0.102	0.005	0.021	
1	63	258	1.486	2.650	2.684	2.686	27751	35780	0.081	0.947	0.115	0.403	0.931	0.045	
2	100	58	1.832	0.909	2.795	2.636	34341	29547	0.309	0.942	0.070	0.079	0.940	0.112	
3	142	59	2.258	0.809	2.798	2.640	34724	29629	0.360	0.939	0.059	0.078	0.939	0.112	
4	75	139	1.783	1.665	2.680	2.689	30307	39572	0.075	0.934	0.111	0.369	0.934	0.049	8000
5	124	47	2.064	0.733	2.685	2.645	29065	29249	0.081	0.943	0.114	0.082	0.943	0.112	
6	140	87	2.181	1.234	2.795	2.639	33930	30022	0.355	0.938	0.061	0.080	0.939	0.111	

7	70	116	1.708	1.487	2.799	2.639	34942	29299	0.370	0.938	0.054	0.077	0.939	0.112	
8	73	61	1.525	0.804	2.801	2.637	34636	30185	0.359	0.940	0.057	0.078	0.937	0.111	
9	154	58	2.385	0.834	2.685	2.646	29230	29462	0.079	0.941	0.113	0.080	0.941	0.112	
10	61	93	1.502	1.258	2.685	2.639	28490	29366	0.077	0.942	0.114	0.078	0.940	0.112	
Avg	100.2	97.6	1.872	1.238	2.741	2.650	31741.6	31211.1	0.215	0.941	0.087	0.140	0.938	0.099	
Std	36.514	63.631	0.331	0.590	0.060	0.020	3001.647	3535.35	0.144	0.004	0.028	0.130	0.003	0.027	
1	89	167	2.019	2.322	2.799	2.699	35932	35189	0.344	0.941	0.057	0.418	0.927	0.042	
2	88	119	2.099	1.793	2.686	2.713	27950	38784	0.082	0.948	0.114	0.354	0.940	0.053	
3	79	61	1.929	1.025	2.797	2.649	35023	29974	0.359	0.940	0.054	0.077	0.937	0.112	
4	83	128	2.010	1.778	2.799	2.652	36525	28979	0.351	0.940	0.056	0.081	0.943	0.113	
5	51	61	1.671	1.044	2.681	2.655	29425	29120	0.078	0.939	0.113	0.082	0.943	0.113	
6	81	66	1.895	1.151	2.797	2.650	35337	29935	0.350	0.940	0.060	0.077	0.937	0.112	
7	79	49	1.781	1.049	2.798	2.655	32727	29348	0.386	0.937	0.054	0.082	0.943	0.112	10000
8	62	55	1.658	0.923	2.686	2.652	27937	29319	0.081	0.946	0.115	0.080	0.942	0.112	
9	91	66	1.956	1.250	2.800	2.651	34577	29705	0.370	0.939	0.055	0.078	0.938	0.113	
10	306	44	4.390	0.829	2.839	2.656	33776	29502	0.205	0.943	0.105	0.081	0.941	0.112	
Avg	100.9	81.6	2.141	1.316	2.768	2.663	32920.9	30985.5	0.261	0.941	0.078	0.141	0.939	0.099	
Std	73.144	41.318	0.804	0.484	0.059	0.023	3292.998	3290.017	0.134	0.003	0.029	0.130	0.005	0.027	

Table 25 Full Comparison of VBGMM on Full dataset, Coreset and Uniform Sampling on Rat 5 Kidney Data

Run	VBGMM Steps		Time		Vbound		Count		Features						samples
									Coreset			Uniform			
	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Int	Blob	Flat	Int	Blob	Flat	
1	37	124	1.358	0.459	3.161	2.282	31150	6	0.286	0.953	0.068	0.095	0.899	0.827	
2	26	71	0.792	0.315	3.236	2.291	36909	70188	0.292	0.941	0.063	0.416	0.900	0.038	
3	108	71	1.124	0.306	3.129	2.198	34907	39405	0.268	0.942	0.069	0.196	0.889	0.061	
4	30	59	0.785	0.276	3.121	2.206	35895	40308	0.222	0.920	0.069	0.203	0.892	0.060	
5	65	90	0.882	0.336	3.174	2.184	32090	39816	0.264	0.949	0.070	0.201	0.894	0.061	
6	23	56	0.755	0.293	3.222	2.264	33985	2	0.282	0.944	0.068	0.119	0.994	0.916	
7	76	85	0.884	0.338	3.121	2.335	35732	5	0.205	0.911	0.068	0.113	0.945	0.849	500
8	40	75	0.801	0.347	3.146	2.326	33449	6	0.252	0.942	0.070	0.095	0.899	0.827	
9	41	64	0.804	0.310	3.168	2.324	35292	69397	0.228	0.924	0.070	0.413	0.904	0.038	
10	51	65	0.847	0.275	3.174	2.212	30858	40758	0.262	0.951	0.071	0.206	0.893	0.060	
Avg	49.7	76	0.903	0.325	3.165	2.262	34026.7	29989.1	0.256	0.938	0.069	0.206	0.911	0.374	
Std	26.500	19.961	0.191	0.053	0.039	0.058	2094.557	28230.9	0.029	0.014	0.002	0.119	0.033	0.415	
1	262	64	1.643	0.373	3.258	2.622	31070	64604	0.325	0.956	0.051	0.421	0.917	0.039	
2	33	44	0.921	0.286	3.267	2.633	33199	63924	0.281	0.945	0.065	0.422	0.918	0.039	
3	45	89	0.850	0.444	3.184	2.542	31745	37975	0.262	0.949	0.070	0.204	0.905	0.062	1000
4	63	207	0.926	0.857	3.192	2.545	29164	37231	0.252	0.953	0.064	0.219	0.919	0.062	

5	46	145	0.850	0.618	3.184	2.579	29864	37152	0.263	0.954	0.069	0.232	0.927	0.062	
6	107	66	1.093	0.374	3.251	2.550	30682	36004	0.301	0.956	0.058	0.207	0.916	0.063	
7	45	113	0.889	0.490	3.172	2.604	31200	64153	0.259	0.950	0.069	0.422	0.918	0.039	
8	47	70	0.867	0.364	3.179	2.559	32735	36159	0.247	0.943	0.071	0.211	0.918	0.063	
9	57	97	0.919	0.458	3.168	2.540	35756	36713	0.226	0.922	0.069	0.204	0.910	0.063	
10	41	103	0.850	0.503	3.170	2.598	35511	35132	0.224	0.923	0.069	0.248	0.939	0.063	
Avg	74.6	99.8	0.981	0.477	3.202	2.577	32092.6	44904.7	0.264	0.945	0.065	0.279	0.919	0.055	
Std	68.964	47.457	0.244	0.162	0.039	0.035	2217.085	13356.74	0.031	0.013	0.007	0.099	0.009	0.011	
1	33	136	0.891	0.764	3.274	2.799	33192	34164	0.272	0.945	0.066	0.242	0.939	0.064	
2	120	68	1.266	0.494	3.181	2.820	34909	33456	0.235	0.932	0.069	0.234	0.938	0.065	
3	61	49	0.995	0.401	3.174	2.817	35141	34088	0.230	0.927	0.070	0.237	0.938	0.064	
4	70	67	1.020	0.541	3.182	2.825	33917	33442	0.255	0.941	0.069	0.243	0.942	0.064	
5	113	77	1.260	0.489	3.199	2.822	31234	32467	0.258	0.950	0.070	0.237	0.942	0.066	
6	50	51	0.978	0.387	3.188	2.811	33514	33899	0.245	0.940	0.070	0.239	0.939	0.064	
7	66	63	1.033	0.427	3.207	2.820	31152	33223	0.270	0.952	0.061	0.233	0.938	0.065	2000
8	71	71	1.033	0.477	3.174	2.800	35409	34296	0.224	0.924	0.069	0.225	0.932	0.064	
9	54	141	0.951	0.809	3.188	2.809	33845	34167	0.245	0.939	0.070	0.241	0.939	0.064	
10	95	142	1.162	0.812	3.175	2.808	35927	34412	0.234	0.926	0.069	0.237	0.937	0.064	
Avg	73.3	86.5	1.059	0.560	3.194	2.813	33824	33761.4	0.247	0.938	0.068	0.237	0.938	0.064	
Std	27.865	37.672	0.128	0.169	0.030	0.009	1639.744	607.5368	0.017	0.010	0.003	0.005	0.003	0.001	
1	103	63	1.580	0.612	3.203	3.018	31307	31880	0.260	0.950	0.070	0.245	0.946	0.065	
2	41	53	1.103	0.554	3.289	3.023	33799	31366	0.281	0.945	0.067	0.247	0.947	0.068	
3	92	69	1.449	0.669	3.179	3.026	34777	31292	0.234	0.930	0.070	0.247	0.947	0.068	
4	82	45	1.380	0.498	3.209	3.028	30959	31261	0.261	0.951	0.070	0.247	0.947	0.069	
5	80	65	1.358	0.690	3.192	3.024	33063	31851	0.250	0.942	0.070	0.243	0.945	0.066	
6	102	52	1.463	0.607	3.209	3.031	31190	31412	0.263	0.951	0.069	0.246	0.947	0.069	
7	101	47	1.552	0.545	3.206	3.030	31777	31386	0.259	0.949	0.070	0.248	0.948	0.067	5000
8	99	58	1.490	0.554	3.207	3.021	30765	31539	0.261	0.952	0.070	0.245	0.946	0.066	
9	51	56	1.167	0.690	3.201	3.017	31177	31713	0.264	0.952	0.069	0.245	0.946	0.069	
10	99	43	1.445	0.537	3.189	3.007	32453	31531	0.251	0.945	0.070	0.246	0.947	0.068	
Avg	85	55.1	1.399	0.596	3.208	3.023	32126.7	31523.1	0.258	0.947	0.069	0.246	0.947	0.067	
Std	22.201	8.762	0.155	0.069	0.030	0.007	1357.902	223.4072	0.012	0.007	0.001	0.001	0.001	0.001	
1	88	56	1.706	0.922	3.181	3.093	36069	31100	0.227	0.923	0.069	0.248	0.948	0.069	
2	75	55	1.566	0.774	3.182	3.091	35586	31086	0.234	0.928	0.069	0.249	0.948	0.069	
3	97	65	1.774	0.971	3.193	3.096	32972	30774	0.252	0.944	0.070	0.248	0.949	0.066	
4	42	54	1.336	0.795	3.180	3.094	35848	30921	0.229	0.925	0.069	0.251	0.949	0.069	
5	127	43	2.121	0.768	3.185	3.097	35457	30972	0.230	0.927	0.069	0.250	0.949	0.069	
6	49	53	1.436	0.767	3.180	3.093	36207	30677	0.225	0.922	0.068	0.249	0.949	0.069	8000
7	86	41	1.650	0.641	3.204	3.098	31536	30885	0.259	0.950	0.070	0.250	0.949	0.069	
8	73	76	1.577	0.982	3.182	3.086	35877	31401	0.228	0.924	0.069	0.250	0.948	0.069	
9	114	55	2.106	0.812	3.183	3.090	35162	30797	0.236	0.931	0.069	0.251	0.949	0.069	
10	218	57	2.771	0.867	3.211	3.100	30673	30854	0.262	0.952	0.069	0.250	0.949	0.069	

Avg	96.9	55.5	1.804	0.830	3.188	3.094	34538.7	30946.7	0.238	0.933	0.069	0.250	0.949	0.069	
Std	49.912	9.936	0.425	0.106	0.011	0.004	2037.792	207.6236	0.014	0.011	0.000	0.001	0.000	0.001	
1	57	42	1.483	0.862	3.181	3.122	36068	30969	0.225	0.922	0.069	0.251	0.949	0.069	
2	57	63	1.551	0.998	3.180	3.119	36075	30864	0.223	0.921	0.069	0.249	0.949	0.069	
3	50	47	1.466	0.783	3.180	3.118	36199	31031	0.224	0.921	0.068	0.250	0.949	0.069	
4	71	48	1.703	0.905	3.183	3.123	35612	30988	0.230	0.926	0.069	0.252	0.949	0.069	
5	103	56	1.966	0.964	3.182	3.121	36055	30746	0.229	0.924	0.069	0.251	0.950	0.069	
6	152	43	2.467	0.758	3.208	3.123	31280	30850	0.265	0.952	0.069	0.251	0.949	0.070	10000
7	62	42	1.567	0.986	3.176	3.124	35770	30986	0.233	0.926	0.069	0.250	0.949	0.069	
8	214	75	3.266	1.231	3.185	3.115	34210	31155	0.243	0.936	0.070	0.253	0.950	0.068	
9	61	47	1.503	0.852	3.183	3.113	35358	31018	0.232	0.928	0.069	0.250	0.949	0.069	
10	83	41	1.846	0.809	3.199	3.126	32653	30976	0.252	0.945	0.070	0.250	0.949	0.069	
Avg	91	50.4	1.882	0.915	3.186	3.120	34928	30958.3	0.236	0.930	0.069	0.250	0.949	0.069	
Std	52.907	11.098	0.576	0.139	0.010	0.004	1695.142	113.1911	0.014	0.011	0.001	0.001	0.000	0.000	

Table 26 Full Comparison of VBGMM on Full dataset, Coreset and Uniform Sampling on Rat 6 Kidney Data

Run	VBGMM Steps		Time		Vbound		Count		Features						samples
	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Coreset			Uniform			
									Int	Blob	Flat	Int	Blob	Flat	
1	32	120	1.406	0.430	2.939	1.941	32745	87722	0.263	0.938	0.066	0.420	0.856	0.033	
2	88	408	1.012	1.176	2.777	2.061	29766	74615	0.196	0.934	0.055	0.448	0.903	0.035	
3	63	102	0.926	0.450	2.828	2.043	28112	76834	0.224	0.951	0.070	0.445	0.897	0.034	
4	110	138	1.047	0.538	2.829	1.886	26751	92249	0.218	0.955	0.071	0.419	0.828	0.033	
5	139	70	1.256	0.352	2.821	2.006	32981	78603	0.319	0.957	0.049	0.447	0.890	0.034	
6	61	186	0.960	0.623	2.769	2.013	39285	76995	0.637	0.900	0.013	0.443	0.897	0.035	
7	67	179	1.025	0.614	2.785	1.932	34536	87509	0.645	0.910	0.012	0.424	0.856	0.033	500
8	52	90	0.952	0.394	2.805	2.063	37570	75357	0.643	0.901	0.013	0.449	0.901	0.035	
9	32	116	0.852	0.482	2.875	1.869	32033	92208	0.257	0.943	0.067	0.419	0.828	0.033	
10	101	73	1.175	0.376	2.792	1.923	44324	3	0.475	0.943	0.033	0.079	0.847	0.936	
Avg	74.5	148.2	1.061	0.543	2.822	1.974	33810.3	74209.5	0.388	0.933	0.045	0.399	0.870	0.124	
Std	34.632	99.500	0.169	0.241	0.051	0.072	5362.569	26979.15	0.192	0.022	0.025	0.113	0.031	0.285	
1	44	59	0.930	0.354	2.939	2.268	33153	73492	0.251	0.940	0.065	0.451	0.905	0.034	
2	75	109	1.030	0.533	2.828	2.277	26478	73618	0.223	0.956	0.062	0.453	0.904	0.034	
3	29	65	0.869	0.373	2.945	2.286	33480	73295	0.273	0.937	0.066	0.458	0.904	0.034	
4	47	98	0.947	0.489	2.827	2.270	26630	74372	0.209	0.952	0.073	0.453	0.902	0.034	
5	175	72	1.369	0.378	2.811	2.244	34791	75423	0.647	0.912	0.013	0.456	0.898	0.033	1000
6	51	61	1.033	0.340	2.949	2.291	35062	72700	0.281	0.936	0.064	0.457	0.906	0.034	
7	62	46	1.008	0.297	2.953	2.296	35456	71232	0.268	0.937	0.063	0.454	0.911	0.035	
8	80	62	1.082	0.344	2.867	2.293	27291	71494	0.210	0.953	0.068	0.453	0.910	0.034	
9	45	60	0.913	0.325	2.934	2.244	31706	33292	0.239	0.938	0.067	0.203	0.933	0.061	

10	71	89	1.003	0.496	2.853	2.307	27276	71022	0.212	0.953	0.070	0.453	0.912	0.035	
Avg	67.9	72.1	1.018	0.393	2.890	2.278	31132.3	68994	0.281	0.941	0.061	0.429	0.909	0.037	
Std	40.883	19.980	0.139	0.082	0.059	0.021	3786.04	12623.2	0.131	0.013	0.017	0.079	0.009	0.008	
1	74	77	1.193	0.541	2.959	2.461	33834	30070	0.267	0.939	0.065	0.195	0.940	0.064	
2	51	70	1.045	0.561	2.866	2.473	27499	29521	0.212	0.952	0.071	0.191	0.940	0.065	
3	1164	84	5.884	0.539	2.848	2.495	28626	65649	0.250	0.954	0.059	0.455	0.922	0.035	
4	96	50	1.216	0.368	2.873	2.499	26059	30890	0.215	0.955	0.068	0.217	0.945	0.064	
5	97	81	1.278	0.557	2.910	2.492	39236	66815	0.368	0.949	0.042	0.455	0.920	0.034	
6	108	69	1.306	0.462	2.912	2.486	42985	32126	0.421	0.947	0.038	0.222	0.943	0.062	
7	51	72	1.081	0.572	2.872	2.482	26642	29656	0.213	0.955	0.070	0.194	0.941	0.065	2000
8	86	98	1.197	1.054	2.889	2.476	34744	67135	0.329	0.952	0.043	0.454	0.920	0.036	
9	186	100	1.627	0.608	2.893	2.470	35563	68218	0.324	0.949	0.044	0.459	0.914	0.032	
10	97	80	1.263	0.633	2.804	2.481	35175	29295	0.644	0.912	0.014	0.191	0.940	0.066	
Avg	201	78.1	1.709	0.589	2.882	2.481	33036.3	44937.5	0.324	0.947	0.051	0.303	0.933	0.052	
Std	340.466	14.525	1.475	0.180	0.041	0.012	5692.158	18975.69	0.133	0.013	0.018	0.132	0.012	0.016	
1	115	53	1.835	0.655	2.895	2.686	33855	31018	0.305	0.951	0.048	0.216	0.946	0.067	
2	109	39	1.588	0.530	2.918	2.704	44117	31628	0.427	0.947	0.037	0.202	0.947	0.068	
3	288	72	2.816	0.728	2.880	2.687	25217	31577	0.213	0.957	0.068	0.218	0.948	0.067	
4	153	35	1.906	0.485	2.897	2.730	33505	31684	0.311	0.953	0.047	0.216	0.947	0.066	
5	136	47	1.779	0.558	2.893	2.726	35411	31900	0.329	0.951	0.044	0.220	0.949	0.066	
6	74	39	1.373	0.573	2.880	2.713	27023	31582	0.210	0.954	0.068	0.214	0.949	0.067	
7	84	61	1.516	0.788	2.910	2.692	43781	31960	0.427	0.948	0.038	0.201	0.948	0.067	5000
8	174	34	2.169	0.462	2.914	2.712	43260	31728	0.428	0.947	0.037	0.217	0.947	0.067	
9	78	33	1.499	0.494	2.971	2.719	34218	32051	0.276	0.938	0.065	0.218	0.947	0.066	
10	93	47	1.494	0.560	2.882	2.723	25815	31833	0.212	0.957	0.065	0.219	0.947	0.066	
Avg	130.4	46	1.798	0.583	2.904	2.709	34620.2	31696.1	0.314	0.950	0.052	0.214	0.948	0.067	
Std	64.510	12.841	0.431	0.108	0.027	0.016	7265.146	288.4323	0.089	0.006	0.013	0.007	0.001	0.001	
1	178	43	2.536	0.829	2.918	2.803	43303	29406	0.428	0.948	0.037	0.221	0.948	0.067	
2	100	37	1.926	0.693	2.915	2.799	43833	29197	0.421	0.948	0.038	0.218	0.948	0.067	
3	113	57	2.127	0.877	2.896	2.758	35006	28387	0.328	0.951	0.045	0.205	0.949	0.067	
4	55	36	1.544	0.745	2.883	2.801	26935	28775	0.211	0.954	0.069	0.218	0.949	0.068	
5	159	37	2.580	0.700	2.895	2.796	35758	29089	0.336	0.951	0.043	0.219	0.948	0.067	
6	74	53	1.725	0.861	2.970	2.761	33076	28132	0.266	0.939	0.066	0.204	0.949	0.067	
7	70	42	1.748	0.749	2.971	2.797	33077	28976	0.267	0.938	0.066	0.219	0.948	0.068	8000
8	64	40	1.600	0.772	2.881	2.799	26987	29649	0.213	0.955	0.070	0.223	0.947	0.067	
9	116	33	2.036	0.673	2.894	2.801	32914	29334	0.316	0.953	0.046	0.220	0.947	0.068	
10	59	49	1.501	0.945	2.971	2.797	34988	29555	0.281	0.936	0.064	0.222	0.948	0.067	
Avg	98.8	42.7	1.932	0.784	2.919	2.791	34587.7	29050	0.307	0.947	0.054	0.217	0.948	0.067	
Std	42.887	7.903	0.388	0.090	0.037	0.017	5645.451	495.2824	0.075	0.007	0.014	0.007	0.001	0.000	
1	95	35	2.149	0.950	2.918	2.831	41604	29413	0.405	0.950	0.038	0.221	0.946	0.068	
2	126	37	2.455	0.942	2.917	2.828	42287	29164	0.424	0.949	0.037	0.221	0.948	0.068	
3	65	44	1.780	1.068	2.881	2.788	26858	27757	0.212	0.955	0.069	0.206	0.951	0.068	10000
4	47	50	1.718	1.029	2.972	2.789	32214	27778	0.256	0.938	0.067	0.205	0.951	0.069	

5	114	35	2.234	0.998	2.916	2.822	42883	28594	0.419	0.948	0.038	0.218	0.949	0.068	
6	60	40	1.587	0.951	2.968	2.823	33005	29130	0.265	0.938	0.066	0.221	0.948	0.068	
7	101	71	2.154	1.235	2.899	2.782	34697	27789	0.326	0.952	0.045	0.204	0.950	0.069	
8	75	34	1.949	0.838	2.970	2.829	33124	28997	0.264	0.939	0.066	0.222	0.948	0.068	
9	49	37	1.641	0.852	2.971	2.831	33432	28968	0.265	0.939	0.066	0.222	0.948	0.068	
10	98	34	2.197	0.843	2.897	2.830	36303	29325	0.338	0.949	0.043	0.219	0.947	0.068	
Avg	83	41.7	1.986	0.971	2.931	2.815	35640.7	28691.5	0.317	0.946	0.054	0.216	0.949	0.068	
Std	27.633	11.490	0.293	0.122	0.035	0.020	5169.908	669.908	0.077	0.006	0.014	0.008	0.002	0.000	

Table 27 Full Comparison of VBGMM on Full dataset, Coreset and Uniform Sampling on Human 1 Kidney Data

	VBGMM Steps		Time		Vbound		Count		Features						
Run									Coreset			Uniform			samples
	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Int	Blob	Flat	Int	Blob	Flat	
1	127	37	40.492	3.196	4.033	3.195	1281760	7837	0.069	0.919	0.079	0.378	0.762	0.219	
2	99	28	39.798	3.142	4.038	3.218	1025904	6265	0.040	0.917	0.084	0.399	0.769	0.223	
3	117	25	39.800	3.182	4.053	3.255	1370061	12991	0.074	0.924	0.076	0.356	0.774	0.194	
4	288	36	40.497	3.172	4.010	3.223	451121	8298	0.123	0.894	0.046	0.373	0.770	0.218	
5	121	26	40.206	3.184	4.014	3.220	973453	6177	0.040	0.918	0.086	0.395	0.765	0.227	
6	50	23	39.563	3.177	4.085	3.240	1438468	6828	0.066	0.899	0.076	0.395	0.754	0.218	
7	152	24	40.352	3.157	4.059	3.231	1586434	6492	0.101	0.901	0.036	0.381	0.770	0.233	500
8	124	35	40.132	3.181	4.023	3.232	1846632	6457	0.097	0.901	0.043	0.391	0.763	0.225	
9	111	25	40.026	3.205	4.053	3.259	668755	6955	0.038	0.937	0.097	0.384	0.769	0.225	
10	140	57	39.817	3.255	4.049	3.239	1004098	7032	0.038	0.914	0.085	0.375	0.768	0.230	
Avg	132.9	31.6	40.068	3.185	4.042	3.231	1164669	7533.2	0.069	0.912	0.071	0.383	0.766	0.221	
Std	60.993	10.373	0.322	0.030	0.023	0.019	423018.9	2034.085	0.031	0.013	0.021	0.013	0.006	0.011	
1	141	91	40.302	3.424	4.102	3.408	1064492	13673	0.045	0.917	0.086	0.329	0.782	0.209	
2	142	54	40.140	3.440	4.096	3.427	1304441	22008	0.044	0.899	0.079	0.286	0.792	0.196	
3	145	118	40.411	3.504	4.096	3.440	1061748	21092	0.045	0.917	0.086	0.288	0.791	0.199	
4	154	97	40.253	3.453	4.049	3.433	2013782	24121	0.103	0.880	0.042	0.281	0.792	0.192	
5	112	93	40.207	3.406	4.100	3.429	1471726	18879	0.057	0.899	0.077	0.295	0.790	0.203	
6	141	107	40.120	3.433	4.062	3.416	1543844	13772	0.054	0.887	0.077	0.322	0.787	0.212	
7	152	98	40.108	3.457	4.087	3.446	870379	24426	0.043	0.930	0.090	0.285	0.791	0.189	1000
8	223	100	40.329	3.611	4.077	3.405	935395	16275	0.051	0.924	0.092	0.310	0.787	0.206	
9	153	76	39.933	3.368	4.082	3.420	1303922	18197	0.068	0.904	0.083	0.308	0.780	0.198	
10	349	89	40.918	3.405	4.079	3.405	913378	19316	0.067	0.885	0.098	0.305	0.783	0.195	
Avg	171.2	92.3	40.272	3.450	4.083	3.423	1248311	19175.9	0.058	0.904	0.081	0.301	0.787	0.200	
Std	68.443	17.436	0.264	0.067	0.017	0.015	357610.5	3839.489	0.018	0.017	0.015	0.017	0.005	0.008	
1	238	195	40.908	3.933	4.112	3.710	1290949	2644485	0.048	0.908	0.079	0.090	0.890	0.053	
2	197	115	40.544	3.662	4.078	3.682	1001179	2743249	0.060	0.914	0.093	0.091	0.885	0.053	2000
3	50	240	39.815	4.125	4.137	3.699	1679588	2434056	0.072	0.898	0.061	0.087	0.898	0.056	

4	111	160	40.412	3.758	4.086	3.696	840799	2805776	0.045	0.927	0.094	0.092	0.882	0.052	
5	139	140	40.287	3.690	4.115	3.684	1330409	2970784	0.052	0.907	0.078	0.093	0.874	0.051	
6	274	99	40.848	3.564	4.109	3.703	1440765	2647263	0.051	0.899	0.077	0.091	0.889	0.053	
7	65	141	40.166	3.693	4.128	3.691	1516706	2744728	0.066	0.900	0.071	0.090	0.885	0.053	
8	256	131	40.757	3.785	4.115	3.706	1316744	2472669	0.047	0.905	0.079	0.088	0.897	0.055	
9	142	83	40.273	3.554	4.070	3.700	926009	2913639	0.055	0.925	0.094	0.092	0.877	0.051	
10	222	96	40.686	3.529	4.137	3.706	1480137	2761518	0.062	0.907	0.069	0.092	0.884	0.052	
Avg	169.4	140	40.470	3.729	4.109	3.698	1282329	2713817	0.056	0.909	0.079	0.091	0.886	0.053	
Std	79.524	48.302	0.344	0.185	0.024	0.009	275177.9	171324.7	0.009	0.010	0.011	0.002	0.008	0.002	
1	355	69	42.442	3.670	4.113	3.927	1441665	1841968	0.048	0.894	0.077	0.075	0.910	0.063	
2	167	38	40.874	3.549	4.116	3.919	1099549	1692455	0.045	0.918	0.084	0.070	0.912	0.067	
3	174	66	40.779	3.637	4.103	3.935	1146237	1855072	0.051	0.911	0.086	0.074	0.909	0.063	
4	122	67	40.679	3.609	4.099	3.929	920588	2136008	0.045	0.929	0.089	0.083	0.906	0.058	
5	338	67	42.058	3.742	4.111	3.924	1182225	2288962	0.049	0.912	0.084	0.086	0.901	0.056	
6	180	63	40.971	3.604	4.117	3.933	1447640	1867728	0.054	0.902	0.076	0.073	0.909	0.063	
7	223	94	40.952	3.903	4.114	3.923	1385883	2008377	0.047	0.899	0.077	0.079	0.907	0.060	5000
8	223	68	41.197	3.651	4.115	3.931	1458243	1703158	0.052	0.900	0.076	0.068	0.911	0.067	
9	240	60	41.738	3.654	4.143	3.933	1665169	1799352	0.068	0.900	0.062	0.073	0.911	0.064	
10	172	62	41.051	3.609	4.110	3.928	1126762	1746090	0.051	0.913	0.086	0.070	0.910	0.066	
Avg	219.4	65.4	41.274	3.663	4.114	3.928	1287396	1893917	0.051	0.908	0.080	0.075	0.909	0.063	
Std	75.275	13.501	0.597	0.098	0.011	0.005	225512.4	194489.2	0.007	0.011	0.008	0.006	0.003	0.004	
1	158	59	41.450	3.784	4.109	4.007	1561785	1577792	0.055	0.895	0.075	0.065	0.912	0.069	
2	422	49	43.606	3.777	4.118	4.006	1379264	1564884	0.054	0.908	0.077	0.064	0.912	0.070	
3	658	96	45.672	4.084	4.119	4.006	1230519	1533427	0.048	0.912	0.080	0.061	0.911	0.070	
4	255	103	42.069	4.179	4.116	4.004	1370380	1500801	0.050	0.904	0.078	0.061	0.912	0.071	
5	225	72	41.994	3.963	4.117	4.010	1417151	1505617	0.051	0.902	0.077	0.061	0.911	0.071	
6	370	68	43.015	3.976	4.115	4.001	1352375	1879920	0.051	0.905	0.079	0.074	0.908	0.062	
7	190	95	41.671	4.144	4.115	4.005	1448653	1648100	0.051	0.902	0.076	0.065	0.910	0.068	8000
8	807	75	47.020	3.973	4.142	4.006	1566019	1690060	0.065	0.903	0.066	0.067	0.909	0.067	
9	505	90	44.363	4.276	4.114	4.004	1365471	1590606	0.048	0.902	0.078	0.065	0.912	0.068	
10	137	102	40.979	4.191	4.116	4.007	1173765	1645936	0.050	0.912	0.084	0.064	0.909	0.068	
Avg	372.7	80.9	43.184	4.035	4.118	4.006	1386538	1613714	0.052	0.905	0.077	0.065	0.911	0.068	
Std	226.238	18.918	1.981	0.170	0.009	0.003	124391.7	112548.3	0.005	0.005	0.005	0.004	0.002	0.003	
1	196	92	41.830	4.709	4.115	4.028	1201801	1493637	0.051	0.911	0.084	0.062	0.913	0.071	
2	218	90	42.352	4.223	4.117	4.039	1184036	1502305	0.048	0.909	0.084	0.060	0.910	0.071	
3	297	101	43.160	4.345	4.117	4.033	1276478	1785993	0.049	0.905	0.081	0.072	0.908	0.064	
4	128	111	41.601	4.527	4.115	4.028	1171293	1379112	0.049	0.911	0.084	0.054	0.910	0.075	
5	251	95	42.344	4.531	4.117	4.031	1430239	1597582	0.049	0.898	0.077	0.063	0.910	0.069	10000
6	161	93	41.479	4.307	4.115	4.035	1170650	1430740	0.047	0.909	0.084	0.058	0.912	0.073	
7	168	91	41.612	4.302	4.116	4.030	1158588	1379955	0.046	0.911	0.084	0.057	0.913	0.074	
8	246	99	42.612	4.373	4.115	4.032	1240230	1662655	0.051	0.911	0.082	0.066	0.909	0.067	
9	804	77	48.456	4.138	4.117	4.033	1359968	1566589	0.050	0.905	0.078	0.063	0.911	0.069	

10	219	99	42.166	4.555	4.117	4.030	1198936	1467514	0.047	0.908	0.083	0.058	0.910	0.072	
Avg	268.8	94.8	42.761	4.401	4.116	4.032	1239222	1526608	0.049	0.908	0.082	0.061	0.911	0.071	
Std	194.376	8.829	2.068	0.175	0.001	0.003	90886.51	129120	0.002	0.004	0.003	0.005	0.002	0.004	

Table 28 Full Comparison of VBGMM on Full dataset, Coreset and Uniform Sampling on Human 2 Kidney Data

Run	VBGMM Steps		Time		Vbound		Count		Features						samples
									Coreset			Uniform			
	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Int	Blob	Flat	Int	Blob	Flat	
1	296	37	33.282	2.614	3.520	2.880	1378847	3911097	0.098	0.881	0.078	0.150	0.716	0.052	
2	155	36	32.109	2.571	3.532	2.871	1024468	3911290	0.109	0.901	0.089	0.150	0.716	0.052	
3	202	40	32.468	2.611	3.532	2.865	1266611	1096	0.141	0.902	0.079	0.364	0.718	0.290	
4	210	33	32.506	2.613	3.527	2.867	1248821	3911288	0.188	0.865	0.037	0.150	0.716	0.052	
5	213	35	32.284	2.588	3.554	2.887	1669439	1306	0.149	0.888	0.055	0.333	0.726	0.295	
6	214	50	32.401	2.647	3.533	2.885	1468217	1916	0.154	0.893	0.054	0.346	0.719	0.264	
7	132	35	32.159	2.593	3.531	2.877	1026338	3910369	0.130	0.910	0.083	0.150	0.716	0.052	500
8	52	38	32.074	2.613	3.499	2.876	1195367	2404	0.136	0.900	0.081	0.359	0.716	0.243	
9	125	28	32.338	2.559	3.545	2.900	886026	1377	0.109	0.883	0.097	0.345	0.718	0.285	
10	123	25	32.050	2.608	3.496	2.900	1046991	1361	0.126	0.903	0.086	0.349	0.719	0.285	
Avg	172.2	35.7	32.367	2.602	3.527	2.881	1221113	1565350	0.134	0.892	0.074	0.270	0.718	0.187	
Std	68.365	6.767	0.361	0.025	0.018	0.012	237964.9	2018823	0.026	0.014	0.019	0.103	0.003	0.117	
1	225	94	32.658	2.847	3.564	3.030	2003197	4270	0.145	0.874	0.055	0.248	0.776	0.264	
2	123	76	32.261	2.794	3.547	3.054	885546	5857	0.108	0.890	0.095	0.208	0.802	0.264	
3	245	467	32.496	4.095	3.541	3.010	858116	3233054	0.084	0.917	0.086	0.149	0.799	0.054	
4	176	338	32.499	3.747	3.540	3.048	822865	4321	0.074	0.911	0.088	0.239	0.782	0.268	
5	194	135	32.227	3.021	3.553	3.036	1068828	2932	0.117	0.898	0.087	0.253	0.771	0.284	
6	256	137	32.855	2.963	3.557	3.019	901548	1085	0.104	0.892	0.095	0.268	0.839	0.319	
7	146	108	32.356	2.937	3.556	3.019	1063118	2301	0.100	0.903	0.085	0.269	0.763	0.291	1000
8	145	85	32.205	2.872	3.551	3.010	842628	2272	0.105	0.898	0.097	0.289	0.743	0.282	
9	167	119	32.591	2.950	3.546	3.019	854013	3629	0.083	0.910	0.090	0.268	0.760	0.265	
10	155	80	32.466	2.790	3.553	3.047	891197	4282	0.108	0.895	0.095	0.244	0.775	0.266	
Avg	183.2	163.9	32.461	3.102	3.551	3.029	1019106	326400.3	0.103	0.899	0.087	0.243	0.781	0.256	
Std	45.411	131.131	0.207	0.446	0.007	0.016	356478.4	1021295	0.020	0.012	0.012	0.040	0.027	0.073	
1	275	106	33.048	2.998	3.557	3.258	879625	2135764	0.075	0.907	0.087	0.140	0.879	0.062	
2	90	211	32.307	3.395	3.569	3.253	1608564	2371778	0.130	0.887	0.062	0.146	0.865	0.059	
3	145	66	32.378	2.844	3.552	3.247	918452	2389466	0.090	0.909	0.088	0.144	0.864	0.059	
4	273	155	32.913	3.185	3.559	3.276	833695	1909459	0.103	0.895	0.097	0.139	0.891	0.064	2000
5	319	106	33.235	2.977	3.545	3.263	1244451	2154769	0.119	0.908	0.077	0.143	0.878	0.061	
6	154	87	32.411	2.965	3.564	3.267	885852	2125750	0.112	0.900	0.095	0.142	0.879	0.062	
7	361	122	33.359	3.054	3.577	3.268	1586715	2133858	0.146	0.891	0.053	0.143	0.879	0.061	

8	174	136	32.827	3.165	3.543	3.269	1077360	1802195	0.116	0.921	0.080	0.134	0.895	0.065	
9	72	124	31.982	3.102	3.562	3.262	1677518	2094183	0.132	0.887	0.063	0.142	0.881	0.061	
10	586	178	34.539	3.272	3.567	3.268	1740699	1616996	0.139	0.886	0.055	0.130	0.902	0.068	
Avg	244.9	129.1	32.900	3.096	3.559	3.263	1245293	2073422	0.116	0.899	0.076	0.140	0.881	0.062	
Std	154.263	43.095	0.726	0.163	0.011	0.008	372399.3	239211.3	0.022	0.012	0.016	0.005	0.012	0.003	
1	196	87	33.483	3.294	3.563	3.416	659494	1937841	0.073	0.923	0.094	0.136	0.887	0.064	
2	198	87	33.134	3.190	3.565	3.426	1064953	1652549	0.106	0.899	0.087	0.125	0.896	0.068	
3	188	96	33.236	3.322	3.566	3.432	912963	1741937	0.095	0.894	0.093	0.131	0.894	0.066	
4	178	91	33.230	3.257	3.561	3.421	697073	1841789	0.074	0.918	0.093	0.136	0.893	0.065	
5	330	86	34.053	3.233	3.564	3.422	778996	1861490	0.108	0.887	0.101	0.131	0.888	0.065	
6	543	108	35.503	3.335	3.566	3.423	852023	1868729	0.098	0.893	0.096	0.138	0.892	0.064	
7	202	125	33.393	3.448	3.568	3.416	1039199	1863613	0.103	0.897	0.089	0.136	0.892	0.064	5000
8	210	105	33.342	3.335	3.562	3.426	770253	1807024	0.085	0.916	0.093	0.129	0.891	0.066	
9	236	87	33.447	3.219	3.563	3.426	649642	1739727	0.080	0.921	0.098	0.136	0.897	0.066	
10	856	87	37.525	3.174	3.554	3.425	455005	1742955	0.068	0.935	0.102	0.130	0.894	0.067	
Avg	313.7	95.9	34.035	3.281	3.563	3.423	787960.1	1805765	0.089	0.908	0.095	0.133	0.892	0.065	
Std	220.519	12.974	1.413	0.083	0.004	0.005	187001.1	84993.32	0.015	0.016	0.005	0.004	0.003	0.001	
1	312	89	34.566	3.622	3.567	3.479	1081143	1763696	0.110	0.892	0.088	0.135	0.895	0.065	
2	150	146	33.435	4.026	3.562	3.477	711499	1841128	0.080	0.916	0.096	0.136	0.892	0.063	
3	211	123	34.000	3.868	3.562	3.474	706822	1670024	0.077	0.921	0.092	0.126	0.895	0.068	
4	307	135	34.523	3.895	3.562	3.473	717717	1664345	0.077	0.920	0.093	0.126	0.895	0.068	
5	143	104	33.487	3.611	3.565	3.474	900553	1687928	0.097	0.902	0.093	0.131	0.897	0.066	
6	256	110	34.484	3.850	3.568	3.477	1005969	1787442	0.105	0.894	0.091	0.134	0.893	0.065	
7	201	115	33.551	3.820	3.567	3.484	980298	1812036	0.096	0.902	0.089	0.135	0.892	0.064	8000
8	182	120	33.685	3.851	3.568	3.469	1010513	1632010	0.107	0.900	0.089	0.120	0.894	0.069	
9	282	104	34.632	3.813	3.568	3.480	946102	1804848	0.105	0.892	0.093	0.133	0.893	0.064	
10	196	84	33.644	3.491	3.568	3.470	1049565	1870216	0.102	0.896	0.088	0.138	0.891	0.064	
Avg	224	113	34.001	3.785	3.566	3.476	911018.1	1753367	0.096	0.903	0.091	0.131	0.894	0.066	
Std	61.738	19.189	0.499	0.160	0.003	0.005	146082.9	83397.69	0.013	0.011	0.003	0.006	0.002	0.002	
1	351	110	35.962	3.839	3.562	3.497	684167	1889640	0.078	0.915	0.097	0.138	0.889	0.062	
2	222	282	34.563	5.553	3.568	3.494	946883	1780771	0.104	0.892	0.093	0.131	0.892	0.065	
3	172	199	33.799	4.994	3.564	3.491	716564	1756764	0.078	0.920	0.093	0.133	0.894	0.065	
4	163	224	33.569	5.286	3.558	3.493	751566	1810429	0.071	0.914	0.090	0.136	0.893	0.064	
5	201	227	34.286	5.103	3.569	3.489	904041	1769251	0.101	0.895	0.094	0.132	0.893	0.065	
6	299	243	35.415	5.133	3.569	3.498	930999	1838040	0.101	0.894	0.093	0.136	0.891	0.063	
7	308	245	35.100	5.192	3.580	3.500	1739286	1757399	0.149	0.883	0.051	0.131	0.892	0.065	10000
8	312	132	35.220	4.352	3.581	3.499	1720794	1773957	0.144	0.886	0.054	0.129	0.892	0.065	
9	221	674	34.397	9.326	3.569	3.475	942677	750303	0.104	0.892	0.093	0.084	0.913	0.095	
10	162	178	33.599	4.733	3.566	3.496	1052255	1799123	0.107	0.903	0.087	0.132	0.892	0.065	
Avg	241.1	251.4	34.591	5.351	3.569	3.493	1038923	1692568	0.104	0.900	0.084	0.128	0.894	0.067	
Std	70.270	157.540	0.816	1.482	0.007	0.007	382324.9	333619.3	0.026	0.013	0.017	0.016	0.007	0.010	

Table 29 Full Comparison of VBGMM on Full dataset, Coreset and Uniform Sampling on Human 3 Kidney Data

Run	VBGMM Steps		Time		Vbound		Count		Features						samples
									Coreset			Uniform			
	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Coreset	Uniform	Int	Blob	Flat	Int	Blob	Flat	
1	98	52	39.466	3.220	3.077	2.038	1444332	382	0.565	0.892	0.025	0.245	0.883	0.366	
2	293	44	39.529	3.146	3.101	2.020	1439993	253	0.260	0.910	0.080	0.240	0.863	0.381	
3	140	24	39.237	3.119	3.122	2.035	1515049	366	0.525	0.897	0.033	0.242	0.878	0.367	
4	116	27	39.064	3.105	3.109	2.018	1218102	245	0.242	0.917	0.088	0.241	0.860	0.382	
5	93	23	39.024	3.090	3.070	2.027	1291536	333	0.231	0.901	0.085	0.244	0.874	0.371	
6	221	57	39.330	3.179	3.061	2.071	793657	68	0.216	0.936	0.103	0.219	0.845	0.443	500
7	152	1125	39.214	6.179	3.122	2.002	1401897	164	0.229	0.892	0.081	0.232	0.835	0.398	
8	264	37	39.636	3.116	3.106	2.046	1209679	349	0.217	0.903	0.087	0.246	0.879	0.369	
9	123	24	39.282	3.083	3.112	2.011	1216222	202	0.217	0.892	0.087	0.239	0.852	0.390	
10	50	24	38.637	3.127	3.112	2.046	1653895	390	0.553	0.880	0.025	0.244	0.882	0.365	
Avg	155	143.7	39.242	3.437	3.099	2.031	1318436	275.2	0.325	0.902	0.070	0.239	0.865	0.383	
Std	79.006	345.024	0.287	0.964	0.022	0.020	235050.9	107.33	0.154	0.016	0.030	0.008	0.017	0.024	
1	107	50	39.125	3.194	3.130	2.182	1493641	3705	0.245	0.889	0.080	0.245	0.912	0.292	
2	78	133	39.110	3.491	3.127	2.283	1301201	341	0.216	0.891	0.084	0.250	0.884	0.370	
3	137	175	39.210	3.691	3.126	2.318	1546300	3168817	0.250	0.894	0.078	0.424	0.883	0.048	
4	191	99	39.315	3.398	3.103	2.305	1524386	351	0.384	0.903	0.045	0.247	0.875	0.368	
5	173	108	39.341	3.433	3.127	2.259	1364369	430	0.568	0.894	0.026	0.244	0.921	0.360	
6	66	162	38.959	3.599	3.119	2.308	1523820	3447161	0.545	0.894	0.028	0.430	0.868	0.046	1000
7	151	123	39.366	3.485	3.115	2.266	1467796	246	0.280	0.894	0.082	0.250	0.871	0.382	
8	162	159	39.562	3.747	3.116	2.294	1584345	3459475	0.287	0.905	0.077	0.428	0.868	0.046	
9	67	90	39.031	3.395	3.127	2.268	901586	414	0.212	0.931	0.098	0.244	0.915	0.362	
10	87	90	39.033	3.375	3.130	2.332	1443551	855262	0.244	0.900	0.081	0.190	0.925	0.098	
Avg	121.9	118.9	39.205	3.481	3.122	2.282	1415100	1093620	0.323	0.899	0.068	0.295	0.892	0.237	
Std	46.637	39.102	0.189	0.163	0.009	0.042	199616	1586881	0.132	0.012	0.025	0.093	0.023	0.156	
1	168	251	39.647	4.289	3.154	2.529	1217715	2690611	0.236	0.918	0.087	0.398	0.901	0.053	
2	121	106	39.305	3.703	3.127	2.574	1437494	1108917	0.275	0.913	0.081	0.211	0.918	0.088	
3	78	159	39.056	3.796	3.145	2.540	1011642	1062592	0.216	0.925	0.093	0.190	0.910	0.087	
4	221	103	39.867	3.500	3.129	2.567	1524608	1099306	0.261	0.900	0.079	0.209	0.917	0.089	
5	103	160	39.307	3.714	3.135	2.568	1422901	1165950	0.561	0.895	0.026	0.208	0.911	0.086	
6	216	177	39.669	3.886	3.138	2.537	1323649	2670252	0.225	0.895	0.084	0.407	0.903	0.053	2000
7	97	121	39.005	3.578	3.146	2.574	1129152	1133794	0.222	0.918	0.089	0.212	0.916	0.088	
8	250	178	39.963	3.818	3.134	2.569	1293030	1134222	0.232	0.903	0.086	0.208	0.914	0.087	
9	139	77	39.547	3.436	3.130	2.574	1573101	1159338	0.284	0.904	0.078	0.212	0.915	0.087	
10	94	93	39.137	3.480	3.136	2.565	1272597	1028160	0.232	0.910	0.086	0.201	0.918	0.091	
Avg	148.7	142.5	39.450	3.720	3.137	2.560	1320589	1425314	0.274	0.908	0.079	0.246	0.912	0.081	
Std	61.467	52.595	0.338	0.253	0.009	0.017	174887.9	662844.5	0.103	0.010	0.019	0.083	0.006	0.015	

1	105	197	39.436	4.362	3.140	2.817	1382570	1058369	0.234	0.897	0.083	0.206	0.918	0.089	
2	73	242	39.578	4.704	3.152	2.820	1080114	1059570	0.219	0.920	0.091	0.198	0.914	0.089	
3	111	165	39.503	4.151	3.142	2.814	1402495	1077857	0.234	0.894	0.082	0.199	0.913	0.088	
4	118	164	39.686	4.191	3.140	2.826	1407542	1104854	0.228	0.890	0.082	0.203	0.913	0.087	
5	114	416	39.799	5.707	3.141	2.815	1340387	1058685	0.222	0.893	0.083	0.199	0.915	0.089	
6	200	869	39.940	8.859	3.130	2.813	1474095	1122303	0.508	0.905	0.034	0.208	0.915	0.087	
7	116	154	39.450	4.096	3.151	2.825	1051815	1081356	0.220	0.924	0.092	0.203	0.915	0.088	5000
8	236	267	40.708	4.791	3.151	2.821	1034855	1110317	0.219	0.924	0.093	0.204	0.914	0.087	
9	233	199	40.593	4.333	3.140	2.817	1241218	1089524	0.211	0.891	0.086	0.203	0.915	0.088	
10	151	329	39.985	5.303	3.143	2.815	1345947	1058004	0.226	0.893	0.084	0.200	0.916	0.089	
Avg	145.7	300.2	39.868	5.050	3.143	2.818	1276104	1082084	0.252	0.903	0.081	0.202	0.915	0.088	
Std	57.302	216.374	0.455	1.439	0.007	0.005	163616.1	24045.18	0.090	0.014	0.017	0.003	0.001	0.001	
1	108	205	39.785	5.292	3.138	2.921	1518410	1110898	0.250	0.892	0.079	0.210	0.916	0.088	
2	121	268	39.946	5.433	3.143	2.908	1422839	1106794	0.234	0.891	0.082	0.210	0.916	0.089	
3	168	230	40.647	5.147	3.140	2.914	1327481	1065915	0.229	0.898	0.084	0.205	0.917	0.090	
4	114	151	39.793	4.413	3.140	2.914	1477388	1087838	0.567	0.888	0.025	0.204	0.915	0.089	
5	104	183	39.991	4.732	3.142	2.912	1438729	998809	0.240	0.895	0.081	0.195	0.917	0.091	
6	143	197	39.952	4.910	3.143	2.919	1325998	1017176	0.222	0.892	0.084	0.197	0.915	0.091	
7	109	175	39.725	4.672	3.142	2.913	1395836	1067340	0.230	0.891	0.082	0.205	0.917	0.090	8000
8	99	193	39.741	4.789	3.142	2.915	1361113	1011253	0.231	0.895	0.083	0.195	0.915	0.091	
9	195	224	40.647	5.066	3.138	2.913	1415154	998833	0.240	0.898	0.082	0.196	0.917	0.091	
10	128	130	40.322	4.336	3.142	2.919	1344229	1055165	0.223	0.891	0.083	0.203	0.917	0.090	
Avg	128.9	195.6	40.055	4.879	3.141	2.915	1402718	1052002	0.267	0.893	0.077	0.202	0.916	0.090	
Std	31.143	39.727	0.357	0.360	0.002	0.004	64603.42	43106.74	0.106	0.003	0.018	0.006	0.001	0.001	
1	126	162	40.395	4.912	3.153	2.953	1030735	1038093	0.214	0.923	0.092	0.202	0.917	0.091	
2	154	145	40.434	4.773	3.143	2.954	1363219	1034491	0.229	0.895	0.083	0.203	0.918	0.091	
3	93	191	39.816	5.088	3.142	2.956	1354890	1092108	0.228	0.896	0.083	0.207	0.916	0.089	
4	85	162	39.690	4.797	3.143	2.955	1349044	1061773	0.227	0.894	0.083	0.202	0.915	0.090	
5	169	139	40.344	4.619	3.140	2.950	1478606	1041176	0.562	0.891	0.026	0.203	0.917	0.090	
6	133	145	40.151	4.639	3.143	2.951	1364013	1041494	0.232	0.895	0.083	0.201	0.915	0.090	
7	138	178	40.302	4.981	3.143	2.959	1330780	1075490	0.222	0.892	0.084	0.207	0.917	0.090	10000
8	132	254	40.332	5.672	3.143	2.952	1329364	1076654	0.223	0.894	0.084	0.201	0.913	0.089	
9	113	159	39.810	4.777	3.143	2.951	1367514	1028604	0.233	0.896	0.083	0.203	0.918	0.091	
10	227	324	41.072	6.388	3.143	2.946	1338795	1119470	0.223	0.892	0.084	0.207	0.913	0.088	
Avg	137	185.9	40.235	5.065	3.144	2.953	1330696	1060935	0.259	0.897	0.079	0.204	0.916	0.090	
Std	40.541	58.887	0.401	0.555	0.003	0.004	113708.5	29613.15	0.107	0.009	0.019	0.002	0.002	0.001	