Structures in High-Dimensional Data: Intrinsic Dimension and Cluster Analysis

Johnsson, Kerstin

2016

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Johnsson, K. (2016). *Structures in High-Dimensional Data: Intrinsic Dimension and Cluster Analysis* (150 ed.). [Doctoral Thesis (compilation), Faculty of Engineering, LTH]. Centre for Mathematical Sciences, Lund University.

Total number of authors:
1

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Structures in High-Dimensional Data: Intrinsic Dimension and Cluster Analysis

Kerstin Johnsson

Lund University

ACADEMIC THESIS

which, by due permission of the Faculty of Engineering at Lund University, will be publicly defended on Friday 9th of September, 2016, at 13:15 in lecture hall MA:1, Annexet, Sölvegatan 20, Lund, for the degree of Doctor of Philosophy in Engineering.

Faculty opponent

Benno Schwikowski, Institut Pasteur, Paris, France.

Structures in High-Dimensional Data: Intrinsic Dimension and Cluster Analysis

Kerstin Johnsson

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematics

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118
SE-221 00 Lund
Sweden

http://www.maths.lth.se/

Doctoral Theses in Mathematical Sciences 2016:6
ISSN 1404-0034

ISBN 978-91-7623-920-9
LUTFMA-1057-2016

Printed in Sweden by MediaTryck, Lund 2016

Abstract

With today's improved measurement and data storing technologies it has become common to collect data in search for hypotheses instead of for testing hypotheses—to do exploratory data analysis. Finding patterns and structures in data is the main goal. This thesis deals with two kinds of structures that can convey relationships between different parts of data in a high-dimensional space: manifolds and clusters. They are in a way opposites of each other: a manifold structure shows that it is plausible to connect two distant points through the manifold, a clustering shows that it is plausible to separate two nearby points by assigning them to different clusters. But clusters and manifolds can also be the same: each cluster can be a manifold of its own.

The first paper in this thesis concerns one specific aspect of a manifold structure, namely its dimension, also called the intrinsic dimension of the data. A novel estimator of intrinsic dimension, taking advantage of "the curse of dimensionality", is proposed and evaluated. It is shown that it has in general less bias than estimators from the literature and can therefore better distinguish manifolds with different dimensions.

The second and third paper in this thesis concern cluster analysis of data generated by flow cytometry—a high-throughput single-cell measurement technology. In this area, clustering is performed routinely by manual assignment of data in two-dimensional plots, to identify cell populations. It is a tedious and subjective task, especially since data often has four, eight, twelve or even more dimensions, and the analysts need to decide which two dimensions to look at together, and in which order.

In the second paper of the thesis a new pipeline for automated cell population identification is proposed, which can process multiple flow cytometry samples in parallel using a hierarchical model that shares information between the clusterings of the samples, thus making corresponding clusters in different samples similar while allowing for variation in cluster location and shape.

In the third and final paper of the thesis, statistical tests for unimodality are investigated as a tool for quality control of automated cell population identification algorithms. It is shown that the different tests have different interpretations of unimodality and thus accept different kinds of clusters as sufficiently close to unimodal.

Populärvetenskaplig sammanfattning

Att söka strukturer i data — Intrinsisk dimension och klustring

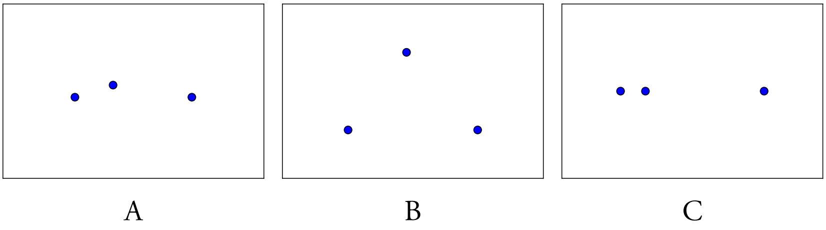
År 1604 lyckades Kepler efter mer än 40 försök med olika ovala figurer anpassa en ellips till de mätningar av Mars planetära bana som Tycho Brahe under åratal nedtecknat. Före detta genombrott var Kopernicus heliocentriska världsbild lika komplicerad som den tidigare geocentriska. Argumenten för Kopernicus världsbild var till stor del teologiska. Nu kunde man börja hävda en vetenskaplig grund i modern bemärkelse. En ellips, en matematisk struktur, gav modellen en enkelhet som inte bara gjorde den mer trolig, utan även gjorde det möjligt för Newton åttio år senare att förklara planeternas rörelser med gravitationskraften och mekanikens lagar.

Letandet efter två olika typer av strukturer i data är huvudtemat i denna avhandling. Den första typen är så kallade mångfalder inbäddade i högre dimensioner. Exempelvis är en kurva en endimensionell mångfald, som vi är vana att se i avbildad i två eller tre dimensioner, men som rent matematiskt likaväl kan finnas i ett sju- eller hundradimensionellt rum. En yta är på motsvarande sätt en tvådimensionell mångfald. Forskningen i avhandlingen handlar om att avgöra dimensionen hos en mångfald beskriven genom data. Det mest förvånande resultatet är att vi med god precision kan avgöra dimensionen hos en mångfald som har högre dimension än antalet datapunkter. Till exempel, om man har tre punkter kan man alltid hitta ett plan som går precis genom de tre punkterna. Hur kan vi då veta att de egentligen kommer från en mångfald av högre dimension?

Den andra typen av struktur är så kallade kluster, med andra ord grupperingar av näraliggande datapunkter. Hur kan man matematiskt avgöra vilka punkter som borde tillhöra samma kluster/grupp? I avhandlingen studeras två frågeställningar relaterade till klustring. Den ena frågeställningen som studeras är hur vi med statistiskt pålitliga metoder kan avgöra huruvida data kommer från ett eller flera kluster. Den andra handlar om klustring av data från upprepade experiment. I de olika experimenten förväntar vi oss att se samma ungefär grupperingar, men variation mellan experimenten gör att varje grupp kan ha förändrats något: den kan ha flyttats, ändrat form och/eller ändrat storlek. Vi har tagit fram en metod som genom en hierarkisk matematisk modell kan dela information om vilka grupperingar som finns mellan datan från de olika experimenten. Modellen kan även ta hänsyn till tidigare erfarenhet om grupperingar man är intresserad av. Att studera dessa klustringsproblem behövs för att förbättra analysmetoder av avancerade mätningar på celler som görs med en så kallad flödescytometer. Vi återkommer till flödescytometri och klustring senare, först ska vi titta närmare på beräkning av dimension.

Dimension av mångfalder

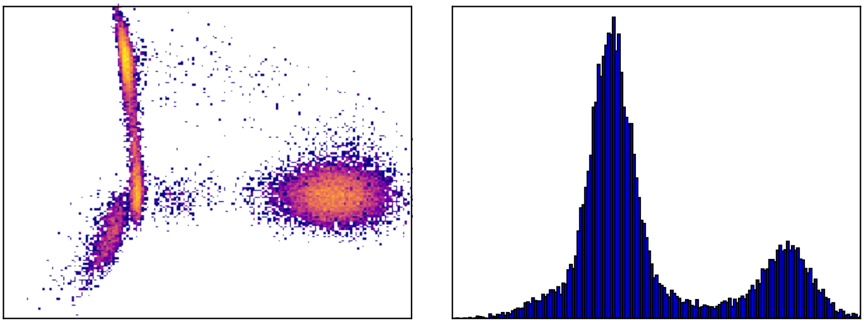
Bilderna A-C visar tre datamängder med tre punkter vardera som är genererade från tre mångfalder med dimension 1, 2 respektive 100 och där bildplanet har anpassats till de tre punkterna. Kan du gissa vilken datamängd som kommer från vilken mångfald? Naturligtvis är det ett omöjligt problem, om vi inte gör några ytterligare antaganden om mångfalderna. Till exempel kan vi för alla tre datamängderna dra en kurva genom alla punkter som passar perfekt. Men om vi antar att mångfalden har låg kurvatur, alltså att den inte kröker särskilt mycket? Då kan du nog direkt gissa att C kommer från en endimensionell mångfald. Men för att komma vidare behöver vi förstå en del om geometri i högdimensionella rum. Anta att du befinner dig i ett rum av dimension n. Allt som befinner sig inom ett avstånd r från dig kallar vi din r-omgivning. Den här omgivningen blir ett klot,



som du befinner dig i centrum av. Rakt ovanför ditt huvud finns en punkt som vi kallar Nordpolen. Om klotet hade varit tredimensionellt, som vanliga klot, så hade du med hjässan fortfarande riktad mot Nordpolen kunnat se rakt ut över en yta, ett tvådimensionellt rum, alltså ett rum med två riktningar som var vinkelräta mot varandra. Men i det n-dimensionella rummet kan du med hjässan mot Nordpolen titta i $n-1$ olika riktningar som är vinkelräta mot varandra. Faktum är att området i de här $n-1$ olika riktningarna som befinner sig inuti klotet som är din r-omgivning bildar ett $n-1$-dimensionellt klot. Ett tvådimensionellt klot är en cirkelskiva, och i det vanliga tredimensionella klotet handlar det om den skiva man får om man gör ett tvärsnitt vid ekvatorn. Ett märkligt fenomen i höga dimensioner är att även om man gör detta tvärsnitt väldigt tunt tar det en allt större andel av det totala klotets volym. I höga dimensioner upptar skivan nästan hela klotets volym. Om man slumpar ut en punkt i klotet kommer den alltså med största sannolikhet att ligga i denna skiva. Det leder till att relativt riktningen till vår referenspunkt, Nordpolen, så kommer den utslumpade punkten att ligga i en riktning som är vinkelrät.

I avhandlingen har vi tagit fram en metod för att skatta dimensionen hos en datamängd baserat på denna idé—att mäta vinklar mellan riktningarna från en referenspunkt till punkter i datamängden och avgöra hur nära vinkelräta riktningarna är. Detta görs med referenspunkter på många ställen i datamängden, och för att antagandet om låg kurvatur ska vara riktigt inkluderas bara punkter nära referenspunkten. På så sätt får man en lokal skattning av dimensionen. Det visar sig att vår metod har betydligt lägre systematiskt fel för skattningar av dimension än andra metoder och att detta leder till att man bättre kan skilja mellan datamängder med olika dimension.

Hur var det då med mängderna A och B? Vi börjar med att lägga en referenspunkt i tyngdpunkten av de tre punkterna i varje mängd. Sedan mäter vi vinklarna mellan riktningarna till de tre datapunkterna. I A får vi två vinklar som är nära 180°och en vinkel som är liten, alltså är riktningarna ganska långt ifrån vinkelräta. I B är vinklarna närmre 90°, vilket innebär att det är mycket troligare att B kommer ifrån en mångfald med hög dimension. Alltså A är 2-dimensionell, B är 100-dimensionell och C är 1-dimensionell!



Tvådimensionellt (vänster) respektive endimensionellt (höger) histogram av flödescytometridata. Datan är hämtad från R-paketet healthyFlowData.

Klustring och flödescytometri

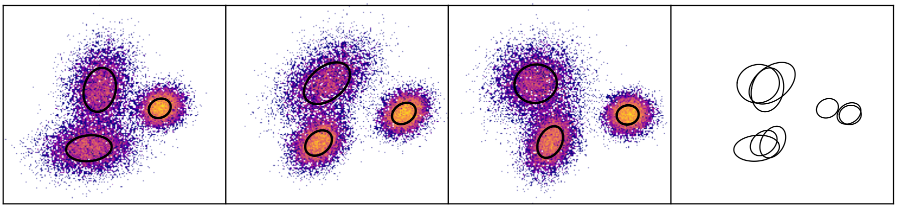
Allt sedan Newtons och Keplers dagar är datainsamling och anpassande av matematiska modeller till insamlade data en hörnsten inom vetenskap. Mer detaljerade data kan avslöja brister i gängse förklaringsmodeller eller leda till nya banbrytande hypoteser. Därför har jakten på nya data varit en motor inom vetenskapen, biomedicin är inget undantag. Mätinstrument och datalagringsteknologi har genomgått en revolutionerande utveckling det senaste decenniet och idag produceras i ett enskilt experiment mer data än någon kan överblicka. En DNA-sekvenserare kan på några timmar läsa av alla tre miljarder baser i en människas genetiska kod. En flödescytometer kan på några minuter ge 15-dimensionella mätningar på cellnivå av 100 000 enskilda celler. Det skulle ta en halvtimme för en forskare att bara scrolla igenom datan från ett enda sådant experiment i ett Excel-ark. Detta innebär att datan måste behandlas och sammanfattas med olika algoritmer innan den presenteras för forskaren. Så länge ett fåtal parametrar är uppmätta per objekt, till exempel om man mäter en eller två egenskaper per cell, har datan låg dimension och kan sammanfattas väl med en- eller tvådimensionella histogram.

I histogrammen av ett flödescytometriprov kan vi se att vi får grupper av celler. Det vanligaste sättet att sammanfatta ett sådant prov på är att rita in en indelning i cellpopulationer i histogrammen och därefter beräkna egenskaper såsom antal celler och medelvärde för de olika parametrarna som är uppmätta, för varje cellpopulation. Ett klassiskt exempel där detta används är övervakning av HIV-infektioner, där en cellpopulation med celler med högt uttryck av två markörer, som kallas CD3 och CD4, blir mindre allt eftersom sjukdomen förvärras. Idag är det vanligt att man använder flödescytometriprov med betydligt fler markörer, ofta runt ett tiotal, både inom forskning kring och diagnostik av blodsjukdomar och sjukdomar hos immunsystemet.

Men att göra en indelning genom att rita i histogram fungerar bara rent praktiskt i upp till två dimensioner. Har man mätt fler parametrar väljer man ut en eller två i taget, gör en indelning, och delar sedan in de populationer man får i subpopulationer genom att välja ut nya parametrar. Man behöver alltså en strategi för vilken sekvens av parametrar man ska välja. På detta sätt blir indelningen i populationer subjektiv och dessutom suboptimal eftersom man inte kan ta hänsyn till alla parametrar samtidigt. Det är också arbetskrävande, särskilt som man kan ha hundratals prover som man behöver göra analysen på. Därför vill man ta fram automatiserade metoder för indelning i populationer. Ett sätt att göra detta är att bygga en modell för hur en cellpopulation ser ut och sedan kombinera sådana modeller för att beskriva hela datamängden. Till exempel i det endimensionella histogrammet över flödescytometridata ovan ser det ut ungefär som att det finns två överlagrade normalfördelningar. Vår modell kan alltså vara att varje cellpopulation beskrivs av en normalfördelning. Det fungerar bra även för högre dimensioner; då använder man sig av den multivariata normalfördelningen. Ett flödescytometriprov i fyra dimensioner kan kanske beskrivas väl av ett tiotal överlagrade multivariata normalfördelningar.

Men, eftersom cellpopulationerna överlappar varandra och normalfördelningen bara är en approximation och inte beskriver datan exakt, så kan det finnas flera modeller som passar datan ungefär lika bra. Av slump kan då en variant väljas för ett flödescytometriprov och en helt annan för ett ett annat prov i samma undersökning. Det innebär att man inte kan jämföra cellpopulationerna från olika prov med varandra. För att hantera detta presenteras i avhandlingen en hierarkisk modell för flödescytometridata, där det finns en grundnivå som är en modell för en enskilt prov och en metanivå som kombinerar parametrarna från olika prov.

Ett exempel på hur modellen fungerar visas i figuren ovan, där modellens grundnivå är illustrerad för tre flödescytometriprov med ellipser som visar de multivariata normalfördelningarnas form. Längst till höger illustreras metanivån som samlar grundmodellerna för de olika proverna. I avhandlingen applicerar vi vår modell på en välkänd referensdatamängd och visar att vi får en indelning i cellpopulationer som är mer jämförbar mellan olika prover än indelningar gjorda antingen manuellt eller med andra automatiserade metoder. Modellen är formu-



Hierarkisk modell för flödescytometridata. Proverna har mycket gemensamt, men är samtidigt lite olika.

lerad i ett Bayesianskt ramverk där vi ovanför metanivån även har en a priori-nivå som beskriver den kunskap vi har om cellpopulationerna i förväg. Vi visar att detta särskilt kan vara till hjälp för att hitta små populationer som annars inte hade detekterats. Som nämndes i inledningen så har ytterligare en frågeställning kring klustring av flödescytometridata studerats — nämligen hur man kan avgöra om en grupp mätningar tillhör en eller flera populationer. Men om två cellpopulationer är väldigt lika för de parametrar man uppmätt, kan man då avgöra att det faktiskt är två populationer? Nej, ibland är det är omöjligt! Vi måste formulera om frågan: Är rimligt att den data vi mäter upp kan komma från en enda cellpopulation? Nu vill vi inte anta att cellpopulationerna följer en viss fördelning, t.ex. normalfördelningen, eftersom detta ofta är för restriktivt, utan ha en mer generell modell för vår population. Den modell vi utnyttjar oss av gör det enda antagandet att fördelningen som en cellpopulation följer ska vara så kallat unimodal. Det betyder ungefär att i ett histogram över datan ska det bara finnas en topp. Mer precist gäller detta bara om man har oändligt mycket data, eftersom man med ändligt mycket data kan få små extra toppar av slump, som man kan se i det endimensionella histogrammet ovan. Eftersom man i praktiken alltid har ändligt mycket data betyder det att man behöver avgöra om de små toppar man har är signifikanta och bryter mot antagandet för unimodalitet. I den statistiska litteraturen finns det framför allt två tester för unimodalitet som har studerats väl: dip-testet och bandbreddstestet. Genom att applicera testerna på data som blivit manuellt indelad i färdiga cellpopulationer kan vi se om de statistiska testerna överensstämmer med den traditionella bilden av hur cellpopulationer ska se ut. Vi kommer fram till att dip- och bandbreddstestet tolkar unimodalitet på olika sätt och att båda sätten kan vara viktiga för att beskriva cellpopulationer i flödescytometridata.

Acknowledgements

First I would like to thank my supervisor Magnus Fontes, who has always been supportive, who has given countless valuable advice during the years and who has connected me to people with interesting research problems.

Three other people have also been especially important for this work. The first is my co-supervisor Charlotte Soneson who played an important part during the first half of my PhD—I've learned a lot about doing research in bioinformatics from you. The second is Jonas Wallin, with whom I've had a lot of scientific discussions, not only regarding the specific problems we've collaborated on, but also on how to perform research in general. You also made me start using Python, for which I am especially grateful. The third person is my husband Magnus Linderoth, with whom I can discuss practically everything, including details of my research, and who has given valuable feedback on many parts of this thesis. The three of you have made me grow as a researcher, programmer and engineer.

In the first part of my thesis work I had much use of the differential and Riemannian geometry that I was taught in an inspiring manner by Sigmundur Gudmundsson. I read his well-written booklets carefully and have turned back to them often for reference.

During my stay at the Pasteur Institute in Paris I learned a lot about the world of immunology and flow cytometry from many people. Especially memorable were the sessions with Matthew Albert, where I got introduced to many fascinating aspects and open problems of the immune system. Alejandra Urrutia, Milena Hasan, Molly Ingersoll and Xiaoyi Chen taught me a lot about flow cytometry and flow cytometry data analysis. With my colleagues in the IGDA research group—Rasmus Henningsson, Jacob Antonsson and J Boussier—I got the opportunity to discuss immunology-related data analysis problems.

Many people at the Centre for Mathematical Sciences and Lund University have supported various aspects of my PhD work. I would like to especially mention Joachim Hein at the Lunarc high-performance computing centre, who has been very helpful in resolving many issues, and our librarian Mikael Abrahamsson who has helped me to get access to resources from near and afar.

My colleagues here at the Centre for Mathematical Sciences have all participated in creating a nice working environment, I would like to especially thank Hanna Källén and Matilda Landgren.

Finally, I would like to thank my family: My wonderful husband Magnus Linderoth who makes me happy and supports me in so many different ways in life, and my parents Karin and Per-Ingvar who are not only great role models, but also have ensured that I got the encouragement and opportunities to train skills I use daily as a researcher and mathematician.

Contents

Abstract iii

Populärvetenskaplig sammanfattning v

Acknowledgements xi

1 Introduction 1

1.1 Contributions . . . 1

2 Intrinsic dimension 5

2.1 Manifold-based definition of intrinsic dimension . . . 6

2.2 Topological dimension . . . 9

2.3 Fractal dimension as intrinsic dimension . . . 11

2.4 The curse of dimensionality . . . 12

2.4.1 The concentration of measure phenomenon . . . 12

3 Cluster analysis 15

3.1 Clustering algorithms . . . 16

3.2 Model-based clustering . . . 18

3.2.1 Bayesian hierarchical models . . . 19

3.2.2 Considerations when using model-based clustering . . . 22

3.3 Evaluating clusterings and finding the number of modes . . . 23

4 Flow cytometry data 25

4.1 What is flow cytometry? . . . 25

4.2 Fluorescence flow cytometry . . . 27

4.3 Mass cytometry . . . 28

4.4 Sources of technical variation 28
4.5 Applications 30
4.6 Data properties 31
4.7 Cell population identification 31

5 Conclusions **35**

Bibliography **37**

Paper I: Low bias local intrinsic dimension estimation from expected simplex skewness **53**
1 Introduction 53
1.1 Related Work 55
2 Methods 57
3 Experiments 60
3.1 Evaluated Estimators 60
3.2 Data Analysis 61
3.3 Dimension Estimation of Synthetic Data Sets 62
3.4 Stratification of Synthetic Data 66
3.5 Dimension Estimation of Real Data 68
3.6 Stratification of Real Data Sets 69
4 Conclusions 70
Bibliography 71
A Closed Expressions for $s_n^{(d)}$ and c_n 76
B Variance of ESS Estimators 77
C Consistency of ESS Estimators 79
D Further Details on Parameters 81
D.1 Estimation of Noise Parameters for TP Estimator 81
D.2 Parameter Selection for Hill, TP, kNN and DANCo . . 82
E Further Details on Results from Experiments in the Main Article 82
F Further Experiments on Synthetic Data Sets 88
G Further Experiments on Real Data Sets 93

Paper II: BayesFlow: Latent modeling of flow cytometry cell populations **99**
1 Introduction 100
2 Methods 104
2.1 Model 104

2.2 Merging latent clusters 107
2.3 Quality control 109
2.4 Experiments 110
2.4.1 Simulated data 110
2.4.2 Flow cytometry data 110
3 Results 113
3.1 Simulated data 113
3.2 Flow cytometry data 117
3.2.1 GvHD 117
3.2.2 healthyFlowData 119
4 Discussion 122
5 Conclusions 124
Bibliography 126
A Posterior 131
B Sampling from the posterior distribution 131
B.1 Label switching 133
B.2 Cluster activation and deactivation 133
C Merging latent clusters 136
D Simulation study 138
D.1 Data generation 138
D.2 Priors 139
D.3 Initialization for small dataset 139
D.4 Initialization for large dataset 140
E Flow cytometry data analysis 141
E.1 Dataset details 141
E.1.1 healthyFlowData 141
E.2 Priors 141
E.3 Point estimates 142
E.4 Quality control 142
E.4.1 Convergence 142
E.4.2 Unimodality 143
E.4.3 Eigenvectors 143
E.5 Parameters and convergence for ASPIRE 147
E.6 GvHD scatterplots 148
E.7 Individual GMM models with EM for healthyFlowData 148

Paper III: What is a 'unimodal' cell population? — Investigating calibrated dip and bandwidth tests for quality control of gating of flow cytometry data **155**

1 Introduction . . . 155
2 Methods . . . 159
2.1 Dip and critical bandwidth . . . 159
2.2 Calibration . . . 159
2.3 Adaptive resampling . . . 160
2.4 Interval selection for bandwidth test . . . 161
2.5 Blurring . . . 161
2.6 Extreme data points . . . 162
3 Experiments . . . 162
3.1 Data . . . 162
3.1.1 Synthetic data . . . 162
3.1.2 Flow cytometry data . . . 163
4 Results . . . 163
4.1 Synthetic data . . . 163
4.2 Flow cytometry data . . . 164
5 Discussion . . . 169
6 Acknowledgments . . . 171
Bibliography . . . 171
A Grid spacing for estimating critical bandwidths . . . 175
B Calibration . . . 175
B.1 Strict unimodality or shoulder? . . . 176
B.2 Probabilistic bisection search . . . 177
B.3 Adaptive probabilistic bisection search . . . 178
C Blurring . . . 178
D Data . . . 180
D.1 Synthetic data . . . 180
E Synthetic data results . . . 180
E.1 Reference distributions . . . 180
E.2 Sensitivity to shoulder shape and number of points . . . 182
E.3 Power . . . 182
F Flow cytometry data results . . . 182

Chapter 1

Introduction

This thesis treats three topics: intrinsic dimension, cluster analysis and flow cytometry data. The chapters 2–4 give an introduction to each of these topics, to prepare for and give context to the three papers which are the main contributions of this thesis.

In Chapter 2 intrinsic dimension is introduced and defined in various ways. The different definitions are related to each other and to estimators of intrinsic dimension. Furthermore, one section is devoted to the "curse of dimensionality" and the concentration phenomenon, since these are the basis for the estimators presented in Paper I. For the non-expert reader a warning should be issued that this is the most technical chapter.

In Chapter 3 an introduction to clustering and clustering algorithms is given, with a focus on model-based clustering. Bayesian hierarchical models and inference for such models are introduced to give background to the clustering algorithm presented in Paper II. Finally cluster evaluation methods are discussed, to put the methods presented in Paper III into perspective.

Chapter 4 gives an introduction to flow cytometry and flow cytometry data analysis, and gives background and motivation for the cell population identification pipeline BayesFlow presented in Paper II.

Finally, Chapter 5 gives an outlook for the methods presented in this thesis.

1.1 Contributions

This thesis is built upon three papers:

Paper I

K. Johnsson, C. Soneson and M. Fontes. Low bias intrinsic dimension estimation from expected simplex skewness. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(1): 196–202, 2015.

In this paper, we propose a set of new estimators of intrinsic dimension. We characterize the estimators on synthetic as well as real data and and prove consistency for the estimators. Compared to other estimators of intrinsic dimension it has lower bias, which makes it better at distinguishing between data sets of different dimension.

Author contributions: MF suggested the topic of study. MF and KJ constructed the ESS estimator. KJ did the literature survey and implemented all the evaluated estimators. KJ, CS and MF designed the experiments and they were run by KJ. KJ developed the theoretical results of the ESS estimator with the help of MF. KJ wrote the paper with the help of MF and CS. All authors read and approved the final version of the manuscript.

Paper II

K. Johnsson, J. Wallin and M. Fontes. BayesFlow: latent modeling of flow cytometry cell populations. *BMC Bioinformatics*, 17(25), 2015.

In this paper, we propose a pipeline for automated cell population identification in flow cytometry data. At the heart of the pipeline is a Bayesian hierarchical model for joint clustering of multiple flow cytometry samples in a batch. We show that the pipeline gives results which are more comparable across samples compared to competing methods.

Author contributions: KJ, JW and MF conceived and planned the study. JW and KJ designed the statistical model and the inference procedure. KJ, JW and MF designed the experiments. JW and KJ implemented BayesFlow and ran the experiments. KJ and JW wrote the article with the help of MF. All authors read and approved the final version of the manuscript. KJ and JW assert equal contributions to the paper.

Paper III

K. Johnsson and M. Fontes. What is a ‘unimodal’ cell population? — Investigating calibrated dip and bandwidth tests for quality control of gating of flow cytometry data. *Submitted for publication.*

In this paper, the calibrated dip and bandwidth tests of unimodality are investigated with the purpose of describing how they can be used for quality control of the cell population identification process of flow cytometry data. The tests are shown to have complementary properties for matching populations assigned by application experts.

Author contributions: KJ conceived and planned the study. KJ did the literature survey and the implementation work. KJ designed the experiments under some discussion with MF. KJ wrote the paper with the help of MF. Both authors read and approved the final version of the manuscript.

Chapter 2

Intrinsic dimension

A data set typically consists of measurements made on a sequence of objects or instances. Each kind of measurement is represented by a variable, and the number of variables is called the (extrinsic) dimension of the data. In many cases the measurements are related; an example is given in Fig. 2.1. For each data point we have an x and a y coordinate, but it could equally well be represented with an angle—which gives a one-dimensional data set. Finding a better representation shows that the data set has lower complexity than what the extrinsic dimension would suggest.

However, to find such a representation is often hard—even when you can manage it, how can you know that there is not an even better one with fewer variables? Estimators of *intrinsic dimension* can measure data complexity without the need to find lower-dimensional representations.

A suite of such estimators is proposed in Paper I. To clarify precisely what is estimated, this chapter discusses definitions of intrinsic dimension. As we will see it is non-trivial to make a stringent definition, but an attempt is made in Definition 2.4. Other definitions used in the literature are discussed in Sections 2.2 and 2.3. The chapter is concluded by introducing "the curse of dimensionality" and the concentration of measure phenomenon, which are not only important for the estimators proposed in Paper I, but also gives some further intuition about high-dimensional data sets.

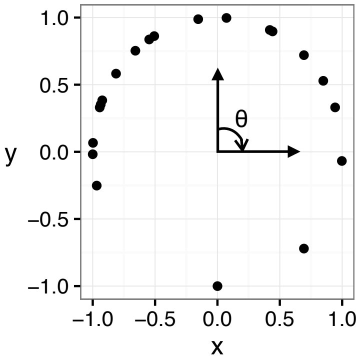


Figure 2.1: A two-dimensional data set with intrinsic dimension 1.

2.1 Manifold-based definition of intrinsic dimension

Following (Bruske and Sommer, 1998, Costa and Hero III, 2004, Hein and Audibert, 2005, Raginsky and Lazebnik, 2005), the definition of intrinsic dimension used in Paper I is based on the notion of a *smooth manifold*. A basic requirement for a set $\mathcal{M}$ to be a manifold is being a topological Hausdorff space, i.e. it needs a topology with the Hausdorff property. In our treatment we always have that $\mathcal{M} \subset \mathbb{R}^D$, where D is the extrinsic dimension of the data, and $\mathcal{M}$ has the topology induced from $\mathbb{R}^D$, which is Hausdorff and even normal.

Definition 2.1. The topological Hausdorff space $\mathcal{M}$ is a *manifold* with *dimension* n if for each $p \in \mathcal{M}$ there is an open set $U \ni p$ and a mapping $x \colon U \to \mathbb{R}^n$ that is continuous with $x(U)$ open, and whose inverse $x^{-1} \colon x(U) \to U$ also is continuous.

This means that for each point p in a manifold with dimension n there is a neighborhood that is a continuous deformation of an open set in $\mathbb{R}^n$. Any point in this neighborhood can be described by n parameters, using the map x^{-1} and the usual coordinates in $\mathbb{R}^n$. But continuity of x and x^{-1} is not enough to ensure that an open ball around p is well approximated by an open ball on an n-dimensional subspace of $\mathbb{R}^d$. For this, the mappings also need to be smooth.

Definition 2.2. A manifold $\mathcal{M}$ is *smooth* if it is equipped with an atlas of local mappings $\mathcal{A} = \{(U_\alpha, x_\alpha)\}, \alpha \in I$ such that

$$\bigcup_{\alpha \in I} U_\alpha \supset \mathcal{M}$$

and for all $\alpha, \beta \in I$

$$x_\beta \circ x_\alpha^{-1}|_{x_\alpha(U_\alpha \cap U_\beta)} \colon x_\alpha(U_\alpha \cap U_\beta) \subset \mathbb{R}^n \to \mathbb{R}^n$$

is smooth, i.e. infinitely differentiable.

Defining intrinsic dimension also requires the concept of a smooth function on $\mathcal{M}$.

Definition 2.3. A function $f \colon \mathcal{M} \to \mathbb{R}$ on a smooth manifold equipped with an atlas $\mathcal{A}$ is *smooth* if for every $(U, x) \in \mathcal{A}$ we have that $f \circ x^{-1} \colon x(U) \subset \mathbb{R}^n \to \mathbb{R}$ is smooth.

The stage is now set for defining intrinsic dimension. It seems natural to define the intrinsic dimension of a data set to be n if it is sampled from a smooth density on a smooth manifold of dimension n. This is not appropriate however, for two reasons. First, the manifold model might not be exact; the data might be close to the manifold instead of directly on the manifold. Second, even when data set is sampled from a smooth density on a smooth manifold of dimension n it is always possible to fit smooth manifolds of dimensions $m < n$ to the data—for example, for any finite set of data points we can draw a smooth curve, i.e. a 1-dimensional manifold, passing through them. Thus we cannot define intrinsic dimension for a data set itself, but only for the process generating it.

Definition 2.4. A process generating data Y_i has *intrinsic dimension* n if Y_i can be written as $Y_i = X_i + \epsilon_i$, where X_i is sampled according to a probability measure with a smooth density that is supported on a smooth n-dimensional manifold $\mathcal{M}$ and ϵ_i is a noise component which is small on a scale where $\mathcal{M}$ is well approximated by an n-dimensional subspace.

To estimate intrinsic dimension for a data set thus actually means estimating the intrinsic dimension of the process that has generated it. We show in Appendix C of Paper I that the proposed estimators are consistent in the case where we have no noise, and illustrate why it is not possible to obtain consistency when we do have noise. The reason for this is that on very small scales the noise will be dominant and the intrinsic dimension of the noise will be estimated instead of the data dimension. In practice, it will be a prerequisite to have a way to define neighborhoods in the data where the noise level is assumed to be small in comparison to the neighborhood size at the same time as the manifold is approximately flat in the neighborhoods. We have empirically investigated the impact of noise to estimators of intrinsic dimension in Section 3.3 and Appendix F of Paper I.

There are two variants of the above definition that have been used frequently in intrinsic dimension estimation papers, and sometimes have been claimed to be equivalent with the above definition (Fukunaga and Olsen, 1971, Camastra, 2003, Carter et al., 2010). In fact they are not, as will be explained below.

The first variant is perhaps most clearly stated by Bennett (Bennett, 1969) who defines intrinsic dimension as "the number of free parameters required in a hypothetical signal generator capable of producing a close approximation to each signal in the collection" (Bennett, 1969, Fukunaga and Olsen, 1971, Levina and Bickel, 2004). In mathematical terms this translates to that each data point Y_i can be written as $Y_i = g(X_i)+\epsilon_i$, where $X_i \in \mathbb{R}^n$ gives the n free parameters and g is the signal generator function. This function is in the general case any continuous function which is sufficiently smooth, but in some cases (Bennett, 1969) a more restrictive set is used. (Levina and Bickel, 2004) allowed any "sufficiently smooth" g, but omitted the noise component ϵ_i.

The second variant of 2.4 is considering the number of parameters to which data can be reduced without losing much information (Carter et al., 2010). This is hard to translate to a precise mathematical formulation, since it depends on how one defines what information is encoded in the data. For example, if we transform the x and y coordinates in Fig. 2.1 to the interval $[0, 2\pi]$ by $t = \arccos y + \pi \cdot I(x < 0)$, where I is the indicator function, we lose the information that data close to 0 or close to 2π in the transformed space are very close to each other in the original data space.

That defining intrinsic dimension by ability to reduce data to a lower-dimensional space is much more restrictive than manifold dimension is shown by the example above. Defining intrinsic dimension by the number of parameters needed to generate the data set is also more restrictive than manifold dimension: A manifold requires around each data point a local chart that parametrizes the neighborhood with n coordinates, but the parametrization does not have to hold for the entire structure. Generating data from n parameters requires a global parametrization.

The manifold-based definition 2.4 uses local properties of the data, whereas the two variants above are based on global properties. Hence any estimators that only use local information, such as local distances, or ability to fit a linear subspaces locally to the data comply better with the manifold-based definition.

2.2 Topological dimension

The notions of manifolds and manifold dimension were developed by Riemann and others from 1850 onwards (Scholz, 1999), but the discoveries of one-to-one correspondences between a line and a square by Cantor in 1877 and the space-filling curve by Peano in 1890 challenged the dimension concept (Crilly, 1999). Was dimension a topological invariant, i.e. constant under homeomorphisms?

This was resolved in the 1910's by Brouwer, who first proved that $\mathbb{R}^n$ is not homeomorphic to $\mathbb{R}^m$ if $n \neq m$ and then constructed a dimension number that agreed with the number of coordinates for Euclidean space and that was topologically invariant. A decade later Urysohn and Menger constructed another invariant which also had these properties. These invariants are called the large and small inductive dimensions respectively (Crilly, 1999). For all separable metric spaces, the large and the small inductive dimensions are equal (Hurewicz and Wallman, 1948, ch. III, prop. 5A).

It is easy to see that manifold dimension agrees with the small inductive dimension, as shown below.

Definition 2.5 (Small inductive dimension)**.** The empty set has dimension -1. A topological space X has dimension n if for any $p \in X$ and closed set $A \subset X, p \notin A$, there is a closed subset Φ with dimension less than n such that $X \backslash \Phi = B \cup C$, where B and C are disjoint open sets and $p \in B$ and $A \subset C$. (Encyclopedia of Mathematics, "Dimension theory")

The induction step can be formulated as that p can be separated from A using a closed set of dimension less than n.

The large inductive dimension is defined in an analogous way, but with a closed set B disjoint from A replacing p.

Proposition 2.1. *A manifold $\mathcal{M}$ with dimension n has small inductive dimension n.*

Proof. Suppose that $p \in \mathcal{M}$ and that A is a closed subset of $\mathcal{M}$ with $p \notin A$. We can find an open set $U \ni p$ so that $x\colon U \to V \subset \mathbb{R}^n$ is a homeomorphism. Since the topology on $\mathcal{M}$ is normal, we can also find disjoint open sets P and Q such that $P \ni p$ and $Q \supset A \cup U^c$. Let $A' = P^c$.

Now $x(A' \cap U)$ is closed in the subset topology of V and V has small inductive dimension n, so we can find a closed set Φ with small inductive dimension less than n and such that $V \backslash \Phi = B \cup C$ with B and C disjoint and open, $x(p) \in B$ and $x(A' \cap U) \subset C$. If $\Psi = x^{-1}(\Phi)$, then

$$\mathcal{M} \backslash \Psi = U \backslash \Psi \cup U^c = U \backslash \Psi \cup Q = x^{-1}(B) \cup (x^{-1}(C) \cup Q),$$

where $x^{-1}(B)$ and $x^{-1}(C) \cup Q$ are disjoint and open, $p \in x^{-1}(B)$ and $A \in x^{-1}(C) \cup Q$. Also, Ψ is clearly closed and has small inductive dimension less than n. □

A third dimension number tracing back to ideas from Lebesgue is the Lebesgue covering dimension (Crilly, 1999). For separable metric spaces this is also equal to the small and large inductive dimensions (Hurewicz and Wallman, 1948, Theorem V 8).

Definition 2.6 (Lebesgue covering dimension)**.** The dimension of a set X is the smallest number n with the property that every open covering has a refinement such that any point appears in at most $n + 1$ of the sets in the refinement. (Encyclopedia of Mathematics, "Dimension theory")

The term topological dimension usually refers to Lebesgue covering dimension (Weisstein). Using the above definitions as a basis for dimension estimators is hard due to their abstractness; a few authors have claimed that their estimators do estimate topological dimension, but in fact they estimate something else such as manifold dimension (Bruske and Sommer, 1998, Kégl, 2002).

The main difference between manifold dimension and topological dimension is that topological dimension is defined for any set in a separable metric space, whereas manifold dimension is only defined for manifolds. From the definition of small inductive dimension and Proposition 2.1 it follows immediately that for a finite disjoint union of manifolds with different manifold dimensions, the topological dimension is the maximal dimension of these.

2.3 Fractal dimension as intrinsic dimension

As set theory and measure theory were developed in the early 20th century, another approach of tackling the problem of defining dimension was proposed by Hausdorff in 1918 (Crilly, 1999, Hausdorff, 1918). Instead of considering topological properties relating to the connectivity structure of sets he considered metric properties, i.e. distances between points, and used a construction by Carathéodory to define a p-dimensional measure that did not depend on the dimension of the embedding space (Hausdorff, 1918). Here p could be non-integer. The Hausdorff dimension of a set was defined as the value of p which either gave a finite measure or for which any $q < p$ would give infinite measure and any $q > p$ would give zero measure (Hausdorff, 1918, Falconer, 1990). The Hausdorff dimension is always larger than or equal to the topological dimension (Hurewicz and Wallman, 1948), but for submanifolds of $\mathbb{R}^d$ they agree (Falconer, 1990). A closely related, but much simpler, definition of dimension is the box counting dimension, which is always larger than or equal to the Hausdorff dimension, but also equal to the manifold dimension for manifolds (Pontrjagin and Schnirelmann, 1932, Falconer, 1990).

The study of certain sets with non-classical geometries, named fractals, was popularized by Mandelbrot in the 1980's (Mandelbrot, 1982, Falconer, 1990). A characteristic of many fractals is that they have non-integer Hausdorff or box-counting dimension; the dimension is an essential feature to investigate (Falconer, 1990).

During the 1980's many fractal sets were studied within the area of dynamical systems. These sets occurred in the form of data generated from so called strange attractors. For these data it were impractical to estimate Hausdorff or box counting dimension because—among other reasons—these dimension concepts are based on the support of the data, when a dimension concept based on the measure generating the data would be preferred. This lead to development of many new measures of fractal dimension such as information dimension and correlation dimension. (Cutler, 1991)

The reader is referred to Cutler (1991) for a comprehensive compilation of different concepts of fractal dimension and how they relate to each other.

2.4 The curse of dimensionality

If you have ever tried to manually optimize something with more than one or two parameters you might have experienced the frustration that is captured in the phrase "the curse of dimensionality" (Bellman, 1961). The number of possible combinations of parameter values is enormous and as the number of parameters increase it soon grows infeasible to do a brute force search for an optimum.

Today the phrase "the curse of dimensionality" is used to describe any kind of phenomenon that becomes problematic when you have a large number of variables or parameters (François et al., 2007), for example when doing numerical integration (Donoho, 2000) or nearest neighbor search (Beyer et al., 1999).

Many "curses of dimensionality" can be traced to what is called the *concentration phenomenon* within mathematics (Pestov, 2008). Pestov (2008) suggested to estimate intrinsic dimension based on how this phenomenon is reflected in data and proposed one estimator of intrinsic dimension along these lines. The estimators in Paper I are also based on this idea, therefore some key elements of concentration of measure are introduced in the next section.

2.4.1 The concentration of measure phenomenon

Concentration of measure is a concept that Vitali Milman started to promote in the beginning of the 1970's (Ledoux, 2005). It been extensively studied within mathematics and applied in many fields such as probability theory, complexity theory and functional analysis. A nice and accessible introduction to the subject viewed from a probabilistic perspective has been written by Talagrand (1996); for a comprehensive review we refer to (Ledoux, 2005).

The amount of concentration of a measurable metric space (X, d, μ) can be quantified by the *concentration function*, defined as

$$\alpha(\epsilon) = \sup\{1 - \mu(A_\epsilon) \colon A \subset X, \mu(A) > \frac{1}{2}\},\ \epsilon > 0,$$

where A_ϵ consists of all points within distance ϵ of A and $\mu(X) = 1$. A space for which $\alpha(\epsilon)$ decreases fast is said to experience concentration of measure.

A family of measurable metric spaces $\{(X_n, d_n, \mu_n)\}_{n=1}^{\infty}$ with the property that $\alpha_n(\mathrm{diam}(X_n)\epsilon) \to 0$ as $n \to \infty$ for any $\epsilon > 0$ is called a Lévy family. There are many Lévy families where n denotes the dimension of the space, for example the n-dimensional unit sphere, or any family of compact Riemannian manifolds with normalized volume measures which also fulfill certain restrictions on their curvatures. This is why the concentration phenomenon is often thought as a phenomenon of high-dimensional spaces.

When (X_n, d_n, μ_n) belongs to a Lévy family and n is high, for any subset $A \subset X_n$ with $\mu(A) > 1/2$, almost all of the remaining measure is concentrated close to the boundary of A. In the case of an n-dimensional sphere, this means that almost all of the measure is concentrated very close to the equator for high n. 'Close' is here defined relative to the sphere's diameter.

An equivalent characterization of the concentration phenomenon can be obtained using Lipschitz functions. Recall that the Lipschitz constant for a real-valued function F is defined as

$$L(F) = \sup_{x \neq y} |F(x) - F(y)|/d(x, y),$$

and when it is finite the function is Lipschitz. Now if m_F is a median of F, it can be shown that (Ledoux, 2005)

$$\mu(\{|F - m_F| \geq r\}) \leq 2\alpha(r/L(F)), \tag{2.1}$$

where α is the concentration function defined as before. Furthermore, if β is any function such that (2.1) holds with β in the place of α, then $\alpha \leq \beta$ (Ledoux, 2005). What the equation (2.1) says is that for spaces with high amount of concentration, i.e. when $\alpha(r)$ decreases fast, any Lipschitz function will be concentrated around its median.

This is in a way similar to the weak law of large numbers, which states that when $X_1, X_2, \ldots, X_n$ are iid random variables with finite expectation, their mean converges in probability to the expected value (Gut, 2009). The concentration of measure phenomenon says that if the product spaces of $X_1, X_2, \ldots, X_n$ equipped with the product metric and the product measure form a Lévy family $\{(X^{\otimes n}, d^{\otimes n}, \mu^{\otimes n})\}_{n=1}^{\infty}$, not only the mean function $F(X_1, X_2, \ldots, X_n) = n^{-1}(X_1 + X_2 + \cdots + X_n)$ converges in probability to its median value, but any Lipschitz function does. And convergence in probability to the median implies that the expected value converges to the median, hence we also get convergence in probability to the expected value.

Donoho (2000) named concentration of measure a "blessing of dimensionality", but it can in many cases be seen as a "curse". Features of data are often measured by Lipschitz functions (Pestov, 2008)—for example projections onto the coordinate axis or distances from a point—and the concentration phenomenon means that these get non-discriminating in high-dimensions. In the example of distances to a given point this means that almost all other points in the space will be almost equidistant to it, making k nearest neighbor methods less meaningful.

It is far from trivial however to show that a sequence of product spaces form a Lévy family, and results such as concentration of distances under certain circumstances have been shown by entirely different means (Beyer et al., 1999). But we conclude this section with a powerful result from the theory of concentration of measure that can explain many curses of dimensionality, for example concentration of distances in a quite general setting.

Theorem 2.1 (Talagrand concentration inequality (Talagrand, 1995, Tao, 2012))**.** *Let $K > 0$, and let $X_1, X_2, \ldots, X_n$ be independent complex random variables with $|X_i| < K$ for $i = 1, \ldots, n$. Let $F\colon \mathbb{C}^n \to \mathbb{R}$ be a 1-Lipschitz convex function. Then for any λ there are constants c, C such that*

$$\Pr(|F(X) - m_F| \geq \lambda K) \leq C \exp(-c\lambda^2)$$

and

$$\Pr(|F(X) - \mathbb{E}(F(X))| \geq \lambda K) \leq C \exp(-c\lambda^2).$$

Chapter 3

Cluster analysis

To group or systematize objects and phenomena is a basic human instinct. It is the basis for interpreting our perceptions and our language is largely constructed from labels we put onto these groups. We put the label "orange" onto things we perceive as similar to 612 nm light. We put the label "walking" onto the act of transporting oneself on foot while always having at least one foot on the ground. Objects have multiple labels and sometimes there is disagreement on whether a certain label should be put on a certain object or not.

For science and science-based professions, grouping and classification is fundamental for organizing and communicating knowledge. Knowing that an entity belongs to a group makes it possible to make predictions about how it will behave under varying circumstances. When a doctor knows that a patient has the diagnosis "diabetes", she will predict that intake of insulin will be instrumental for the patient's well-being. Had the diagnosis been "leukemia", insulin would not have been prescribed.

Sometimes it is not the members of the group themselves that are of interest, but rather a collective property, such as group size, that is the predictor. If a disproportionate amount of blood cells in a blood sample belongs to the group of white blood cells, a doctor might predict that the patient has leukemia.

The problem of defining reasonably well-separated groups in data sets is called clustering. There should be reason to believe that the objects in one group are more similar to each other than to objects of other groups, that the grouping reveals some hidden structure. This problem is significantly harder than the classification problem, where the possible set of labels is known and the task is to choose which one is most appropriate. Grouping plants and animals into species, symptoms into syndromes, events into epochs are all examples of clustering. When a clustering has been made, new objects can be assigned to groups by classification.

Milligan (1996) and Hennig and Liao (2013) have made good overviews of the different steps of a cluster analysis. The first things to consider are what data to base the grouping on and how differences between measurements are valued. What characteristics of the objects are believed to give a partitioning that is relevant for the area of research? Are all variables equally important? How should differences across variables be combined? The next step is to decide on a method for grouping the data—the clustering algorithm. The choice of algorithm will have crucial effect on the result. The final steps in a cluster analysis are evaluation and interpretation of the obtained partition. The evaluation can be used to compare different clustering algorithms against each other, or to select parameters.

Paper II describes a clustering algorithm tailored for grouping measurements of cells in flow cytometry data—cell population identification. Paper III addresses the last (and often under-appreciated) part of the cluster analysis—the evaluation—for the specific case of cell population identification, by investigating tests for unimodality.

3.1 Clustering algorithms

There are hundreds of clustering algorithms in the literature. A 2005 survey included almost 300 references, with algorithms grouped into ten categories (Xu and Wunsch, 2005). The book *Cluster analysis* (Everitt et al., 2011) devotes five out of nine chapters to clustering algorithms, and has around 600 references. One reason for this abundance is that clustering is central to many fields of science, which have developed their own algorithms (Xu and Wunsch, 2005). But there are also many well-motivated reasons for the plethora of algorithms from a data analysis perspective: 1) The algorithms can have different objectives. Is a soft clustering sought after, where each data point is given a probability of belonging to each cluster, or is a hard partitioning the goal? Or a hierarchical tree of clusters? (Jain et al., 1999) 2) The measured variables can be continuous, discrete, binary and/or categorical (Jain et al., 1999, Hennig and Liao, 2013). 3) Different cluster shapes can be acceptable depending on the type of data (Banfield and Raftery, 1993, Jain et al., 1999). 4) High-dimensional data might need special treatment due to the curse of dimensionality (Parsons et al., 2004, Bouveyron and Brunet-

Saumard, 2014). 5) The size and dimension of the data can put restrictions on algorithm complexity (Xu and Wunsch, 2005, Berkhin, 2006).

The main reason that the algorithm presented i Paper II stands out is that it accomplishes parallel clustering of many related data sets. The clusters in the different data sets are expected to be similar, but vary in a number of pre-specified ways.

The algorithm in Paper II belongs to the group of model-based clustering algorithms (Fraley and Raftery, 2002). Properties of this group of algorithms will be further discussed below—to enlighten this discussion a brief overview of groups of clustering algorithms is given first. There are of course many ways to do such a grouping, but here is a rough division into four categories:

Model-based methods: For these algorithms a statistical model is built for the data. This is based on separate models for each cluster, which are combined in a *mixture model*. Parameter inference leads to detection and description of the clusters.

Density-based methods: Density estimation is a non-parametric statistical approach. These methods do not necessarily estimate the density explicitly, but rather use some specific aspects of the density, such as local maxima or low-density regions, to form clusters. Examples include mean-shift clustering (Fukunaga and Hostetler, 1975) and DBSCAN (Ester et al., 1996).

Objective-based methods: These methods define an objective function that describes how well-separated or how tight the clusters are in a given clustering and tries to optimize for that. For example, the classical k-means algorithm tries to minimize the total squared distance to the cluster centers (MacQueen, 1967) and graph-based methods such as min-cut (Papadimitriou and Steiglitz, 1982) and normalized cut (Shi and Malik, 2000), which form the basis for spectral clustering, tries to find the best separation.

Algorithm-based methods: This category includes clustering methods that are most easily described by how the algorithm is designed. Hierarchical clustering (Everitt et al., 2011)—possibly the most widely used clustering methodology—is an example of this. Agglomerative versions of hierarchical clustering start with each data point as a separate cluster and then in each step uses some rule to decide which two clusters to merge next. Divisive versions work the other way, starting with a single cluster. Either the entire tree is given as the clustering result, or some stopping criteria, such as a given number of sought-after clusters, is used to determine when the final clustering is obtained. An entirely different algorithm falling into this category is affinity propagation (Frey and Dueck, 2007).

3.2 Model-based clustering

The central element of a model-based clustering algorithm is the *finite mixture model* (McLachlan and Peel, 2000, Frühwirth-Schnatter, 2006). With this model the probability density describing the data set can be written as

$$f(y) = \sum_{k=1}^{K} \pi_k f_k(y), \tag{3.1}$$

where f_k is the probability density function for mixture component k and π_k is the weight of component k. The mixture density (3.1) is thus a combination of K classes, where the variation within each class k is described by f_k, and π_k describes the relative size of class k. Usually component densities are assumed to come from the same parametric family, i.e. $f_k(y) = g(y; \Theta_k)$, where Θ_k are the parameters. A common choice for g is the normal distribution, then $\Theta_k = (\mu_k, \Sigma_k)$. This important special case is called a *Gaussian mixture model* (GMM).

When no further constraints are taken into account, i.e. when (3.1) is the complete description of the model, and when K is assumed to be known, the parameters Θ_k and π_k can be estimated by maximum likelihood through the *expectation-maximization* (EM) algorithm (Dempster et al., 1977, McLachlan and Peel, 2000). The idea behind the EM algorithm is to add cluster allocation variables x_i for each data point y_i, $i = 1, \ldots, n$ and treat them as missing data. In the expectation step the conditional distribution for each x_i given the current set of parameters $\{\Theta_k\}_{k=1}^{K}$ is computed. This means that one computes the probabilities for y_i to belong to each component, denoted $p_{i1}, p_{i2}, \ldots, p_{iK}$. Based on this, one can write the expected log-likelihood as

$$L(\Theta_1, \ldots, \Theta_K) = \sum_{i=1}^{n} \sum_{k=1}^{K} p_{ik} \log f(y_i, \Theta_k). \tag{3.2}$$

The expected log-likelihood (3.2) is then maximized over $\{\Theta_k\}_{k=1}^{K}$ component-wise in the maximization step.

Dempster et al. (1977) showed that each EM step gives a monotonically increasing likelihood for (3.1) and the algorithm is run until the likelihood has converged to a local maximum. But in many cases—for example for Gaussian mixture models—the likelihood is unbounded, which for the Gaussian case happens when some eigenvalue of one Σ_k approaches zero. McLachlan and Peel (2000) discusses how to detect if the algorithm is trapped in a spurious local maximum on the way to infinite likelihood, so that such solutions can be discarded.

When the number of components K is not known in advance a common solution is to fit the model for multiple values of K and use some model selection criterion to choose among them (Frühwirth-Schnatter, 2006). Another solution is to use *Dirichlet mixtures* (Escobar and West, 1995), which can automatically determine the value of K.

To get a clustering, each data point is either assigned to the component that it is most likely to belong to, or the probabilities of belonging to different components are returned to give a soft clustering. In some cases, for example in Paper II, components are combined to form clusters (Baudry et al., 2010, Hennig, 2010), otherwise each component corresponds to a separate cluster.

The component density function g can have a variety of forms, which gives mixture models much flexibility. Even for Gaussian mixture models there are many possibilities since one can put restrictions on μ_k and Σ_k. Examples include using a common Σ_k for all components, or letting Σ_k be diagonal or even proportional to the identity matrix (Banfield and Raftery, 1993). Other options are using skew probability densities and densities with fat tails (Frühwirth-Schnatter and Pyne, 2010) or densities describing discrete or binary data, possibly combined with continuous densities to describe so called mixed data (Hennig and Liao, 2013).

Another type of flexibility comes from the possibility of incorporating (3.1) into a larger model if we have some additional information about Θ_k, as is done in Paper II through the use of a *Bayesian hierarchical model*, also called a *Bayesian network*.

3.2.1 Bayesian hierarchical models

Bayesian hierarchical models are typically used to model data in multiple related experiments, or in data that has some hierarchical structure (Gelman et al., 2014). They have multiple levels, so that the variables parametrizing the data model, for example μ_k, Σ_k and π_k in a Gaussian mixture model, are themselves modeled by

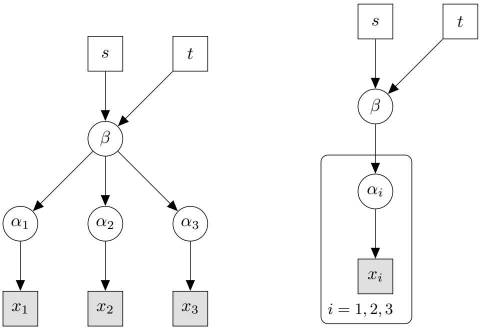


Figure 3.1: Two illustrations of the same directed acyclic graph of a simple Bayesian hierarchical model, where the right one uses plate notation indicating repetitions of the variables inside the plate. Square nodes represent observed data (shaded) or fixed values, and round nodes represent latent variables.

a distribution parametrized by latent variables. The latent variables can be dependent on yet other latent variables or on a prior distribution. Bayesian hierarchical models can be used to describe deep hierarchies and complicated dependence structures. To convey the structure of a Bayesian hierarchical model it is usually illustrated by a directed acyclic graph (DAG), where an arrow from A to B means that when A is given, B is conditionally independent to those parts of the model to which it is only connected through A. An example of a DAG is Fig. 1 in Paper II, a much simpler one is given here for illustration in Fig. 3.1.

Despite the complicated structures of Bayesian hierarchical models there are ways to estimate the posterior distribution, the most common being Markov chain Monte Carlo sampling (Gelman et al., 2014). An excellent introduction to the theory behind Markov chain Monte Carlo is (Geyer, 2011), some of which is recaptured next.

The aim of Monte Carlo methods is to compute expected values using sampling. Suppose that we have observed the data $\mathbf{Y}$ and our model has the latent variables $\mathbf{\Theta}$. If we can estimate $E[g(\mathbf{\Theta})|\mathbf{Y}]$ for any g, we can learn things such as expected values and variance of the posterior for $\mathbf{\Theta}$.

Now if $\{\mathbf{\Theta}^{(m)}\}_{m=1}^{M}$ are independent samples from the distribution of $\mathbf{\Theta}|\mathbf{Y}$, the central limit theorem gives that

$$\frac{1}{M}\sum_{m=1}^{M} g(\mathbf{\Theta}^{(m)}) \approx N(\boldsymbol{\mu}_g, \frac{\mathbf{\Sigma}_g}{M}) \tag{3.3}$$

for large M, where $\boldsymbol{\mu}_g$ is the expected value of $g(\mathbf{\Theta})|\mathbf{Y}$ and $\mathbf{\Sigma}_g$ is the covariance matrix, so with large M we will get close to the expected value.

The idea behind Markov chain Monte Carlo is to generate samples approximating the posterior distribution using a Markov chain, i.e. a sequence of random variables $X_1, X_2, \ldots$, where X_{n+1} only depends on X_n. The conditional distribution $\pi(X_{n+1}|X_n)$ is called the transition probabilities, and together with the initial distribution $\pi(X_1)$ this defines the Markov chain.

Under certain conditions on the transition probabilities and the initial distribution the *Markov chain central limit theorem* (Markov chain CLT) holds, which can replace the central limit theorem when $\{\mathbf{\Theta}^{(m)}\}_{m=1}^{\infty}$ is a realization of a Markov chain $\{\mathbf{\Theta}_m\}_{m=1}^{\infty}$ that has $\mathbf{\Theta}|\mathbf{Y}$ as a stationary distribution (meaning that the transition probabilities preserve the distribution). In the Markov chain CLT, (3.3) also holds, but with

$$\mathbf{\Sigma}_g = \mathrm{Cov}[g(\mathbf{\Theta}_m|\mathbf{Y})] + 2\sum_{k=1}^{\infty} \mathrm{Cov}[g(\mathbf{\Theta}_m|\mathbf{Y}), g(\mathbf{\Theta}_{m+k}|\mathbf{Y})].$$

The Metropolis-Hastings algorithm, Gibbs sampling (which is a special case of Metropolis-Hastings) and the reversible jump method (Green, 1995) are ways to construct transition distributions for which the posterior distribution is reversible and thus stationary.

For countable state spaces it is sufficient that the Markov chain is irreducible (any state can be reached from any other state), aperiodic (each state can be revisited after an arbitrary number of time steps) for the Markov chain CLT to hold for any initial distribution (Billingsley, 1986). But for uncountable state spaces it is not possible to have irreducibility; these conditions can then be replaced with a condition called Harris recurrence (Roberts and Rosenthal, 2006).

If as for the EM algorithm, the component allocation indicators x_i are added as variables, Gibbs sampling can be used to study the posterior distribution (Tanner and Wong, 1987). In Gibbs sampling the conditional distributions of each of the variables given all other variables are used as transition distributions.

3.2.2 Considerations when using model-based clustering

As seen above, model-based clustering algorithms are very flexible. Many types of data can be modeled, and the shapes of the clusters can be controlled. For example, in a Gaussian mixture model that allows covariance matrices that are not proportional to the identity matrix, data with different scales is automatically handled. This stands especially in contrast to objective-based methods, where data have to be scaled carefully and distance metrics chosen so that distances between data points truly reflect how likely it is that they occur in the same cluster (Jain et al., 1999).

Hierarchical clustering and density-based clustering allow different kinds of cluster shapes than model-based clustering. In Gaussian mixture models clusters are approximately ellipsoidal, and though using skew distributions the set of possible shapes can be enlarged, cluster shapes will still be blob-like. Hierarchical clustering and density based clustering can result in banana-shaped or spiral clusters not attainable by mixture models unless you combine a very large number of components. On the other hand, Gaussian mixture models can handle overlapping clusters and clusters with disparate densities well. A good overview of different kinds of cluster shapes that the most used clustering algorithms allow is given in documentation of the Python machine learning package scikit-learn (Pedregosa et al., 2011), the relevant parts being the submodules sklearn.mixture and sklearn.cluster (scikit-learn developers, 2015).

In model-based clustering all assumptions on the data are specified through the model. After clustering, model checking procedures can be used to validate the results (Gelman et al., 2014). In other types of clustering, parameters need to be set that also imply assumptions on the data, but it is not always clear how to relate them to the specific data set under study and how assumptions can be checked.

The downside of assuming a specific model is the well-known fact that all models are simplifications (Box, 1976), and when the model does not fit the data well, unexpected results can be obtained when maximizing the likelihood. It is therefore crucial to validate results from model-based clusterings, as is discussed in Paper II.

3.3 Evaluating clusterings and finding the number of modes

Data sets from different application areas have different kinds of patterns of interest and therefore different cluster types. Thus there does not exist any universally good clustering algorithm (Guyon et al., 2009, Jain et al., 1999). Cluster algorithms or clustering results thus need to be evaluated with the application in mind. Model-based clusterings can be evaluated with model selection criteria such as BIC (Schwarz, 1978) and AIC (Akaike, 1974), but one has to remember that these are based on the fit of the assumed model to the data. In Paper II another approach is taken, the simplest model (i.e. with the least number of components) that describes the data sufficiently well (as determined by some quality criteria) is chosen.

Guyon et al. (2009) argues that clustering algorithms have to be evaluated based on the purpose of the clustering. They mention two kinds of purposes for clustering: data preprocessing and exploratory analysis. Both of these are for example relevant for the cell population identification problem in flow cytometry treated in Paper II and Chapter 3. Cell population identification often acts as a preprocessing step, where the main interest is in the cell population sizes, which are then related to other variables, or used for diagnosis (O'Neill et al., 2013). In other cases, the researcher is interested in finding new populations of interest and describe their properties. In this case the population identification is exploratory.

When the purpose is data preprocessing, Guyon et al. (2009) hold that the end result should be used to evaluate the performance of the clustering algorithm—the better clustering is the one which gives the better end results. But when the purpose is exploratory, measures of statistical significance for clusterings can be valuable to sort out those clusterings which are worthy of the researcher's attention.

There are many different evaluation criteria for clusterings, many of them designed especially to determine the number of clusters; a review from a model-based perspective is given in (Frühwirth-Schnatter, 2006, Sec. 7.1.4). However, it should be noted that the problem of determining the number of cell populations in flow cytometry data is slightly different from the typical interpretation of the problem of determining the number of clusters. In clustering, a partition of the data that has little overlap between clusters is sought for. Criteria for determining the number of clusters often fails with highly overlapping clusters (Frühwirth-

Schnatter, 2006, Tibshirani et al., 2001) as can be found in flow cytometry data.

To evaluate the evidence for multiple cell populations, Paper III proposes to do testing for unimodality. It has been proven that it is impossible to give statistical upper bounds on the number of modes of a probability density (Donoho, 1988). With finite data you cannot exclude the possibility that a small bump actually is due to multimodality of the density, however it can be possible to determine that such a bump could have occurred by chance from a unimodal distribution with high probability. On the other hand Donoho (1988) also proved that it is possible to determine lower bounds for the number of modes; which is natural since if you have two clear bumps in the data the number of modes must be at least two.

Chapter 4

Flow cytometry data

Paper II and III in this thesis concern analysis of flow cytometry data. This chapter gives an introduction to flow cytometry, including a brief overview of its applications and an introduction to the cell population identification problem. The rationale for introducing the automated population identification method presented in Paper II is to handle variation between flow cytometry data samples. To give a deeper understanding of how variation can arise one section is devoted to various sources of technical variation that can occur in flow cytometry analyses.

4.1 What is flow cytometry?

Cytometry refers to quantitative measurements on single cells. Ever since the 17th century, when Robert Hooke looked at cork in a microscope and coined the term *cell* to describe the honey-comb like structures he saw, depicted in Fig. 4.1, the technology for looking at cells and using the properties one sees to learn something about the organism they are taken from, have advanced steadily. But since the 1930's (Shapiro, 2005, ch. 1) in parallel to development of microscopes and data analysis of images from these, flow cytometry has developed as a technology to make measurements on single cells without actually looking at them. The basic idea is that the cells pass by a measurement apparatus one by one in a fluid stream. Today the most common way to make these measurements is by attaching fluorescent probes to cells and using lasers to detect them.

One of the advantages of flow cytometry as compared to microscopy is that one can get a very high throughput—tens of thousands of cells can be processed

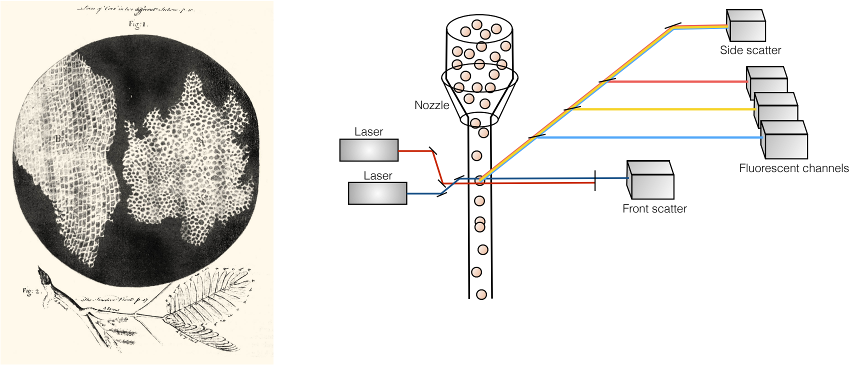


Figure 4.1: Left: Cells in cork sample, drawn by Robert Hooke. From *Micrographia*, 1665. Right: A flow cytometer. Cells enter the fluid stream through a nozzle. The light from one or multiple lasers of different wavelengths hits each cell, exciting fluorophores attached to the cells. For each fluorophore the emitted light is measured by a fluorescent channel. Front scatter is the amount of non-direct light scattered by the cell in the direction of the laser beam; side scatter is the amount of light scattered by the cell in directions orthogonal to the laser beam.

each second (BD Biosciences, 2013). Another advantage is that fluorescence flow cytometry technology can be used for sorting cells into groups, which in itself has a wide array of applications. However, one loses all structural information, on the tissue level as well as on the subcellular level. Therefore it is most common to use flow cytometry analysis on non-structured tissues, i.e. fluid—most notably blood (Shapiro, 2005, ch. 1).

Other quantitative single-cell measurement technologies include single-cell PCR of targeted transcripts and single-cell RNA or DNA sequencing. However, the throughput of these are many orders of magnitude lower than for flow cytometry (Bendall and Nolan, 2012).

4.2 Fluorescence flow cytometry

Flow cytometry is based on attaching probes to specific cell structures and measuring how many probes that attach to each cell. One can use different specificities of binding of the probes, but today it is most common to use antibodies binding to a specific protein on the cell surface as probes (Shapiro, 2005, Ch. 7). For example, T cells (a kind of immune cell), have thousands of T cell receptor (TCR) and co-receptor proteins on their cell surface. This receptor is used for signaling to the cell when it should activate (Murphy et al., 2012). Using an antibody that attaches to the T cell co-receptor one can determine if a cell is a T cell or not.

To detect the antibodies that are used as probes, they have to be labeled with fluorescent markers. This is typically either small organic molecules that react with amines on the antibodies, or certain fluorescent proteins derived from algae, phycobiliproteins (Invitrogen, 2010, Ch. 1), (Shapiro, 2005, Ch. 7). Recently, the repertoire of labels have been expanded by so called quantum dots made from semiconductors (Perfetto et al., 2004, Invitrogen, 2010), and certain conducting organic polymers (Chattopadhyay et al., 2012), which enables more sensitive detection of probes due to brighter fluorescence as well as the use of more fluorescent markers in parallel due to a better use of the spectrum.

A schematic drawing of a flow cytometer is shown in Fig. 4.1. Lasers are used to excite the fluorophores, and the amount of each fluorescent marker is measured by filtering the emitted and scattered light from the cell so that mainly light from this fluorophore is obtained. The filtered light is then amplified by a photomultiplier tube before reaching a detector. In addition to fluorescent markers the amount of scattered light in the direction of the laser beam (front scatter) and in directions orthogonal to the laser beam (side scatter) are measured. The front scatter roughly increases with cell size and the side scatter increases with internal complexity and granularity of the cell. Front and side scatter measurements are often used to distinguish major cell types. (Shapiro, 2005, Ch. 1)

When fluorescence flow cytometers were first developed in the late 1960's, only one fluorescent marker could be measured in addition to front and side scatter. The technology gradually developed so that in the mid 1980's four colors could be measured simultaneously by the most advanced instruments. However, few laboratories saw the need to use instruments measuring more than two markers. It was the AIDS epidemic that triggered the more widespread use of three- and four-color instruments, since certain cell subsets relevant for studying AIDS could only be detected using three or four fluorescent markers. (De Rosa et al., 2003)

The number of possible markers that could be measured simultaneously increased steadily during the 1990's and 2000's and today's state of the art instruments can measure up to 18 colors (Perfetto et al., 2004, BD Biosciences, 2013). The major obstacle when adding more markers is overlap of the emission spectra of the fluorophores (Perfetto et al., 2004). The most common way to account for this overlap is by first using calibration beads to measure how large spill into other channel each fluorophore gives. Then when doing measurements on cells the measured data is multiplied with the inverse of the spill matrix, which amounts to subtracting an estimate of the spillover (Bagwell and Adams, 1993). This process, called compensation, is problematic if there is high spectral overlap or if the cells have high autofluorescence.

A more efficient way to use the information in the emitted light is to measure the entire emission spectrum and use spectral deconvolution algorithms to estimate the abundance of each marker (Nolan and Condello, 2013). This approach is called *spectral flow cytometry*. The first commercially available spectral flow cytometers were released in 2013; they can measure up to 15 fluorescent markers simultaneously (Sony, 2013).

4.3 Mass cytometry

Instead of using fluorescent labels to tag probes it is possible to use metal particles with differential mass. To detect these, mass spectrometers are used. This technology is called *mass cytometry* or CyToF (Bandura et al., 2009). The main advantage of mass cytometry is that more markers can be studied in parallel, today in a single experiment more than 40 parameters can be measured for each cell (Fluidigm, 2015). In this thesis we will only consider data generated using fluorescence flow cytometry, but the methods applies in principle also to mass cytometry data; the data characteristics are quite similar. However, one must beware the curse of dimensionality.

4.4 Sources of technical variation

There are many possible sources of technical variation in flow cytometry data acquisition that can affect downstream data analysis. The major ones are listed below.

Sample handling: Biological samples, e.g. blood or bone marrow, are treated in various ways before running it through the flow cytometer. Factors that can affect the flow cytometry data is whether they are frozen or preserved in other ways, for example to enable delayed analysis; the time between sample acquisition and analysis; if subsets of cells are extracted, for example through centrifugation or by lysis (dissolving) of certain cell types such as red blood cells; how staining with the markers is performed, i.e. how the probes are attached to the cells; and how solid tissue is cut up and homogenized (Maecker et al., 2010, Kalina et al., 2012, Hasan et al., 2015).

Panel design: A panel is a predefined set of probe targets combined with a specific set of fluorescent labels. Panels have to be carefully designed to minimize spectral overlap and to get optimal signals. As the number of colors increase this becomes increasingly important (De Rosa et al., 2003). For example a protein that has low abundance requires a bright label to be detected. Certain dyes are also non-stable (Hasan et al., 2015).

Probe selection: The probes that attach to the cells are *monoclonal antibodies*, i.e. antibodies produced from cells which have been cloned (Murphy et al., 2012). However, clones from different manufacturers, and even different clones from the same manufacturer, have different protein binding properties (Kalina et al., 2012, Hasan et al., 2015). Certain antibodies are also incompatible with each other (Chattopadhyay and Roederer, 2012).

Calibration and compensation settings: The settings for the laser, photomultiplier tubes and detectors have to be adjusted regularly in order to detect as much signal as possible; this is called calibration. To minimize variation this should be done according to standard operating procedures, preferably with as much automation as possible, for example using calibration beads to which fluorophores are attached (Maecker et al., 2010, Kalina et al., 2012, Hasan et al., 2015). However, automated procedures for calibrating instruments might be different between instruments from different manufacturers (Maecker et al., 2010).

Using many colors simultaneously reduces the need to use multiple panels for the same biological sample, thus minimizing sample handling variation (Maecker et al., 2010). On the other hand it makes the data more affected by the spillover matrix, and even when this is correctly estimated non-intuitive artifacts can occur (De Rosa et al., 2003). To estimate the amount of spillover in other fluorescent channels, compensation beads to which the fluorescence labeled probes attach can be used, or a stained reference sample (Kalina et al., 2012, Hasan et al., 2015). Compensation settings also have to be updated regularly.

Instrument performace: Regular quality control needs to be performed on the flow cytometer; when the quality criteria is not met the instrument might need to be cleaned or serviced (Kalina et al., 2012).

The cell population identification pipeline presented in Paper II is designed to handle variation in location and shape of the cell populations in the measurement space, so that samples despite this can be analyzed simultaneously. A variation in location means that the measured mean fluorescence intensities are changed, and a variation in shape means that the fluorescence pattern is changed differently for different cells. Such changes are most likely to be due to sample handling, probe selection or calibration and compensation settings.

4.5 Applications

Studying properties of single cells enables the understanding of heterogeneity of seemingly homogeneous groups (Bendall and Nolan, 2012). One such group is lymphocytes—a kind of white blood cell that looks quite boring in a microscope: it is fairly round with a large nucleus and not much structure in its cytoplasm. Lymphocytes look like inactive cells and it was long before any of their functions were discovered (Murphy et al., 2012). Today they are the most studied cell type within immunology, since they are responsible for the adaptive immune system.

It is primarily the need for better analyses of the lymphocyte cell population that has driven the technical development of flow cytometers (Perfetto et al., 2004), and they have grown into a formidable tool for this. Studying the adaptive immune system through lymphocyte subpopulations is key to understanding infectious diseases such as HIV (Betts et al., 2006), hepatitis C (Evans et al., 2007) and tuberculosis (Fuhrmann et al., 2008); autoimmune diseases such as allergies (Cheung et al., 2008), diabetes (Tang et al., 2006) and multiple sclerosis (Du et al., 2009); for developing vaccines (Kool et al., 2008) and for monitoring organ transplants (Maguire et al., 2014, Jaye et al., 2012).

Understanding and monitoring the adaptive immune system is also crucial in cancer immunotherapy, where the patient’s own immune system is triggered to attack the tumor. But even though the basic idea is simple, the immune system–tumor interactions are very complex and thus hard to control (Gupta et al., 2016).

Cancer immunotherapy is an area that has been studied for decades, but recently it has received much increased interest due to discoveries of successful therapies for melanoma and prostate cancer (Mellman et al., 2011, Pardoll, 2012).

Another area where flow cytometry has major importance is studying, diagnosing and monitoring leukemias and lymphomas (Vardiman et al., 2009, Swerdlow et al., 2016, Van Dongen et al., 2012). For evaluation of *minimal residual disease* after treatment, which is an important prognostic factor, the high throughput of flow cytometry has proven especially useful (van Dongen et al., 2015).

The applications of flow cytometry have also spread widely beyond the study of lymphocytes. Some recent examples are research on stem cells (Mich et al., 2014, Kumar et al., 2015), discovery of new taxa of marine microbes (Petersen et al., 2012) and analysis of extracellular vesicles with potential use as biomarkers, e.g. for thrombosis (van der Vlist et al., 2012, Mooberry and Key, 2016)

4.6 Data properties

A flow cytometry data set has as many dimensions as the number of fluorescent markers plus three: front scatter, side scatter and time. The time variable can be used to detect problems during a run, for example due to variations in fluid dynamics, and can be used for data cleaning (Fletez-Brant et al., 2016), but it is otherwise typically not considered. For each of the the other variables there are actually three measurements: the area, the height and the width of the measured pulse. The height and the width measurements are usually only used for preprocessing though. The variables for fluorescent markers are typically transformed using a log-like transform (Finak et al., 2010), for example logicle (Parks et al., 2006), that also can handle negative values (negative values can arise due to compensation). This facilitates viewing the data in scatter plots and histograms.

Each data point is called an *event* and corresponds usually to a single cell, but could also be multiple cells clogged together, a doublet, or debris (Shapiro, 2005, ch. 1). A flow cytometry sample has typically from 10^4 to 10^6 events. In the course of a study, everything from a few samples to thousands of flow cytometry samples can be analyzed (Hasan et al., 2015).

4.7 Cell population identification

Hasan et al. (2015) describes typical data analysis procedures for flow cytometry data: First doublets are removed by considering the area versus the height, or the width, of the front and/or side scatter pulse, then the group of cells of interest is singled out, e.g. T-cells, by using markers for this cell type. These two first steps can also be integrated. After this the cells of interest are partitioned into subpopulations using some of the remaining markers. Finally, for the obtained cell populations, cell population size and mean fluorescence intensities (MFI) are reported. In each analysis step one or two-dimensional scatter plots or histograms are used to draw a *gate*, marking which cells to use for further analysis.

The process described above is called *gating* or *manual gating*. It has been recognized that the gating process induces much non-biological variation in cell population sizes, since it is a subjective process to choose the gating strategy, i.e. in which order markers should be considered, and to draw the gates (Maecker et al., 2005, Welters et al., 2012).

To remove the subjectivity and the gating variation, many methods for *automated gating* have been developed. Such methods can be devised for specific tasks such as finding small cell populations (Naim et al., 2014) or finding populations that differ most between two samples (Bruggner et al., 2014). Through the FlowCAP project, `http://flowcap.flowsite.org`, (Aghaeepour et al., 2013), different algorithms for flow cytometry data analysis have been compared through a number of challenges. These challenges give a good overview of the types of analysis that are aimed at for automated gating.

The first FlowCAP round, in 2010, was aimed at reproducing manual gating. Four data sets gated by manual operators were used as ground truth. There were four different challenges, where different amounts of information were given, such as the true number of populations or some cells in each population. In FlowCAP II, the challenges were to do sample classification, for example to classify blood samples taken from patients with acute myeloid leukemia and normal blood samples. FlowCAP III contained four challenges on different topics: to identify a rare cell population in new samples given samples where it had been identified, to predict survival time for a patient based on a flow cytometry sample and training data from other patients, to classify flow cytometry samples into one of two categories given training data, and to reproduce a manual gating given the gating strategy. The last round up to date, FlowCAP IV, had one challenge where time until progression to AIDS should be predicted given two flow cytometry samples from each patient and a population of training samples.

For many of the FlowCAP challenges it is not necessary to actually identify the relevant cell populations, for example in FlowCAP II many machine learning methods using other types of features than cell population sizes performed well (Aghaeepour et al., 2013). However, typically it is not the classification itself that is of scientific interest, but finding features or cell populations that can help classification, i.e. that are biomarkers for a specific condition. In FlowCAP IV participants were asked not only to provide a prediction of time until progression, but also to describe the features that were most important for the prediction.

Chapter 5

Conclusions

We are in an era of massive data collection, and many times data are collected without knowing exactly what one is looking for. Donoho (2000) notes that: "[...] it has become much cheaper to gather data than to worry much about what data to gather". The sheer amount of data means that to learn something from it, it has to be structured and summarized. The methods presented in this thesis can support various aspects of this process.

The dimension estimator presented in Paper I has the aim to give an understanding of data complexity. In an exploratory data analysis pipeline, it will probably serve its most important role as an initial diagnosis tool, guiding other data analysis methods. The tests for unimodality investigated in Paper III will probably have their largest roles in the other end of the pipeline—in the quality control of acquired results.

BayesFlow, the clustering algorithm presented in Paper II integrates many central elements of analysis of flow cytometry data: fitting a mixture model, merging components to form clusters and doing quality checks and supportive visualizations of the result. Cell population identification in flow cytometry data provides an exciting opportunity for further development of clustering algorithms and cluster evaluation methods. Despite much work on automated methods, manual gating is still the state of the art—for good reasons. BayesFlow shows an important direction to be explored further, where information can be shared between samples during the clustering process, and prior knowledge can be integrated.

Guyon et al. (2009) hold that when clustering is used as a preprocessing step—as can be argued is the case for most cases of cell population identification—it should be evaluated based on the usefulness of the result. The straightforward way to define this is as how well the cluster features can be used to predict or diagnose medical conditions of interest. But communicating results to a wider audience, including researchers in the field and the person giving out diagnoses, requires understanding and understanding requires something more than a machine-learning feature. When a certain cell population, as defined by a cluster of cells expressing a certain combination of markers, is seen to be the predictor for a medical condition, this can guide further research. This is why finding biologically plausible cell populations is so important. When automated cell population identification methods are developed this always has to be kept in mind.

Bibliography

N. Aghaeepour, G. Finak, The FlowCAP Consortium, The DREAM Consortium, H. Hoos, T. R. Mosmann, R. Brinkman, R. Gottardo, and R. H. Scheuermann. Critical assessment of automated flow cytometry data analysis techniques. *Nature Methods*, 10(3):228–238, 2013.

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

C. B. Bagwell and E. G. Adams. Fluorescence spectral overlap compensation for any number of flow cytometry parameters. *Annals of the New York Academy of Sciences*, 677(1):167–184, 1993.

D. R. Bandura, V. I. Baranov, O. I. Ornatsky, A. Antonov, R. Kinach, X. Lou, S. Pavlov, S. Vorobiev, J. E. Dick, and S. D. Tanner. Mass cytometry: technique for real time single cell multitarget immunoassay based on inductively coupled plasma time-of-flight mass spectrometry. *Analytical Chemistry*, 81(16):6813–6822, 2009.

J. D. Banfield and A. E. Raftery. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, pages 803–821, 1993.

J.-P. Baudry, A. E. Raftery, G. Celeux, K. Lo, and R. Gottardo. Combining mixture components for clustering. *Journal of Computational and Graphical Statistics*, 19(2), 2010.

BD Biosciences. *BD LSRFortessa X-20*, 2013.

R. Bellman. *Adaptive control processes: a guided tour*. Princeton university press, Princeton, New Jersey, 1961.

S. C. Bendall and G. P. Nolan. From single cells to deep phenotypes in cancer. *Nature Biotechnology*, 30(7):639–647, 2012.

R. Bennett. The intrinsic dimensionality of signal collections. *IEEE Transactions on Information Theory*, 15(5):517–525, 1969.

P. Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer, 2006.

M. R. Betts, M. C. Nason, S. M. West, S. C. De Rosa, S. A. Migueles, J. Abraham, M. M. Lederman, J. M. Benito, P. A. Goepfert, M. Connors, et al. HIV nonprogressors preferentially maintain highly functional HIV-specific CD8+ T cells. *Blood*, 107(12):4781–4789, 2006.

K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *International conference on database theory*, number 1540 in Lecture Notes in Computer Science, pages 217–235. Springer, 1999.

P. Billingsley. *Probability and measure*. John Wiley & Sons, New York, 2nd edition, 1986.

C. Bouveyron and C. Brunet-Saumard. Model-based clustering of high-dimensional data: A review. *Computational Statistics & Data Analysis*, 71:52–78, 2014.

G. E. Box. Science and statistics. *Journal of the American Statistical Association*, 71(356):791–799, 1976.

R. V. Bruggner, B. Bodenmiller, D. L. Dill, R. J. Tibshirani, and G. P. Nolan. Automated identification of stratifying signatures in cellular subpopulations. *Proceedings of the National Academy of Sciences*, 111(26):E2770–E2777, 2014.

J. Bruske and G. Sommer. Intrinsic dimensionality estimation with optimally topology preserving maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):572–575, 1998.

F. Camastra. Data dimensionality estimation methods: A survey. *Pattern Recognition*, 36(12):2945–2954, 2003.

K. M. Carter, R. Raich, and A. O. Hero III. On local intrinsic dimension estimation and its applications. *IEEE Trans. Signal Process.*, 58(2):650–663, 2010.

P. K. Chattopadhyay and M. Roederer. Cytometry: today's technology and tomorrow's horizons. *Methods*, 57(3):251–258, 2012.

P. K. Chattopadhyay, B. Gaylord, A. Palmer, N. Jiang, M. A. Raven, G. Lewis, M. A. Reuter, N.-u. Rahman, D. A. Price, M. R. Betts, et al. Brilliant violet fluorophores: a new class of ultrabright fluorescent compounds for immunofluorescence experiments. *Cytometry Part A*, 81(6):456–466, 2012.

P. F. Cheung, C. K. Wong, and C. W. Lam. Molecular mechanisms of cytokine and chemokine release from eosinophils activated by IL-17A, IL-17F, and IL-23: implication for Th17 lymphocytes-mediated allergic inflammation. *The Journal of Immunology*, 180(8):5625–5635, 2008.

J. A. Costa and A. O. Hero III. Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Trans. Signal Process.*, 52(8): 2210–2221, 2004.

T. Crilly. The emergence of topological dimension theory. In I. M. James, editor, *History of Topology*. Elsevier, Amsterdam, 1999.

C. D. Cutler. Some results on the behavior and estimation of the fractal dimensions of distributions on attractors. *Journal of Statistical Physics*, 62(3-4): 651–708, 1991.

S. C. De Rosa, J. M. Brenchley, and M. Roederer. Beyond six colors: a new era in flow cytometry. *Nature Medicine*, 9(1):112–117, 2003.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

D. L. Donoho. One-sided inference about functionals of a density. *The Annals of Statistics*, 16(4):1390–1420, 1988.

D. L. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. AMS Challenges Lecture, 2000.

C. Du, C. Liu, J. Kang, G. Zhao, Z. Ye, S. Huang, Z. Li, Z. Wu, and G. Pei. MicroRNA miR-326 regulates TH-17 differentiation and is associated with the pathogenesis of multiple sclerosis. *Nature Immunology*, 10(12):1252–1259, 2009.

Encyclopedia of Mathematics. Dimension theory. P.S. Aleksandrov (originator) `http://www.encyclopediaofmath.org/index.php?title=Dimension_theory&oldid=14136`.

M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowlege Discovery and Data Mining*, pages 226–231, 1996.

M. J. Evans, T. von Hahn, D. M. Tscherne, A. J. Syder, M. Panis, B. Wölk, T. Hatziioannou, J. A. McKeating, P. D. Bieniasz, and C. M. Rice. Claudin-1 is a hepatitis C virus co-receptor required for a late step in entry. *Nature*, 446 (7137):801–805, 2007.

B. S. Everitt, D. Stahl, M. Leese, and S. Landau. *Cluster Analysis*. Wiley Series in Probability and Statistics. John Wiley & Sons, Chichester, UK, 2011.

K. Falconer. *Fractal Geometry—Mathematical foundations and applications*. John Wiley & Sons, Chichester, 1990.

G. Finak, J.-M. Perez, A. Weng, and R. Gottardo. Optimizing transformations for automated, high throughput analysis of flow cytometry data. *BMC Bioinformatics*, 11(1):546, 2010.

K. Fletez-Brant, J. Špidlen, R. R. Brinkman, M. Roederer, and P. K. Chattopadhyay. flowClean: Automated identification and removal of fluorescence anomalies in flow cytometry data. *Cytometry Part A*, 89(5):461–471, 2016.

Fluidigm. *Helios — A CyToF system*, 2015.

C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

D. François et al. *High-dimensional data analysis: optimal metrics and feature selection*. PhD thesis, Université catholique de Louvain, Louvain-la-Neuve, Belgium, 2007.

B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

S. Frühwirth-Schnatter. *Finite mixture and Markov switching models*. Springer Series in Statistics. Springer-Verlag, New York, 2006.

S. Frühwirth-Schnatter and S. Pyne. Bayesian inference for finite mixtures of univariate and multivariate skew-normal and skew-t distributions. *Biostatistics*, 11(2):317–336, 2010.

S. Fuhrmann, M. Streitz, and F. Kern. How flow cytometry is changing the study of TB immunology and clinical diagnosis. *Cytometry Part A*, 73(11):1100–1106, 2008.

K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.

K. Fukunaga and D. R. Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Trans. Comput.*, C-20:176–183, Feb. 1971.

A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC Boca Raton, FL, USA, 3 edition, 2014.

C. J. Geyer. Introduction to markov chain monte carlo. In S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, editors, *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

S. K. Gupta, T. Jaitly, U. Schmitz, G. Schuler, O. Wolkenhauer, and J. Vera. Personalized cancer immunotherapy using systems medicine approaches. *Briefings in Bioinformatics*, 17(3):453–467, 2016.

A. Gut. *An Intermediate Course in Probability*. Springer, New York, 2nd edition, 2009.

I. Guyon, U. Von Luxburg, and R. C. Williamson. Clustering: Science or art. In *NIPS 2009 workshop on clustering theory*, pages 1–11, 2009.

M. Hasan et al. Semi-automated and standardized cytometric procedures for multi-panel and multi-parametric whole blood immunophenotyping. *Clinical Immunology*, 157(2):261–276, 2015.

F. Hausdorff. Dimension und äußeres maß. *Mathematische Annalen*, 79(1-2): 157–179, 1918.

M. Hein and J.-Y. Audibert. Intrinsic dimensionality estimation of submanifolds in R^d. In *Proc. 22nd Int. Conf. Machine Learning*, pages 289–296. ACM, 2005. data generator available at `http://www.ml.uni-saarland.de/code/IntDim/IntDim.htm`.

C. Hennig. Methods for merging gaussian mixture components. *Advances in data analysis and classification*, 4(1):3–34, 2010.

C. Hennig and T. F. Liao. How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 62(3):309–369, 2013.

W. Hurewicz and H. Wallman. *Dimension Theory*. Princeton university press, 1948.

Invitrogen. *Molecular Probes Handbook*. Thermo Fisher Scientific, 11th edition, 2010.

A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

D. L. Jaye, R. A. Bray, H. M. Gebel, W. A. Harris, and E. K. Waller. Translational applications of flow cytometry in clinical practice. *The Journal of Immunology*, 188(10):4715–4719, 2012.

T. Kalina, J. Flores-Montero, V. H. J. Van Der Velden, M. Martin-Ayuso, S. Böttcher, M. Ritgen, J. Almeida, L. Lhermitte, V. Asnafi, A. Mendonca, et al. EuroFlow standardization of flow cytometer instrument settings and immunophenotyping protocols. *Leukemia*, 26(9):1986–2010, 2012.

B. Kégl. Intrinsic dimension estimation using packing numbers. In *Proc. Advances in Neural Information Processing Systems 15*, pages 681–688, 2002.

M. Kool, T. Soullié, M. van Nimwegen, M. A. Willart, F. Muskens, S. Jung, H. C. Hoogsteden, H. Hammad, and B. N. Lambrecht. Alum adjuvant boosts adaptive immunity by inducing uric acid and activating inflammatory dendritic cells. *The Journal of experimental medicine*, 205(4):869–882, 2008.

N. Kumar, J. Richter, J. Cutts, K. T. Bush, C. Trujillo, S. K. Nigam, T. Gaasterland, D. Brafman, and K. Willert. Generation of an expandable intermediate mesoderm restricted progenitor cell line from human pluripotent stem cells. *eLife*, 4:e08413, 2015.

M. Ledoux. *The concentration of measure phenomenon*. Number 89 in Mathematical Surveys and Monographs. American Mathematical Society, Providence, Rhode Island, 2005.

E. Levina and P. J. Bickel. Maximum likelihood estimation of intrinsic dimension. In *Proc. Advances in Neural Information Processing Systems 17*, pages 777–784, 2004.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297, Oakland, CA, USA., 1967.

H. T. Maecker, A. Rinfret, P. D'Souza, J. Darden, E. Roig, C. Landry, P. Hayes, J. Birungi, O. Anzala, M. Garcia, et al. Standardization of cytokine flow cytometry assays. *BMC Immunology*, 6(13), 2005.

H. T. Maecker, J. P. McCoy Jr, F. H. I. Consortium, et al. A model for harmonizing flow cytometry in clinical trials. *Nature Immunology*, 11(11):975–978, 2010.

O. Maguire, J. D. Tario Jr, T. C. Shanahan, P. K. Wallace, and H. Minderman. Flow cytometry and solid organ transplantation: a perfect match. *Immunological investigations*, 43(8):756–774, 2014.

B. Mandelbrot. *The fractal geometry of nature*. W. H. Freeman and Company, New York, 1982.

G. McLachlan and D. Peel. *Finite mixture models*. Wiley series in probability and statistics. John Wiley & Sons, New York, 2000.

I. Mellman, G. Coukos, and G. Dranoff. Cancer immunotherapy comes of age. *Nature*, 480(7378):480–489, 2011.

J. K. Mich, R. A. Signer, D. Nakada, A. Pineda, R. J. Burgess, T. Y. Vue, J. E. Johnson, and S. J. Morrison. Prospective identification of functionally distinct stem cells and neurosphere-initiating cells in adult mouse forebrain. *eLife*, 3: e02669, 2014.

G. W. Milligan. Clustering validation: results and implications for applied analyses. In G. D. P Arabie, L J Hubert, editor, *Clustering and Classification*, pages 341–376. World Scientific, 1996.

M. J. Mooberry and N. S. Key. Microparticle analysis in disorders of hemostasis and thrombosis. *Cytometry Part A*, 89(2):111–122, 2016.

K. Murphy, P. Travers, and M. Walport. *Janeway's immunobiology*. Garland Science, London, 2012.

I. Naim, S. Datta, J. Rebhahn, J. S. Cavenaugh, T. R. Mosmann, and G. Sharma. Swift—scalable clustering for automated identification of rare cell populations in large, high-dimensional flow cytometry datasets, part 1: Algorithm design. *Cytometry Part A*, 85(5):408–421, 2014.

J. P. Nolan and D. Condello. Spectral flow cytometry. *Current Protocols in Cytometry*, pages 1–27, 2013.

K. O'Neill, N. Aghaeepour, J. Špidlen, and R. Brinkman. Flow cytometry bioinformatics. *PLoS Computational Biology*, 9(12):e1003365, 2013.

C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Dover Publications, Inc., Mineola, New York, 1982.

D. M. Pardoll. Immunology beats cancer: a blueprint for successful translation. *Nature immunology*, 13(12):1129, 2012.

D. R. Parks, M. Roederer, and W. A. Moore. A new "logicle" display method avoids deceptive effects of logarithmic scaling for low signals and compensated data. *Cytometry Part A*, 69(6):541–551, 2006.

L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explorations Newsletter*, 6(1):90–105, 2004.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

S. P. Perfetto, P. K. Chattopadhyay, and M. Roederer. Seventeen-colour flow cytometry: unravelling the immune system. *Nature Reviews Immunology*, 4(8): 648–655, 2004.

V. Pestov. An axiomatic approach to intrinsic dimension of a dataset. *Neur. Netw.*, 21(2-3):204–213, 2008.

T. W. Petersen, C. B. Harrison, D. N. Horner, and G. van den Engh. Flow cytometric characterization of marine microbes. *Methods*, 57(3):350–358, 2012.

L. Pontrjagin and L. Schnirelmann. Sur une propriété métrique de la dimension. *Annals of Mathematics*, pages 156–162, 1932.

M. Raginsky and S. Lazebnik. Estimation of intrinsic dimensionality using high-rate vector quantization. In *Advances in Neural Information Processing Systems*, pages 1105–1112, 2005.

G. O. Roberts and J. S. Rosenthal. Harris recurrence of metropolis-within-gibbs and trans-dimensional markov chains. *The Annals of Applied Probability*, 16(4): 2123–2139, 2006.

E. Scholz. The concept of manifold, 1850-1950. In *History of Topology*. Elsevier, Amsterdam, 1999.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2): 461–464, 1978.

scikit-learn developers. *scikit-learn user guide*, release 0.17 edition, November 2015.

H. M. Shapiro. *Practical flow cytometry*. John Wiley & Sons, Hoboken, New Jersey, 2005.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

Sony. *SP6800 Spectral Analyzer*, 2013.

S. H. Swerdlow, E. Campo, S. A. Pileri, N. L. Harris, H. Stein, R. Siebert, R. Advani, M. Ghielmini, G. A. Salles, A. D. Zelenetz, et al. The 2016 revision of the world health organization classification of lymphoid neoplasms. *Blood*, 127 (20):2375–2390, 2016.

M. Talagrand. Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathématiques de l'Institut des Hautes Etudes Scientifiques*, 81(1):73–205, 1995.

M. Talagrand. A new look at independence. *The Annals of Probability*, 24(1): 1–34, 1996.

Q. Tang, J. Y. Adams, A. J. Tooley, M. Bi, B. T. Fife, P. Serra, P. Santamaria, R. M. Locksley, M. F. Krummel, and J. A. Bluestone. Visualizing regulatory t cell control of autoimmune responses in nonobese diabetic mice. *Nature Immunology*, 7(1):83–92, 2006.

M. A. Tanner and W. H. Wong. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540, 1987.

T. Tao. *Topics in random matrix theory*, volume 132 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, Rhode Island, 2012.

R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

E. J. van der Vlist, E. N. Nolte, W. Stoorvogel, G. J. Arkesteijn, and M. H. Wauben. Fluorescent labeling of nano-sized vesicles released by cells and subsequent quantitative and qualitative analysis by high-resolution flow cytometry. *Nature protocols*, 7(7):1311–1326, 2012.

J. J. van Dongen, V. H. van der Velden, M. Brüggemann, and A. Orfao. Minimal residual disease diagnostics in acute lymphoblastic leukemia: need for sensitive, fast, and standardized technologies. *Blood*, 125(26):3996–4009, 2015.

J. J. M. Van Dongen, L. Lhermitte, S. Böttcher, J. Almeida, V. Van der Velden, J. Flores-Montero, A. Rawstron, V. Asnafi, Q. Lecrevisse, P. Lucio, et al. Euroflow antibody panels for standardized n-dimensional flow cytometric immunophenotyping of normal, reactive and malignant leukocytes. *Leukemia*, 26(9):1908–1975, 2012.

J. W. Vardiman, J. Thiele, D. A. Arber, R. D. Brunning, M. J. Borowitz, A. Porwit, N. L. Harris, M. M. Le Beau, E. Hellström-Lindberg, A. Tefferi, et al. The 2008 revision of the World Health Organization (WHO) classification of myeloid neoplasms and acute leukemia: rationale and important changes. *Blood*, 114(5):937–951, 2009.

E. W. Weisstein. Dimension. From MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/Dimension.html`.

M. J. Welters, C. Gouttefangeas, T. H. Ramwadhdoebe, A. Letsch, C. H. Ottensmeier, C. M. Britten, and S. H. van der Burg. Harmonization of the intracellular cytokine staining assay. *Cancer Immunology, Immunotherapy*, 61 (7):967–978, 2012.

R. Xu and D. Wunsch. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.

Papers

PAPER I

Published in *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2015.

Low bias local intrinsic dimension estimation from expected simplex skewness

Kerstin Johnsson, Charlotte Soneson and Magnus Fontes

Abstract

In exploratory high-dimensional data analysis, local intrinsic dimension estimation can sometimes be used in order to discriminate between data sets sampled from different low-dimensional structures. Global intrinsic dimension estimators can in many cases be adapted to local estimation, but this leads to problems with high negative bias or high variance. We introduce a method that exploits the curse/blessing of dimensionality and produces local intrinsic dimension estimators that have very low bias, even in cases where the intrinsic dimension is higher than the number of data points, in combination with relatively low variance. We show that our estimators have a very good ability to classify local data sets by their dimension compared to other local intrinsic dimension estimators; furthermore we provide examples showing the usefulness of local intrinsic dimension estimation in general and our method in particular for stratification of real data sets.

1 Introduction

High-dimensional data sets are now collected at an unprecedented rate in many areas of science and engineering. Unsupervised learning methods with the purpose to find structures in such data are thus naturally an area of much interest. When non-linear functional relations exist between variables it is natural to use a manifold corrupted with noise as a model for the data. The manifold model can be used in various ways: in manifold learning it is used in order to find a lower-dimensional representation for the data (Lee and Verleysen, 2007), intrinsic dimension estimation methods use it to find a measure of the local complexity of the data, and in computational topology topological features such as holes are used to gain qualitative information about the data (Carlsson, 2009).

Local intrinsic dimension estimation can also be used if multiple manifolds build the underlying structure of the data. Many interesting applications of local dimension estimation have recently been put forward (Carter et al., 2010, Haro et al., 2008), where clustering based on local dimensionality (stratification) plays a central role. They include image segmentation, image classification, and network anomaly detection. Using a small number of data points for dimension estimation is necessary for local intrinsic dimension estimation, but it means that we get a large bias and/or high variance for most estimators, especially if the data has a high intrinsic dimension.

In this paper we present new estimators based on angular information, which are almost unbiased and also have a relatively low variance. Consequently, in comparison to other estimators they have a better or similar ability to distinguish between data sets of different dimensions and can estimate dimension more accurately. The new estimators can in particular accurately estimate dimensions higher than the number of sample points in a local data set. The estimators are computationally fast, simple to implement and have no parameters to tune.

The method that we use to derive these estimators we call Expected Simplex Skewness (ESS); it exploits the concentration phenomenon, the mathematical equivalent of one of the curses (or blessings) of dimensionality (Donoho, 2000). Using the concentration phenomenon for dimension estimation opens up many possibilities, of which some have recently been explored (Pestov, 2008, Rozza et al., 2012, Ceruti et al., 2012). One implication of the concentration phenomenon is that Lipschitz functions from the unit sphere S^n or $S^n \times \cdots \times S^n$ to $\mathbb{R}^d$ concentrate around their medians to a higher and higher degree when the dimension n increases. In high dimensions this means that Lipschitz functions are essentially constant. One example is the projection of S^n to any coordinate axis: As $n \to \infty$ the pushforward measure of the uniform measure on S^n approaches the Gaussian measure that has probability density function $\sqrt{n/2\pi}e^{-nx^2/2}$ (Gromov, 1999). This implies two things that we use in the ESS method: 1) The average length of vectors going from the origin to S^n projected onto any coordinate axis will approach zero as $n \to \infty$ and 2) pairs of vectors going from the origin to S^n will tend to be close to orthogonal as $n \to \infty$.

The asymptotic distribution of angles as $n \to \infty$ was used in (Ceruti et al., 2012) for construction of the DANCo dimension estimator, but with the ESS method instead of using an asymptotic distribution we use distributions that are valid for finite dimensions n.

It might seem paradoxical that we can estimate intrinsic dimensions that are higher than the number of data points, since N data points always can be embedded into an $(N-1)$-dimensional subspace. However, even though the data points can be embedded into an $(N-1)$-dimensional subspace it can sometimes be seen that it is very unlikely that they are generated from an $(N-1)$-dimensional distribution. In our case this happens when vectors going from the centroid to the data points in the local data set are on average closer to orthogonal than would be expected if they were generated from an $(N-1)$-dimensional distribution.

1.1 Related Work

We restrict this overview to methods that can be used for local dimension estimation, which means that among other things we exclude methods based on manifold learning. We also exclude methods that require a very large number of data points, such as those who estimate box-counting (capacity) dimension. For a survey which includes many of the dimension estimation methods which are omitted here we refer to (Rozza et al., 2012).

Almost all local dimension estimation methods fall into three categories: 1) methods that use eigenvalues of the local covariance matrix (local PCA), 2) methods that use the local distances between points, and 3) methods that use geometric objects constructed from data points, such as vectors with data points as endpoints and Voronoi cells. The second category is the largest and includes many methods for fractal dimension estimation. One recent method (Ceruti et al., 2012) integrates information from both distances and angles between vectors, but on the large this is an area that is yet to be explored. A more general approach is taken by Pestov (Pestov, 2008) who defines dimension from the concentration phenomenon in a way that applies also to binary data and relational data. He presents an estimator of dimension that fits into this framework and another such estimator is presented in (Chávez et al., 2001).

Fukunaga and Olsen were the first to propose to use the local covariance matrix/local PCA to determine intrinsic dimension (Fukunaga and Olsen, 1971).

The main issue for these methods is how to determine how many of the eigenvalues that are significant; the dimension estimate is the number of significant eigenvalues. This means that if a hyperplane exactly fit the local data set the dimension estimate can at most be the dimension of that hyperplane, hence if the intrinsic dimension is higher than the number of data points we cannot get an accurate estimate.

The original Fukunaga-Olsen method says that any eigenvalue that is at least α (= 5 %) of the largest eigenvalue is significant. A recent method by Fan et al. (Fan et al., 2010) uses both gaps in the eigenvalues and total variance to determine which eigenvalues that are significant. Based on the probabilistic formulation of PCA (Tipping and Bishop, 1999) there are many recent Bayesian methods for determining the number of significant singular values in PCA, which could also be used locally. However these methods usually assume latent Gaussian data, which is not applicable to local data, which we assume is approximately uniform. The method by Hoff (Hoff, 2007) does not assume this, but it is very computationally demanding.

There are numerous ways of using the distribution of local distances that have been exploited for dimension estimation. The first was the maximum-likelihood estimator by Hill (Hill, 1975, Harte, 2001); a similar method much used for (correlation) dimension estimation of fractals was developed later by Takens (Takens, 1985). Hill's and Takens' estimators were also studied in (Levina and Bickel, 2004). Another common method for correlation dimension estimation of fractals is the method by Grassberger and Procaccia (Grassberger and Procaccia, 1983). Other methods that use the distribution of local distances include (Carter et al., 2010, Rozza et al., 2012, Pettis et al., 1979, Judd, 1994, Camastra and Vinciarelli, 2002, Costa and Hero III, 2004, 2006). All these methods are based on a model where the data is uniformly distributed on a manifold that is well approximated by its tangent plane on the scale where we measure local distances. Some methods have also included Gaussian noise in the model (Haro et al., 2008, Smith, 1992, Schouten et al., 1994, Diks, 1996, Oltmans and Verheijen, 1997), and Schreiber (Schreiber, 1997) has reviewed how Gaussian noise influences the distribution of local distances.

A problem when using the distribution of local distances is boundary effects, which distort the distribution and lead to a negative bias especially for high dimensions where any manifold with a boundary has almost all of its volume concentrated close to the boundary.

Methods that use other geometric properties than distances are in general less tested than methods from the two first categories. For some of them it has not been shown that they work when the intrinsic dimension is higher than 3. An early method by Trunk (Trunk, 1976) considered for each data point the angle between the vector going from the data point to its kth nearest neighbor and the hyperplane spanned by the vectors to the $k-1$ nearest neighbors. The dimension estimate was the minimum value of k for which the average angle exceeded a threshold. However the threshold is dependent on both the number of points and k, and no algorithm for determining the thresholds in general was described. Recent geometric methods for dimension estimation include analyzing shapes of cells in Voronoi tessellations (Dey et al., 2002), measuring distances to linear subspaces of best fit in certain neighborhoods (Giesen and Wagner, 2004) and analyzing shapes of simplices constructed with data points as vertices (Cheng and Chiu, 2009). The method using simplices (Cheng and Chiu, 2009) showed promising results for some high-dimensional data sets, but the dependence on input parameters is unclear and the method is very computationally demanding.

2 Methods

Local dimension estimation means that we first define neighborhoods in the data set and then estimate the dimension of these. A natural way to define a neighborhood is to start at a data point and take all other data points within a certain distance, or take the k nearest neighbors as the neighborhood. In this paper we use the k nearest neighbors in Euclidean distance so that we control the number of data points in the neighborhood. If the data set follows a model of an n-manifold with noise—under the conditions that the manifold is sufficiently smooth, the sampling is dense enough and the noise is not too large—the data set will locally be well approximated by a uniform distribution on the tangent plane to the manifold. This means that the neighborhood, or local data set as we also call it, will approximately follow a uniform distribution on an n-dimensional hyperball.

The Expected Simplex Skewness (ESS) method takes local data sets and gives intrinsic dimension estimates for them. The method has two closely related versions: ESSa and ESSb. We will start with describing ESSa since it has given ESS its name.

In ESSa we begin by choosing a target dimension d, which is arbitrary except that it has to be lower than the intrinsic dimension that we want to estimate.

Choosing $d = 1$ is our default option but we want to stress that it is a particular case of a more general method. For the local data set we consider simplices with one vertex in the centroid and the other $d+1$ vertices in data points. We use such simplices to estimate the expected value of what we call the simplex skewness measure. The estimation is done by a weighted mean and the result is compared to the true expected simplex skewness for uniformly distributed n-dimensional data for various n.

The simplex skewness measure is the volume of a simplex constructed as above divided by the volume it would have had if the edges incident to the centroid vertex were orthogonal. A very skew simplex has low simplex skewness measure.

As noted in the Introduction, pairs of vectors going from the origin to S^n will more often be close to orthogonal when n increases, and it follows directly that this will also be the case for the edges in the simplex incident to the centroid vertex if the local data set is uniformly distributed in an n-dimensional hyperball. Hence, the expected simplex skewness measure will approach 1 as the dimension increases. Noise and curvature will cause the expected skewness measure to deviate from this, but as we see in experiments the deviation is not very large.

If the target dimension is $d = 1$ the simplex skewness measure is $\sin\theta$, where θ is the angle between the two edges incident to the centroid vertex. The expected simplex skewness measure for uniformly distributed data on the unit ball B^n is then

$$s_n^{(1)} = \frac{1}{V(n)} \int_{B^n} |\sin\theta(x)|\, dV(x) \ , \tag{1}$$

where $V(n) = \pi^{n/2}/\Gamma(n/2+1)$ is the volume of the unit n-ball and $\theta(x)$ is the angle between the line through the origin and $x \in B^n$ and a fixed coordinate axis.

With target dimension $d > 1$ the expected simplex skewness measure for uniformly distributed data on B^n is

$$s_n^{(d)} = \frac{1}{V(n)^d} \int_{B^n \times \cdots \times B^n} |u \wedge \frac{v_1}{|v_1|} \wedge \cdots \wedge \frac{v_d}{|v_d|}| \\ dV(v_1)dV(v_2)\ldots dV(v_d), \tag{2}$$

where u is the unit vector along a reference coordinate axis and $\wedge$ denotes the exterior product.

For the ESSb estimator we consider only target dimension 1 here, but the method can be generalized to higher target dimensions. As in ESSa we assume that we have a local data set and we consider pairs of vectors going from the centroid to data points. The quantity that we use for dimension estimation is the expected length of the projection of one vector onto the other after the vectors are scaled to unit length. As in ESSa this is estimated by a weighted mean and compared to the expected values for uniformly distributed data. If the angle between two normalized vectors is θ the length of the projection is $\cos\theta$, thus the expected projection for uniformly distributed data in B^n is

$$c_n = \frac{1}{V(n)} \int_{B^n} |\cos\theta(x)| \, dV(x), \tag{3}$$

where $V(n)$ and $\theta(x)$ are defined as above. c_n approaches zero as $n \to \infty$, as was discussed in the Introduction. More precisely, $s_n^{(d)}$ increases monotonically with n and it approaches 1 as $n \to \infty$ and c_n decreases monotonically with n and it approaches 0 as $n \to \infty$. In Appendix A in the Supplemental Material, available online at `http://doi.ieeecomputersociety.org/10.1109/TPAMIxxxxxxx`, we derive closed expressions for $s_n^{(d)}$ and c_n; the resulting expressions are

$$s_n^{(d)} = \frac{\Gamma\left(\frac{n}{2}\right)^{d+1}}{\Gamma\left(\frac{n+1}{2}\right)^{d} \Gamma\left(\frac{n-d}{2}\right)} \quad \text{and} \quad c_n = \frac{2V(n-1)}{A(n-1)}, \tag{4}$$

where $A(n)$ is the area of the unit n-sphere, i.e. $A(n) = (n+1)V(n+1)$.

In order to reduce the impact of noise and the exact position of the centroid we want points far from the centroid to have higher weights than points that are close to the centroid. Hence we give to each simplex or vector pair a weight that equals the product of the lengths of the edges incident to the centroid or the product of the vectors' lengths respectively. After centering the local data set so that the centroid coincides with the origin, the simplex skewness is computed from the wedge product while the projection length is computed from the dot product. For a local data set X our estimators of the expected simplex skewness with $d = 1$, $s^{(1)}$, and the expected projection length, c, respectively are thus

$$\hat{s}^{(1)} = \frac{\sum_{x,y\in X} |\bar{x} \wedge \bar{y}|}{\sum_{x,y\in X} |\bar{x}||\bar{y}|} \quad \text{and} \quad \hat{c} = \frac{\sum_{x,y\in X} |(\bar{x}, \bar{y})|}{\sum_{x,y\in X} |\bar{x}||\bar{y}|}, \tag{5}$$

where $\bar{x}$ is the vector from the centroid of the local data set to the data point x. The estimator of $s^{(d)}$ for $d > 1$ is an obvious generalization of (5). If the local data set is big, and we have more than 5,000 simplices or vector pairs, we sample 5,000 of them to use for the estimation in order to save computation time, and we find that the precision is still good enough.

Now we only need to define functions that take $\hat{s}^{(d)}$ or $\hat{c}$ and return an estimate of the dimension. Given $\hat{s}^{(d)} = s_n^{(d)}$ or $\hat{c} = c_n$ respectively we know from (4) that the dimension estimate should be n. For $s_n^{(d)} < \hat{s}^{(d)} < s_{n+1}^{(d)}$ or $c_n < \hat{c} < c_{n+1}$ we use linear interpolation to determine the dimension estimate, i.e.

$$\hat{n} = n + \frac{\hat{s}^{(d)} - s_n^{(d)}}{s_{n+1}^{(d)} - s_n^{(d)}} \quad \text{or} \quad \hat{n} = n + \frac{\hat{c} - c_n}{c_{n+1} - c_n}.$$

Given a local data set with N data points and extrinsic dimension D, the time complexity of ESSa is $O(N^{d+1}D(d+1)^2)$ since we get $O(N^{d+1})$ simplices for which we need to compute the volume through $\mathrm{vol}(S) = \sqrt{\det(SS^T)}/d!$, where S is a $(d+1) \times D$-matrix containing the coordinates of the vertices away from the origin. The time complexity of ESSb is the same with $d = 1$.

3 Experiments

We first use five groups of carefully designed synthetic data sets to assess the ability of different estimators to distinguish between data sets with different intrinsic dimensions under different circumstances and to assess the precision of the estimators. Then we use a wide range of manifolds previously studied in the context of global dimension estimation (Rozza et al., 2012) to assess the versatility of the estimators. We also evaluate stratification of a synthetic data set consisting of data sampled along two manifolds with different dimensions. Finally we consider three real data sets, where we see the potential of using intrinsic dimension estimates for stratification. Further experiments on real and synthetic data are presented in Appendices F–G in the Supplemental Material, available online.

3.1 Evaluated Estimators

We have compared the ESS estimator to two estimators based on local PCA: Fukunaga-Olsen (F-O) (Fukunaga and Olsen, 1971) and Fan (Fan et al., 2010); three estimators based on distributions of distances: Hill (Hill, 1975, Harte, 2001) (also known as MLE (Levina and Bickel, 2004)), TP (Haro et al., 2008) and kNN (Carter et al., 2010); and one estimator based on both distance and angular information: DANCo (Ceruti et al., 2012).[1] If there is no (estimated) noise the TP method is the same as the Hill method. We have modified the TP and kNN estimators slightly: For the TP estimator we utilize the noncentral χ distribution instead of the Gaussian approximation of it used in (Haro et al., 2008) to describe the translation of distances due to noise. For the kNN estimator we use regular bootstrapping instead of block bootstrapping since block bootstrapping requires a spatial ordering which is hard to achieve and has little meaning in high dimensions.

The TP estimator needs as input data an estimate of the noise, both of its dimension and of its variance. How we compute this is described in Appendix D in the Supplemental Material, available online. Two parameters, $\hat{n}$ and $\tilde{n}$, which are different rough pre-estimates of dimension are used for the computation.

For the F-O and Fan methods we used the parameters in (Fukunaga and Olsen, 1971) and (Fan et al., 2010) respectively. We experimented with a large number of different parameters for the Hill, TP, kNN and DANCo estimators for each neighborhood size. Parameters were chosen to maximize the number of correct dimension estimates after calibration as described in Section 3.2 for uniformly distributed local data sets of dimensions 3–9 and the same size as the neighborhood. For a neighborhood size of 50, this objective is what is reported in Table 1, column U_{2-10}. Appendix D in the Supplemental Material, available online, describes the selection process in more detail. The ESS estimators do not require any tuning of parameters. We use two different values of d (1 and 2) for ESSa just in order to show that both work well.

3.2 Data Analysis

In order to characterize the estimators we do dimension estimation on groups of data sets, with the data sets in each group generated from the same underlying model. The probability density of the dimension estimates for each group is estimated with a kernel density estimator. The estimated densities are used for visualization and densities estimated from ideal local data sets—data sets sampled from uniform distributions on hyperballs—are used for calibrating estimators with large or moderate bias. This means that we can compare classification

[1] `R` implementations of the evaluated estimators, including the ESS estimators, are available at `http://www.maths.lu.se/staff/kerstin-johnsson/research/manifold-dimension-estimation/`.

performance of estimators with different biases by the frequency of correct calibrated dimension estimates.

What the calibration does is to set the thresholds, i.e. decision boundaries, for classification of dimension so that the probability of classifying the ideal local data sets correctly is maximized. This means that if we get η as a dimension estimate of a data set using an estimator that needs calibration (F-O with $D \gtrsim N$, Hill, TP, kNN, Fan), after calibration the data set is classified as having dimension n, where n is the dimension of the group of ideal local data sets which gave the highest kernel density estimate at η. If the estimator does not need calibration (ESS[2], F-O with $D \ll N$, DANCo), n is taken as the one of the dimensions considered that is closest to η. The TP estimator is the same as the Hill estimator for the ideal data sets since they are noise-free, hence the same thresholds are used. Examples of kernel density estimates of ideal local data sets, and the resulting thresholds, are shown in Fig. S4 in the Supplemental Material, available online.

We use the kernel density estimator from the *ks* package for R with a Gaussian kernel (Duong, 2007). We use the plug-in bandwidth selection provided with the package, except that when we evaluate integer-valued dimension estimators we set the bandwidth to 0.1 if the discrete structure is dominant and smoothing gives bad classification performance.

3.3 Dimension Estimation of Synthetic Data Sets

We use four sets of groups of synthetic data sets for testing classification performance and one set of groups of data sets for testing precision. For the sets of groups used to test classification performance, each group consists of 100 data sets with a given dimension and each data set has 50 data points. The groups in the first two sets, U_{2-10} and U_{20-100}, consist of data sets sampled with uniform density in hyperballs of dimensions $2, 3, \ldots, 10$ and $20, 30, \ldots, 100$ respectively. These data sets are what we call uniformly distributed local data sets and they are the ideal data sets for local dimension estimation. This is because when we cut out a piece of an n-manifold with a cut-off radius r it will look approximately like $B^n(r)$ if the manifold is flat enough or if r is small enough.

The third set of data set groups, N_{2-10}, represents the ideal case corrupted with noise. We can think of the data sets as generated the following way: Assume

[2] Even though there is a bias for high dimensions in the ESS estimators, we think that since it is relatively small it is better to disregard it so that we can avoid thresholds that depend on specific realizations of ideal data sets.

Table 1: Classification performance P based on dimension estimation of local data sets with 50 points.

	U_{2-10}	U_{20-100}	N_{2-10}	S_{2-10}
ESSa, $d = 1$	0.97	0.89	0.58	0.91
ESSa, $d = 2$	0.98	0.89	0.45	0.92
ESSb	0.96	0.86	0.73	0.89
Hill, $k = 35$	0.94	0.89	0.58	0.88
TP (exact noise var.) $k = 21$	0.89	0.81	0.78[b]	—
TP (estimated noise var.) $k = 20$	—	—	0.59[b]	0.88[c]
kNN $k = 9, N = 10, \gamma = 2$ $p = \{25, 28, 31, \ldots, 49\}$	0.52[a]	0.63[a]	0.43	0.51
F-O, $\alpha = 0.05$	1[a]	0.86[a]	1	0.79
Fan, $\alpha = 10, \beta = 0.8$	0.38[a]	0.76[a]	0.27	0.37
DANCo, $k = 35$	0.83[a]	0.75	0.34	0.76

[a] Bandwidth set to 0.1 in order to keep the discrete structure.

[b] $\tilde{n} = 6, \hat{n} = 10$.

[c] $\tilde{n} = \hat{n} = D - 1$, where D is the extrinsic dimension of data.

that we can sample an infinite number of data points with uniform density on a hyperplane through the origin. To these data points 15-dimensional Gaussian noise with distribution $\mathcal{N}(0, 0.05^2 I)$ is added, and those data points that end up within distance 1 from the origin are candidates for the local data set. Among the candidates we randomly choose 50 data points for the local data set. If there are not enough candidates we sample more data points uniformly from the hyperplane and repeat the procedure. The data generation is described in more detail in (Johnsson, 2013). Also for these data sets the intrinsic dimension, i.e. the dimension of the hyperplane, ranges from 2 to 10.

The fourth set of groups, S_{2-10}, is designed to test how well the estimators can handle curvature. Local data sets consisting of 50 points each are cut out from a data set with 1000 points sampled with a uniform density on the surface of 2–10-dimensional hyperspheres. The procedure for finding a local data set is to choose a point p at random from the whole data set and using its nearest neighbors as the local data set. The point p is not put into the local data set since we want to assume that the local data is drawn from a uniform distribution on a hyperball, and p will be in the center of the hyperball, not drawn randomly.

When computing classification performance we exclude groups of data sets with highest and lowest intrinsic dimension (2 and 10 or 20 and 100), since any arbitrarily low or high dimension estimate will be classified as these dimensions. Hence the classification performance P is the proportion of correct dimension estimates for data sets of dimension 3–9 or 30–90, after calibration when needed (see Section 3.2). In Table 1 P is shown for each estimator for the four sets of groups of data sets. The ESS methods perform very well in comparison to the others, especially for U_{20-100} and S_{2-10}. The methods that give integer estimates (kNN, F-O, Fan, DANCo) have a disadvantage in classification especially when the discrete structure is dominant. This can be seen in Fig. S4 in the Supplemental Material, available online.

The fifth set of groups of synthetic data sets are uniformly distributed local data, as in U_{2-10} and U_{20-100}, but with dimensions 5 and 70 and with varying number of points. To increase interpretability the parameters for the estimators were the same for all data sets, they were optimized for 30 data points. Fig. 1 shows the average bias of the ESSa, Hill, F-O and DANCo estimators on these groups of data sets. The ESSa, F-O and DANCo estimators are approximately unbiased for the 5-dimensional data sets, but for the 70-dimensional data set only ESSa and DANCo are approximately unbiased. The DANCo estimator includes

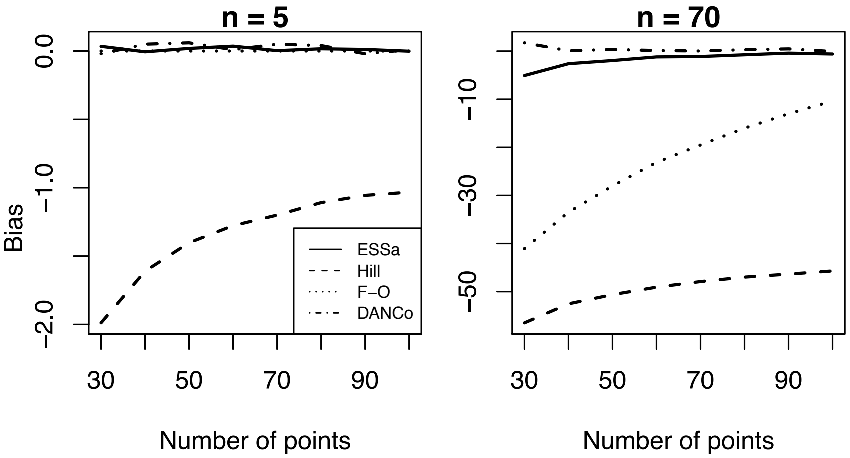


Figure 1: Biases of the ESSa ($d = 1$), Hill ($k = 24$), F-O ($\alpha = 0.05$) and DANCo ($k = 26$) estimators for local data sets with uniform distribution and with intrinsic dimensions $n = 5$ and $n = 70$. The number of number of data points vary between 30 and 100.

calibration with dimension estimates on uniformly distributed local data sets as a final step and has therefore no bias.

To study the estimators' versatilities, a wide range of data sets generated from smooth densities on manifolds was used. We refer to (Rozza et al., 2012) and (Hein and Audibert, 2005) for detailed descriptions of the 17 datasets ($\mathcal{M}_1$–$\mathcal{M}_{15}$, with three versions of $\mathcal{M}_{10}$: $\mathcal{M}_{10a}$, $\mathcal{M}_{10b}$, $\mathcal{M}_{10b}$). The intrinsic dimensions of the data ranged from 1 to 24 and the ratio between intrinsic and extrinsic dimension ranged from 1:1 to 1:13.

To do local dimension estimation, 100 neighborhoods for each of three different neighborhood sizes (30, 50, 100) were sampled from each of the 17 data sets, all having 2000 points. For a given estimator, data set and neighborhood size, relative bias (i.e. $|\text{bias}|/n$) and coefficient of variation (i.e. $|\text{sdev}|/n$) was computed. The median and maximum relative bias and coefficient of variation for the estimators are shown in Table 2. The ESS and DANCo estimators have similar performance, DANCo has smaller maximal relative bias. The Hill, kNN and F-O estimators have small CV, but their biases are large. Further results from this experiment are shown in Fig. S5 in the Supplemental Material, available online.

It takes a few minutes on a standard desktop computer to compute the dimen-

Table 2: Relative bias and coefficient of variation for local dimension estimation with neighborhood sizes 30, 50 and 100 of 17 different manifolds studied in (Rozza et al., 2012).

	Relative bias Median/Max	CV Median/Max
ESSa, d = 1	0.05/0.65	0.07/0.17
Hill, $k_{30} = 24, k_{50} = 35, k_{100} = 63$	0.44/0.70	0.03/0.06
kNN, $N = 10, \gamma = 2$ $k_{30} = 15, p_{30} = \{16, 19, \ldots, 28\}$ $k_{50} = 9, p_{50} = \{25, 28, \ldots, 49\}$ $k_{100} = 6, p_{100} = \{50, 53, \ldots, 98\}$	0.33/0.71	0.04/0.22
F-O	0.09/1.00	0.05/0.40
Fan	0.44/6.00	0.37/6.03
DANCo, $k_{30} = 26, k_{50} = 35, k_{100} = 35$	0.05/0.40	0.06/0.17

sion estimates of the 900 local data sets in U_{20-100} with the ESSa estimator for $d = 1$ or the ESSb estimator. The ESSa estimator for $d = 2$ needs approximately 15 minutes. The Hill, F-O and Fan estimators need only a few seconds, whereas the DANCo estimator needs approximately twice the time of the ESSa ($d = 1$) estimator if its calibration data are reused. However, if the number of data points is very large, the time needed to find the local data sets will be dominant.

3.4 Stratification of Synthetic Data

Stratification means discriminating between points that lie on different manifolds. Stratification based on local dimension estimates has been extensively studied in (Haro et al., 2008), where a mixture model and regularization were used together with local dimension estimates from the TP method. The mixture model and regularization could be used together with any local dimension estimation method, and it should benefit from more accurate and well separated estimates. In this section we study how fit the output of different estimators of local dimension is for applying stratification methods. Note that the neighborhoods now might contain points sampled from two different manifolds.

Table 3: Area under ROC-curve (AUC-ROC) and root mean squared error (RMSE) for 5-ball inside 8-sphere.

	AUC-ROC	$\mathrm{RMSE}_{\mathrm{ball}}/\mathrm{RMSE}_{\mathrm{sphere}}$
ESSa, $d = 1$	0.92	0.68/1.06
Hill, $k = 24$	0.72	1.91/4.55
TP, $k = 12, \tilde{n} = 8, \hat{n} = 8$	0.79	1.90/3.71
kNN, $k = 15, N = 10, \gamma = 2$ $p = \{16, 19, 22, \ldots, 48\}$	0.29	2.02/4.69
F-O, $\alpha = 0.05$	0.84	1.94/0.83
Fan, $\alpha = 10, \beta = 0.8$	0.67	1.73/4.00
DANCo, $k = 23$	0.70	0.77/1.41

We consider a data set consisting of 200 points on a 5-dimensional hyperball centered at the origin, together with 200 points on an 8-dimensional hypersphere also centered at the origin. We estimate local dimension at each of the 400 data points using neighborhoods with 30 points. To evaluate how well separated the dimension estimates from the two manifolds are we use classifiers that use a threshold for intrinsic dimension to separate the two manifolds. Points with a lower dimension than the threshold are supposed to be on the 5-ball and points with a higher dimension than the threshold are supposed to be on the 8-sphere. We use the receiver operating characteristic (ROC) for such classifiers based on the intrinsic dimension estimates of each estimator. The area under the ROC-curve is used as a measure of how well separated the two groups are; this and the mean square errors of the dimension estimates are shown in Table 3. Kernel density estimates of for the estimators are shown in Fig. S6 in the Supplemental Material, available online.

ESSa outperforms all the other estimators both in terms of area under ROC-curve and root mean square error.

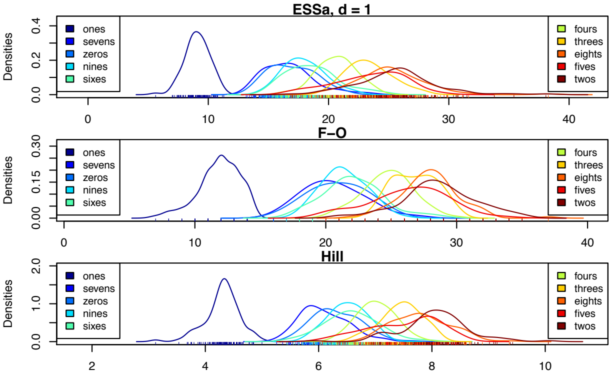


Figure 2: MNIST digits 0-9. For each digit we used 500 images of it as the data set and constructed 100 neighborhoods with 50 points each where the dimension was estimated. The extrinsic dimension is 784.

3.5 Dimension Estimation of Real Data

The first real data set we consider is the 28×28 pixel images of handwritten digits in the MNIST database[3]. This data set has for example been studied in (Haro et al., 2008), where stratification was used to distinguish handwritten ones from handwritten twos. From the training data set of 60,000 images we use 500 images of each digit 0-9. This is to ensure that we get the same number of images for each digit and to make the neighborhoods cover a relatively large portion of the whole data set, so that the manifold structure can be seen in neighborhoods and not only noise. We estimate the dimension of the neighborhood of each image in our sample, the neighborhood consists of the 50 nearest neighbors of the image.

In Fig. 2 we show dimension estimates from the ESSa, the Hill and the F-O method for all the ten digits and we see that it is not surprising that ones and twos are easily distinguished from each other. The overall pattern of the estimates is very similar for the different estimators, although the values of the estimates differ. For the digit 1, the dimension estimates of the ESSa and F-O estimators are similar to what have previously been estimated with global intrinsic dimension estimators (Rozza et al., 2012). We see that even with noise and high extrinsic dimension, the ESSa estimator behaves in a reasonable way.

[3] `http://yann.lecun.com/exdb/mnist/`

3.6 Stratification of Real Data Sets

Finally we use dimension estimation for two stratification applications: image segmentation and gene expression classification. It has been proposed that intrinsic dimensionality can be used for segmentation based on textures since it is a measure of the complexity of the data set (Carter et al., 2010). Following (Carter et al., 2010) we divide an image of a panda (Fig. 3, top left) into patches with 12 × 12 pixels and consider each patch as a 144-dimensional data point. Then we make local dimension estimation using the 30 nearest neighbors.

The results are scaled to $[0, 255]$ and printed in Fig. 3. We see that patches that are on edges in the image have very low dimensionality, but more importantly the patches in most of the panda fur, both the black and the white parts, have similar dimensionality, whereas the background in general has lower dimensionality. This means that intrinsic dimensionality actually can be used to find parts with similar texture. The dark blob in the bottom right is a region where identical (black) patches occur in the original image. For dimension estimators that cannot handle a situation with identical patches we remove duplicates. In Fig. 3 we see that all the tested dimension estimators yielded similar results. The ESSa estimator handles also this situation in a reasonable way. The Fan estimator, which compensates for noise, seems to have the best results though.

Our final example is classification of gene expression data. The classification task that is considered here—distinguishing between tumor and normal samples—is not a very hard classification task, however it is a task where we for some estimators can distinguish groups solely based on dimension estimation. In general we do not intend to use dimension estimation on its own for classification, but as an input to a clustering or classification method (Carter et al., 2010, Haro et al., 2008).

The data set that we consider is a gene expression data set from ovarian cancer samples collected through the TCGA project (Bell et al., 2011). The data set consists of 570 tumor samples and 8 samples collected from normal tissue next to the tumor. The extrinsic dimension of the data is 12981. We perform local dimension estimation with neighborhoods of size 30, 50 and 100 around each sample. For all estimators and neighborhood sizes the adjacent normal samples have in general lower estimated intrinsic dimension of their neighborhoods. However, in most cases the two groups are not clearly distinguishable. We use the receiver operating characteristic (ROC) for classification through thresholding at different dimensions to quantify how well the groups are distinguished for each estimator

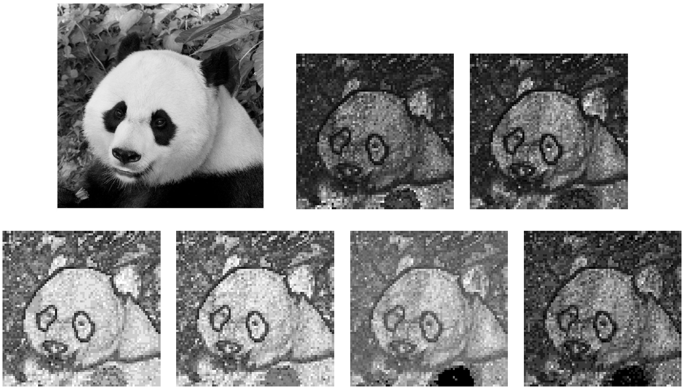


Figure 3: Local dimension estimation of patches in a picture of a panda, using neighborhoods with 30 points. Light patches means high dimension, dark patches low dimension. Top row: Original image, ESSa, DANCo; bottom row: Fan, F-O, Hill and kNN estimators. Estimator parameters as in Table 3. Image source: `http://newsdesk.si.edu/sites/default/files/photos/nzp_Mei_Xiang.jpg`

and neighborhood size. The results are shown in Table 4. The ESSa estimator performs very well for all three neighborhood sizes, only the kNN estimator is slightly better for neighborhood size 100. Kernel density estimates are shown in Fig. S7 in the Supplemental Material, available online.

4 Conclusions

We have seen that the ESS estimators perform better than other estimators for many local dimension estimation tasks. Some estimators have similar performance for classification of synthetic data sets, but of these estimators only the ESS estimators are approximately unbiased even for high dimensions. Moreover, the biases for most other estimators depend heavily on the number of points used for estimation. The DANCo estimator has less bias than the ESS estimators regardless of local data size, but its classification performance is worse.

The ESS estimators have a simple formulation, which make them both easy to implement and amenable to mathematical analysis. Some statistical properties, including consistency, are discussed in the Appendices B and C of the Supple-

Table 4: 1 - AUC (area under ROC curve) for discrimination using dimension estimates of gene expression data (570 tumor samples, 8 adjacent normal samples).

Neighborhood size	30	50	100
ESSa, d = 1	1e-3	0	3e-3
Hill, $k_{30} = 24, k_{50} = 35, k_{100} = 63$	2e-3	1e-3	8e-3
kNN, $N = 10, \gamma = 2$	8e-3	2e-4	2e-3
$k_{30} = 15, p_{30} = \{16, 19, 22, 25, 28\}$			
$k_{50} = 9, p_{50} = \{25, 28, 31, \ldots, 49\}$			
$k_{100} = 6, p_{100} = \{50, 53, 56, \ldots, 98\}$			
F-O	0.14	5e-3	0.01
Fan	5e-3	5e-3	6e-3
DANCo, $k_{30} = 26, k_{50} = 35, k_{100} = 35$	0.01	4e-3	0.09

mental Material, available online. Furthermore, they do not require any tuning of parameters.

The high classification performance and low bias of ESS show that it has good potential to work as a basis for stratification through mixture modeling. ESSa is shown to be clearly superior to other methods on a synthetic data example with two manifolds of different dimensions. We have real data sets where the ESSa estimator as it is performs better than other dimension estimation methods, and other data sets where local PCA performs better. A feasible explanation to this is that ESS is more sensitive to noise than local PCA and it should be investigated whether noise filtering could be used to improve performance.

Bibliography

L. A. Aguirre, G. G. Rodrigues, and E. M. A. M. Mendes. Nonlinear identification and cluster analysis of chaotic attractors from a real implementation of Chua's circuit. *Int. J. Bifurcation and Chaos*, 7(06):1411–1423, 1997.

K. Bache and M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

D. Bell, A. Berchuk, M. Birrer, J. Chien, D. Cramer, F. Dao, R. Dhir, P. DiSaia, H. Gabra, P. Glenn, et al. Integrated genomic analyses of ovarian carcinoma.

Nature, 474(7353):609–615, 2011. data are available through Bioconductor via the R package curatedOvarianData.

F. Camastra and A. Vinciarelli. Estimating the intrinsic dimension of data with a fractal-based method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(10):1404–1407, 2002.

G. Carlsson. Topology and data. *Bull. Amer. Math. Soc.*, 46(2):255–308, 2009.

K. M. Carter, R. Raich, and A. O. Hero III. On local intrinsic dimension estimation and its applications. *IEEE Trans. Signal Process.*, 58(2):650–663, 2010.

C. Ceruti, S. Bassis, A. Rozza, G. Lombardi, E. Casiraghi, and P. Campadelli. DANCo: Dimensionality from angle and norm concentration. arXiv preprint:1206.3881, 2012.

E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín. Searching in metric spaces. *ACM Computing Surv.*, 33(3):273–321, 2001.

S.-W. Cheng and M.-K. Chiu. Dimension detection via slivers. In *Proc. 20th Annu. ACM-SIAM Symp. Discrete Algorithms*, pages 1001–1010, 2009.

J. A. Costa and A. O. Hero III. Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Trans. Signal Process.*, 52(8): 2210–2221, 2004.

J. A. Costa and A. O. Hero III. Determining intrinsic dimension and entropy of high-dimensional shape spaces. In *Statistics and Analysis of Shapes*, pages 231–252. Birkhäuser, Boston, 2006.

T. K. Dey, J. Giesen, S. Goswami, and W. Zhao. Shape dimension and approximation from samples. In *Proc. 13th Annu. ACM-SIAM Symp. Discrete Algorithms*, pages 772–780, 2002.

C. Diks. Estimating invariants of noisy attractors. *Physical Review E*, 53(5):4263–4266, 1996.

D. L. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. AMS Challenges Lecture, 2000.

T. Duong. ks: Kernel density estimation and kernel discriminant analysis for multivariate data in R. *J. Stat. Software*, 21(7):1–16, 2007.

M. Fan, N. Gu, H. Qiao, and B. Zhang. Intrinsic dimension estimation of data by principal component analysis. arXiv preprint:1002.2050, 2010.

K. Fukunaga and D. R. Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Trans. Comput.*, C-20:176–183, Feb. 1971.

N. A. Gershenfeld and A. S. Weigend. The future of time series. In N. A. Gershenfeld and A. S. Weigend, editors, *Time series prediction: Forecasting the future and understanding the past*. Addison-Wesley, 1994. data available at `http://www-psych.stanford.edu/~andreas/Time-Series/SantaFe.html`.

J. Giesen and U. Wagner. Shape dimension and intrinsic metric from samples of manifolds. *Discrete and Computational Geometry*, 32(2):245–267, 2004.

P. Grassberger and I. Procaccia. Measuring the strangeness of strange attractors. *Physica D: Nonlin. Phen.*, 9(1-2):189–208, 1983.

M. Gromov. *Metric structures for Riemannian and non-Riemannian spaces*, volume 152 of *Progress in Mathematics*. Birkhäuser, Basel, 1999. chapter 3 1/2.

G. Haro, G. Randall, and G. Sapiro. Translated Poisson mixture model for stratification learning. *Int. J. Comput. Vision*, 80(3):358–374, 2008.

D. Harte. *Multifractals — Theory and applications*. Chapman and Hall/CRC, 2001.

M. Hein and J.-Y. Audibert. Intrinsic dimensionality estimation of submanifolds in R^d. In *Proc. 22nd Int. Conf. Machine Learning*, pages 289–296. ACM, 2005. data generator available at `http://www.ml.uni-saarland.de/code/IntDim/IntDim.htm`.

B. M. Hill. A simple general approach to inference about the tail of a distribution. *Ann. Stat.*, 3(5):1163–1174, 1975.

P. D. Hoff. Model averaging and dimension selection for the singular value decomposition. *J. Amer. Stat. Assoc.*, 102(478), 2007.

K. Johnsson. *R package manifgen: Data Sets on Manifolds*, 2013. URL `http://www.maths.lu.se/staff/kerstin-johnsson/research/manifold-dimension-estimation/`. function cuthplane.

K. Judd. Estimating dimension from small samples. *Physica D: Nonlin. Phen.*, 71 (4):421–429, 1994.

I. Kivimäki, K. Lagus, I. T. Nieminen, J. J. Väyrynen, and T. Honkela. Using correlation dimension for analysing text data. In *Artificial Neural Networks–ICANN 2010*, volume 6352 of *Lecture Notes in Computer Science*, pages 368–373. Springer, 2010.

A. M. G. Klein Tank, J. B. Wijngaard, G. P. Können, R. Böhm, G. Demarée, A. Gocheva, M. Mileta, S. Pashiardis, L. Hejkrlik, C. Kern-Hansen, et al. Daily dataset of 20th-century surface air temperature and precipitation series for the European Climate Assessment. *Int. J. Climatology*, 22(12):1441–1453, 2002. data and metadata available at `http://www.ecad.eu`.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998. data available at `http://yann.lecun.com/exdb/mnist/`.

J. A. Lee and M. Verleysen. *Nonlinear dimensionality reduction*. Information Science and Statistics. Springer, New York, 2007.

E. Levina and P. J. Bickel. Maximum likelihood estimation of intrinsic dimension. In *Proc. Advances in Neural Information Processing Systems 17*, pages 777–784, 2004.

H. Oltmans and P. J. T. Verheijen. Influence of noise on power-law scaling functions and an algorithm for dimension estimations. *Physical Review E*, 56(1): 1160–1170, 1997.

V. Pestov. An axiomatic approach to intrinsic dimension of a dataset. *Neur. Netw.*, 21(2-3):204–213, 2008.

K. W. Pettis, T. A. Bailey, A. K. Jain, and R. C. Dubes. An intrinsic dimensionality estimator from near-neighbor information. *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-1:25–37, Jan. 1979.

A. Rozza, G. Lombardi, C. Ceruti, E. Casiraghi, and P. Campadelli. Novel high intrinsic dimensionality estimators. *Mach. Learn.*, 89(1-2):37–65, 2012.

J. C. Schouten, F. Takens, and C. M. van den Bleek. Estimation of the dimension of a noisy attractor. *Physical Review E*, 50(3):1851–1861, 1994.

T. Schreiber. Influence of Gaussian noise on the correlation exponent. *Physical Review E*, 56(1):274–277, 1997.

R. L. Smith. Estimating dimension in noisy chaotic time series. *J. Roy. Stat. Soc. Ser. B (Methodological)*, 54(2):329–351, 1992.

F. Takens. On the numerical determination of the dimension of an attractor. In *Dynamical Systems and Bifurcations*, volume 1125 of *Lecture Notes in Mathematics*. Springer, 1985.

J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. data available at `http://isomap.stanford.edu/`.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *J. Roy. Stat. Soc. Ser. B (Stat. Meth.)*, 61(3):611–622, 1999.

G. V. Trunk. Stastical estimation of the intrinsic dimensionality of a noisy signal collection. *IEEE Trans. Comput.*, C-25:165–171, Feb. 1976.

A Closed Expressions for $s_n^{(d)}$ and c_n

Let $V(n) = \pi^{n/2}/\Gamma(n/2+1)$ denote the volume of the unit n-ball and let $A(n) = (n+1)V(n+1)$ denote the area of the unit n-sphere.

c_n is computed from

$$c_n = \frac{1}{V(n)} \int_{B^n} |\cos\theta|\, dV = \frac{1}{A(n-1)} \int_{S^{n-1}} |\cos\theta|\, dS$$
$$= \frac{1}{A(n-1)} \int_0^\pi |\cos\theta|\, A(n-2) \sin^{n-2}\theta\, d\theta$$
$$= \frac{2A(n-2)}{(n-1)A(n-1)} = \frac{2V(n-1)}{A(n-1)}.$$

Now let $\hat{v}_i = v_i/|v_i|$. In order to compute

$$s_n^{(d)} = \frac{1}{V(n)^d} \int_{B^n\times\cdots\times B^n} |u \wedge \hat{v}_1 \wedge \cdots \wedge \hat{v}_d| \\ dV(v_1)dV(v_2)\dots dV(v_d),$$

we first note that

$$|u \wedge \hat{v}_1 \wedge \cdots \wedge \hat{v}_d| = |u \wedge \hat{v}_1 \wedge \cdots \wedge \hat{v}_{d-1}| \cdot |\sin\theta(v_d|u, v_1, \dots v_{d-1})|,$$

where $\theta(v_d|u, v_1, \dots v_{d-1})$ is the angle between v_d and the hyperplane spanned by $u, v_1, v_2, \dots, v_{d-1}$. Thus

$$s_n^{(d)} = \frac{1}{V(n)^d} \int_{B^n\times\cdots\times B^n} |u \wedge \hat{v}_1 \wedge \cdots \wedge \hat{v}_d| \\ dV(v_1)dV(v_2)\dots dV(v_d)$$
$$= \frac{1}{V(n)^d} \int_{B^n\times\cdots\times B^n} \\ \left(\int_{B^n} \sin\theta(v_d|u, v_1, \dots, v_{d-1})\, dV(v_d)\right) \\ \cdot |u \wedge \hat{v}_1 \wedge \cdots \wedge \hat{v}_{d-1}|\, dV(v_1)dV(v_2)\dots dV(v_{d-1})$$
$$= s_n^{(d-1)} \cdot \frac{1}{V(n)} \int_{B^n} \sin\theta(v|\pi_d)\, dV(v)$$

where π_d is the d-dimensional hyperplane defined by $x_{d+1} = x_{d+2} = \cdots = x_n = 0$. Hence it is sufficient to know

$$b_n^m = \frac{1}{V(n)} \int_{B^n} \sin\theta(v|\pi_m)\, dV(v) \\ = \frac{1}{A(n-1)} \int_{S^{n-1}} \sin\theta(v|\pi_m)\, dS(v)$$

for $m = 1, \ldots, d$. The set of points on S^{n-1} with distance $r = \sin\theta$ to π_m is $\{x \in \mathbb{R}^n : x_1^2 + \cdots + x_m^2 = 1 - r^2, x_{m+1}^2 + \cdots x_n^2 = r^2\}$, i.e. the product of an $(m-1)$-sphere with radius $\cos\theta$ and an $(n-m-1)$-sphere with radius $\sin\theta$. This means that

$$b_n^m = \frac{1}{A(n-1)} \int_0^{\pi/2} \sin\theta\, A(n-m-1) \sin^{n-m-1}\theta \\ \cdot A(m-1)\cos^{m-1}\theta\, d\theta = \frac{\Gamma\left(\frac{n}{2}\right)\Gamma\left(\frac{n+1-m}{2}\right)}{\Gamma\left(\frac{n+1}{2}\right)\Gamma\left(\frac{n-m}{2}\right)},$$

and so

$$s_n^{(d)} = \prod_{m=1}^{d} b_n^m = \frac{\Gamma\left(\frac{n}{2}\right)^{m+1}}{\Gamma\left(\frac{n+1}{2}\right)^m \Gamma\left(\frac{n-m}{2}\right)}.$$

B Variance of ESS Estimators

In order to further characterize the ESS estimators we want to find the variance of

$$\hat{s}^{(1)} = \frac{\sum_{x,y\in\mathcal{X}} |\bar{x} \wedge \bar{y}|}{\sum_{x,y\in\mathcal{X}} |\bar{x}||\bar{y}|} \quad \text{and} \quad \hat{c} = \frac{\sum_{x,y\in D} |(\bar{x},\bar{y})|}{\sum_{x,y\in\mathcal{X}} |\bar{x}||\bar{y}|},$$

in the ideal case where we assume that $\mathcal{X}$ is a realization of N i.i.d. variables $\{X_1, \ldots, X_n\}$, where each X_i has uniform distribution on an n-ball for some n. $\bar{x}$ denotes the vector going from the center of the ball to the data point x. Without loss of generality we may assume that the center of the ball is the origin. Let $\theta\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be the function taking (x, y) to the angle between $\bar{x}$ and $\bar{y}$. The estimators $\hat{s}^{(1)}$ and $\hat{c}$ can be seen as weighted averages of $\sin(\theta(X_i, X_j))$ and $\cos(\theta(X_i, X_j))$ respectively. The weights depend only on the lengths of $\bar{X}_i$ and

$\bar{X}_j$ and are hence independent of $\sin(\theta(X_i, X_j))$ and $|\cos(\theta(X_i, X_j))|$. This means that

$$\mathrm{Var}(\hat{s}^{(1)}) = \mathrm{Var}\left(\binom{N}{2}^{-1} \sum_{1\le i<j\le N} \sin(\theta(X_i, X_j))\right)$$

and that

$$\mathrm{Var}(\hat{c}) = \mathrm{Var}\left(\binom{N}{2}^{-1} \sum_{1\le i<j\le N} |\cos(\theta(X_i, X_j))|\right).$$

Let Θ_k, $k = 1, \dots N(N-1)/2$ be an enumeration of $\theta(X_i, X_j)$, $1 \le i < j \le n$. The marginal probability density function for each Θ_k is $f(\theta) = \sin^{n-2}\theta A(n-2)/A(n-1)$. In order to see this, first note that since $\theta(X_i, X_j)$ is not dependent on the lengths of $\bar{X}_i$ and $\bar{X}_j$ we can assume that X_i and X_j lie on the unit $(n-1)$-sphere. Because of rotation invariance we can also assume that X_j coincides with a coordinate axis. Finally, the points on a unit $(n-1)$-sphere that form an angle θ to a fixed axis is a unit $(n-2)$-sphere with radius $\sin\theta$.

The variables Θ_k are pairwise independent; this is obvious if $\Theta_{k_1} = \theta(X_i, X_j)$ and $\Theta_{k_2} = \theta(X_m, X_n)$ with distinct i, j, m and n. However, even if $i = m$ we have independence because of rotational invariance and independency of X_j and X_m. Analogously we have independence if $j = n$ but $i \neq m$. This means that if $Z = (\sin\Theta_k)_{k=1}^{N(N-1)/2}$, then

$$\mathrm{Var}(\hat{s}^{(1)}) = \binom{N}{2}^{-2} \mathbf{1}^T \mathrm{Cov}(Z)\mathbf{1} = \frac{2}{N(N-1)}\mathrm{Var}(\sin\Theta_1).$$

Similarly $\mathrm{Var}(\hat{c}) = 2\mathrm{Var}(|\cos\Theta_1|)/N(N-1)$. Finally we have

$$\begin{aligned}\mathrm{Var}(\sin\Theta_1) = \int_0^\pi \sin^2\theta f(\theta)\, d\theta - \left(\int_0^\pi \sin\theta f(\theta)\, d\theta\right)^2 \\ = \frac{A(n-2)}{A(n-1)} \int_0^\pi \sin^n\theta\, d\theta \\ - \left(\frac{A(n-2)}{A(n-1)} \int_0^\pi \sin^{n-1}\theta\, d\theta\right)^2\end{aligned}$$

and

$$\begin{aligned}\operatorname{Var}(|\cos\Theta_1|) &= \int_0^{\pi} \cos^2\theta f(\theta)\,d\theta \\ &\qquad - \left(\int_0^{\pi} |\cos\theta| f(\theta)\,d\theta\right)^2 \\ &= \frac{A(n-2)}{A(n-1)}\int_0^{\pi}\cos^2\theta\sin^{n-2}\theta\,d\theta \\ &\qquad - \left(\frac{A(n-2)}{A(n-1)}\int_0^{\pi}|\cos\theta|\sin^{n-2}\theta\,d\theta\right)^2 \\ &= \frac{2A(n-2)}{A(n-1)}\int_0^{\pi/2}(\sin^{n-2}\theta - \sin^n\theta)\,d\theta \\ &\qquad - \left(\frac{2V(n-1)}{A(n-1)}\right)^2\end{aligned}$$

for which we can get closed expressions using

$$\int_0^{\pi/2}\sin^m\theta\,d\theta = \frac{\sqrt{\pi}}{2}\frac{\Gamma(\frac{m+1}{2})}{\Gamma(\frac{m+2}{2})}.$$

In Fig. S1 we see the expected value and standard deviation of $\hat{s}^{(1)}$ when $N = 50$, and we can clearly see how it gets harder to estimate higher intrinsic dimensions.

C Consistency of ESS Estimators

From Appendix B we see that with $M = \binom{N}{2}$,

$$\hat{s}^{(1)} = \frac{1}{M}\sum_{k=1}^{M}\sin\Theta_k \quad \text{and} \quad \hat{c} = \frac{1}{M}\sum_{k=1}^{M}|\cos\Theta_k|,$$

where the Θ_k are pairwise independent and the $\sin\Theta_k$ have finite variance as well as the $|\cos\Theta_k|$. In the ideal case with data uniformly distributed on a ball we thus get consistency for both estimators from the weak law of large numbers.

Suppose now that data is sampled with a measure μ with smooth non-uniform density on a smooth n-submanifold $\mathcal{M}$ of $\mathbb{R}^D$. Since $\mathcal{M}$ is embedded in $\mathbb{R}^D$ we

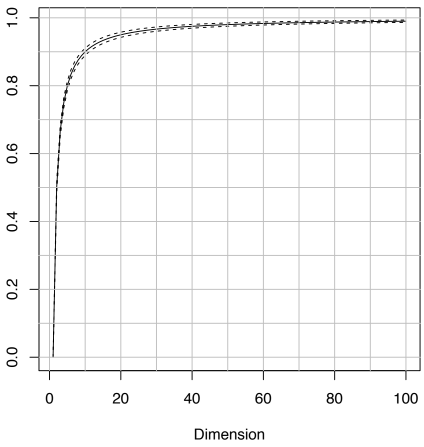


Figure S1: Mean value of $\hat{s}^{(1)}$ and mean value plus/minus one standard deviation (dashed lines) for a local data set with 50 data points.

can view the tangent space at $p \in \mathcal{M}$, $T_p\mathcal{M}$, as the n-dimensional subspace of $\mathbb{R}^D$ consisting of all tangents at p to curves in $\mathcal{M}$ going through p. Let $P\colon \mathcal{M} \to T_p\mathcal{M}$ be the orthogonal projection of a point on $\mathcal{M}$ onto $T_p\mathcal{M}$. Let $U = \mathcal{M} \cap B_\epsilon(p)$ for any ϵ that is small enough so that $P|_U$ is one-to-one and let $V = P(U)$. Also, let ν be the pushforward measure of μ under P.

If $q = p + tv \in V \subseteq T_p\mathcal{M}$, where v is a tangent vector of length one and $t \in \mathbb{R}$, then $|q - P^{-1}(q)| = O(t^2) \leq O(\epsilon^2)$. This means that the difference of estimating c_n and $s_n^{(d)}$ from μ on U instead of from ν on V will be $O(\epsilon^2)$. Furthermore we get from the triangle inequality that $B_{\epsilon - C\epsilon^2}(p) \subseteq V \subseteq B_\epsilon(p)$. Since $(\epsilon - C\epsilon^2)^n/\epsilon^n \to 1$ when $\epsilon \to 0$ we get that $\nu(V) \to \nu(B_{\epsilon - C\epsilon^2}(p))$ as $\epsilon \to 0$, so the estimation will asymptotically be the same as if we only consider points within $B_{\epsilon - C\epsilon^2}(p)$. Since $B_{\epsilon - C\epsilon^2}(p)$ is a sphere we now almost have the ideal case, the difference being that ν is not uniform on $B_{\epsilon - C\epsilon^2}(p)$. However, since P is smooth and μ has smooth density, so has ν, which means that when $\epsilon \to 0$, ν gets arbitrarily close to uniform measure on $B_{\epsilon - C\epsilon^2} \cap T_p\mathcal{M}$.

It should be noted though that in the presence of noise the ESS estimators, as well as any estimator of local intrinsic dimension, will not be consistent since then the local intrinsic dimension is not well defined. This is illustrated in Fig. S2, where we can see that with denser sampling and a fixed number of neighbors

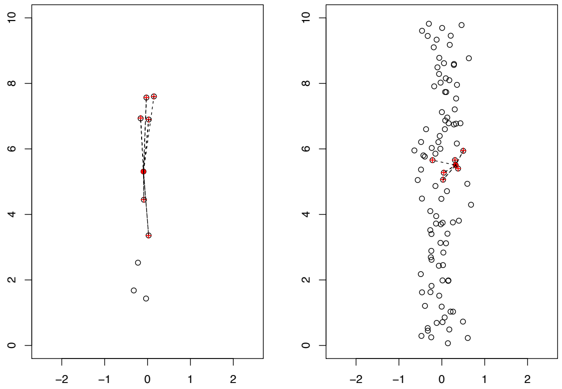


Figure S2: To the left we have 10 samples from a uniform one-dimensional distribution with two-dimensional noise, to the right we have 100 samples from the same distribution. The six nearest neighbors of a point are marked.

(or a smaller cut-off radius) the local data set will eventually get the same intrinsic dimension as the noise.

D Further Details on Parameters

D.1 Estimation of Noise Parameters for TP Estimator

The TP estimator needs two parameters for the noise in the data: its dimension and its variance. It is a natural idea to use the extrinsic dimension of the data as the dimension of the noise, but cutting out a local set from a manifold means that the part of the noise that is parallel to the manifold is canceled out if the manifold is flat. Hence we subtract a rough pre-estimate $\tilde{n}$ of the dimension of the manifold from the extrinsic dimension of the data set and use the result as the estimated dimension of the noise. The variance of the noise is estimated from the local covariance matrix as follows: Given a preliminary estimate of the dimension $\hat{n}$, the estimate of the noise variance is the mean of the $D - \hat{n}$ smallest eigenvalues of the covariance matrix for the local data set, where D is the extrinsic dimension of the data. If $\hat{n}$ is smaller than n, some relatively large eigenvalues corresponding to directions along the manifold can result in a significant overestimate of the noise variance; in order to avoid this one can choose $\hat{n}$ larger than what is believed to be the intrinsic dimension.

We have chosen $\hat{n}$ and $\tilde{n}$ for each experiment in a way that we think can represent a reasonable guess from a researcher not knowing the true intrinsic dimension. For dimension estimation of synthetic data with noise we use both the exact value of the variance of the noise and the estimate.

D.2 Parameter Selection for Hill, TP, kNN and DANCo

A systematic search was used to find the best parameters for the Hill, TP (exact variance), TP (estimated variance), kNN and DANCo estimators. For the kNN estimator and neighborhood size M we limited the search to the case $N = 10, \gamma = 2$ and $p \in \{p_{\min}, p_{\min} + 3, \ldots, p_{\min} + 3m\}$ where $p_{\min} = \max(k + 1, M/2)$ and m was chosen such that $p_{\min} + 3m \in (M - 4, M - 1)$. Hence only the parameter k needed to be optimized.

The best parameters were defined as those who seemed to give the best classification performance (i.e. proportion of correct estimates) of uniformly distributed local data sets of dimension 3, 4, . . . , 9, with thresholds based on dimension estimates of uniformly distributed local data sets of dimension 2, 3, . . . 10 (see main paper Section 3.1). A coarse search was first made over a large interval and then a refined search in the range which gave high classification performances in the first search. We studied the results in bar charts such as those shown in Fig. S3. Due to random effects some judgment was still needed to pick parameters. In Fig. S3 the kNN estimator had the best classification performance in the range from $k = 8$ to $k = 11$, so we selected $k = 9$ as the parameter. We observed that the bias increased as k increased, which was a reason to rather choose $k = 9$ than $k = 10$. For the DANCo estimator classification performance was fairly constant from k = 30 to k = 40, so we selected $k = 35$ since it was in the middle of this interval.

E Further Details on Results from Experiments in the Main Article

The purpose of this section is to provide further details on the results from some of the experiments in the main article, namely the dimension estimation of U_{2-10}, U_{20-100}, N_{2-10} and S_{2-10}, the dimension estimation of the 5-ball together with the 8-sphere, the dimension estimation of the wide range of synthetic data sets

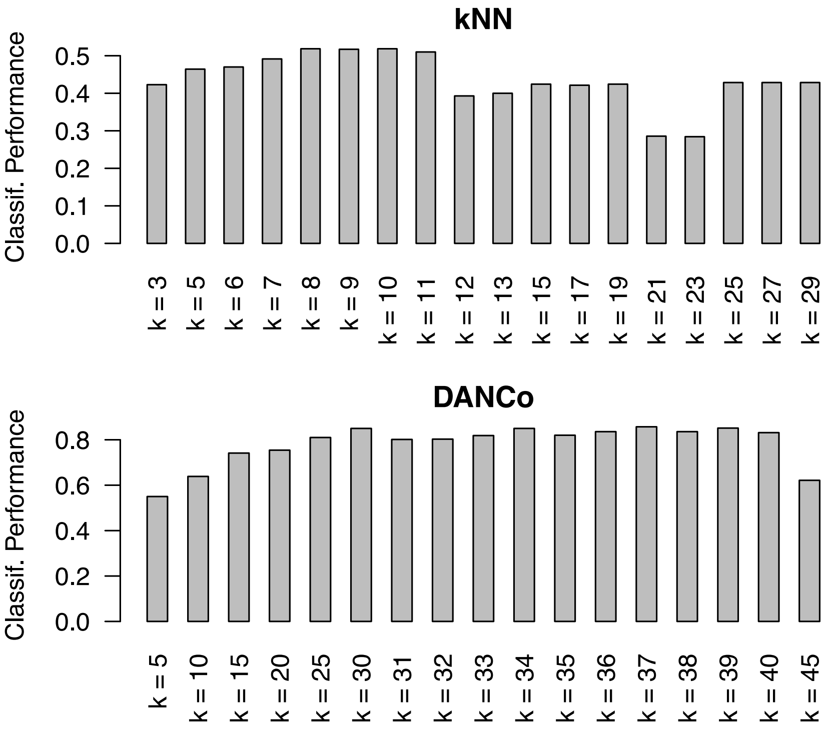


Figure S3: Examples of bar charts used for parameter selection. The neighborhood size used for the charts above is 50.

Table 5: Parameters for Hill, TP, kNN and DANCo estimators for varying neighborhood sizes.

Neighborhood size	30	50	100
Hill	$k = 24$	$k = 35$	$k = 63$
TP (exact variance)	$k = 9$	$k = 21$	$k = 32$
TP (estimated variance)	$k = 12$	$k = 20$	$k = 36$
kNN	$k = 15$	$k = 9$	$k = 6$
DANCo	$k = 26$	$k = 35$	$k = 35$

and the dimension estimation of the ovarian cancer data set.

In Fig. S4 the kernel density estimates for the high-dimensional set of groups, U_{20-100} are shown for the estimators we consider. Here we can see the disadvantage of only providing integer estimates when it comes to classification. Also note that the decision boundaries are not set optimally for the ESS estimators since we consider these estimators as unbiased when we do classification. Even though there is a bias for high dimensions in the ESS estimators, we think that since it is relatively small it is better to disregard it so that we can avoid decision boundaries that depend on specific realizations of ideal data sets.

Table 6 shows average bias and standard deviation for the sets U_{2-10}, U_{20-100}, N_{2-10} and S_{2-10}. The Hill, TP and Fan estimators have a large bias, but the Hill and TP estimators have small average standard deviation. The kNN estimator has higher average standard deviation and somewhat smaller bias. The F-O estimator has large bias for the high-dimensional set. The DANCo and ESS estimators all have small bias, but the ESS estimators have smaller average standard deviation.

An overview of the wide range of synthetic data sets previously used for global dimension estimation (Rozza et al., 2012) can be found in Table 7. The data sets are sampled from smooth manifolds with a known intrinsic dimension n ranging from 1 to 24. The extrinsic dimension of the data, D, ranges from 3 to 96. Many of the data sets are sampled with a non-uniform density over the manifold, but the density is always smooth.

Results from dimension estimation of these data sets are shown in Fig. S5. The ESS and DANCo estimators have in general the most accurate results with a few exceptions where there ESS estimators have a slightly larger bias than other estimators ($\mathcal{M}_6, \mathcal{M}_8$ and $\mathcal{M}_{13}$). The ESS estimators are versatile and are never very far off. An interesting feature is that the ESS and DANCo estimators very seldom underestimates the dimension, and then very slightly, whereas the Hill estimator never overestimates the dimension. This means that we get an upper and a lower bound for the intrinsic dimension.

Fig. S6 shows kernel density estimates for the 5-ball/8-sphere data set that was used to investigate stratification properties in the main article (Section 3.3). Also here we see the advantage of providing non-integer dimension estimates. In the left part of Fig. S6 the true identities of the data points have been used to compute kernel density estimates and decision boundaries, but we see that for the estimators which give approximately bimodal kernel density estimates when all neighborhoods are considered together (the ESS, Hill, kNN and Fan estimators),

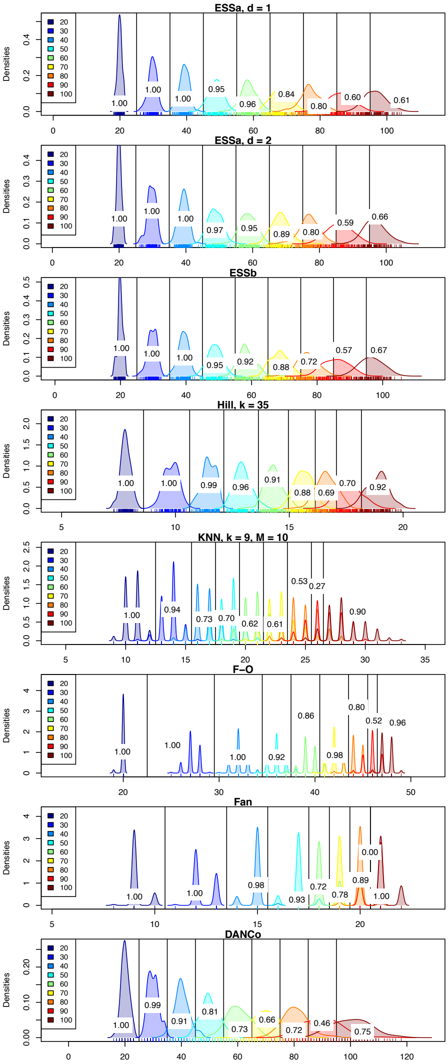


Figure S4: Kernel density estimates of dimension estimates for U_{20-100}, i.e. local uniformly distributed data sets with 50 data points each, with intrinsic dimension 20, 30, . . . , 100 in each group.

Table 6: Average bias and average standard deviation for local data sets with 50 points (cf. main article Table 1).

	U_{2-10}	U_{20-100}	N_{2-10}	S_{2-10}	
ESSa, $d = 1$	0.02	-1.74	0.43	0.19	bias
	0.19	2.2	0.21	0.21	sd
ESSa, $d = 2$	0.02	-1.73	0.51	0.22	bias
	0.17	2.2	0.19	0.19	sd
ESSb	0.02	-1.77	0.36	0.16	bias
	0.22	2.4	0.23	0.23	sd
Hill	-2.46	-46.1	-2.28	-2.40	bias
	0.10	0.38	0.10	0.095	sd
TP (ex. var.)	-1.9	-42	-1.9[a]	—	bias
	0.16	0.66	0.18	—	sd
TP (est. var.)	—	—	-2.0[a]	-1.9[b]	bias
	—	—	0.42	0.16	sd
kNN	-1.64	-39.7	-1.39	-1.61	bias
	0.34	1.1	0.39	0.35	sd
F-O	0	-22.8	0	0.19	bias
	0	0.71	0	0.27	sd
Fan	-2.73	-43.0	-1.98	-2.31	bias
	0.34	0.41	0.20	0.25	sd
DANCo	0.02	0.21	0.71	0.21	bias
	0.38	4.0	0.5	0.43	sd

[a] $\tilde{n} = 6, \hat{n} = 10$.

[b] $\tilde{n} = \hat{n} = D - 1$, where D is the extrinsic dimension of data.

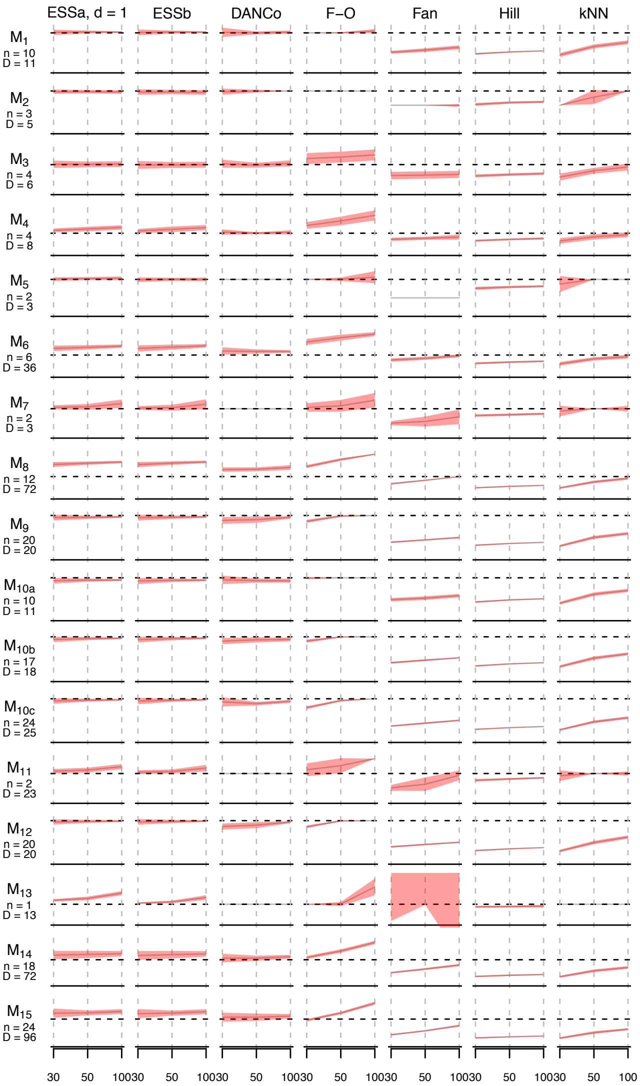


Figure S5: Intrinsic dimension estimation on synthetic data sets with intrinsic dimension n and extrinsic dimension D for neighborhood size $N = 30, 50, 100$. Parameters for the DANCo, Hill and kNN estimators are given in Table 5. The shaded red area shows the region within one standard deviation from the mean of the local intrinsic dimension estimates. The dashed line shows the true intrinsic dimension.

Table 7: Synthetic data sets previously used for global dimension estimation.

Data set	Intr. dim.	Size	Source	Comment
$\mathcal{M}_1 - \mathcal{M}_{13}$	1–24	$2000 \times D$	(Hein and Audibert, 2005)	
$\mathcal{M}_{14} - \mathcal{M}_{15}$	18, 24	$2000 \times D$		[a]

[a] Generated as described in (Rozza et al., 2012).

a similar decision boundary would have been obtained only using bimodality.

In Fig. S7 the dimension estimates of the ovarian cancer gene expression data set are shown. The large green bars which correspond to adjacent normal samples have on average lower dimension estimates than the tumor samples.

F Further Experiments on Synthetic Data Sets

In this section we use a few more sets of groups of local data to characterize the estimators.

Two aspects that were not studied separately in the main article were the effect of scaling of the data and edge effects. We also want to study the effect of high-dimensional noise more in detail. Hence we consider four additional sets of groups of data. C_{2-10} consists of one group of 100 data sets for each dimension 2, 3, . . . 9. Each data set in the group with intrinsic dimension n is first generated from a uniform distribution on the unit ball and then half of the variables (rounded down) are scaled with a factor 2. But since we consider local dimension estimation where neighborhoods are determined by a cut-off radius (the cut-off radius is the distance to the Nth neighbor when we consider data sets of size N) we exclude data points that are moved outside the unit ball. If we do not get enough data points inside the unit ball we generate more data points in the same way. E_{2-10} consists of data points uniformly distributed in the upper hemisphere of the unit ball (i.e. the first coordinate has a positive value), which simulates a situation close to a manifold boundary. Hence edge effects should be more pronounced in this data set. To study high-dimensional noise we use the sets $N^{0.01}_{20-100}$ and $N^{0.05}_{20-100}$, where $N^{0.05}_{20-100}$ is generated in the same way as N_{2-10} but with intrinsic dimensions 20, 30, . . . , 100 and noise dimension 150, and $N^{0.01}_{20-100}$ is also generated in this way, but with covariance matrix $0.01^2 I$ instead

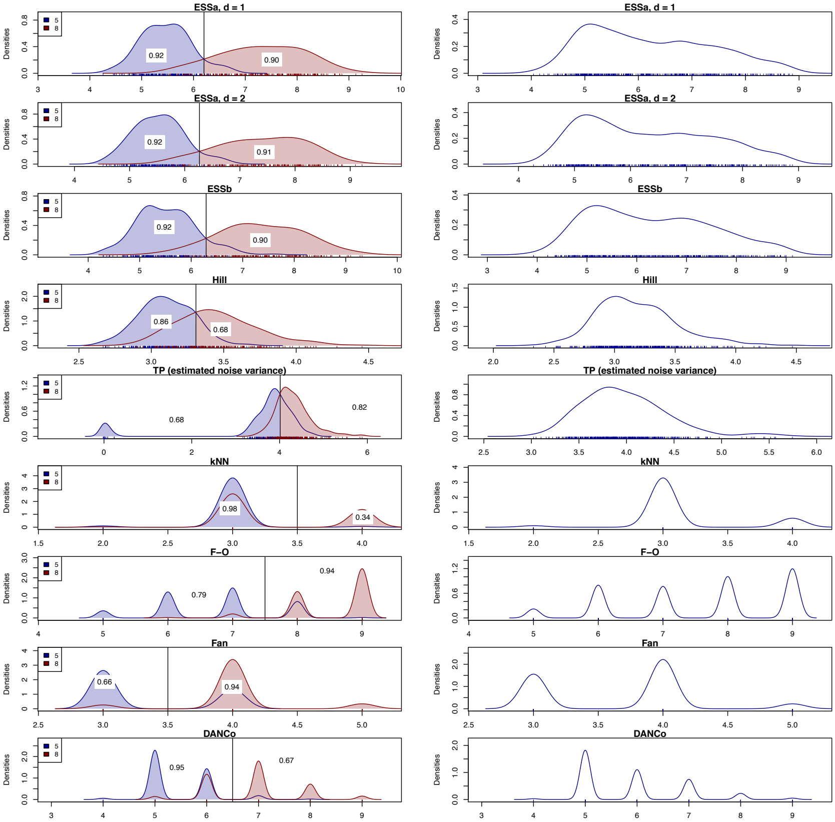


Figure S6: Kernel density estimates of local dimension estimates of 5-ball inside 8-sphere. In the left image prior knowledge about which manifold each data point belongs to is used to construct kernel density estimates. The right image shows kernel density estimates of the dimension estimates of all data points.

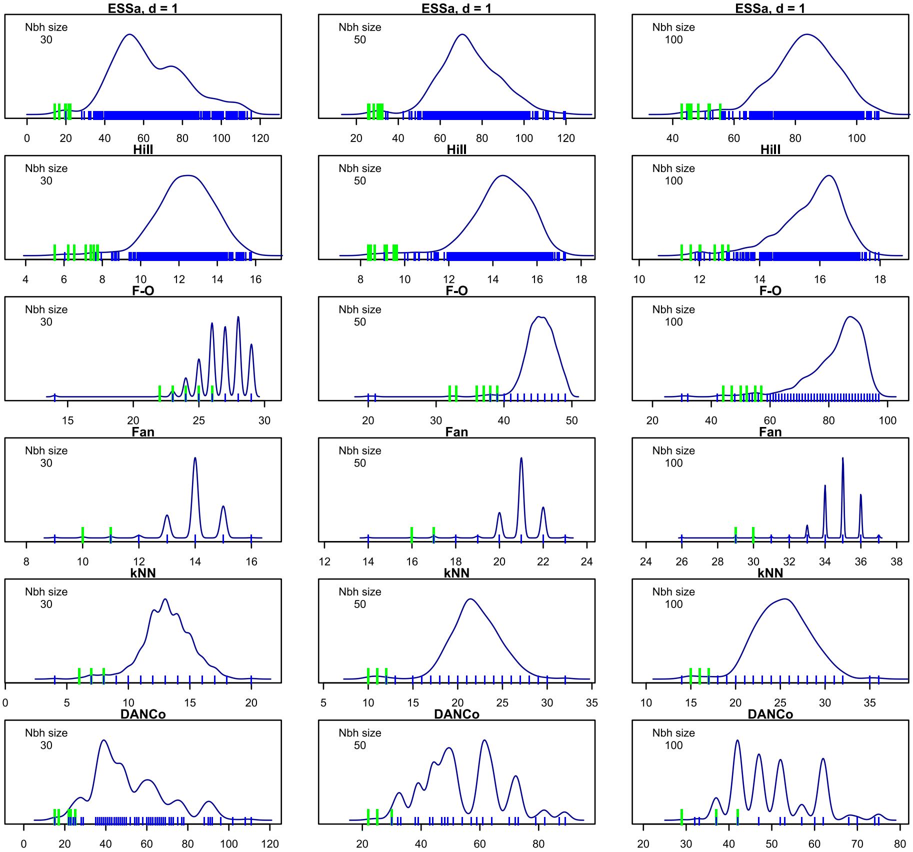


Figure S7: Dimension estimates of ovarian cancer data and kernel density estimates of these. The large green bars show dimension estimates of adjacent normal samples and the small blue bars show dimension estimates of tumor samples. The curve is the kernel density estimate of all dimension estimates together.

Table 8: Classification performance P based on dimension estimation of local data sets with 50 points.

	C_{2-10}	E_{2-10}	$N_{20-100}^{0.01}$
ESSa, $d = 1$	0.97	0.81	0.92
ESSa, $d = 2$	0.97	0.86	0.91
ESSb	0.96	0.72	0.89
Hill, $k = 35$	0.84	0.76	0.83
TP (exact noise var.) $k = 21$	—	—	0.58[a]
TP (estimated noise var.) $k = 20$	—	—	0.78[a]
kNN $k = 9, N = 10,$ $p = \{25, 28, 31, \ldots, 49\}$	0.50	0.48	0.60
F-O, $\alpha = 0.05$	1	1	
Fan, $\alpha = 10, \beta = 0.8$	0.39	0.41	0.59
DANCo, $k = 35$	0.85	0.67	0.73

[a] $\tilde{n} = 60, \hat{n} = 100.$

of $0.05^2 I$. The classification performance of the estimators on C_{2-10}, E_{2-10} and $N_{20-100}^{0.01}$ are shown in Table 8.

The classification performance P on $N_{20-100}^{0.05}$ is zero for all estimators, due to high bias. The dimension estimation results are shown in Fig. S8. We can see that for most estimators (not F-O, Fan and TP with exact noise variance) the classification performance would be decent if the decision boundaries were set based on the noisy data sets themselves, instead of data sets without noise. This means that when comparing between noisy data sets with similar noise but different intrinsic dimensions there is still a good chance of distinguishing between them.

It is interesting to note that for the data sets with high-dimensional noise the TP method with estimated noise variance works much better than the TP method with exact noise variance. This shows that the estimate of the dimension of the noise is important, and not only the variance.

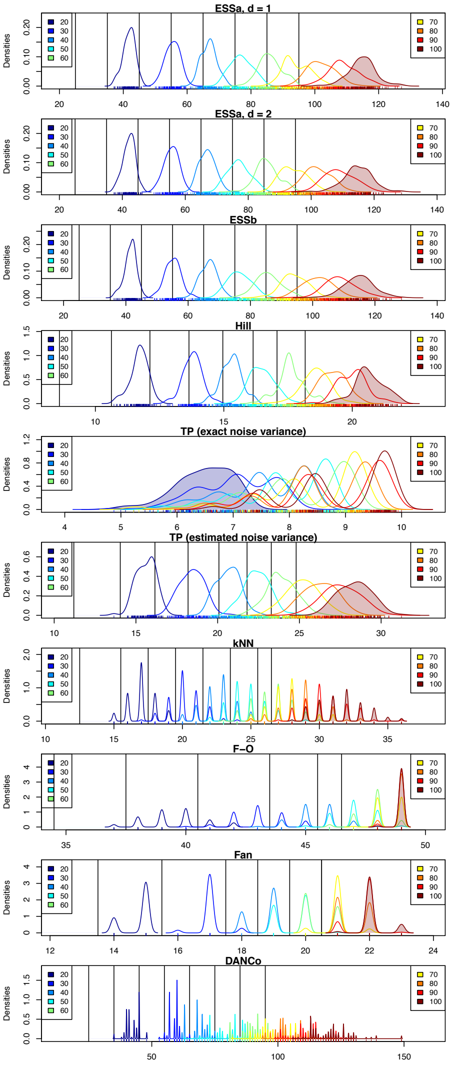


Figure S8: Dimension estimation of $N^{0.05}_{20-100}$. Classification boundaries are based on U_{20-100}. The filled areas correspond to estimates that are correctly classified.

G Further Experiments on Real Data Sets

In this section we use six real data sets previously used for global intrinsic dimension estimation (Rozza et al., 2012) to further test ESS and other local intrinsic dimension estimators. As in the previous section we adopt the experimental setup for local dimension estimation, which means that we pick 100 local data sets with N points from each data set, with $N = 30, 50, 100$. An overview of the data sets are shown in Table 9.

Table 9: Real data sets.

Data set	Intr. dim.	Size	Source	Comment
$\mathcal{M}_{\text{Faces}}$	3	698×4096	(Tenenbaum et al., 2000)	
$\mathcal{M}_{\text{MNIST1}}$		6742×784	(LeCun et al., 1998)	Training points for digit 1.
$\mathcal{M}_{\text{SantaFe}}$	9	1000×50	(Gershenfeld and Weigend, 1994)	Data set D2. [a]
$\mathcal{M}_{\text{Isolet}}$		7797×617	(Bache and Lichman, 2013)	
$\mathcal{M}_{\text{DSVC1}}$		250×20	(Aguirre et al., 1997)	[a]
$\mathcal{M}_{\text{Paris14e}}$		785×20	(Klein Tank et al., 2002)	[a, b]

[a] As in (Rozza et al., 2012) the method of delays is used so that the one-dimensional time series data is embedded into a higher-dimensional space.

[b] Daily mean temperatures from Paris 14-E Parc Montsouris, from January 1st 1958 to December 25 2000. The end date is earlier than reported in (Rozza et al., 2012), but this is necessary to get the correct number of samples.

The data sets $\mathcal{M}_{\text{Faces}}$ and $\mathcal{M}_{\text{SantaFe}}$ have a ground truth since they are actually synthetically generated, but not from a manifold model and hence considered as real data. For the other data sets, $\mathcal{M}_{\text{MNIST1}}$, $\mathcal{M}_{\text{Isolet}}$, $\mathcal{M}_{\text{DSVC1}}$ and $\mathcal{M}_{\text{Paris14e}}$, we can only compare with other estimators; here we use global dimension estimates from (Rozza et al., 2012) as a reference (we remove outliers and results from estimators based on global linear embeddings).

The $\mathcal{M}_{\text{Faces}}$ set is also known as the ISOMAP face data set (Tenenbaum et al., 2000), it consists of 698 64×64 pixel images of faces. Each pixel is considered as a variable, so the extrinsic dimension is 4096. This data set has three degrees of freedom (two degrees of freedom for pose and one degree of freedom for lightning direction), this is the intrinsic dimension. The $\mathcal{M}_{\text{SantaFe}}$ set is the D2 time series from the Santa Fe time series competition (Gershenfeld and Weigend, 1994). It has 50,000 1-dimensional data points. The data set describes a particle in motion and it has nine degrees of freedom (four for position, four for velocity and one for time). The degrees of freedom in the Santa Fe data set correspond to the intrinsic dimension of the data set when it is embedded in a higher-dimensional space by the method of delays.

The $\mathcal{M}_{\text{MNIST1}}$ data set we also studied in the main article, it is the images of handwritten ones from the training set of the MNIST data (LeCun et al., 1998). However, the setting is not exactly the same since in the main article we looked at a sample of 500 points from the MNIST training data set. Here we take neighborhoods based on the whole data set.

The $\mathcal{M}_{\text{DSVC1}}$ data set consists of a time series from a chaotic system with a strange attractor. The model for the data is not a smooth distribution along a manifold, but a smooth distribution along a fractal. The various definitions of fractal dimension coincide with manifold dimension for data sets that are manifolds, but otherwise they can differ among themselves. The Hill estimator is constructed so that it measures the fractal dimension known as correlation dimension, but the other estimators considered here have not a well-defined behavior for fractal sets since they are based on manifold models.

The $\mathcal{M}_{\text{Isolet}}$ data set contains features computed from recordings of spoken letters (Bache and Lichman, 2013), and the $\mathcal{M}_{\text{Paris14e}}$ data set consists of a long time of temperatures from Paris, spanning over a period of approximately 40 years (Klein Tank et al., 2002).

The results of dimension estimation on these data sets are shown in Fig. S9.

The ESS and DANCo estimators are the only estimators with correct mean for the $\mathcal{M}_{\text{SantaFe}}$ data set (the ESS estimators have sligthly larger variance than DANCo). For the other data set with ground truth, $\mathcal{M}_{\text{Faces}}$, these estimators overestimate dimension (ESS estimators more than DANCo). Also estimators which in most cases underestimate dimension (Hill and kNN) overestimate the dimension of the $\mathcal{M}_{\text{Faces}}$ data set, however not as much as ESS and DANCo. When increasing the neighborhood size the estimated dimensions increase for all estimators on this data set. Since the extrinsic dimension of the data set is very high (4096) a plausible explanation for this is that the tangent space approximation of the manifold is not good inside the neighborhood, i.e. that the manifold is curved inside the neighborhood. When the neighborhood size increases the approximation gets worse.

For three of the other data sets ($\mathcal{M}_{\text{MNIST1}}$, $\mathcal{M}_{\text{Isolet}}$ and $\mathcal{M}_{\text{Paris14e}}$) the dimension estimates from the ESS and DANCo estimators are higher than or in the top part of the reference interval. The reference is based on global dimension estimators, which since they use more information than local dimension estimators can be considered more accurate when the whole data set has the same dimension, but many of them still suffer from negative bias (Rozza et al., 2012) and this has to be considered when looking at the reference.

For the $\mathcal{M}_{\text{MNIST1}}$ data set we get slightly higher dimension estimates here than in the main article, which is due to that the whole data set is used when constructing neighborhoods. This means that the sampling of the manifold is denser and the effect of noise gets more pronounced, cf. Fig. S2. When neighborhood size is increased the dimension estimates decrease since noise gets less dominant. For the Fan and kNN estimators the estimated dimension increases with neighborhood size, but these estimator have in general a negative bias which decreases with increasing neighborhood size (see Fig. S5).

Also for the $\mathcal{M}_{\text{Isolet}}$ data we see that estimated dimension decreases with increasing neighborhood size for ESSa, ESSb and DANCo. In (Kivimäki et al., 2010) it was reported that with smaller neighborhoods the dimension increased a lot for this data set.

For the $\mathcal{M}_{\text{DSVC1}}$ data set the ESS estimators are within the reference interval. For this fractal data set many estimators that cannot incorporate fractal sets in the model that they are built upon still gave good results.

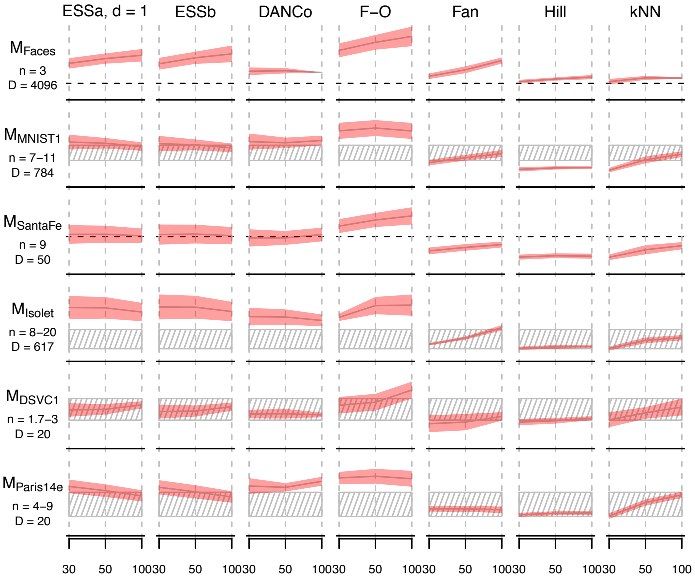


Figure S9: Intrinsic dimension estimation on real data sets with extrinsic dimension D for neighborhood size $N = 30, 50, 100$. Parameters for the DANCo, Hill and kNN estimators are given in Table 5. The ground truth or reference for intrinsic dimension is n. The shaded red area shows the region within one standard deviation from the mean of the local intrinsic dimension estimates. The dashed line shows the true intrinsic dimension and the area shaded with lines shows the the interval where the reference global dimension estimates are for data sets without ground truth.

PAPER II

Published in *BMC Bioinformatics* 2015.

BayesFlow: Latent modeling of flow cytometry cell populations

Kerstin Johnsson, Jonas Wallin and Magnus Fontes

Abstract

Background: Flow cytometry is a widespread single-cell measurement technology with a multitude of clinical and research applications. Interpretation of flow cytometry data is hard; the instrumentation is delicate and can not render absolute measurements, hence samples can only be interpreted in relation to each other while at the same time comparisons are confounded by inter-sample variation. Despite this, most automated flow cytometry data analysis methods either treat samples individually or ignore the variation by for example pooling the data. A key requirement for models that include multiple samples is the ability to visualize and assess inferred variation, since what could be allowed as technical variation in one setting are different phenotypes in another.

Results: We introduce BayesFlow, a pipeline for latent modeling of flow cytometry cell populations built upon a Bayesian hierarchical model. The model systematizes variation in location as well as shape. Expert knowledge can be incorporated through informative priors and the results can be supervised through compact and comprehensive visualizations.

BayesFlow is applied to two synthetic and two real flow cytometry data sets. The first real data set is from the FlowCAP I challenge. BayesFlow does not only give a gating which would place it among the top performers in FlowCAP I for this dataset, it also gives a more consistent treatment of different samples than either manual gating or other automated gating methods. The second real data set contains replicated flow cytometry measurements of samples from healthy individuals. BayesFlow gives here cell populations with clear expression patterns and small technical intra-donor variation as compared to biological inter-donor variation.

Conclusions Modeling latent relations between samples through BayesFlow enables a systematic analysis of inter-sample variation. As opposed to other joint gating methods, effort is put at ensuring that the obtained partition of the data corresponds to actual cell populations, and the result is therefore directly biologically interpretable. BayesFlow is freely available at GitHub.

1 Introduction

In a flow cytometer a number of characteristics for each individual cell in a sample of $\sim 10^4$ to $\sim 10^6$ cells are quantified as they pass through the cytometer in a fluid stream. The data that are obtained are most often summarized by grouping cells into cell populations; properties of these cell populations are used in many clinical applications—for example monitoring HIV infection and diagnosing blood cancers—and in many branches of medical research (Shapiro, 2005, Nolan and Yang, 2007). Defining the cell populations based on the measured characteristics is in state-of-the-art analyses still done manually by trained operators looking at two-dimensional projections of the data. The importance of automated methods has risen along with an increase of the dimension of typical flow cytometry data sets due to developments in flow cytometry technology (O'Neill et al., 2013) and the emergence of studies with large numbers of flow cytometry samples (Chen et al., 2015). Furthermore, manual so called gating of cell populations is a subjective process where operators have to take more or less arbitrary decisions for example when there are overlapping populations (Welters et al., 2012).

Automatic cell population identification is hard since flow cytometry measurements are not absolute, while at the same time different samples cannot be directly compared due to technical variation—especially apparent when samples are analyzed at different laboratories (Welters et al., 2012)—and intrinsic biological variation within and between subjects. Despite this, research into automated population identification methods has focused on individual or pooled flow cytometry samples, sometimes attempting to align data at first through normalization procedures (Hahne et al., 2010).

Automated methods with the aim to replace manual gating must be able to treat multiple samples jointly and take variation between samples into account, while at the same time make it possible for the user to monitor that variation so that it is not too high for the application at hand. For example it needs to be decided if a shift in location of a population in a sample can be seen as technical variation and accepted or if the changed marker expression means that it is a different cell phenotype. These kinds of methods also need to be able to take prior information into account—in manual gating the experience of the operator can be necessary to define a population. We have developed BayesFlow, a method which models variation in cell population location as well as shape, can include prior information for example about cell population location, and gives a result that can be assessed in compact and comprehensive visualizations.

Partitioning the cell measurements in a sample into cell populations is essentially a clustering problem. In the context of flow cytometry data analysis clustering is called automated gating, as opposed to the manual gating performed by operators. Model-based clustering using mixture models has been the most used approach for automated gating (Lo et al., 2008, Boedigheimer and Ferbas, 2008, Chan et al., 2008, Pyne et al., 2009, Hu et al., 2013, Naim et al., 2014). Mixture models are very well suited to describe flow cytometry data because they have a natural biological interpretation based on the cell populations. Examples of other approaches that have been used for automated gating are grid based density clustering (Qian et al., 2010), spectral clustering (Zare et al., 2010), hierarchical clustering (Qiu et al., 2011, Bruggner et al., 2014) and k-means clustering (Aghaeepour et al., 2011, Ge and Sealfon, 2012). An evaluation of a wide range of automated gating methods was performed in the FlowCAP I challenge (Aghaeepour et al., 2013). The discrepancy with manual gating was often quite large even for the best methods, with average F-measures around 0.9 for both completely automated and manually tuned methods. Large discrepancies between manual and automatically gated samples can be acceptable since the arbitrary decisions taken in manual gating means that the gates could just as well have been set another way. However, it is important that the gating is consistent between samples so that they can be compared against each other.

Joint identification of cell populations in a collection of samples can be accomplished by pooling the samples (Qiu et al., 2011, Naim et al., 2014) or matching populations identified separately in the samples (Pyne et al., 2009, Azad et al., 2013). However, in the first approach no variation between samples is taken into account and in the second approach no information is shared between samples.

Recently a third approach has been explored, where a Bayesian hierarchical model is used to share information between samples while at the same time allowing for variation. This was first utilized for flow cytometry gating by Cron et al. (Cron et al., 2013), with a hierarchical Dirichlet process model with fixed locations and shapes of cell populations. An extension of this model, also modeling variation in cell population locations has been used to create ASPIRE, a method for anomalous sample detection (Dundar et al., 2014).

BayesFlow follows this third approach, but use a differently structured model than what has been used previously, favoring explicit modeling instead of implicit, parametric instead of non-parametric (or massively parametric). This follows the philosophy that mathematical models can never perfectly fit reality, thus it is important to be able to convey the constructed model and its parameters and in what ways it simplifies the data.

For example, in addition to variation in location BayesFlow explicitly models variation in cell population shape, whereas ASPIRE models shape variations implicitly by combining Gaussian components with the same shape. This means that an aberrant shape variation of a cell population in a sample can be detected in BayesFlow by examining the parameters of the model, which is not possible in ASPIRE. Perhaps more importantly, BayesFlow gives a parsimonious model which much fewer parameters—each individual parameter for the components in BayesFlow can be assessed through compact visualizations and thus undesired behaviors can be detected and corrected for by change of setup. Moreover, a restriction in ASPIRE which is avoided by BayesFlow is that the variation of component location within and between samples is connected to the shape of the components.

In BayesFlow, the cells in a sample are clustered using a multivariate Gaussian mixture model (GMM), where K components describe true and artificial cell populations and one component describes outliers. Artificial cell populations are measurements that cluster together and behave otherwise like real cell populations, but arise for example from dead cells, non-specific binding of markers or doublets; doublets are pairs or groups of cells that pass through the flow cytometer at the same time. Measurements which are not clearly grouped but spread out over the measurement space, for example due to measurement noise, are modeled as outliers.

For each component not representing outliers its mean and covariance matrix is linked to a latent cluster which collects corresponding components across all samples. In practice this is done by assuming a normal prior for the means and an inverse Wishart prior for the covariance matrices of the components linked to a given latent cluster. The parameters of sample and latent components are jointly estimated by Markov Chain Monte Carlo (MCMC) sampling. The variation in location and shape between corresponding mixture components across samples is controlled by the priors on parameters of the latent clusters. The location of component means and shape of components can also be restricted if there is prior information supporting this. To allow for that flow cytometry data frequently have missing cell populations, we include the possibility that not all components are present in every sample.

A challenge that has to be addressed when analyzing flow cytometry data is that cell populations can be skewed and/or have heavy tails and are then not well described by a single Gaussian component (Lo et al., 2008, Pyne et al., 2009, Frühwirth-Schnatter and Pyne, 2010). To handle this we use multiple components to model such populations, an approach that have often been employed for flow cytometry data (Finak et al., 2009, Chan et al., 2008, Baudry et al., 2010, Naim et al., 2014) and has the further advantage that the number of cell populations can be automatically detected. We merge Gaussian components into super components with a procedure based on a systematic study of methods for merging mixture components (Hennig, 2010).

Results from the MCMC sampling and subsequent merging are evaluated in a number of quality tests. This is a crucial step since what is deemed as a good clustering is application dependent. In some settings a given amount of variation in location or shape is expected from biological or technical reasons, whereas in others the same variation would indicate a different population. This also means that it is necessary for the user to choose prior parameters for their application. To simplify this process we have derived parametrizations so that the same value of the parameters gives a similar effect of the prior on data sets of different sizes.

We verified the ability of the sampling scheme to recover model parameters by fitting the model to a small three-dimensional synthetic data set with 1.2 million cells in total and a large synthetic data set with in total 28 million cells in 8 dimensions. Then we applied BayesFlow to one of the datasets in the FlowCAP I challenge, the GvHD dataset, which contains samples from patients who have had organ transplants and might have early signs of graft-versus-host disease. We show that BayesFlow does not only give a result which has the same degree of accordance with manual gating as the best performing methods in FlowCAP I—

which is much higher than what is obtained for other methods based on joint gating with Bayesian hierarchical models—it does also give a more similar treatment of different samples than manual gating and the best methods from FlowCAP I. Finally we applied BayesFlow, ASPIRE (Dundar et al., 2014) and HDPGMM (Cron et al., 2013) to a data set with replicated samples from four healthy individuals. The ratio between intra-donor technical variation and inter-donor biological variation was similar between BayesFlow and HDPGMM, which was lower than for ASPIRE. Moreover, BayesFlow was the only of the three methods which gave cell populations with clear expression patterns.

2 Methods

2.1 Model

Let $\mathbf{Y}_{ij}$ denote vector valued measurement number i in sample j. Here $i \in \{1, \ldots, n_j\}$, where n_j is the number of cells in sample j, and $j \in \{1, \ldots, J\}$, where J is the number of samples. We let the dimension of the observations be denoted d. With K mixture components describing cell populations the probability density for cell measurement i of a flow cytometry sample j is modeled as

$$f(\mathbf{Y}_{ij}) = \sum_{k=1}^{K} \pi_{jk} N(\mathbf{Y}_{ij}; \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}) + \pi_{j0} N(\mathbf{Y}_{ij}; \boldsymbol{\mu}_{j0}, \boldsymbol{\Sigma}_{j0}), \tag{1}$$

where $N(\mathbf{Y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the probability density function of the normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ evaluated at $\mathbf{Y}$. To facilitate interpretation, the number K should be chosen as small as possible, given that the model pass quality requirements (described under Quality control). The last component represents outliers and its parameters $\boldsymbol{\mu}_{j0} = \boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_{j0} = \boldsymbol{\Sigma}_0$ are identical across samples. Outliers are often modeled by a uniform density over the measurement space (Fraley and Raftery, 1998); however due to the curse of dimensionality (Lee and Verleysen, 2007), this is not well behaved when we have more than a few dimensions, in which case a Gaussian should perform better. Noise coming from for example dead cells can also be captured in artificial cell populations, and can be excluded in downstream analyses based on the expression patterns.

The vector $\boldsymbol{\pi}_j = \{\pi_{j0}, \ldots, \pi_{jK}\}$ contains the mixing proportions, i.e. the proportion of cells described by the component. To connect cell populations between samples we use a latent layer, assuming that for a given k each $\boldsymbol{\mu}_{jk}$ and $\boldsymbol{\Sigma}_{jk}$ is drawn from a normal and an inverse Wishart distribution respectively. Specifically, in our model, for $k = 1, \ldots, K$,

$$\boldsymbol{\mu}_{jk}|\boldsymbol{\theta}_k, \boldsymbol{\Sigma}_{\theta_k} \sim N(\boldsymbol{\theta}_k, \boldsymbol{\Sigma}_{\theta_k}), \quad \boldsymbol{\Sigma}_{jk}|\boldsymbol{\Psi}_k, \nu_k \sim IW(\boldsymbol{\Psi}_k, \nu_k) \tag{2}$$

where $\boldsymbol{\theta}_k$, $\boldsymbol{\Sigma}_{\theta_k}$, $\boldsymbol{\Psi}_k$ and ν_k are hyper-parameters describing latent cluster k. These parameters describes the variability between flow cytometry samples, in contrast to $\boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}$ which describe the distribution of cell measurements within a sample. The normal and inverse Wishart distributions are conjugate priors to the mean and the covariance respectively of the normal distribution, enabling efficient sampling, however they are not jointly conjugate.

We call $\boldsymbol{\theta}_k$ and $\boldsymbol{\Psi}_k/(\nu_k - d - 1)$ the latent cluster mean and latent cluster covariance matrix respectively, since they are the a priori expected values of $\boldsymbol{\mu}_{jk}$ and $\boldsymbol{\Sigma}_{jk}$.

For the hyper-parameters describing the latent clusters and the mixing proportions we use the following prior distributions:

$$\begin{aligned} \boldsymbol{\theta}_k|\mathbf{t}_k, \mathbf{S}_k &\sim N(\mathbf{t}_k, \mathbf{S}_k), & \boldsymbol{\pi}_j &\sim D(\mathbf{a}), \\ \boldsymbol{\Sigma}_{\theta_k}|\mathbf{Q}_k, n_{\theta_k} &\sim IW(\mathbf{Q}_k, n_{\theta_k}), & \nu_k|\lambda_k &\sim \exp(-\lambda_k), \\ \boldsymbol{\Psi}_k|\mathbf{H}_k, n_{\Psi_k} &\sim W(\mathbf{H}_k, n_{\Psi_k}), \end{aligned} \tag{3}$$

where W denotes the Wishart distribution and D denotes the Dirichlet distribution, which is conjugate prior to the multinomial distribution. For each ν_k we assign a exponential prior on the positive natural numbers. The complete structure of the model is displayed through a directed acyclic graph (DAG) in Fig. 1.

The parameters $\mathbf{t}_k$ and $\mathbf{S}_k$ define the prior belief of the locations of the latent means $\boldsymbol{\theta}_k$, whereas the parameters $\mathbf{Q}_k$ and n_{θ_k} control the spread of mixture component means within a latent cluster and are hence important to control the variation across samples. A large n_{θ_k} along with a small $\mathbf{Q}_k$ forces the $\boldsymbol{\mu}_{jk}$ together; it makes large deviations between $\boldsymbol{\Sigma}_{\theta_k}$ and $\mathbf{Q}_k$ unlikely. The parameters $\mathbf{H}_k$ and n_{Ψ_k} control the expected values and the variation of latent covariance matrices as well as the variation among mixture component covariance matrices in a latent cluster. If n_{Ψ_k} is large each $\boldsymbol{\Sigma}_{jk}$ will be close to $\boldsymbol{\Psi}_k/(\nu_k - d - 1)$ for any k, since a high n_{Ψ_k} makes high ν_k more probable.

Finally, to simplify sampling from the posterior distribution of the parameters, we add an component assignment variable $x_{ij} \in \{0, 1, \ldots, K\}$ describing

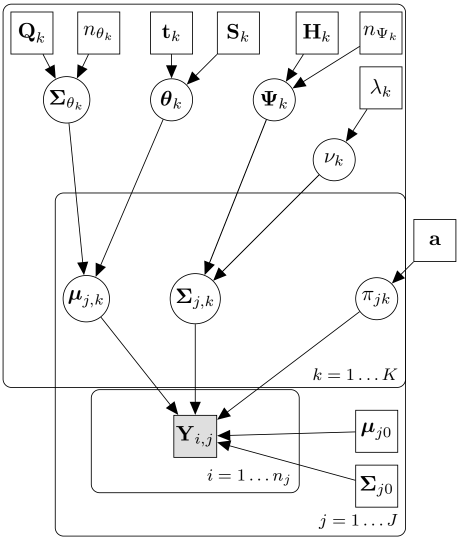


Figure 1: Directed acyclic graph describing the Bayesian hierarchical model. Square boxes indicate that the values are known.

which component $\mathbf{Y}_{ij}$ is drawn from. To comply with (1), the a priori uncertainty of component membership is modeled by $x_{ij} \sim Mult(\boldsymbol{\pi}_j, 1)$, where $Mult$ denotes the multinomial distribution.

The resulting posterior distribution of all the parameters, denoted jointly by $\boldsymbol{\Theta}$, and $\mathbf{x}$ given the data $\mathbf{Y}$ is given in the Supplementary material, Section A. In Section B we describe the Markov chain Monte Carlo (MCMC) sampling scheme used to generate posteriors for our model parameters.

The computational bottleneck of the sampling scheme is the sampling of $\mathbf{x}$, with a computational complexity bounded by $\mathcal{O}(Jd^3K \max_j n_j)$. To handle high dimensions diagonal covariance matrices can be used instead, in which case the complexity is bounded by $\mathcal{O}(JdK \max_j n_j)$. However, for datasets with more than 20 dimensions the mathematical feasibility of using Gaussian mixture models without any prior dimension reduction needs to be seriously considered first, due to the curse of dimensionality (Lee and Verleysen, 2007).

Absent components

In some flow cytometry data sets not all cell populations are present in all samples. In our model this corresponds to that $\pi_{jk} = 0$ for some (j, k). However, mixture component parameters for empty clusters will still affect the mixing of the MCMC for the parameters of the latent cluster. It can also happen that if a cluster is empty that the mixture component moves and split a neighboring cluster in two. To avoid this in such data sets we extend the model by introducing a variable $\mathbf{Z}_j \in \{0,1\}^K$ that says which components are active in sample j. This has the further advantage that when sampling from the posterior distribution of the model we get the probability for each cluster that it is present in a sample. We impose a prior on $\mathbf{Z}_j$ which is proportional to $\exp(-c_s \sum_{k=1}^K \mathbf{Z}_j) I(\sum_{k=1}^K \mathbf{Z}_j > 0)$ where I denotes the indicator function and $c_s > 0$. The prior makes the model prefer fewer activated clusters so that if there is a very small cluster the likelihood will be larger if it is inactivated, which prevents spurious clusters. The strength of this prior can be adjusted to the expected size of the smallest clusters.

The changes to (1)–(3) required by this extension are straightforward but inference of the model becomes a bit more involved since removing components reduces the dimension of the model. To accommodate for this we have included a reversible jump step in our sampling algorithm. Details are given in the Supplementary Material, Section B.

2.2 Merging latent clusters

To determine the "correct" number of clusters in a data set directly from the data is an ill-defined problem, since what should be considered to be a separate cluster depends on the interpretation of the data. Nevertheless, there are many different criteria which can be used to guide the decision about the number of populations (Frühwirth-Schnatter, 2006, Hennig, 2010). We use overlap between components—measured by Bhattacharyya distance—and unimodality of the resulting super clusters—measured by Hartigan's dip test (Hartigan and Hartigan, 1985)—to determine which latent clusters to merge and to indicate our confidence in the mergers.

In an evaluation of criteria for merging Gaussian components to represent more complex distributions, the Bhattacharyya distance performed well (Hennig, 2010). Bhattacharyya distance merges clusters according to a pattern-based cluster concept as opposed to a modality-based concept (Hennig, 2010). With a pattern-based cluster concept a small dense cluster inside a sparse cluster—for example a well specified cell population inside a region with sparse outliers—will be considered to be different clusters. This would not be the case for the modality-based cluster concept as long as the generating probability density is unimodal.

The Bhattacharyya distance between $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ is

$$d_{\text{bhat}} = 1/8 \cdot (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \bar{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + 1/2 \cdot \log\left(|\bar{\boldsymbol{\Sigma}}| / \sqrt{|\boldsymbol{\Sigma}_2||\boldsymbol{\Sigma}_2|} \right), \quad (4)$$

where $\bar{\boldsymbol{\Sigma}} = (\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)/2$ (Fukunaga, 1990). In order to measure Bhattacharyya distance between mixtures of Gaussian distributions, which is necessary for deciding if super clusters should be merged with other clusters, we approximate each mixture with a Gaussian distribution. The means and the covariance matrices are estimated using a soft clustering of the data points inferred from the sampling of x_{ij}, detailed in the Supplementary material, Section C.

However, it is not obvious how to set a threshold for d_{bhat}, since the appropriate threshold depends on the distribution of the data (Hennig, 2010), which is unknown. Because of this we use a low soft threshold d_1 and a high hard threshold d_2. Two clusters closer to each other than d_1 are always merged, two clusters whose distance is between d_1 and d_2 are only merged if they fulfill an additional criterion based on Hartigan's dip test for unimodality.

Unimodality is an appealing heuristic for defining cell populations, and it has frequently been used for automated gating (Chan et al., 2008, Ge and Sealfon, 2012, Naim et al., 2014). It has two main limitations. The first one, that populations intuitively should be separate if they have very different densities—even when they overlap so that their combined distribution is unimodal—can be bypassed by combining unimodality with a pattern-based merging criterion such as Bhattacharyya distance. The second one, that it is difficult to determine if a multi-dimensional empirical distribution is multimodal, is usually handled by considering one-dimensional projections (Hennig, 2010, Naim et al., 2014). This is the approach we take here, using Hartigan's dip test of unimodality for each of the projections onto the coordinate axes where Bhattacharyya overlap is low, and for the projection onto Fisher's discriminant coordinate. If for a proposed merger, any of these projections is found to be multimodal, the clusters are not merged. Further details of the merging procedure are given in the Supplementary material, Section C.

2.3 Quality control

To verify that the output of BayesFlow fulfills the user's requirements, a number of checks are performed:

- Convergence of the MCMC sampler is established by viewing trace plots of sampled parameters, such as in Fig. S1.

- To ensure that variation of the two different populations are not confused with each other, we require that the Bhattacharyya distance as well as the Euclidean distance from each sample component to its corresponding latent component should be smaller than these distances to any other latent component which does not belong to the same super cluster.

- To ensure that the obtained clusters should not be divided further, Hartigan's dip test is computed for the projections onto the coordinate axes of all super clusters. Projections which have a dip test p-value below 0.28—the threshold for merging components (see Supplemental Material Section C)—are visualized using histograms of quantiles of the weighted data belonging to the cluster, as in Fig. S3 in the Supplemental material.

- To ensure that the model fits the data reasonably well, samples from the posterior predictive is compared to the true data in one- and two-dimensional histograms such as Fig. 3, Fig. 4 and Fig. 12.

- To ensure that there are no outliers among the cluster centers, the centers for each cluster are plotted together along one dimension, such as in Fig. 13 (a).

- Additionally, to detect components with aberrant shapes, the eigenvectors corresponding to the largest eigenvalues, multiplied with the corresponding eigenvalues, can be viewed as in Fig. S4 in the Supplemental Material.

If any of the quality criteria is not met, the simulation should be rerun, either using the same or different parameters. Even if the same parameters are used a different result can be obtained due to randomness in the initialization.

2.4 Experiments

2.4.1 Simulated data

In order to verify that the proposed sampling scheme can find the correct model parameters, the MCMC algorithm was applied to two simulated datasets. The first dataset was three-dimensional, which enables direct visual evaluation. It had four latent clusters across eighty artificial flow cytometry samples; each sample had 15,000 cells giving a total of 1.2 million cells. One of the latent clusters was present only in eight samples and another one was present in 24 samples, so that the ability to find rare cell populations was tested. Moreover, the cluster which was present in only eight samples contained only 1% of the total number of cells, thus also the ability to find small cell populations was tested. The parameters and the algorithm used for generating the data are given in the Supplementary material, Section D.1.

The second data set was designed to test the ability to handle large data. It was eight-dimensional, with eleven latent clusters and 192 artificial flow cytometry samples. Each sample had measurements of 150,000 cells, giving a total of 28 million cells. Four of the eleven clusters were missing in half of the samples.

Prior parameters and initial values for the MCMC sampler are given in the Supplementary material, Section D.1. All priors were chosen to be non-informative. The outlier component was not used for inference in the small dataset, but it was used for the large dataset. The MCMC sampler ran first for a number of burn-in iterations, then the posterior distribution was explored in a number of production iterations. During the production iterations, apart from sampling parameters of the model, a value of $\mathbf{Y}$ was also drawn, i.e. a sample from the posterior predictive. For the first synthetic data set 10,000 burn-in and 100,000 production iterations were used. For the second, larger, data set we used 5000 burn-in iterations and 5000 production iterations.

For the second data set the MCMC sampler was run on Amazon Cloud, using 192 cores. Each iteration took on average one second, so that about 2.7 hours was needed in total.

2.4.2 Flow cytometry data

We analyze two flow cytometry data sets with BayesFlow: the data set GvHD from the FlowCAP I challenge—with four markers, 12 samples and approximately 13,000 cells per sample—and a data set obtained from the R package

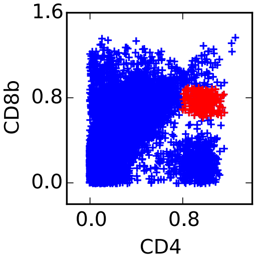


Figure 2: Cell population which is hard to detect.

healthyFlowData (Azad, 2013) with technical replicates of PBMC samples from healthy donors—in total 20 samples with approximately 20,000 cells, also measured with four markers. In the GvHD dataset we can compare the gating obtained from BayesFlow with manual gating provided from FlowCAP as well as automated gating from a wide range of other methods. In the healthyFlowData we can instead compare gating between technical replicates to see if samples are treated in a consistent manner.

For the healtyFlowData dataset we used an exploratory approach with non-informative priors. We ran multiple simulations and gradually increased the number of components until we passed the quality criteria described under Quality control; we finally arrived at using $K = 25$ components. For the GvHD data set we started with an exploratory approach and gradually increased the number of components, but in the quality checks we noted one population in one of the samples which was very hard to capture. Then we decided to use an informative approach for this population. Using a scatter plot, Fig. 2, we set boundaries for this population in the dimensions given by the CD4 and the CD8b marker and computed its mean and empirical covariance matrix. We used the mean to set an informative prior for $\boldsymbol{\theta}_k$ and the mean and the empirical covariance to initialize the component. Prior parameters in both the informative and non-informative case are described in the Supplemental Material, Section E.2.

BayesFlow applies three data preprocessing steps: 1) Data points with extreme values in at least one dimension (larger than 0.999 times the largest data point or smaller than 1.001 times the smallest data point) are removed. Such data points can lead to components with singular covariance matrices, and a well designed flow cytometry experiment should not have significant populations with such values. 2) The data is scaled using the 1% and 99% percentiles $q_{0.01}$ and $q_{0.99}$ of the pooled data, with the same scaling for all samples, so that $q_{0.01} = 0$ and $q_{0.99} = 1$ for each marker for the pooled data. This is done in order to be able to set informative priors in an intuitive way. 3) Before testing which components should be merged, a very small amount of noise is added to the data (standard deviation 0.003). This is since the discreteness of the original flow cytometry measurements can lead to a striped pattern in the flow cytometry data (Roederer, 2001) and also when it is not visible to the human eye it disturbs the dip test.

After preprocessing, parameters for the MCMC sampler were initialized by running the EM algorithm on the pooled data, followed by the initialization scheme used for the large synthetic dataset, detailed in the Supplemental Material, Section D.4. We ran 16,000 burn-in iterations and 4000 production iterations of the MCMC sampler for both experiments. The burn-in period consisted of five phases: In the first phase, the priors on variation in location and shape were modified to force clusters together. Before the second phase, priors parameters were set to normal again. After the second phase, components which were considered to be outliers were turned off. They were forced to stay off during a short third phase, but from the forth phase and onwards components were allowed to turn on and off. Label switching was allowed during the initial four phases in order to escape non-desired local minima, but then disallowed. The values of parameters controlling the simulation during the burn-in and production period are given in Table S1 in the Supplemental Material.

We also applied the two other joint gating methods based on Bayesian hierarchical models: ASPIRE (Dundar et al., 2014) and HDPGMM (Cron et al., 2013). For ASPIRE parameters were chosen according to the strategy recommended by Dundar et al. (Dundar et al., 2014); details are given in the Supplementary Material, Section E.5. For each run we used in total 15,000 iterations, of which 14,000 were set as burn in iterations. For HDPGMM default parameters were used, with a burn-in period of 3000 iterations and a production period of 100 iterations.

We ran BayesFlow and ASPIRE on a 3.2 GHz quad core CPU. A BayesFlow run took 0.5 h for the GvHD dataset and 1.4 h for the healthyFlowData dataset. ASPIRE took in total 2.4 h for the GvHD dataset and 6.6 h for the healthyFlowData dataset per run. Four runs of ASPIRE was needed to determine the κ_i parameters. HDPGMM was run on a dual core GPU. It needed 0.72 h for the GvHD dataset and approximately 1 h for the healthyFlowData dataset.

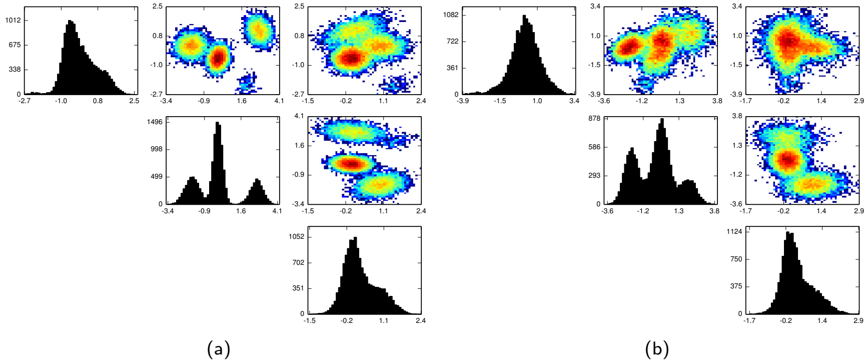


Figure 3: (a) One and two dimensional histograms for one synthetic flow cytometry sample containing 15,000 data points; (b) histograms of 15,000 data points drawn uniformly from the pooled data from the synthetic data experiment.

3 Results

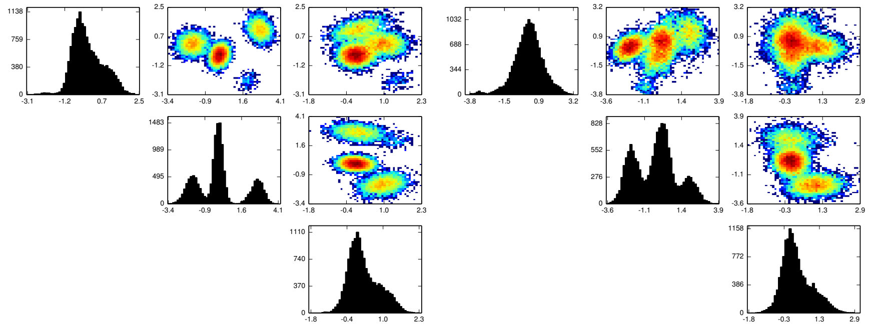
3.1 Simulated data

We begin by analyzing the smaller data set. In Fig. 3 we show univariate and bivariate histograms of all synthetic cell measurements pooled together, as well as the corresponding histograms of the data from a single flow cytometry sample where all four clusters are present. Note that the data when pooled together has a complicated density, as it is in fact a mixture of 232 multivariate normal densities.

In Fig. 4 we show the same univariate and bivariate histograms, but this time with samples from the posterior predictive distribution of $\mathbf{Y}$. From the synthetic cell measurements generated from the inferred models of the datasets it is clear that the inferred models are accurate and capture the variation across samples, which a model only of pooled data cannot do.

Fig. 5 displays dots at the posterior mean locations of the mixture component centers $\boldsymbol{\mu}_{jk}$ whose posterior probability of being active is greater than 1%; the true locations of the active clusters are displayed as circles. The model is able to detect which clusters that are active and which are not, and to find the location of the component means.

Finally in Fig. 7 and Fig. 9, the marginal posterior distributions of the latent cluster parameters $\boldsymbol{\theta}_k$ and $\boldsymbol{\Psi}_k$, subtracted by their true values, are presented. In Fig. 7 the dot represents the difference between the median of posterior dis-



(a)

(b)

Figure 4: (a) One and two dimensional histograms of 15,000 posterior draws of $\mathbf{Y}$ for the flow cytometry sample displayed in Fig. 3 (a); (b) histograms of 15,000 posterior draws of $\mathbf{Y}$ drawn uniformly from all the flow cytometry samples, thus matching Fig. 3 (b).

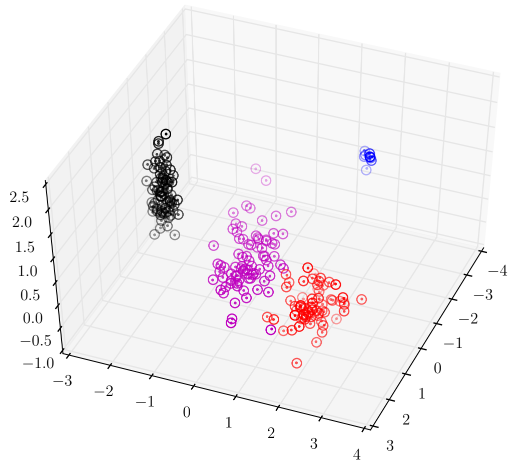


Figure 5: The posterior mean of the mixture component centers, $\boldsymbol{\mu}_{jk}$ (dots), and the true cluster centers (circles) in the small synthetic data experiment.

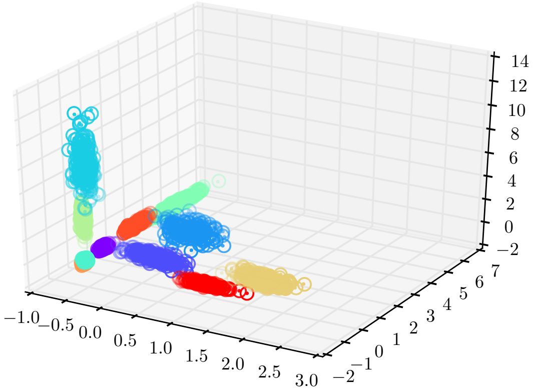


Figure 6: The posterior mean of the mixture component centers, $\boldsymbol{\mu}_{jk}$ (dots), and the true cluster centers (circles) in the large synthetic data experiment for the first three dimensions.

tribution and the true value of each $\boldsymbol{\theta}_k$. The vertical lines represent the 2.5% and 97.5% quantiles. Fig. 9 displays results for each latent covariance matrix $\boldsymbol{\Psi}_k/(\nu_k - 4)$ in the same way. From Fig. 7 and Fig. 9 we see that the true parameters of both the means and the covariances are all between the 2.5% and 97.5% quantiles of the posterior distribution.

The true and estimated cluster centers of the 8-dimensional data set cannot be displayed efficiently with just three dimensions at hand, but a 3-dimensional projection is shown in Fig. 6. The average error in Euclidean distance in the full

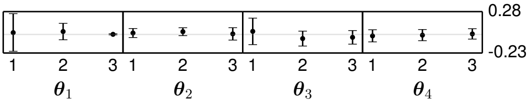


Figure 7: The difference between the true value of each entry in each $\boldsymbol{\theta}_k$ and the approximated marginal posterior distribution generated by the MCMC sampler in the small synthetic data experiment. The black dot represents the median and the vertical line goes between the 2.5% and 97.5% quantiles. The light gray horizontal line is the 0 line.

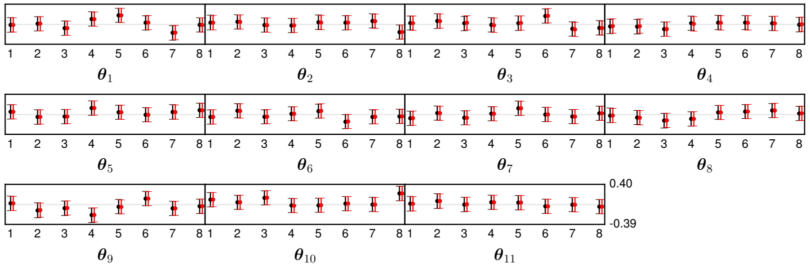
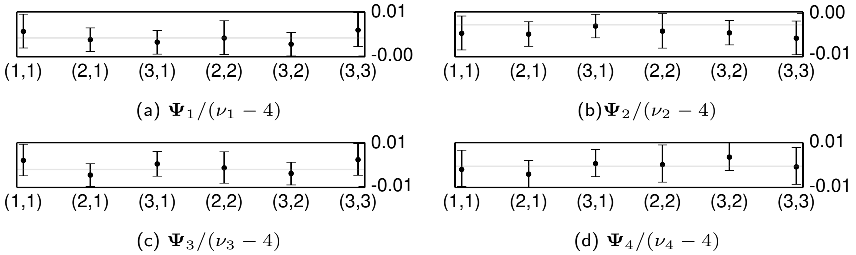


Figure 8: The difference between the true value of each entry in each $\boldsymbol{\theta}_k$ and the approximated marginal posterior distribution generated by the MCMC sampler in the large synthetic data experiment. The black dot represents the median and the vertical line goes between the 2.5% and 97.5% quantiles. To get the axis on the same scale for all the clusters, they are scaled by the standard deviation of $\boldsymbol{\mu}_k$. The light gray horizontal line is the 0 line. The red dot and lines is the same however where one uses the true $\boldsymbol{\mu}_k$ to estimate $\boldsymbol{\theta}_k$, rather then the $\boldsymbol{\mu}_k$ obtained by taking the posterior means of the mixtures.



(a) $\boldsymbol{\Psi}_1/(\nu_1 - 4)$

(b)$\boldsymbol{\Psi}_2/(\nu_2 - 4)$

(c) $\boldsymbol{\Psi}_3/(\nu_3 - 4)$

(d) $\boldsymbol{\Psi}_4/(\nu_4 - 4)$

Figure 9: The difference between the true value of each of the entries in $\boldsymbol{\Psi}_k/(\nu_k - 4)$ and the approximated marginal posterior distribution generated by the MCMC sampler in the synthetic data experiment. The black dot shows the median, and the black vertical line goes between the 2.5% and 97.5% quantiles. The light gray horizontal line is the 0 line.

8-dimensional space is 0.007, which can be compared to the average error had the latent mean across samples been used, namely 0.110, which is the best that could have been obtained from a model not including variation between samples. The outlier component was used for inference in the results presented here, but omitting it has very small effect.

In Figure 8, we show the posterior distribution of the latent cluster means where again the dot represents the difference between the median of posterior distribution and the true value of each $\boldsymbol{\theta}_k$. The vertical lines are the 2.5% and 97.5% quantiles. The posterior samples have been divided by the standard deviation of the true $\boldsymbol{\theta}_k$ so that the scales across the clusters are equal. Some of the credibility intervals do not contain zero, but this is explained when studying the intervals that would have been obtained if the true μ_k were used (shown in red), since they are almost identical.

We thus see that cluster centers and credibility intervals for latent clusters are captured well in both synthetic data sets.

3.2 Flow cytometry data

3.2.1 GvHD

For the analysis of the GvHD dataset we did twelve runs of BayesFlow in the informed setup described above. Seven were excluded due to confusion between populations, i.e. at least one sample component was closest to the wrong latent component; of the remaining five, one more run was excluded since it has not converged, and another two because of multimodal clusters. This leaved two runs that passed the quality control.

Table 1 reports the accordance with manual gating for the two BayesFlow runs as well as what is obtained from ASPIRE and HDPGMM, as well as the top two performing methods for this data set in FlowCAP: flowMeans and SamSPECTRAL.

One of the two BayesFlow runs has the highest accordance with manual gating, the other one is on par with flowMeans and SamSPECTRAL, which is considerably higher than ASPIRE and HDPGMM. However, as can be seen in Fig. 10, the gating of different samples is arguably most consistent for BayesFlow as compared to manual gating, flowMeans and SamSPECTRAL.

To get a further understanding of the variability between samples in BayesFlow, summary statistics for the obtained components and cell populations are shown

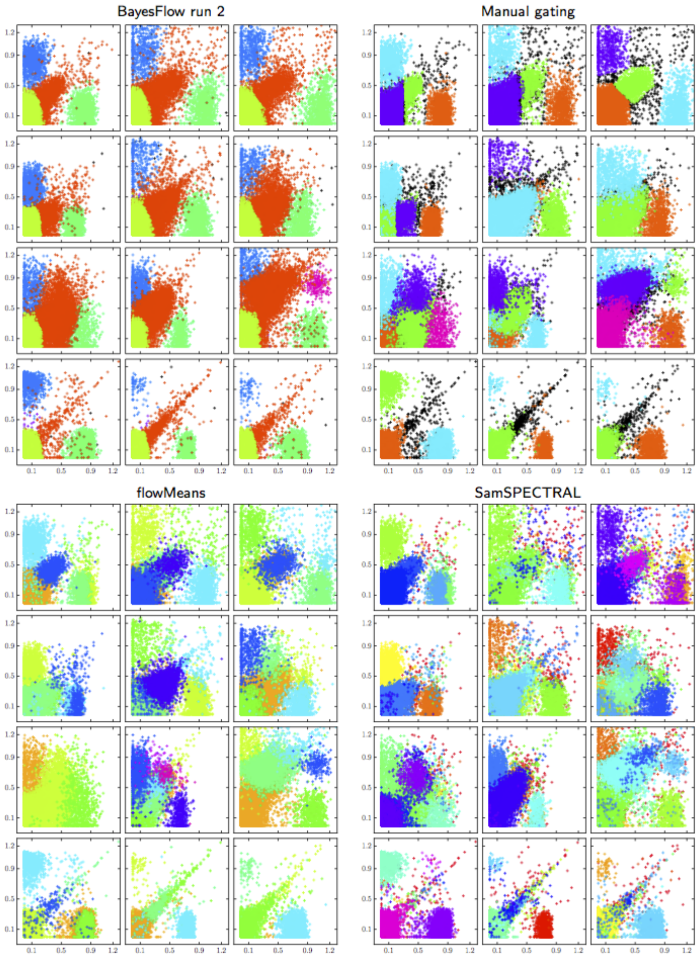


Figure 10: Gated events according to four methods (BayesFlow, manual and the two top performers in FlowCAP I) of the twelve samples in the GvHD dataset, projected onto the two first dimensions. For BayesFlow, the run with least accordance with manual gating, run 2, is shown. Similar plots for ASPIRE and HDPGMM as well as BayesFlow run 1 are shown in the Supplemental Material, Fig. S6.

Table 1: Accordance with manual gating for GvHD data set. For HDPGMM we also report the result when components are merged according to our merging procedure. When this procedure is applied to the results obtained by ASPIRE, no components are merged, i.e. the original result is identical to what is obtained after merging.

Method	F-measure	Precision	Recall
BayesFlow run 1	0.91 (0.86, 0.95)	0.96	0.89
BayesFlow run 2	0.87 (0.82, 0.92)	0.95	0.84
ASPIRE	0.67 (0.63, 0.72)	0.86	0.63
HDPGMM	0.35 (0.30, 0.39)	0.98	0.23
HDPGMM merged	0.60 (0.54, 0.66)	0.95	0.48
FlowMeans	*0.88 (0.82, 0.93)*	*0.93*	*0.86*
SamSpectral	*0.87 (0.81, 0.93)*	*0.96*	*0.83*
Ensemble FlowCAP	*0.88*		

in Fig. 11.

3.2.2 healthyFlowData

We did 18 runs of BayesFlow with $K = 25$. Ten of these were excluded due to confusion between populations, moreover two runs were excluded since they had clusters with clearly multimodal distributions. For the six runs that passed the quality control, 3-6 components were turned off across all samples; they are excluded from visualizations.

In Fig. 12 we visualize model fit and inter-sample variation for the first of the six runs that passed the quality control by plotting latent and sample components as well as histograms of real data and synthetic data generated from the model, for two different samples and for the pooled data. We can thus see how shape variations are captured by the model.

The output of BayesFlow, ASPIRE and HDPGMM can be compared in Fig. 13. The merging procedure we used for BayesFlow has been applied for both ASPIRE and HDPGMM, however for ASPIRE no components were merged by this. In BayesFlow each of the populations correspond to clear expression patterns, which is not the case for the other methods. For example the first population is clearly CD4+CD8- T-cells whereas for both ASPIRE and HDPGMM this

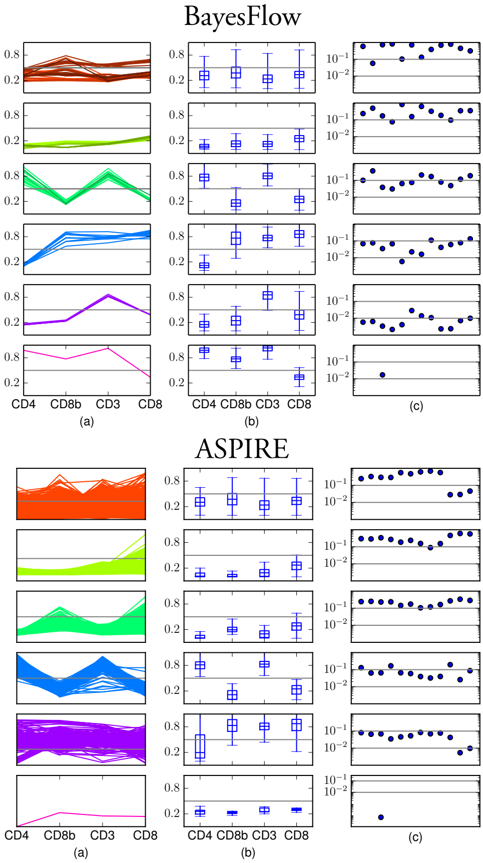


Figure 11: Summary statistics of the six cell populations obtained by BayesFlow (run 2) in the dataset GvHD. The outlier component has at most 0.0019 of the cells in a sample. (a) Each panel displays the locations $\boldsymbol{\mu}_{jk}$ of all mixture components that represent the population, across all samples. Different shades of a color represent different latent components k. (b) Box plots of the soft clusters in the pooled data. The boxes go between the quantiles $q_{km,0.25}$ and $q_{km,0.75}$, the whiskers extend to $q_{km,0.01}$ and $q_{km,0.99}$. The α-quantile for (merged) component k in dimension m, $q_{km,\alpha}$, is here defined as $q_{km,\alpha} = \min_{i'j'}\{Y_{i'j'm} : \alpha < \sum_{ij:Y_{ijm}<Y_{i'j'm}} w_{ijk}\}$. (c) Population proportions in each of the twelve flow cytometry samples.

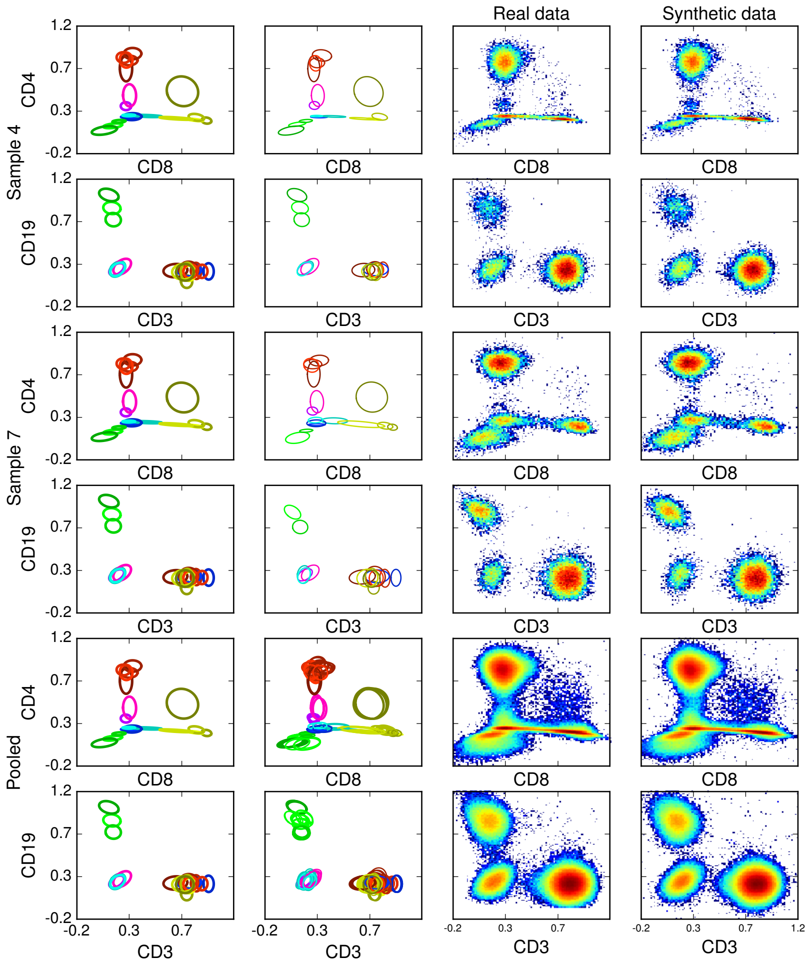


Figure 12: BayesFlow component parameter representations of inferred latent clusters (first column) and mixture components (second column) together with histograms of real data (third column) and synthetic data generated from the model (fourth column) for the healthyFlowData. The center of each ellipse is the mean and each semi-axis is an eigenvector with length given by the corresponding eigenvalue of the projected covariance matrix. For the latent clusters the parameters $(\boldsymbol{\theta}_k, \frac{1}{(\nu_k - d - 1)}\boldsymbol{\Psi}_k)$ are shown, for the mixture components the parameters $(\boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk})$ are shown. Each component or cluster is depicted with the same color as in Fig. 13; different shades of same color corresponds to latent clusters that have been merged.

population contains both components which are CD8- and components which are CD8+.

We also compare intra-donor variation of cell population size to inter-donor variation for the six BayesFlow runs, as well as for ASPIRE and HDPGMM in Fig. 14. For ASPIRE there are inter-donor distances which are clearly smaller than some intra-donor distances, which is not the case for BayesFlow and HDPGMM.

4 Discussion

From different runs of BayesFlow we can get different representations of data, as in the case of the GvHD dataset. This is because with highly overlapping populations there might be multiple models representing the data equally well. But since all samples are gated jointly in every run, the gated populations can still be compared across samples. The user might have a preference for one representation or the other though, and informative priors can be used to guide BayesFlow to a preferred representation.

BayesFlow is not aimed at discovery of rare cell populations, but it can be used together with an algorithm specifically designed for detecting rare cell populations in a sample, such as SWIFT (Naim et al., 2014), and then use informative priors to find how this population occurs across an entire set of samples, in a similar way as was done in the GvHD dataset.

How much clusters should be merged is a decision that needs to be taken by the interpreter of the data. In some settings one might want to be restrictive with merging and then use higher thresholds. In others one might want additional mergers after viewing joint 1-dimensional projections of the clusters.

The BayesFlow pipeline does not in itself include any compensation or any of the non-linear transformations which are often used for flow cytometry data, such as logicle. Compensation is a linear transformation and Gaussian Mixture Models are invariant under linear transformations, so they perform equally well on uncompensated and compensated data. Non-linear transformations such as logicle can make Gaussian populations non-Gaussian, which makes inference harder. The flow cytometry data we used for the experiments had already been compensated, the healthyFlowData data set had also been transformed with an asinh transform; details are given in the Supplemental Material, Section E.1.

BayesFlow finds a joint representation of an entire set of samples. In order for this representation to be reasonable there has to be sufficient correspondences be-

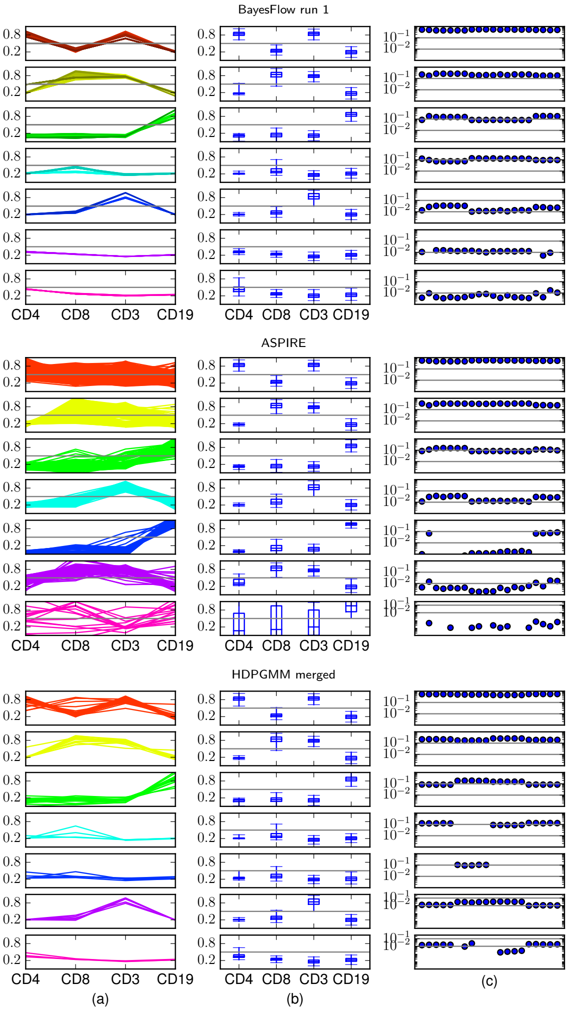


Figure 13: Summary statistics of inferred cell populations in BayesFlow, ASPIRE and HDPGMM, ordered by population size. For HDPGMM, the six largest components after merging are shown, the remaining components have together at most 0.0013 of the cells in a sample. The noise component in BayesFlow has at most 0.004 of the cells in a sample. (a) Locations $\boldsymbol{\mu}_{jk}$ of mixture components that represent the each population, in each samples, cf. Fig. 11. (b) Box plots of the soft clusters in the pooled data cf. Fig. 11. (c) Population proportions across flow cytometry samples.

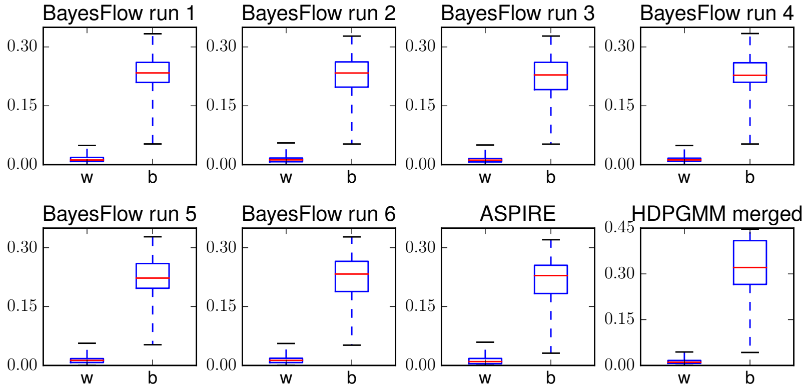


Figure 14: Distances within (w) and between (b) donors as measured by ℓ_1 distance between vectors of population sizes. For the six BayesFlow runs and HDPGMM there is very little or no overlap between within-donor and between-donor distances, whereas for ASPIRE there is clear overlap.

tween samples. Even if for a data set with very little correspondences a joint model could be obtained by using a very large number of components, it would hard to gain any insights from such a model. In such a case an entirely computational pipeline without the cell population identification step would be preferred.

BayesFlow can be computationally intensive if many runs are needed to pass the quality control. For these cases it would be desirable to complement BayesFlow e.g. with initialization methods that would allow passing the quality control more often, so that few runs in BayesFlow would be needed. Fast initialization methods and early quality checks aiming at this would therefore be of interest for the community and is something that we propose for further study.

5 Conclusions

In this paper we have presented a new Bayesian hierarchical model designed for joint cell population identification in many flow cytometry samples. The model captures the variability in shapes and locations of the populations between the samples and we have demonstrated its use in an exploratory as well as in a partly informed setting with some prior information. We showed that for synthetic datasets generated from the model, the parameters were recovered with high accuracy through a MCMC sampling scheme. The model was then applied to a real flow cytometry data set where a manual gating was available, and it was shown to have very high accordance with manual gating as compared to other automated gating methods, while at the same time the gating was more consistent across samples than either the manual gating or other automated gating methods. When applied to another flow cytometry data set with technical replicates of blood from healthy donors, BayesFlow gave a parsimonious representation of the data, which enables visualization and monitoring of its parameters. The obtained cell populations had clear expression patterns as opposed to the clusters obtained by ASPIRE and HDPGMM, where for example $CD4+CD8-$ T-cells where in the same cluster as $CD4+CD8+$ T-cells. The population sizes obtained by BayesFlow and HDPGMM respectively had lower intra-donor variation compared to inter-donor variation than what was obtained from ASPIRE.

Many approaches of automated gating of multiple flow cytometry samples in parallel have been aimed at finding features of the data so that either samples can be classified into groups, e.g. cancer or normal, or they can be used to predict an outcome such as expected time to progression of disease. Features are often designed based on characteristics of cell populations, but usually not so much attention has been given to ensure that they represent actual cell populations. BayesFlow takes the opposite approach and gives a representation of the data according to cell populations, with the same cell populations across the entire set of samples (except when some populations only occurs in a subset of the samples). The advantages to this approach are among others that the result is directly biologically interpretable and that a rich output is given which can be explored in many different ways which are familiar to someone who is used to manual gating. In this way we can join the objectivity and ability to work in high dimensions and with many samples of automated gating with the flexibility in interpretation of manual gating.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

KJ, JW and MF conceived and planned the study. JW and KJ designed the statistical model and the inference procedure. KJ, JW and MF designed the experiments. JW and KJ implemented BayesFlow and ran the experiments. KJ and JW wrote the article with the help of MF. All authors read and approved the final version of the manuscript.

Acknowledgements

We would like to thank Bartek Rajwa and Ariful Azad for collecting and sharing the healthyFlowData data set and for providing information about the preprocessing of the data.

The first author gratefully acknowledges a travel grant from the Royal Swedish Academy of Sciences' foundations. The second author is supported by Knut and Alice Wallenbergs stiftelse.

Bibliography

N. Aghaeepour, R. Nikolic, H. H. Hoos, and R. R. Brinkman. Rapid cell population identification in flow cytometry data. *Cytometry Part A*, 79(1):6–13, 2011.

N. Aghaeepour, G. Finak, The FlowCAP Consortium, The DREAM Consortium, H. Hoos, T. R. Mosmann, R. Brinkman, R. Gottardo, and R. H. Scheuermann. Critical assessment of automated flow cytometry data analysis techniques. *Nature Methods*, 10(3):228–238, 2013.

A. Azad. *healthyFlowData: Healthy dataset used by the flowMatch package*, 2013. R package version 1.2.0.

A. Azad. *flowVS: Variance stabilization in flow cytometry (and microarrays)*, 2015. R package version 1.1.0.

A. Azad, A. Khan, B. Rajwa, S. Pyne, and A. Pothen. Classifying immunophenotypes with templates from flow cytometry. In *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*, page 256. ACM, 2013.

J.-P. Baudry, A. E. Raftery, G. Celeux, K. Lo, and R. Gottardo. Combining mixture components for clustering. *Journal of Computational and Graphical Statistics*, 19(2), 2010.

M. J. Boedigheimer and J. Ferbas. Mixture modeling approach to flow cytometry data. *Cytometry Part A*, 73(5):421–429, 2008.

R. V. Bruggner, B. Bodenmiller, D. L. Dill, R. J. Tibshirani, and G. P. Nolan. Automated identification of stratifying signatures in cellular subpopulations. *Proceedings of the National Academy of Sciences*, 111(26):E2770–E2777, 2014.

C. Chan, F. Feng, J. Ottinger, D. Foster, M. West, and T. B. Kepler. Statistical mixture modeling for cell subtype identification in flow cytometry. *Cytometry Part A*, 73(8):693–701, 2008.

X. Chen, M. Hasan, V. Libri, A. Urrutia, B. Beitz, V. Rouilly, D. Duffy, É. Patin, B. Chalmond, L. Rogge, L. Quintana-Murci, and M. L. Albert. Automated flow cytometric analysis across large numbers of samples and cell types. *Clinical Immunology*, 2015.

A. Cron, C. Gouttefangeas, J. Frelinger, L. Lin, S. K. Singh, C. M. Britten, M. J. Welters, S. H. van der Burg, M. West, and C. Chan. Hierarchical modeling for rare event detection and cell subset alignment across flow cytometry samples. *PLoS Computational Biology*, 9(7):e1003130, 2013.

M. Dundar, F. Akova, H. Z. Yerebakan, and B. Rajwa. A non-parametric Bayesian model for joint cell clustering and cluster matching: Identification of anomalous sample phenotypes with random effects. *BMC Bioinformatics*, 15:314, 2014.

G. Finak, A. Bashashati, R. Brinkman, and R. Gottardo. Merging mixture components for cell population identification in flow cytometry. *Advances in Bioinformatics*, 2009.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

C. Fraley and A. E. Raftery. How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

S. Frühwirth-Schnatter. *Finite Mixture and Markov Switching Models: Modeling and Applications to Random Processes*. Springer, New York, 2006. Chapter 4.

S. Frühwirth-Schnatter and S. Pyne. Bayesian inference for finite mixtures of univariate and multivariate skew-normal and skew-t distributions. *Biostatistics*, 11(2):317–336, 2010. doi: 10.1093/biostatistics/kxp062. URL `http://biostatistics.oxfordjournals.org/content/11/2/317.abstract`.

K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic press, San Diego, 1990.

Y. Ge and S. C. Sealfon. flowPeaks: a fast unsupervised clustering for flow cytometry data via k-means and density peak finding. *Bioinformatics*, 28(15): 2052–2058, 2012.

P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

F. Hahne, A. H. Khodabakhshi, A. Bashashati, C.-J. Wong, R. D. Gascoyne, A. P. Weng, V. Seyfert-Margolis, K. Bourcier, A. Asare, T. Lumley, et al. Per-channel basis normalization methods for flow cytometry data. *Cytometry Part A*, 77(2): 121–131, 2010.

J. A. Hartigan and P. M. Hartigan. The dip test of unimodality. *The Annals of Statistics*, pages 70–84, 1985.

C. Hennig. Methods for merging Gaussian mixture components. *Advances in Data Analysis and Classification*, 4(1):3–34, 2010.

X. Hu, H. Kim, P. J. Brennan, B. Han, C. M. Baecher-Allan, P. L. De Jager, M. B. Brenner, and S. Raychaudhuri. Application of user-guided automated cytometric data analysis to large-scale immunoprofiling of invariant natural killer T cells. *Proceedings of the National Academy of Sciences*, 110(47):19030–19035, 2013.

J. A. Lee and M. Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, New York, 2007.

K. Lo, R. R. Brinkman, and R. Gottardo. Automated gating of flow cytometry data via robust model-based clustering. *Cytometry Part A*, 73(4):321–332, 2008.

I. Naim, S. Datta, J. Rebhahn, J. S. Cavenaugh, T. R. Mosmann, and G. Sharma. SWIFT—scalable clustering for automated identification of rare cell populations in large, high-dimensional flow cytometry datasets, part 1: Algorithm design scalable clustering for automated identification of rare cell populations in large, high-dimensional flow cytometry datasets, part 1: Algorithm design. *Cytometry Part A*, 85(5):408–421, 2014.

J. P. Nolan and L. Yang. The flow of cytometry into systems biology. *Briefings in Functional Genomics and Proteomics*, 6(2):81–90, 2007.

K. O'Neill, N. Aghaeepour, J. Špidlen, and R. Brinkman. Flow cytometry bioinformatics. *PLoS Computational Biology*, 9(12):e1003365, 2013.

S. Pyne, X. Hu, K. Wang, E. Rossin, T.-I. Lin, L. M. Maier, C. Baecher-Allan, G. J. McLachlan, P. Tamayo, D. A. Hafler, et al. Automated high-dimensional flow cytometric data analysis. *Proceedings of the National Academy of Sciences*, 106(21):8519–8524, 2009.

Y. Qian, C. Wei, F. Eun-Hyung Lee, J. Campbell, J. Halliley, J. A. Lee, J. Cai, Y. M. Kong, E. Sadat, E. Thomson, et al. Elucidation of seventeen human peripheral blood B-cell subsets and quantification of the tetanus response using a density-based method for the automated identification of cell populations in multidimensional flow cytometry data. *Cytometry Part B: Clinical Cytometry*, 78(S1):S69–S82, 2010.

P. Qiu, E. F. Simonds, S. C. Bendall, K. D. Gibbs Jr, R. V. Bruggner, M. D. Linderman, K. Sachs, G. P. Nolan, and S. K. Plevritis. Extracting a cellular hierarchy from high-dimensional cytometry data with spade. *Nature Biotechnology*, 29(10):886–891, 2011.

S. Richardson and P. J. Green. On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(4):731–792, 1997. ISSN 1467-9868.

C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer, New York, 2004. ISBN 9780387212395.

G. O. Roberts and J. S. Rosenthal. Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18(2):349–367, 2009.

M. Roederer. Spectral compensation for flow cytometry: Visualization artifacts, limitations, and caveats. *Cytometry*, 45(3):194–205, 2001.

H. M. Shapiro. *Practical flow cytometry*. John Wiley & Sons, Hoboken, New Jersey, 2005.

M. J. Welters, C. Gouttefangeas, T. H. Ramwadhdoebe, A. Letsch, C. H. Ottensmeier, C. M. Britten, and S. H. van der Burg. Harmonization of the intracellular cytokine staining assay. *Cancer Immunology, Immunotherapy*, 61 (7):967–978, 2012.

H. Zare, P. Shooshtari, A. Gupta, and R. R. Brinkman. Data reduction for spectral clustering to analyze high throughput flow cytometry data. *BMC Bioinformatics*, 11:403, 2010.

Figures

Additional Files

Additional file 1 — Supplementary material

The supplementary material contains the posterior in BayesFlow, the MCMC sampling scheme, additional details on the merging of components, information about the data generation, priors and initialization for the synthetic data example; parameters used for ASPIRE, additional details on healthyFlowData, the priors and the initialization procedure used when studying this data set and further results pertaining to the real flow cytometry data set, including fitting Gaussian mixture models to individual samples of healthyFlowData with the EM algorithm and scatter plots of GvHD for ASPIRE, HDPGMM and BayesFlow run 1.

Additional file 2 — Data generation files

A Python script for generating the large synthetic dataset, along with means, covariances and weights needed for this.

A Posterior

The posterior distribution given the model (1), (2), the priors (3) and data $\mathbf{Y}$ is

$$
\begin{aligned}
&\pi(\boldsymbol{\Theta}|\mathbf{Y},\mathbf{x})\\
&\propto \left(\prod_{j=1}^{J} \prod_{i=1}^{n_j} |\boldsymbol{\Sigma}_{j\mathbf{x}_{ij}}|^{-1/2} \exp\left(-\frac{1}{2} (\mathbf{Y}_{ij} - \boldsymbol{\mu}_{j\mathbf{x}_{ij}})^\top \boldsymbol{\Sigma}_{j\mathbf{x}_{ij}}^{-1} (\mathbf{Y}_{ij} - \boldsymbol{\mu}_{j\mathbf{x}_{ij}}) \right) \pi_{j\mathbf{x}_{ij}} \right) \cdot \\
&\left(\prod_{j=1}^{J} \prod_{k=1}^{K} \pi_{jk}^{a_{jk}} |\boldsymbol{\Sigma}_{\theta_k}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} (\boldsymbol{\mu}_{jk} - \boldsymbol{\theta}_k)^\top \boldsymbol{\Sigma}_{\theta_k}^{-1} (\boldsymbol{\mu}_{jk} - \boldsymbol{\theta}_k) \right) \right.\\
&\qquad\qquad\left. \frac{|\boldsymbol{\Sigma}_{jk}|^{-\frac{\nu_k+d+1}{2}} |\boldsymbol{\Psi}_k|^{\frac{\nu_k}{2}}}{2^{\frac{\nu_k d}{2}} \Gamma_d(\frac{\nu_k}{2})} \exp\left(-\mathrm{tr}(\boldsymbol{\Psi}_k \boldsymbol{\Sigma}_{jk}^{-1})/2 \right) \right) \cdot \\
&\left(\prod_{k=1}^{K} \exp\left(-\frac{1}{2} (\boldsymbol{\theta}_k - \mathbf{t}_k)^\top \mathbf{S}_k^{-1} (\boldsymbol{\theta}_k - \mathbf{t}_k) \right) |\boldsymbol{\Psi}_k|^{\frac{n_{\Psi_k}-d-1}{2}} \exp\left(-\mathrm{tr}(\mathbf{H}_k^{-1} \boldsymbol{\Psi}_k)/2 \right) \right.\\
&\qquad\qquad\left. |\boldsymbol{\Sigma}_{\theta_k}|^{-\frac{n_{\theta_k}}{2}} \exp\left(-\mathrm{tr}(\mathbf{Q}_k \boldsymbol{\Sigma}_{\theta_k}^{-1})/2 \right) \exp(-\lambda_k \nu_k) \right) .
\end{aligned}
\tag{5}
$$

B Sampling from the posterior distribution

We use a Markov Chain Monte Carlo (MCMC) algorithm to generate samples from the posterior distribution of the parameters Robert and Casella (2004). In each iteration we draw a value of each of the parameters $\boldsymbol{\Theta}$ and of $\mathbf{x}$. The backbone of our algorithm is a Gibbs sampler, but we need a Metropolis-Hastings step to sample ν_k. We also use Metropolis-Hastings steps to enable label-switching—which improves the mixing of the Gibbs sampler—and to turn on and off mixture components in the extended model with absent clusters.

In a Gibbs sampler samples from the full posterior distribution is obtained by successively sampling from the conditional posterior distributions of each of the variables given all other variables. First we sample the component assigment variables, $\mathbf{x}$, fixing all other parameters. The posterior from which we sample is a multinomial distribution with

$$
\pi(x_{ij} = k| \ldots) \propto \frac{N(\mathbf{Y}_{ij}; \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk})\pi_{jk}}{\sum_{h=0}^{K} N(\mathbf{Y}_{ij}; \boldsymbol{\mu}_{jh}, \boldsymbol{\Sigma}_{jh})\pi_{jh}},
$$

where '...' denotes conditioning on all parameter except the one of interest.

Let n_{jk} denote the number of i such that $x_{ij} = k$ and let $\mathbf{Y}_{.jk}$ denote the vector joining all $\mathbf{Y}_{ij}$ such that $x_{ij} = k$. The following Gibbs steps are derived from the posterior distribution (5)

$$\begin{aligned}
\boldsymbol{\pi}_j|\ldots &\sim D(a+n_{j1},\ldots,a+n_{jK}), \qquad (6)\\
\boldsymbol{\Sigma}_{jk}|\ldots &\sim IW(\boldsymbol{\Psi}_k + \sum_{i=1}^{n_{jk}}(\mathbf{Y}_{ijk}-\boldsymbol{\mu}_{jk})(\mathbf{Y}_{ijk}-\boldsymbol{\mu}_{jk})^\top, n_{jk}+\nu_k),\\
\boldsymbol{\mu}_{jk}|\ldots &\sim N_C(\boldsymbol{\Sigma}_{\theta_k}^{-1}\boldsymbol{\theta}_k + \boldsymbol{\Sigma}_{jk}^{-1}\sum_{i=1}^{n_{jk}}\mathbf{Y}_{ijk}, \boldsymbol{\Sigma}_{\theta_k}^{-1} + n_{jk}\boldsymbol{\Sigma}_{jk}^{-1}),\\
\boldsymbol{\Sigma}_{\theta_k}|\ldots &\sim IW(\mathbf{Q}_k + \sum_{j=1}^{J}(\boldsymbol{\mu}_{jk}-\boldsymbol{\theta}_k)(\boldsymbol{\mu}_{jk}-\boldsymbol{\theta}_k)^\top, J+n_{\theta_k}),\\
\boldsymbol{\Psi}_k|\ldots &\sim W\left(\left(\mathbf{H}_k^{-1} + \sum_{j=1}^{J}\boldsymbol{\Sigma}_{jk}^{-1}\right)^{-1}, n_{\Psi_k} + J\nu_k\right),\\
\boldsymbol{\theta}_k|\ldots &\sim N_C(\mathbf{S}_k^{-1}\mathbf{t}_k + \boldsymbol{\Sigma}_{\theta_k}^{-1}\sum_{j=1}^{J}\boldsymbol{\mu}_{jk}, \mathbf{S}_k^{-1} + J\boldsymbol{\Sigma}_{\theta_k}^{-1}).
\end{aligned}$$

Here N_C denotes the canonical parameterization of the normal distribution, which means that if $\mathbf{x} \sim N_C(\mathbf{b},\mathbf{Q})$, then $\pi(\mathbf{x}) \propto \exp\left(-\frac{1}{2}\mathbf{x}^\top\mathbf{Q}\mathbf{x} + \mathbf{b}^\top\mathbf{x}\right)$.

To handle the non-standard conditional distribution of ν_k, we utilize a Metropolis-Hastings (MH) algorithm. A proposal ν_k^* is generated by sampling $\nu_k^* \sim \nu_k + Z$, where Z is uniformly distributed on $\{-r, -r+1, \ldots, r\}$ for some $r \in \mathbb{N}^+$. Hence the transition density $q(\nu_k, \nu_k^*) = q(\nu_k|\nu_k^*)$ is constant on its support. The proposed ν_k^* is accepted with probability

$$\alpha(\nu_k,\nu_k^*) = \min\left(1, \frac{\pi(\nu_k^*)q(\nu_k^*,\nu)}{\pi(\nu_k)q(\nu,\nu_k^*)}\right) = \min\left(1, \frac{\pi(\nu_k^*)}{\pi(\nu_k)}\right),$$

where $\pi(\nu_k)$ denotes the posterior distribution of ν_k given all other parameters and data. Using (5) we get that

$$\alpha(\nu_k,\nu_k^*) = \min\left(1, \prod_{j=1}^{J}\frac{\Gamma_d(\frac{\nu_k}{2})}{\Gamma_d(\frac{\nu_k^*}{2})}\left(2^d|\boldsymbol{\Sigma}_{jk}||\boldsymbol{\Psi}_k|^{-1}\right)^{\frac{\nu_k-\nu_k^*}{2}}\exp\left(\lambda_k(\nu_k^*-\nu_k)\right)\right).$$

If ν_k^* is accepted it will be the new sample, otherwise the new sample will be ν_k. The parameter r is updated adaptively to get a desired acceptance rate of 0.3, according to an algorithm by Roberts and Rosenthal Roberts and Rosenthal (2009).

B.1 Label switching

An issue that frequently occurs, especially with poor starting values, is that a cluster $\{\boldsymbol{\mu}_{jk_1}, \boldsymbol{\Sigma}_{jk_1}, \pi_{jk_1}\}$ is incorrectly assigned to the latent cluster k_1 when it clearly should belong to k_2. When the number cells is large the first row of (5) will dominate the posterior so that $\{\boldsymbol{\mu}_{jk_1}, \boldsymbol{\Sigma}_{jk_1}, \pi_{jk_1}\}$ or $\{\boldsymbol{\mu}_{jk_2}, \boldsymbol{\Sigma}_{jk_2}, \pi_{jk_2}\}$ does not change much at all in the updating step and thus in practice the clusters will never move close enough to each other in order to switch locations.

To remedy this issue, we introduce an extra MH step where labels can be switched between clusters in each sample j in each iteration. The proposed MH algorithm has a symmetric transition kernel, where two labels k_1 and k_2 are sampled from $\{1, \ldots, K\}$ with equal probability. The proposed switch is accepted with probability

$$\alpha(k_1, k_2) = \min\left(1, \frac{\pi(\mu_{jk_2}|\boldsymbol{\theta}_{k_1}, \boldsymbol{\Sigma}_{\theta_{k_1}})\pi(\mu_{jk_1}|\boldsymbol{\theta}_{k_2}, \boldsymbol{\Sigma}_{\theta_{k_2}})}{\pi(\mu_{jk_1}|\boldsymbol{\theta}_{k_1}, \boldsymbol{\Sigma}_{\theta_{k_1}})\pi(\mu_{jk_2}|\boldsymbol{\theta}_{k_2}, \boldsymbol{\Sigma}_{\theta_{k_2}})}\right.$$
$$\left.\frac{\pi(\boldsymbol{\Sigma}_{jk_1}|\boldsymbol{\Psi}_{k_2}, \nu_{k_2})\pi(\boldsymbol{\Sigma}_{jk_2}|\boldsymbol{\Psi}_{k_1}, \nu_{k_1})}{\pi(\boldsymbol{\Sigma}_{jk_1}|\boldsymbol{\Psi}_{k_1}, \nu_{k_1})\pi(\boldsymbol{\Sigma}_{jk_2}|\boldsymbol{\Psi}_{k_2}, \nu_{k_2})}\right). \quad (7)$$

B.2 Cluster activation and deactivation

In the extended model where components can be absent in some samples we use a reversible jump MH-algorithm Green (1995) to enable changes to the dimension of the model. We use the indicator variable $\mathbf{Z}_j$ to keep track of which components that are active; $Z_{jk} = 1$ if component k is active in sample j and $Z_{jk} = 0$ otherwise.

Activation or deactivation is proposed as the last step of each iteration of the MCMC algorithm. Throughout the activation/deactivation step the component assignment variables x_{ij} are integrated out of the posterior.

A deactivation of an active component is proposed with probability p_d and an activation of a component that is not active is proposed with probability p_a. The component that is proposed to be deactivated/activated is chosen randomly among the clusters that are active or not active respectively with equal probability. The probability of proposing to deactivate component k in sample j is

$$q(Z_{jk} = 1 \to 0) = \frac{p_d}{\sum_{l=1}^{K} Z_{jl}}.$$

The probability of proposing to activate component k in sample j is

$$q(Z_{jk} = 0 \to 1) = \frac{p_a}{K - \sum_{l=1}^{K} Z_{jl}}.$$

If an activation step is proposed it is necessary to generate parameters for the new component; they are obtained in the following way:

$$\begin{aligned} \pi_{jk}^* &\sim \text{Beta}(\alpha, \beta), \\ \boldsymbol{\mu}_{jk}^* &\sim N(\boldsymbol{\theta}_k, \boldsymbol{\Sigma}_{\theta_k}), \\ \boldsymbol{\Sigma}_{jk}^* &\sim IW(\boldsymbol{\Psi}_k, \nu_k). \end{aligned}$$

Here α and β is chosen so that the probability π_{jk}^* is typically close to zero. The transition density $q_{k_j}(\boldsymbol{\mu}_{jk}^*, \boldsymbol{\Sigma}_{jk}^*, \pi_{jk}^*)$ is the joint density of these new parameters when they are sampled as above. For the remaining components we keep the mean and covariance parameters, $\boldsymbol{\mu}_{jl}^* = \boldsymbol{\mu}_{jl}$ and $\boldsymbol{\Sigma}_{jl}^* = \boldsymbol{\Sigma}_{jl}$ for $l \neq k$, but the probabilities $\boldsymbol{\pi}_j$ have to be modified. In the reversible jump algorithm this is done in a dimension matching transform. When activating a cluster we set $\pi_{jl}^* = (1 - \pi_{jk}^*)\pi_{jl}$ for $l \neq k$ in the transform and when deactivating a cluster we set $\pi_{jl}^* = \pi_{jl}/(1 - \pi_{jk})$ for $l \neq k$.

In order to make the Markov chain reversible it is necessary to add the Jacobian of the variable change in the dimension matching transform as a factor in the acceptance probability. Let $\boldsymbol{\Theta}^*$ denote the set of parameters in the proposed model and let $\boldsymbol{\Theta}$ denote the set of current parameters. In an activation step we get Richardson and Green (1997)

$$\left| \frac{\partial(\boldsymbol{\Theta}^*)}{\partial\left(\boldsymbol{\Theta}, \pi_{jk}^*, \boldsymbol{\mu}_{jk}^*, \boldsymbol{\Sigma}_{jk}^*\right)} \right| = (1 - \pi_{jk}^*)^{\sum_{l=1}^{K} Z_{jl}},$$

and in a deactivation step the Jacobian is the inverse.

We are now ready to define the acceptance probability for a proposed $\boldsymbol{\Theta}^*$ which implies activation of component k in sample j. The acceptance probability equals

$$\alpha\left(\boldsymbol{\Theta},\, \boldsymbol{\Theta}^*\right) = \min\left\{1, \frac{\pi(\boldsymbol{\Theta}^*|\mathbf{Y})q(Z_{jk}=1\to 0)}{\pi(\boldsymbol{\Theta}|\mathbf{Y})q_{k_j}(\boldsymbol{\mu}^*_{jk}, \boldsymbol{\Sigma}^*_{jk}, \pi^*_{jk})q(Z^*_{jk}=0\to 1)} \right.$$
$$\left. \left| \frac{\partial(\boldsymbol{\Theta}^*)}{\partial\left(\boldsymbol{\Theta}, \pi^*_{jk}, \boldsymbol{\mu}^*_{jk}, \boldsymbol{\Sigma}^*_{jk}\right)} \right| \right\}, \qquad (8)$$

where $\pi(\boldsymbol{\Theta}^*|\mathbf{Y})$ is the posterior distribution (5) with x_{ij} integrated out. This can be written as

$$\alpha\left(\boldsymbol{\Theta},\, \boldsymbol{\Theta}^*\right) = \min\left\{1, \frac{\prod_{i=1}^{n_j}\sum_{l=1}^{K} Z^*_{jl}\pi^*_{jl}N(\mathbf{Y}_{ij}; \boldsymbol{\mu}^*_{jl}, \boldsymbol{\Sigma}^*_{jl})}{\prod_{i=1}^{n_j}\sum_{l=1}^{K} Z_{jl}\pi_{jl}N(\mathbf{Y}_{ij}; \boldsymbol{\mu}_{jl}, \boldsymbol{\Sigma}_{jl})} \cdot \right.$$
$$\left. \frac{D(\boldsymbol{\pi}^*_j; \mathbf{a})\exp(-c_s)}{\mathrm{Beta}(\pi^*_{jk}; \alpha, \beta)D(\mathbf{p}_j; \mathbf{a})} \frac{\frac{p_d}{\sum_{l=1}^{K} Z_{jl}}}{\frac{p_b}{K-\sum_{l=1}^{K} Z_{jl}}}(1-\pi^*_{jk})^{\sum_{l=1}^{K} Z_{jl}} \right\}.$$

The acceptance probability for a deactivation step is obtained from the same expression but with inverse ratio.

When we extend the model and introduce $\mathbf{Z}_j$ the posterior changes so that the sampling of the other variables has to be modified. As an example the conditional distribution of $\boldsymbol{\Psi}_k$ changes to

$$W\left(\left(\mathbf{H}_k + \sum_{h=1}^{J} Z_{hk}\boldsymbol{\Sigma}^{-1}_{jk}\right)^{-1},\ \nu^* + \nu_k\sum_{h=1}^{J} Z_{hk}\right).$$

We do not display all the changes since they are notationally complicated but otherwise straightforward, except for the label switching step. Suppose we propose to change k_1 to k_2 where k_1 is an inactive cluster. Then the acceptance probability (7) changes to

$$\alpha(k_1, k_2) = \min\left(1, \frac{\pi(\boldsymbol{\mu}_{jk_2}|\boldsymbol{\theta}_{k_1}, \boldsymbol{\Sigma}_{\theta_{k_1}})\pi(\boldsymbol{\Sigma}_{jk_2}|\boldsymbol{\Psi}_{k_1}, \nu_{k_1})}{\pi(\boldsymbol{\mu}_{jk_2}|\boldsymbol{\theta}_{k_2}, \boldsymbol{\Sigma}_{\mu_{k_2}})\pi(\boldsymbol{\Sigma}_{jk_2}|\boldsymbol{\Psi}_{k_2}, \nu_{k_2})}\right). \qquad (9)$$

C Merging latent clusters

The merging of latent clusters is done in a hierarchical fashion. In each step we have a number of latent super clusters comprising of one or more latent clusters. The corresponding super components in each sample are mixtures of Gaussians, a representation which is hard to work with. It is useful to instead use the data perspective, i.e. to consider the soft clustering of the data induced by the GMM of each sample.

For each sample we define super cluster k from the probabilities for each of the data points in that sample to belong to any of the components linked to the latent super cluster k. We denote cluster k in sample j by $\Gamma_{k,j} = (\mathbf{Y}_{ij}, w_{ijk})_{i=1}^{n_j}$, where w_{ijk} is the probability that $\mathbf{Y}_{ij}$ belongs to super cluster k. The parameter w_{ijk} can be estimated from the sampling of x_{ij}.

To determine candidates for the subsequent merger, Bhattacharyya distance is computed between all pairs of current clusters in each sample. To do this we approximate each $\Gamma_{k,j}$ with a Gaussian distribution with parameters

$$\boldsymbol{\mu}^{(kj)} = \sum_{i=1}^{n_j} w_{ijk}\mathbf{Y}_{ij}, \qquad \boldsymbol{\Sigma}^{(kj)} = \sum_{i=1}^{n_j} w_{ijk}(\mathbf{Y}_{ij} - \boldsymbol{\mu}^{(kj)})(\mathbf{Y}_{ij} - \boldsymbol{\mu}^{(kj)})^\top$$

and use formula (4), so

$$\begin{aligned} d_{\text{bhat}}(\Gamma_{k,j}, \Gamma_{l,j}) = 1/8 \cdot (\boldsymbol{\mu}^{(kj)} - \boldsymbol{\mu}^{(lj)})^\top \bar{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{\mu}^{(kj)} - \boldsymbol{\mu}^{(lj)}) \\ + 1/2 \cdot \log\left(|\bar{\boldsymbol{\Sigma}}|/\sqrt{|\boldsymbol{\Sigma}^{(kj)}||\boldsymbol{\Sigma}^{(lj)}|}\right), \end{aligned}$$

where $\bar{\boldsymbol{\Sigma}} = (\boldsymbol{\Sigma}^{(kj)} + \boldsymbol{\Sigma}^{(lj)})/2$. The candidates for the subsequent merger are the pair of clusters (k, l)—which among those pairs who have not previously been evaluated for merging—has highest minimal value of $\exp(-d_{\text{bhat}}(\Gamma_{kj}, \Gamma_{lj}))$ across samples j. It is natural to consider $\exp(-d_{\text{bhat}})$ instead of d_{bhat} when comparing Bhattacharyya distances since $\exp(-d_{\text{bhat}})$ is an upper bound of the misclassification probability between the components Fukunaga (1990).

If $\min_j(\exp(-d_{\text{bhat}}(\Gamma_{kj}, \Gamma_{lj}))) > h_1$ latent clusters k and l are immediately merged. On the other hand, if $h_1 > \min_j(\exp(-d_{\text{bhat}}(\Gamma_{kj}, \Gamma_{lj}))) > h_2$, they are merged only if the resulting cluster does not have sufficient evidence of being multimodal. Finally, if $\min_j(\exp(-d_{\text{bhat}}(\Gamma_{kj}, \Gamma_{lj}))) < h_2$ they are not merged and the procedure is stopped.

To evaluate multimodality of potential mergers we apply Hartigan's dip test of unimodality Hartigan and Hartigan (1985) to the projection of the merged cluster onto the coordinate axes which have 1-dimensional Bhattacharyya overlap below a threshold $h^{(1)}$ and to the projection onto Fisher's discriminant coordinate separating the two clusters, namely $u = (\mathbf{\Sigma}^{(kj)} + \mathbf{\Sigma}^{lj)})^{-1}(\boldsymbol{\mu}^{(kj)} - \boldsymbol{\mu}^{(lj)})$ Fisher (1936). Hartigan's dip statistic is computed from the empirical distribution function, which can readily be computed for these soft clusters from $(\mathbf{Y}_{ij}, w_{ijk})_{i=1}^{n_j}$. If for any of the projections in any of the samples where the total weight of the cluster $\sum_i^{n_j} w_{ijk}$ is at least 10, we get a p-value below the threshold h_d we do not merge.

To determine the thresholds h_1, h_2 and h_d we use results from two experiments performed by Hennig Hennig (2010). Synthetic data were generated from distributions which naturally represent a single cluster and a number of Gaussian components were fitted to the data. For different criteria, threshold values for merging the components to one cluster in 95% of the cases, were then reported. The experiments were performed over a range of different dimensions and number of data points. To determine h_1, h_2 and h_d, we consider only results for distributions of dimension two to five and for at least 100 and at most 500 points, since for most of the flow cytometry samples in the data sets studied in Section 3.2 a small cluster containing 1% of the data points would have about 100–200 data points.

In the first experiment two components were fitted to data generated from a unimodal mixture of two Gaussian distributions with the property that if the means were further apart the density would be bimodal. In the second experiment six Gaussian components were fitted to data generated from uniform distributions on hypercubes. The merging of the six components were made in a hierarchical procedure similar to ours.

When Bhattacharyya distance was used as merging criterion the threshold for $\exp(-d_{\text{bhat}})$ varied between 0.40 and 0.53 for the relevant 2- and 5-dimensional data sets in the first experiment. For the second experiment we considered four combinations of dimension and number of data points and for these the thresholds were 0.12, 0.17, 0.01 and 0.11 respectively. This lead us to use $h_1 = 0.47$ as the soft threshold and $h_2 = 0.08$ as the soft threshold.

Hartigan's dip test was also evaluated as a criterion for merging, but only the first of the experiments is relevant for our use of it, since we only use the dip test to evaluate proposed mergers and not select candidates for merging. Only projections onto Fisher's discriminant coordinate were considered in the experiment. The threshold for the p-value varied between 0.15 and 0.41, so we chose $h_d = 0.28$. It should be noted that this cannot be translated into a significance level since the tests are done in a data-dependent way.

The threshold $h^{(1)}$ was set based on the results in the first experiment for one-dimensional data sets. For data sets with 50 data points, the threshold for $\exp(-d_{\text{bhat}})$ was 0.201, for data sets with 200 points it was 0.39 and for data sets with 500 data points it was 0.49. Therefore we let $h^{(1)}$ be dependent on the weight of the cluster in the following way:

$$h^{(1)}(w) = \begin{cases} 0.201 & \text{if} \quad w \leq 50 \\ 0.390 & \text{if} \quad 50 < w \leq 200 \\ 0.490 & \text{if} \quad w > 200. \end{cases}$$

D Simulation study

D.1 Data generation

In this section the method for generating the small synthetic dataset is presented. The Additional file `article_simulatedata.py` contains the method for generating the large synthetic dataset. The four latent means are

$$\boldsymbol{\theta}_1 = [0,0,0],\ \boldsymbol{\theta}_2 = [0,-2,1],\ \boldsymbol{\theta}_3 = [1,2,0],\ \boldsymbol{\theta}_4 = [-2,2,1.5].$$

Each $\boldsymbol{\mu}_{jk}$ in the simulation is generated by

$$\begin{aligned} \boldsymbol{\mu}_{jk} &= \boldsymbol{\theta}_k + \mathbf{Z}_{jk},\ k = 1,2,3,4 \\ \mathbf{Z}_{jk} &\sim N(\mathbf{0}, \boldsymbol{\Sigma}_{\mu_k}), \end{aligned}$$

where

$$\boldsymbol{\Sigma}_{\mu_1} = \begin{bmatrix} 1.27 & 0.25 & 0 \\ 0.25 & 0.27 & -0.001 \\ 0 & -0.001 & 0.001 \end{bmatrix}, \quad \boldsymbol{\Sigma}_{\mu_2} = \begin{bmatrix} 0.06 & 0.04 & -0.03 \\ 0.04 & 0.05 & 0 \\ -0.03 & 0. & 0.09 \end{bmatrix},$$

$$\boldsymbol{\Sigma}_{\mu_3} = \begin{bmatrix} 0.44 & 0.08 & 0.08 \\ 0.08 & 0.16 & 0 \\ 0.08 & 0 & 0.16 \end{bmatrix}, \qquad \boldsymbol{\Sigma}_{\mu_4} = 0.01\mathbf{I}.$$

The covariance matrices are generated through

$$\boldsymbol{\Sigma}_{jk} \sim IW((\nu_k - 3)\boldsymbol{\Psi}_k, \nu_k),\ k = 1, 2, 3, 4,$$

where

$$\boldsymbol{\Psi}_1 = 0.1\mathbf{I}, \qquad \boldsymbol{\Psi}_2 = 0.1 \begin{bmatrix} 2.0 & 0.5 & 0 \\ 0.5 & 2.0 & 0.5 \\ 0 & 0.5 & 2.0 \end{bmatrix},$$

$$\boldsymbol{\Psi}_3 = 0.1 \begin{bmatrix} 2.0 & -0.5 & 1.0 \\ -0.5 & 2.0 & -0.5 \\ 1.0 & -0.5 & 2.0 \end{bmatrix}, \qquad \boldsymbol{\Psi}_4 = 0.1 \begin{bmatrix} 1.0 & 0.3 & 0.3 \\ 0.3 & 1.0 & 0.3 \\ 0.3 & 0.3 & 1.0 \end{bmatrix},$$

and $\nu_k = 100$ for all k. Finally, $\boldsymbol{\pi}_j = [0.49, 0.3, 0.2, 0.01]$ if all clusters are present. If one or two clusters are not present the ratio of the probabilities for the present clusters remains the same.

D.2 Priors

The priors are set to represent non informative priors; the priors are set equal for all classes. The exact values are:

$$\begin{aligned} \mathbf{S}_k &= 10^6\mathbf{I}_d, \mathbf{t}_k = \mathbf{0}, \\ \mathbf{H}_k &= 10^{-6}\mathbf{I}_d, n_{\psi_k} = d, \\ \mathbf{Q}_k &= 10^{-6}\mathbf{I}_d, n_{\theta_k} = d, \\ l_k &= 0.01, \end{aligned}$$

for $k = 1, 2, 3, K$ with $K = 4$ for the small dataset and $K = 11$ for the large dataset. For the small dataset the outlier component was not used for inference.

D.3 Initialization for small dataset

Before running the MCMC sampler to get samples from the posterior distribution, we utilize the following initialization to get suitable initial parameter values. First we set all mean parameters $\boldsymbol{\mu}_{jk}$ and $\boldsymbol{\theta}_k$ to $\mathbf{0}$ and all covariance and precision matrices $\boldsymbol{\Sigma}_{jk}$, $\boldsymbol{\Sigma}_{\theta_k}$ and $\boldsymbol{\Psi}_k$ to $\mathbf{I}$. Then after letting the MCMC sampler run for 5000 iterations, without the option of turning off components, we link all the components across samples through the following procedure:

1. The first sample is left unchanged.

2. For the second sample the components are first sorted by $\boldsymbol{\pi}_2$, so we get ordered components $(\boldsymbol{\mu}_{2(i)}, \boldsymbol{\Sigma}_{2(i)}, \pi_{2(i)})$ for $i = 1, 2, 3, 4$, where $\pi_{2(1)} \geq \pi_{2(2)} \geq \pi_{2(3)} \geq \pi_{2(4)}$. Then the first component $(\boldsymbol{\mu}_{2(1)}, \boldsymbol{\Sigma}_{2(1)}, \pi_{2(1)})$ is matched to the component k whose mean $\boldsymbol{\mu}_{1k}$ is closest to $\boldsymbol{\mu}_{2(1)}$. If for example we have that $\boldsymbol{\mu}_{13}$ is closest to $\boldsymbol{\mu}_{2(1)}$ we set $(\boldsymbol{\mu}_{23}, \boldsymbol{\Sigma}_{23}, \pi_{23}) = (\boldsymbol{\mu}_{2(1)}, \boldsymbol{\Sigma}_{2(1)}, \pi_{2(1)})$. This is repeated for $(\boldsymbol{\mu}_{2(i)}, \boldsymbol{\Sigma}_{2(i)}, \pi_{2(i)})$, $i = 2, 3, 4$, but indices which have already been assigned to components are excluded from consideration.

3. For the remaining samples we proceed as for the second sample, with the exception that the matching of $\boldsymbol{\mu}_{j(k)}$ is now done to the average of the $j - 1$ previously matched clusters means, namely $(j-1)^{-1} \sum_{l=1}^{j-1} \boldsymbol{\mu}_{lk}$ for $k = 1, 2, 3, 4$.

D.4 Initialization for large dataset

Before starting the actual MCMC sampler, we run an initialization scheme that is designed to make the sampler jump out of local maxima of the likelihood. The method we use does not give a reversible Markov chain and thus cannot be part of the actual MCMC run. We do the following steps about ten times for each GMM without updating the latent parameters:

1. Sample $\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}_j$ using the regular Gibbs sampler for ten iterations.

2. Calculate the likelihood for the current parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}_j$. Randomly select a cluster $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_{jk})$ and then select a dimension d_1 at random. Remove the cluster $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_{jk})$ and the cluster closest to it in d_1, draw two random points and use them as initial points for two new clusters. Run the Gibbs sampler for ten iterations. If the new parameters has higher likelihood then the old keep the new, otherwise go back to the old.

3. Calculate the likelihood for the current parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}$. Randomly select a cluster $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k)$, with a probability of choosing cluster k proportional to $\frac{1}{\pi_{jk}}$ so that the smaller the cluster the more likely it is to be chosen. Remove $\boldsymbol{\mu}_k$, draw a random point and use it as $\boldsymbol{\mu}_k$ and set $\boldsymbol{\Sigma}_k$ to the old $\boldsymbol{\Sigma}_k$ times ten. Then run the Gibbs sampler for ten iterations. If the

new parameters has higher likelihood then the old keep the new otherwise go back to the old.

The two last steps works quite well for destroying clusters that have been stuck in the wrong shape or removing small clusters that is at the wrong location of the space.

E Flow cytometry data analysis

E.1 Dataset details

E.1.1 healthyFlowData

Here follows a description of the dataset healthyFlowData: how it was obtained and how it was preprocessed before we downloaded it from the R package healthyFlowData.

Antibodies against CD45, CD19, CD3, CD8 and CD4 linked to fluorochromes were used to mark the PBMC and when passed through the flow cytometer the expression of these markers were measured along with front and side scatter. A standard transformation called compensation was used to remove effects of spectral overlap Azad et al. (2013). Following this the data was transformed using the function $\operatorname{asinh}(y/c)$, where c was chosen to minimize Bartlett's statistic, with the purpose to stabilize variance between markers; functions for this transformation are available in the R package flowVS Azad (2015). Measurements corresponding to lymphocytes were selected using front and side scatter by fitting a bivariate normal distribution and filtering based on a likelihood threshold using the norm2Filter function in the flowCore R package (Azad, personal communication). This resulted in between 6172 and 19,554 cell measurements for each sample. Since all lymphocytes are CD45+, only the other four markers were retained.

E.2 Priors

Priors should be set depending on the application, since they specify our tolerance to variation. However, to simplify this process we want to be able to translate prior parameters between data sets with different number of samples, cells and components. To do this, we consider the sampling scheme (6). Looking at the sampling of $\boldsymbol{\theta}_k$, we see that the effect of $\mathbf{S}_k$ decreases proportionally to the number of samples J. Thus we set $\mathbf{S}_k = \mathbf{I} \cdot s_k/J$ for those k for which we want an informative prior on location. Based on the sampling of $\boldsymbol{\mu}_{jk}$ we see that $\boldsymbol{\Sigma}_{\theta_k}$ should be proportional to $1/n_{jk}$, thus from the sampling of $\boldsymbol{\Sigma}_{\theta_k}$, n_{θ_k} should be proportional to n_{jk}. The value n_{jk} can be estimated by n/K, where n is the total number of cells across samples. Furthermore, $\mathbf{Q}_k$ should be proportional to J.

Moving over to shape variation, from the sampling of $\boldsymbol{\Sigma}_{jk}$ we see that $\boldsymbol{\Psi}_k$ should be proportional to n_{jk} and from the sampling of $\boldsymbol{\Psi}$ we see that to achieve this $n_{\boldsymbol{\Psi}_k}$ should also be proportional to n_{jk}. Furthermore $\mathbf{H}_k$ should be proportional to $1/J$. In summary,

$$\begin{aligned} n_{\boldsymbol{\theta}_k} &= nt_k \cdot n/K, & n_{\boldsymbol{\Psi}_k} &= np_k \cdot n/K, \\ \mathbf{Q}_k &= q_k \cdot J, & \mathbf{H}_k &= h_k/J, \\ \mathbf{S}_k &= s_k/J \cdot \mathbf{I}, \end{aligned}$$

where the parameters nt_k, np_k, q_k, h_k and s_k can be reused across data sets of different sizes and with different number of components. Note that $\mathbf{S}_k$ should only be set as above when informative priors on latent locations of clusters are wanted. For the flow cytometry data sets considered in this work we use $nt_k = 0.75$, $np_k = 0.25$, $h_k = 10^3$ and $q_k = 10^{-3}$ for all components k. $\mathbf{S}_k$ was uninformative in most cases and set to 10^6, but for the rare phenotype in the GvHD data set we used $s_k = 0.01^2$.

E.3 Point estimates

During the production iterations of the MCMC sampler we get samples of

$$\boldsymbol{\Theta}^{(r)} = \left(\boldsymbol{\mu}_{jk}^{(r)}, \boldsymbol{\Sigma}_{jk}^{(r)}, \boldsymbol{\theta}_k^{(r)}, \boldsymbol{\Psi}_k^{(r)}, \nu_k^{(r)}, \boldsymbol{\pi}_j^{(r)}\right), \; r = 1, \dots, R.$$

We use the means of $\boldsymbol{\mu}_{jk}^{(r)}, \boldsymbol{\Sigma}_{jk}^{(r)}, \boldsymbol{\theta}_k^{(r)}$ and $\boldsymbol{\Psi}_k^{(r)}/(\nu_k^{(r)} - d - 1)$ to get point estimates of sample component and latent cluster means and covariance matrices; the means of $\boldsymbol{\pi}_j^{(r)}$ are used to get point estimates of the mixing proportions.

E.4 Quality control

E.4.1 Convergence

We assess the convergence of the MCMC sampler in BayesFlow by looking at trace plots for $\boldsymbol{\theta}_k$ and ν_k, where $k \in \{1, \dots, K\}$. The trace plots for the first

	p_{sw}	p_a=p_d
Burn-in phase 1a	0.1	0
Burn-in phase 1b	0.1	0
Burn-in phase 2a	0.1	0
Burn-in phase 2b	0.1	0.1
Burn-in phase 3	0	0.1
Production phase	0	0.1

Table 2: Simulation parameters for MCMC sampling for real flow cytometry data. During the phase 1a, the prior parameters n_θ and n_Ψ are increased by a factor of 100. After phase 1b, outlying sample components are turned off, i.e. sample components which are closer in Bhattacharyya distance to another latent component than the one to which they are connected.

accepted run of healthyFlowData and GvHD are shown in Fig. S1 and Fig. S2 respectively.

E.4.2 Unimodality

We want to detect if the distribution of data assigned to a single component or super component is not unimodal, since it indicates that the latent cluster maybe should be divided into two or more components. To do this we use Hartigan's dip test Hartigan and Hartigan (1985) of unimodality for the one-dimensional marginal distributions. For cluster–dimension combinations which give dip tests below 0.28 (our threshold for merging clusters) we consider histograms of quantiles of the clusters as shown in Fig. S3 (usual histograms are less useful since the clusters are soft). When there are tendencies of bimodality it can be accepted when it seems unlikely that dividing the cluster further would result in a new interesting population. This can for example be the case if this tendency exist in a single sample and it is not in the midrange of expression (around 0.5) where important splits between positive and negative cells are often made.

E.4.3 Eigenvectors

Thanks to that we explicitly model component shapes we can find patterns among the shapes by studying the eigenvectors of the sample component covariance ma-

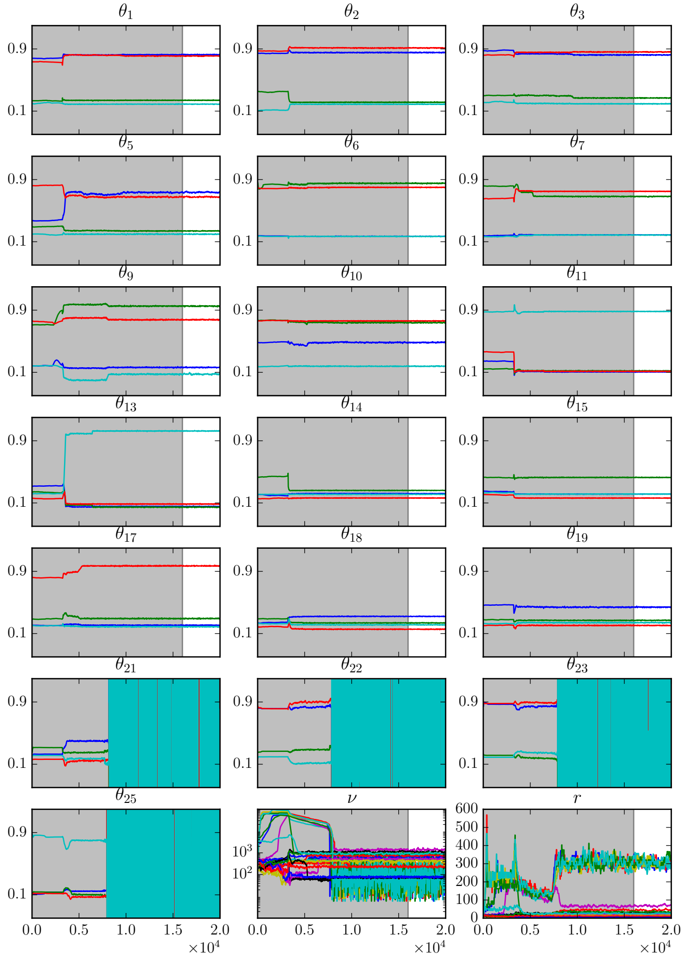


Figure S1: Trace plots of latent means θ_k for $k = 1, \ldots, 25$, ν and MH sampling interval r, for the first accepted BayesFlow run on healthyFlowData. Burn-in iterations are plotted on gray background. As can be seen the clusters 20-25 were turned of during the burn-in iterations.

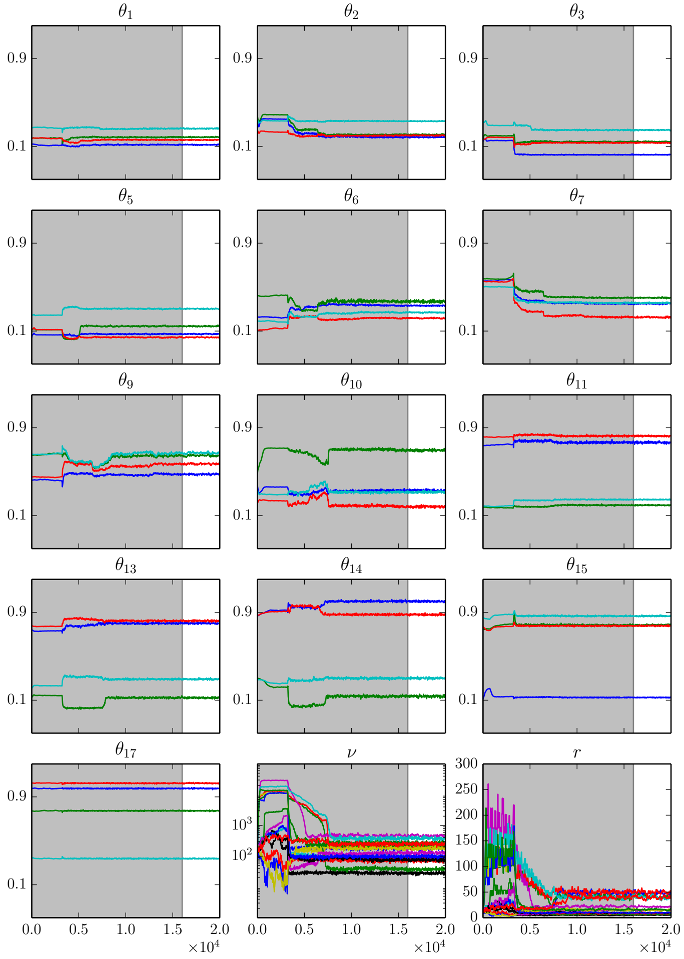


Figure S2: Trace plots of latent means θ_k for $k = 1, \ldots, 25$, ν and MH sampling interval r, for the first accepted BayesFlow run on GvHD. Burn-in iterations are plotted on gray background.

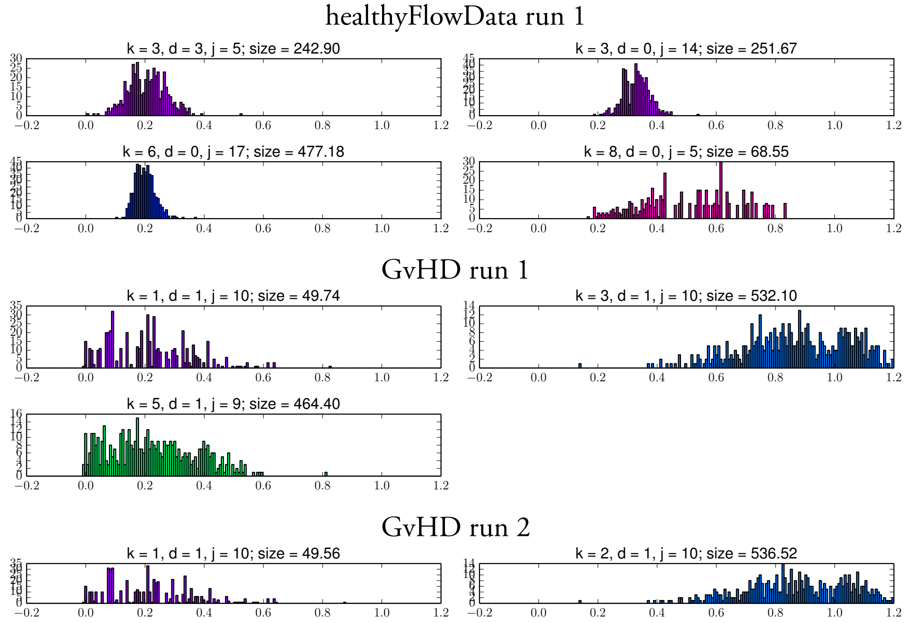


Figure S3: Histograms of quantiles of soft clusters in one dimension. Only dimension–cluster combinations which gives dip tests below 0.28 are shown. Evaluating these is part of the quality control and all the above have been seen as acceptable. Even if there are tendencies of bimodality it can be accepted when it seems likely that the cluster consists of a single population based on the expression.

trices, as in Fig. S4.

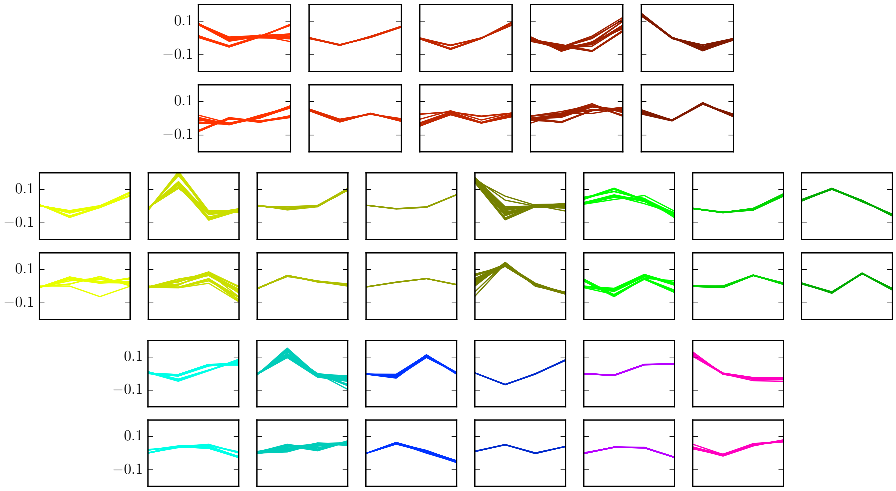


Figure S4: The two first eigenvectors scaled by their corresponding eigenvalues of the 19 active components in the first accepted run for the healthyFlowData dataset. For most components the eigenvectorsj—i.e. the shapes—are very similar across samples, but we can for example also see that for some components there are two groups of shapes.

E.5 Parameters and convergence for ASPIRE

As recommended, we first standardize the pooled data and then use the parameter values $s = 150\log(d+1)/d$, $m = d+2$, $\kappa_0 = 0.05$ and $\alpha = \gamma = 1$. To decide κ_i we tried four different recommended values, $\{0.1, 0.25, 0.5, 1\}$. The highest mean likelihood during the production iterations was obtained for $\kappa_i = 0.1$ for both healthyFlowData and GvHD (see Fig. S5), thus we used results from this run as the final results. However, we observed that the likelihood increased monotonically when decreasing κ_i, so for healthyFlowData we also explored additional, smaller values of κ_i, namely 0.05, 0.25 and 0.01. We noted a continued increase in mean likelihood and noted that this was accompanied by a decrease in the number of latent components and an increase in the number of mixture components corresponding to each latent component. For $\kappa_i = 0.01$, essentially all data points (> 99.99%) were assigned to the two largest latent components.

This led us to stick to the value $\kappa_i = 0.1$.

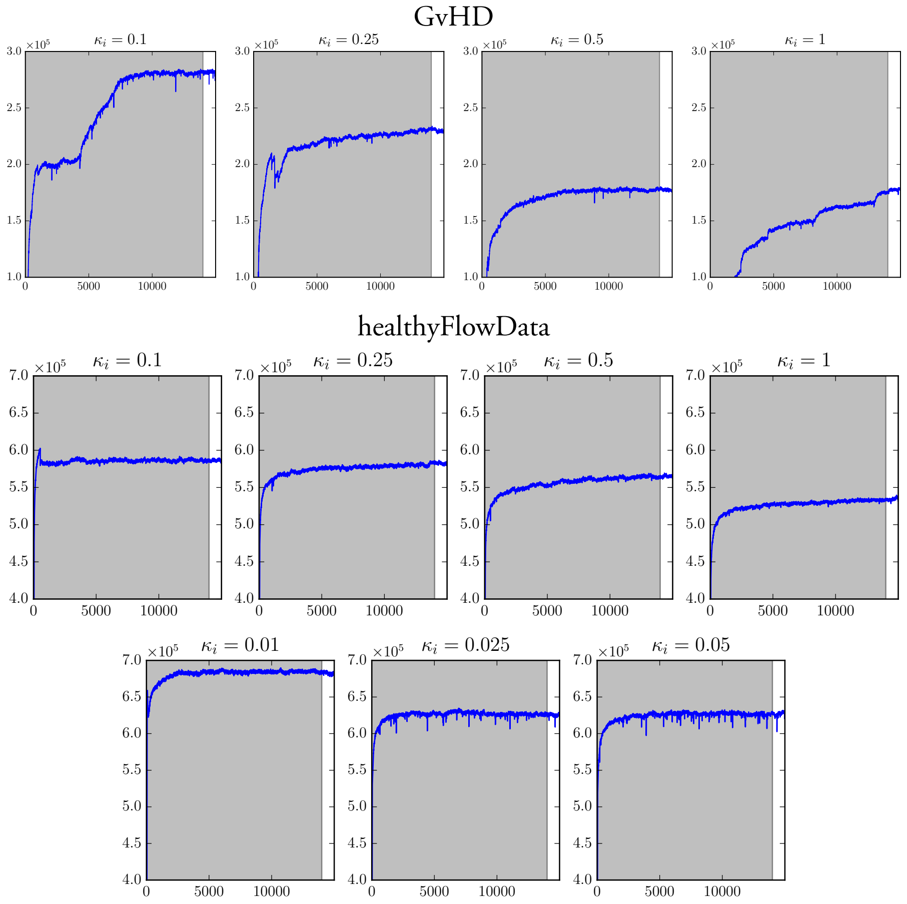


Figure S5: Trace plot of likelihoods for ASPIRE runs with different κ_i. The shaded areas show burn-in iterations.

E.6 GvHD scatterplots

E.7 Individual GMM models with EM for healthyFlowData

The variation between flow cytometry samples is systematized in the hierarchical model, results of this can be seen in Fig. 12 and Fig. 13 (a). For comparison, we fit

Gaussian mixture models using the expectation-maximization algorithm to each flow cytometry sample separately. In this case there are no clear correspondences among the mixture components between samples, as seen in Fig. S7. When the data set was studied previously with an algorithm matching populations found by separate analysis of the samples, this was only done with a coarse partition of the cell measurements, with four cell populations Azad et al. (2013).

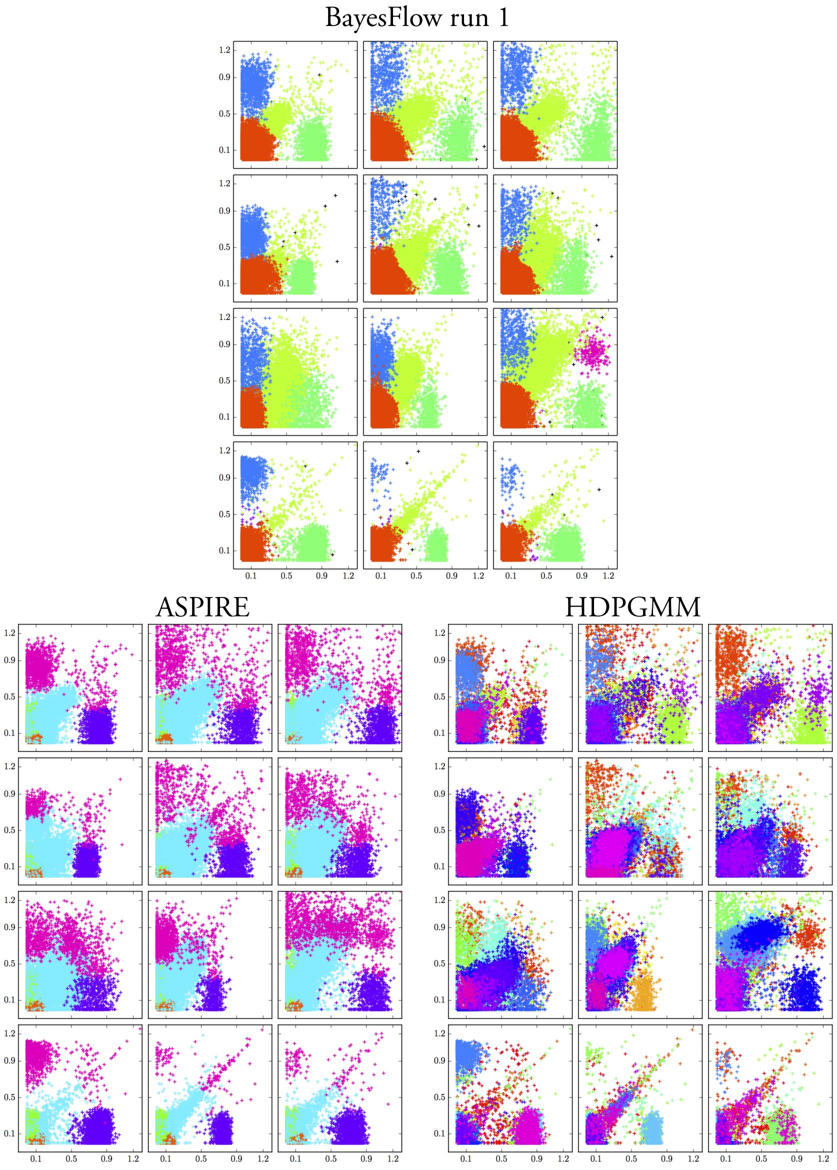


Figure S6: Gated events according to BayesFow run 1, ASPIRE and HDPGMM of the twelve samples in the GvHD dataset, projected onto the two first dimensions.

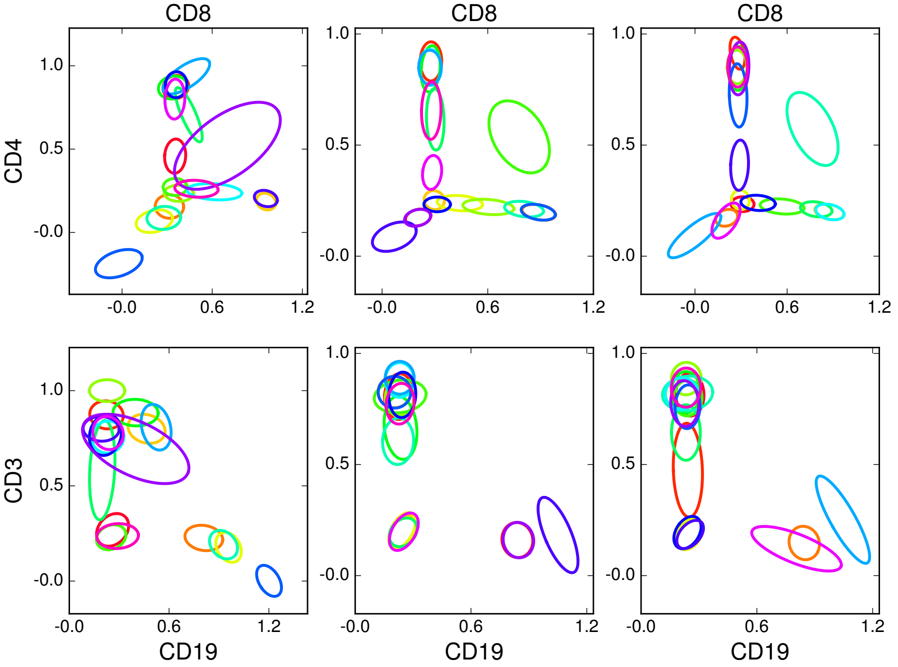


Figure S7: Component parameter representations of inferred mixture components in independent Gaussian mixture models of three flow cytometry samples. The two samples depicted in the two right columns are technical replicates. Note that there is no correspondence between colors between columns.

PAPER III

Submitted for publication.

What is a ‘unimodal’ cell population? — Investigating calibrated dip and bandwidth tests for quality control of gating of flow cytometry data

Kerstin Johnsson and Magnus Fontes

Abstract

Many automated gating algorithms for flow cytometry data are based on the concept of unimodal cell populations. This generally means that one-dimensional density estimates, such as histograms, have just one mode. It is an intuitively appealing notion that has potential applications for automated gating quality control as well as for manual gating protocols. However, there is no canonical way to make the density estimate—thus defining unimodality is not straightforward. In the statistics literature this problem has been approached from two perspectives: the dip test measuring the probability mass that needs to be transferred to render a unimodal distribution function, and the bandwidth test measuring the critical smoothing required for obtaining a unimodal density. In this paper we empirically investigate calibrated versions of the dip and bandwidth tests. We illustrate how they can be applied to flow cytometry data and show how they have complementary properties.

1 Introduction

A key problem in gating of flow cytometry data is how to distinguish when a cell subset represents one or multiple—possibly overlapping—cell populations.

In manual gating this is left to the judgment of the analyst, supported by various visualizations. In automated gating the decision is often based on the concept of unimodality or that the population should represent a single density peak (Ge and Sealfon, 2012, Naim et al., 2014, Malek et al., 2015). This allows for a large variety of cell population shapes—in particular no assumptions about Gaussianity or other distributional assumptions are made. The number of density peaks has also been used for normalization of flow cytometry data (Hahne et al., 2010) and in initial steps during automated gating (Aghaeepour et al., 2011). But measuring unimodality is not trivial. That a data set is unimodal really means that the probability density function generating the data is unimodal, so that if we had unlimited amounts of data, any histogram describing it would be unimodal. But in practice data is limited, and with a sufficiently small bin widths there will always be bumps in the histogram. There is no canonical way to choose the bin width or the smoothing when estimating the density. This paper investigates the two main approaches evaluating unimodality from the statistics literature—the dip test (Hartigan and Hartigan, 1985) and the bandwidth test (Silverman, 1981)—and use manually gated flow cytometry data from the FlowCAP I challenge (Aghaeepour et al., 2013) to relate what is seen as acceptable cell populations by the analysts to how deviations from unimodality are assessed by the tests.

Tests for unimodality have been used as part of quality control of automated gating (Johnsson et al., 2016), but could equally well be applied to manually gated populations. Quality control of identified cell populations should be an integral part of any automated gating procedure—the common practice of evaluating automated gating based on resemblance to manual gating (Aghaeepour et al., 2013) has to be substituted if automated gating is to replace manual gating. Validation must be based on gating-independent automatically computed measures; testing for unimodality gives one such measure.

The tests that are evaluated in this paper apply to univariate data. Multivariate flow cytometry data is therefore projected onto one dimension at a time before the tests are applied. This paper uses projections onto the coordinate axes, i.e. each measured dimension is considered separately. Other linear projections (Naim et al., 2014, Hennig, 2010) or non-linear projections (Ahmed and Walther, 2012) could also be used. There are approaches to direct evaluation of multimodality for multivariate data (Hartigan, 1987, Polonik, 1995, Hartigan and Mohanty, 1992, Rozál and Hartigan, 1994, Burman and Polonik, 2009, Hennig, 2010), but it is a much harder problem and it has not been shown that such methods

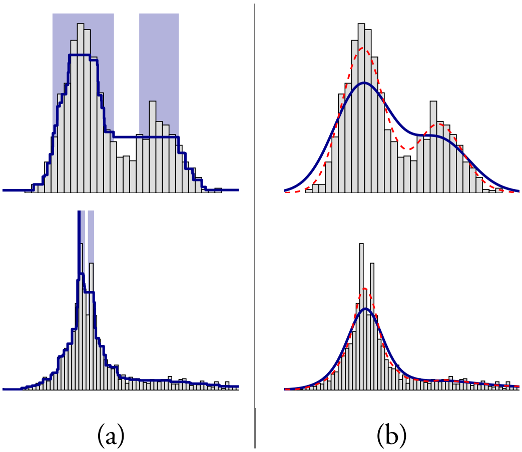


Figure 1: (a) Dip test illustration. The blue curve is the closest unimodal function according to the dip test. The shaded regions indicate the two most significant modes. (b) Bandwidth test illustration. The blue curve is the closest unimodal function according to the bandwidth test, i.e. a kernel density estimate with the critical bandwidth. The dashed red curve is the kernel density estimate with a bandwidth which is half the critical bandwidth.

have desirable theoretical properties, for example possibility to calibrate the tests to obtain a correct significance level (Cheng and Hall, 1998b, 1999, Hall and York, 2001).

The dip test and the bandwidth test takes two fundamentally different approaches to overcome the problem of density estimation. The dip test does not compute the density at all, but rather uses the empirical distribution function (the cumulative density), which is easy to estimate. The unimodal function which has the distribution function that best approximates the empirical distribution function (as measured by maximal pointwise distance) is taken as the closest unimodal function according to the dip test. The density of this unimodal function is illustrated for two data sets in Fig. 1 (a). The maximal distance between the empirical distribution function and the distribution function of the unimodal density is called the dip. Using an equivalence between the dip test and another test called excess mass test (Müller and Sawitzki, 1991, Cheng and Hall, 1998a) it is possible to find the two most influential modes in the data, shown as shaded regions in Fig. 1 (a). The dip equals half the difference in probability mass between the data and the unimodal function in the smallest mode.

The bandwidth test estimates the probability density using a kernel density estimator with a Gaussian kernel (Silverman, 1986) with the *critical bandwidth*, i.e. the smallest bandwidth that gives a unimodal density. According to the bandwidth test, this density is the closest unimodal density for the data; it is illustrated in 1 (b). As can be seen by comparing the dip and bandwidth tests for the bottom data set in Fig. 1 the two tests treat deviations from unimodality differently. The dip test finds the two most significant modes as two thin modes with high probability mass close to each other. The bandwidth test uses a critical bandwidth which smooths the entire data set to unimodality. This bandwidth is not determined by these thin modes, but by the “shoulder” in the right part of the data. Even with half the critical bandwidth the two thin modes seen by the dip test are a single mode in the kernel density estimate. In the experiments on FlowCAP I data we will later see how this “global” property of the bandwidth test often agrees with manual gater’s interpretations, but it also makes the bandwidth test sensitive to outliers and long flat regions. Outliers can be handled though by only counting modes inside a selected interval (Hall and York, 2001); in this paper we use a data-adaptive approach for selecting this interval.

To perform the calibrated dip and bandwidth tests at significance level α, data is resampled repeatedly from the respective closest unimodal densities. If the dip or the critical bandwidth for the resampled data is larger than the original dip or critical bandwidth scaled by a calibration constant λ_α in less than α of the new samples, then the data is seen having disproportionately large deviations from unimodality and the null hypothesis of unimodality is rejected (Cheng and Hall, 1999). The classical dip test (Hartigan and Hartigan, 1985) instead compares the dip of the data to dips resampled from the uniform distribution. The classical bandwidth test (Silverman, 1981) uses resampling, but instead of a calibration constant uses a rescaling to compensate for that the resampled data will have slightly higher variance.

In this paper we explain what the theoretical results regarding calibration means and make an empirical investigation of how the calibrated tests compare to the classical tests in practice. To find the necessary calibration constants we have devised a probabilistic bisection search procedure.

We also go through the special considerations that need to be taken into account when applying the dip and bandwidth tests to flow cytometry data, with regards to truncated and saturated data. Three different ways of blurring the data to counter the effects of truncation are investigated. Finally, the tests are applied to manually gated data from the FlowCAP I challenge (Aghaeepour et al., 2013). The relation between the dip test and excess mass test is used to find the size of the modal region and relate it to rejections by the dip and bandwidth tests. A test looking for flat regions in the data is applied to find differences in rejection patterns of the dip and bandwidth tests for cell populations with 'shoulders'.

In summary, this paper makes a comprehensive investigation of dip and bandwidth tests, with the aim of furthering the use of these in quality control of automatically as well as manually gated cell populations. The methods used are implemented in the Python package *modality*, available at `https://github.com/kjohnsson/modality/`.

2 Methods

2.1 Dip and critical bandwidth

We have implemented algorithms for finding the optimal dip and the critical bandwidth based on (Hartigan and Hartigan, 1985), (Hartigan, 1985) and (Silverman, 1981). The dip algorithm has been modified to explicitly find and return the closest unimodal function as well as the dip. Moreover, the connection to the excess mass test (Müller and Sawitzki, 1991, Cheng and Hall, 1998a) has been exploited to find the two most influential modes for the dip. For the classical dip test, p-values are computed using interpolation based on a table from the R package diptest (Maechler, 2015).

Implementing the bandwidth test requires computation of the number of modes of a kernel density estimate. In our implementation this is done by evaluating the density on a grid. The grid cells have width $0.05h$, where h is the bandwidth. It is shown in the Supplemental Material Section A that this ensures that no modes higher than $9h^{-1} \cdot 10^{-4}$ are missed. Kernel density estimates are computed with the Python package scikit-learn (Pedregosa et al., 2011).

2.2 Calibration

The hardest part in constructing tests for unimodality is assessing how extreme an estimated deviation is—how likely is it to occur by chance? The classical dip and bandwidth tests are conservative for most unimodal densities (as long as the bandwidth test is adjusted to disregard outliers), i.e. a p-value of 0.05 means that the probability that the deviation could have occurred by chance from a unimodal density is in fact smaller (Hartigan and Hartigan, 1985, Silverman, 1981). For the classical dip test the situation is dramatic, especially with a large number of points (Cheng and Hall, 1998b). This is expected to lead to loss of power and failure to detect multimodal distributions, which is evaluated in the experiments on synthetic data.

However, it has been shown that it is possible to calibrate the tests for restricted classes of unimodal functions, namely a class of strictly unimodal distributions (i.e. with one maximum but without inflection points) and a class of unimodal distributions with a shoulder (i.e. with one maximum and one inflection point) (Cheng and Hall, 1998b, 1999, Hall and York, 2001). What this means this that if S is the statistic under study (dip or bandwidth), S^* is the resampled statistic and λ_α is a calibration constant depending on the significance level α, the test that rejects unimodality when $P(S^*/S > \lambda_\alpha) < \alpha$—i.e. when $\lambda_\alpha S$ is among the α most extreme statistics—has asymptotically correct level. Asymptotically correct level means in practice that the null hypothesis is rejected in precisely α of the cases when S is sampled from the correct class of unimodal functions and the number of points is large. The definition of a 'large' number of points varies from test to test and between classes of unimodal functions; this is investigated below in experiments on synthetic data.

The calibration constants λ_α depend not only on α, but also on the type of test and the class of unimodal functions that is used—strictly unimodal or unimodal with shoulder. We have developed a probabilistic bisection search algorithm to determine calibration constants, detailed in the Supplemental material Section B. Following (Hall and York, 2001) we use data with 10,000 points for the calibration.

2.3 Adaptive resampling

To find out when to reject the calibrated dip or bandwidth test one needs to estimate if $P(S^*/S > \lambda_\alpha) < \alpha$, where S is the dip or critical bandwidth. The standard simplest method is to sample S^* a fixed number of times and base the rejection on this. However, some data sets require only a few samples to determine with high confidence that $P(S^*/S > \lambda_\alpha) < \alpha$, others require many more. To determine this is part of the probabilistic bisection search method for finding calibration constants (Supplementary material, Section B.2), which we reuse with the modification that we use confidence level 0.05 as a standard instead of 0.01.

2.4 Interval selection for bandwidth test

The bandwidth test is sensitive to data far out in the tails of a distribution, and if the density does go to zero fast enough, as is for example the case for the Student's t-distribution, data in the tails will very often lead to rejection of the null hypothesis (Hall and York, 2001). By only considering an interval where the probability density is sufficiently large when counting modes in the density, this can be avoided (Hall and York, 2001).

We use k-nearest-neighbor estimation of the density to select such an interval (Silverman, 1986). The interval boundaries are defined as the leftmost and rightmost points respectively where the estimated density exceeds a threshold. This density estimate is adaptive to local variations in the true data density, and is therefore a better alternative than kernel density estimation for this purpose. The parameter k should be selected such that if there are fewer than k points in a tail, they can be disregarded. For this paper we use $k = 5$ and the threshold is set to 0.2 divided by the size of the range of the data.

2.5 Blurring

Due to storage limitations, flow cytometry data are truncated. For example, the FlowCAP I data used in this paper has at most 1024 different values in each dimension. Even when the truncation is much smaller in comparison to the dynamic range, if there are clusters which only span a small part of the dynamic range they will still be heavily affected by truncation.

To counter the effect of truncation it has been recommended for flow cytometry data to add uniform noise corresponding to the size of the bins (Roederer, 2001, Bagwell and Adams, 1993), blurring the data. Minnotte (1997) proposed instead to use an algorithm called frequency polygon blurring in the context of testing for modes, motivated by that this should better preserve the modal structure of the data.

We have investigated the effects of truncation together with standard blurring, frequency polygon (FP) blurring, or a deterministic variant of FP blurring. The results are summarized in Supplemental Figure S1. In short, there are no experimental evidence for favoring any of the three kinds of blurring. We chose FP blurring for further experiments, due to attractive theoretical properties, see Supplemental Material Section C.

2.6 Extreme data points

The dip test and the bandwidth test are non-parametric tests, meaning that we assume no particular model for the data. This also means that saturated measurements or measurements below the limit of detection, i.e. measurements equal to zero or the largest possible value in the channel, will rarely add any information. On the other hand, they can mislead the tests to find modes at the extreme values. Hence we remove all such data.

3 Experiments

3.1 Data

3.1.1 Synthetic data

We use data generated from known distributions to evaluate the effects of calibration. The purpose of calibration is to ensure a given rejection rate for the class of strictly unimodal distributions or the class of distributions with a shoulder and at the same time increase the power to detect multimodal distributions. The theoretical results ensuring a certain number of rejections hold when data have a 'large' number of points (Cheng and Hall, 1999), but how many points this really is might vary from case to case and is therefore investigated empirically.

We study rejection rates for data with 100, 1000 and 10,000 points generated from seventeen unimodal distributions—the standard normal distribution and sixteen unimodal distributions with differently shaped shoulders. The shoulder distributions can be seen as limiting cases of strictly unimodal distributions—strictly unimodal distributions can get arbitrarily close to them—so they can be seen as a worst case for this class.

How the power of the dip and bandwidth tests are affected by calibration are studied on data sets—also with 100, 1000 and 10,000 points—which are close to unimodal distributions with a shoulder, but instead of a shoulder have a small bump. The size b of the bump is measured as the area between the density curve and the horizontal line through the lowest point, the valley, between the major mode and the bump, and the sizes vary between 10^{-4} and 0.03. This size can be interpreted as the proportion of data points which needs to be removed to make the distribution unimodal.

Details of all distributions used to generate data are given in the Supplemental material Section D.1. Since the distributions are known we can select appropriate intervals for the bandwidth test beforehand, how this is done is also described in the Supplemental Material Section D.1.

3.1.2 Flow cytometry data

We use the GvHD and StemCell data sets from the FlowCAP I challenge, in total 42 samples, and the additional labels provided by eight independent analysts given the same gating instructions (Aghaeepour et al., 2013), to see how the dip and bandwidth tests agree with traditional gating practice regarding what is seen as acceptable distributions. The gating instructions included that the analysts should separate out any discernible cell populations, so the individual cell populations should be homogeneous and a reasonable ground truth for unimodality. That we have labels from many analysts means that they can be compared to find disagreements and detect cell populations that are not considered homogeneous by all of the analysts.

We define a measure of gater concurrence for a given cluster by considering how the data in this cluster are partitioned by the other analysts. For each of the other analysts the proportion of the largest subcluster is taken as how much they concur with the cluster. The final concurrence value is defined as the median of this across analysts. The clusters are put into three categories: low, medium and high gater concurrence as defined by concurrence values below 0.8, between 0.8 and 0.95 and above 0.95 respectively.

Only populations with at least 10 non-saturated data points are considered and any dimension where all the non-saturated data are collapsed into a single point is disregarded. For the GvHD data this means that we study in total 2367 clusters (1462 low/610 medium/295 high concurrence) and for the StemCell data 2132 clusters (658/627/847). Both data sets have six dimensions (GvHD: FSC, SSC, CD4, CD8b, CD3, CD8 , StemCell: FSC, SSC, CD45.1, Ly64/Mac 1, Dead cells, CD45.2), to which the tests for unimodality are applied independently.

4 Results

4.1 Synthetic data

When calibration works as intended, different distributions in the calibration class generate data with similar rejection rates when the number of points is large. As seen in Fig. 2 the calibrated dip test does indeed have similar rejection rates for different shoulder distributions and for data sets with 1000 and 10,000 data points. For data sets 100 data points, rejection rates are slightly higher, but still comparable. The calibrated bandwidth test on the other hand show a large variation both for shoulder shapes and for different number of points. It seems that 10,000 is not a large enough number for the calibration of the bandwidth test.

Fig. 2 also shows that the bandwidth test calibrated against a shoulder distribution performs on average similar to the uncalibrated bandwidth test (data on individual distributions confirming this is shown in Supplemental Material, Fig. S5), whereas the uncalibrated dip test has much lower rejection rates than the calibrated versions of it. Furthermore, the dip test calibrated against a normal distribution is similar to the dip test calibrated against a shoulder distribution, as could be expected from theory (see Supplemental material Section B.1), but the different calibrations for the bandwidth test are quite different.

In general, increasing the number of data points increases the power of statistical tests, meaning that data not following the null hypothesis more often will lead to rejections. That the rejection rate for the bandwidth test increases with the number of data points, as also can be seen in Fig. 2, is aligned with the pattern one would obtain if data did not follow the null hypothesis. In a way the bandwidth test treats the data generated from shoulder distributions as if they did not follow the hypothesis of unimodality.

The main advantage of calibration, expect that one “knows what one gets” in terms of the number of rejections of the null hypothesis, is that the power to detect multimodal distributions increases. Fig. 3 shows that the calibrated dip test does indeed much more often reject unimodality for distributions that are only slightly bimodal, i.e. have a small second mode. The bandwidth test calibrated against a shoulder distribution is also in this regard similar to the uncalibrated test, but the version calibrated against the normal distribution has higher power. Fig. 3 also shows that the expected number of data points in the bump to a large extent can predict the rejection rate.

4.2 Flow cytometry data

The uncalibrated dip and bandwidth tests, as well as the tests calibrated against a shoulder distribution were applied to the flow cytometry data, using significance level 0.05. For the resampling-based tests (i.e. all tests except the uncalibrated dip test) adaptive sampling was used with a maximum of 12,800 resampled data

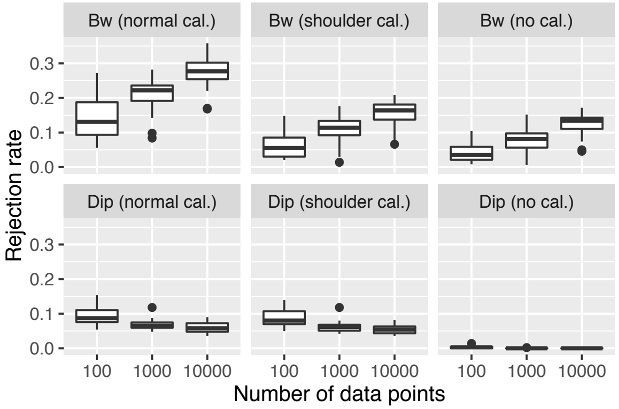


Figure 2: Rejection rates at level 0.05 for data from 16 different shoulder distributions, for dip and bandwidth (bw) tests, uncalibrated or calibrated against a normal (i.e. strictly unimodal) distribution or a shoulder distribution. Each box plot summarizes the 16 rejection rates for the different distributions, where each rejection rate is computed based on 500 data sets sampled from one distribution. Each test is performed at significance level 0.05, with 1000 resamples for the resampling-based tests (i.e. no adaptive sampling). The used distributions are shown in Fig. S2 (a) in the Supplemental material.

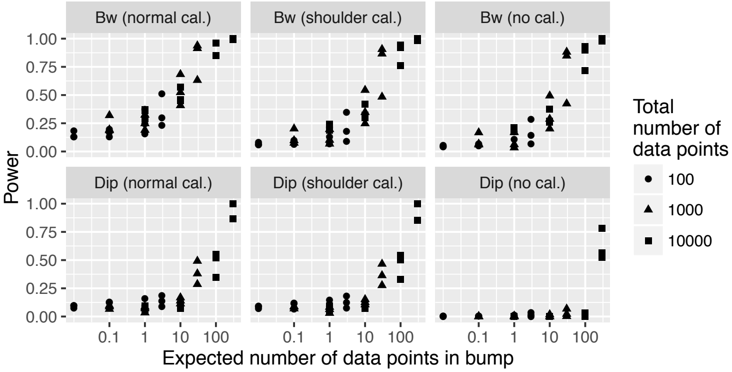


Figure 3: Power (rejection rate) to detect multimodality in data sets where the second mode is a small bump, for dip and bandwidth tests performed at significance level 0.05 with different calibrations. The bump size b is defined in Section 3.1.1 and takes the values 0.001, 0.001, 0.01 and 0.03. The expected number of points in the bump is b times the total number of data points. All distributions used are shown in the Supplemental material, Fig. S2 (b). For the resampling-based tests, 1000 resamples were used (i.e. no adaptive sampling).

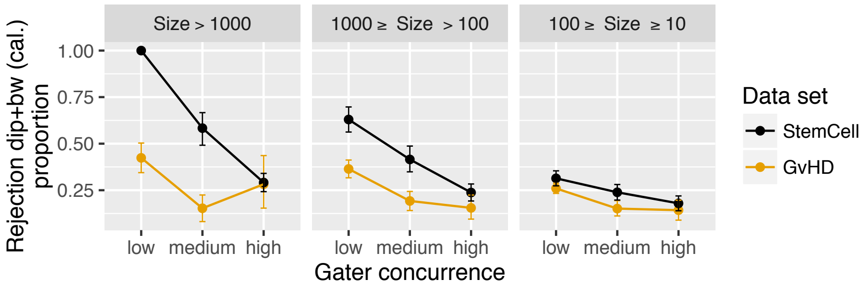


Figure 4: Rates of rejection by both calibrated dip and calibrated bandwidth tests (shoulder reference) in at least one dimension, split by gater concurrence and cluster size. The gater concurrence categories are defined in Section 3.1.2. The error bars show Jeffrey's confidence interval (Brown et al., 2001) computed with the R package binom (Dorai-Raj, 2014).

sets. The proportion of undecided tests at the maximum number of samples was 0.7%.

A great majority of the cluster dimensions are accepted for each test (Supplemental material Fig. S7). For the most part the tests agree whether to reject or accept unimodality of a cluster dimension: The calibrated bandwidth and dip test disagree on whether to accept or reject in 14% of cluster dimensions, the calibrated and uncalibrated dip test disagree in 7.6% of cases, and the calibrated and uncalibrated bandwidth test disagree in 1.5% of cases. The disagreements between the dip and the bandwidth tests are investigated further below. How the simultaneous rejections by the dip and bandwidth tests relate to cluster size and gater concurrence is shown in Fig. 4.

Fig. 4 shows that when gater concurrence increase, there is a significant reduction in the proportion of clusters that are rejected by both the calibrated dip and bandwidth tests, for all size categories and both data sets except for the clusters with more than 1000 points for the GvHD data set. However, this category is small, with only 39 clusters. Also, the differences in rejections between low and high concurrence increase with the number of data points—additional data points give more power to detect multimodality (cf. Fig. 3). Fig. S8 in Supplemental material shows that similar patterns hold for each of the four tests independently.

To investigate what characteristics of the clusters that lead the tests to reject unimodality we have investigated the size of modal regions and evidence for flat

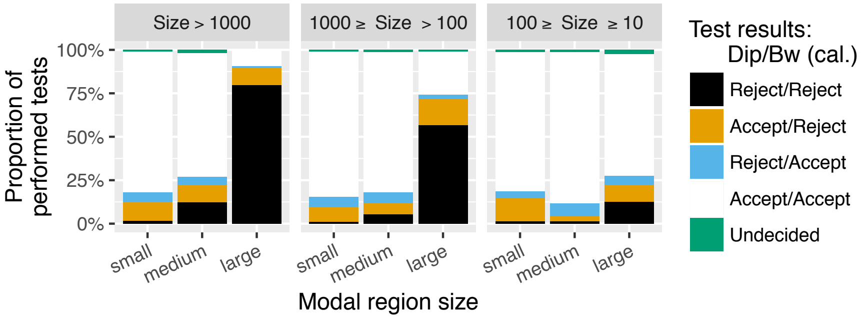


Figure 5: Comparison of calibrated dip and bandwidth tests (shoulder reference) for FlowCAP I data, split by modal region size and data set size. The modal region is defined as the span from the leftmost part of the left mode to the rightmost part of the right mode, where the modes are found using excess mass (Müller and Sawitzki, 1991, Cheng and Hall, 1998a). A small modal region is less than 30% of the data range, a medium modal region is between 30% and 70% of the data range, and a large modal region is more than 70% of the data range.

parts in the data. The modal region size is based on the modes found by the excess mass test (equivalent to the dip test) (Müller and Sawitzki, 1991), and it is defined as the span from the leftmost part of the left mode to the rightmost part of the right mode. Flat parts of the data are found by comparing portions of the data to the uniform distribution using the Anderson-Darling test (Anderson and Darling, 1952).

Fig. 5 shows that for data sets with more than a hundred data points there is a striking difference in rejection patterns between those with differently sized modal regions. A large modal region means high rates of concurrent rejections and that rejection by dip test together with acceptance by bandwidth test is uncommon. Furthermore, for data sets with small modal regions it is uncommon with concurrent rejections by dip and bandwidth test. These are the data sets where rejections are most different between the bandwidth and dip test.

This subset of data sets with small modal regions and for which the calibrated dip and bandwidth tests disagree is tested for flat regions, the result is shown in Fig. 6. For data sets with more than a hundred data points, those who have a part which resembles the uniform distribution more ($p > 10^{-5}$), i.e. show evidence of having a flat part or a shoulder, are often rejected by the bandwidth test while

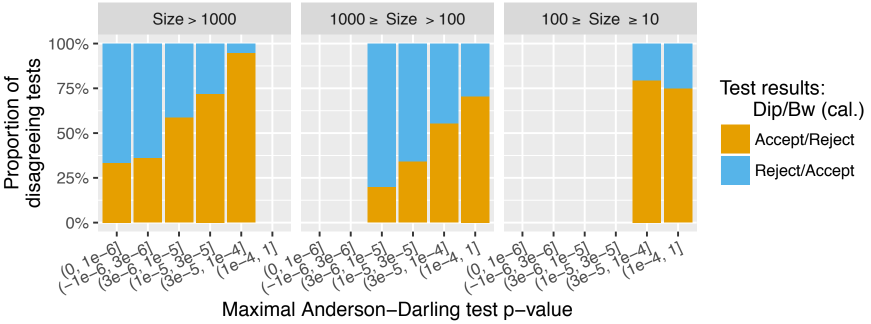


Figure 6: Test results for calibrated dip and bandwidth test when they disagree, for data sets with small modal regions, partitioned by evidence of flat regions as measured by Anderson–Darling test p-values for comparing empirical distributions of cluster portions to the uniform distribution. The p-values are computed using the R package goftest (Faraway et al., 2015). The cluster portions considered span at least 20% of the interval selected as described in Section 2.4 (which excludes outliers) and contain at least 1% of the data and at least 5 data points. The maximal p-value over all such portions inside the selected interval is used for the partitioning.

accepted by the dip test. For those who do not have such a part the situation is reversed.

5 Discussion

Unimodality is an attractive concept for describing homogeneous cell populations, that has been used in automated gating in a variety of settings. In the statistics literature unimodality has been well studied, mainly through the dip test and bandwidth test approaches. The standard dip and bandwidth tests are more conservative than the given significance level suggests though, especially the dip test. From theoretical results for the dip and bandwidth tests (Cheng and Hall, 1999), it should be possible to calibrate them to give accurate levels, at least for data sets with a large number of points. Experiments on synthetic data show that for the dip test this works well with data set sizes one can expect in flow cytometry data, whereas the bandwidth test requires a larger number of data points for the calibration to work as intended. Further experiments showed that calibration increased the power to detect multimodal distributions for the dip test, whereas for the bandwidth test the power did not change so much.

Calibrating the dip test with a shoulder distribution as reference or the normal distribution did not make a large difference. However, for the bandwidth test the choice of reference distribution had a large impact, showing that it is sensitive to whether the distribution generating the data has a shoulder or not; sensitivity to the shape of the shoulder is further shown in the Supplemental Material, Fig. S4.

Before applying the dip and bandwidth tests to flow cytometry data careful considerations need to be taken regarding truncation, saturated data, measurements below the limit of detection and outliers. The steps outlined in this paper can serve as a guide. Also the usual considerations regarding preprocessing and transformations of flow cytometry data have to be taken into account. It is important to note that transformations like logicle (Parks et al., 2006) does not necessarily preserve the number of modes of the density.

Both the dip test and the bandwidth test agreed to a large extent with the gaters' assessment that the tested flow cytometry clusters were homogeneous, and that both tests rejected unimodality showed a strong relation to whether the majority of the gaters agreed that a cluster was homogeneous. For large or moderately sized clusters, a large modal region, i.e. two modes far apart or two wide modes, was a strong indication that both tests would reject unimodality. For small modal regions there were more disagreements—the bandwidth test gave more rejections when there were more evidence for a flat region in the data, thus causing the modal region to be squished into another region, and the dip test gave more rejections with less evidence of a flat part, thus indicating that the modal region was part of a larger mode. Example histograms where the dip test and the bandwidth test differ are shown in Figs. S9 and S10 in the Supplemental Material.

Clusters with flat regions, which were accepted only by the dip test, as well as clusters with local modes inside a larger mode, which were accepted only by the bandwidth test, had been accepted in the manual gating. This showed that the dip and the bandwidth tests have complementary elements for matching manually gated populations.

However, one might want to use a different interpretation of homogeneous clusters than used by the manual gaters of the FlowCAP I data. The results in this paper can be used to guide selection of a test for unimodality that agrees with the desired interpretation. For example, if distributions with shoulders are not acceptable, the bandwidth test should be used, calibrated with a strictly uniform distribution as a reference.

In this paper the tests were performed consistently at significance level 0.05. A researcher might want to use other levels depending on whether it is more important to detect multimodal populations or to avoid false positives.

6 Acknowledgments

KJ would like to thank Magnus Linderoth, Dragi Anevski and Jonas Wallin for interesting discussions regarding the dip test and Magnus Linderoth for discussion of experimental design and adaptive probabilistic bisection search.

The computations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at Lunarc, Lund University.

The authors declare no conflict of interest.

Bibliography

N. Aghaeepour, R. Nikolic, H. H. Hoos, and R. R. Brinkman. Rapid cell population identification in flow cytometry data. *Cytometry Part A*, 79(1):6–13, 2011.

N. Aghaeepour, G. Finak, The FlowCAP Consortium, The DREAM Consortium, H. Hoos, T. R. Mosmann, R. Brinkman, R. Gottardo, and R. H. Scheuermann. Critical assessment of automated flow cytometry data analysis techniques. *Nature Methods*, 10(3):228–238, 2013.

M. O. Ahmed and G. Walther. Investigating the multimodality of multivariate data with principal curves. *Computational Statistics & Data Analysis*, 56(12): 4462–4469, 2012.

T. W. Anderson and D. A. Darling. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *The annals of mathematical statistics*, 23(2):193–212, 1952.

C. B. Bagwell and E. G. Adams. Fluorescence spectral overlap compensation for any number of flow cytometry parameters. *Annals of the New York Academy of Sciences*, 677(1):167–184, 1993.

L. D. Brown, T. T. Cai, and A. DasGupta. Interval estimation for a binomial proportion. *Statistical science*, pages 101–117, 2001.

P. Burman and W. Polonik. Multivariate mode hunting: Data analytic tools with measures of significance. *Journal of Multivariate Analysis*, 100(6):1198–1218, 2009.

M.-Y. Cheng and P. Hall. On mode testing and empirical approximations to distributions. *Statistics & Probability Letters*, 39(3):245–254, 1998a.

M.-Y. Cheng and P. Hall. Calibrating the excess mass and dip tests of modality. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(3): 579–589, 1998b.

M.-Y. Cheng and P. Hall. Mode testing in difficult cases. *The Annals of Statistics*, 27(4):1294–1315, 1999.

S. Dorai-Raj. *binom: Binomial Confidence Intervals For Several Parameterizations*, 2014. URL `https://CRAN.R-project.org/package=binom`. R package version 1.1-1.

J. Faraway, G. Marsaglia, J. Marsaglia, and A. Baddeley. *goftest: Classical Goodness-of-Fit Tests for Univariate Distributions*, 2015. URL `https://CRAN.R-project.org/package=goftest`. R package version 1.0-3.

Y. Ge and S. C. Sealfon. flowpeaks: a fast unsupervised clustering for flow cytometry data via k-means and density peak finding. *Bioinformatics*, 28(15): 2052–2058, 2012.

F. Hahne, A. H. Khodabakhshi, A. Bashashati, C.-J. Wong, R. D. Gascoyne, A. P. Weng, V. Seyfert-Margolis, K. Bourcier, A. Asare, T. Lumley, et al. Per-channel basis normalization methods for flow cytometry data. *Cytometry Part A*, 77(2): 121–131, 2010.

P. Hall and M. York. On the calibration of silverman's test for multimodality. *Statistica Sinica*, 11(2):515–536, 2001.

J. A. Hartigan. Estimation of a convex density contour in two dimensions. *Journal of the American Statistical Association*, 82(397):267–270, 1987.

J. A. Hartigan and P. M. Hartigan. The dip test of unimodality. *The Annals of Statistics*, 13(1):70–84, 1985.

J. A. Hartigan and S. Mohanty. The runt test for multimodality. *Journal of Classification*, 9(1):63–70, 1992.

P. M. Hartigan. Algorithm AS 217: Computation of the dip statistic to test for unimodality. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 34(3):320–325, 1985.

C. Hennig. Methods for merging gaussian mixture components. *Advances in data analysis and classification*, 4(1):3–34, 2010.

K. Johnsson, J. Wallin, and M. Fontes. Bayesflow: latent modeling of flow cytometry cell populations. *BMC Bioinformatics*, 17(25):1–16, 2016.

M. Maechler. *diptest: Hartigan's Dip Test Statistic for Unimodality - Corrected*, 2015. URL `https://CRAN.R-project.org/package=diptest`. R package version 0.75-7.

M. Malek, M. J. Taghiyar, L. Chong, G. Finak, R. Gottardo, and R. R. Brinkman. flowdensity: reproducing manual gating of flow cytometry data by automated density-based cell population identification. *Bioinformatics*, 31(4):606–607, 2015.

E. Mammen, J. S. Marron, and N. I. Fisher. Some asymptotics for multimodality tests based on kernel density estimates. *Probability Theory and Related Fields*, 91(1):115–132, 1992.

M. C. Minnotte. *A test of mode existence with applications to multimodality*. PhD thesis, Rice University, 1993.

M. C. Minnotte. Nonparametric testing of the existence of modes. *The Annals of Statistics*, 25(4):1646–1660, 1997.

D. W. Müller and G. Sawitzki. Excess mass estimates and tests for multimodality. *Journal of the American Statistical Association*, 86(415):738–746, 1991.

I. Naim, S. Datta, J. Rebhahn, J. S. Cavenaugh, T. R. Mosmann, and G. Sharma. Swift—scalable clustering for automated identification of rare cell populations in large, high-dimensional flow cytometry datasets, part 1: Algorithm design. *Cytometry Part A*, 85(5):408–421, 2014.

D. R. Parks, M. Roederer, and W. A. Moore. A new "logicle" display method avoids deceptive effects of logarithmic scaling for low signals and compensated data. *Cytometry Part A*, 69(6):541–551, 2006.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

W. Polonik. Measuring mass concentrations and estimating density contour clusters-an excess mass approach. *The Annals of Statistics*, 23(3):855–881, 1995.

M. Roederer. Spectral compensation for flow cytometry: visualization artifacts, limitations, and caveats. *Cytometry*, 45(3):194–205, 2001.

G. P. M. Rozál and J. A. Hartigan. The map test for multimodality. *Journal of classification*, 11(1):5–36, 1994.

B. W. Silverman. Using kernel density estimates to investigate multimodality. *Journal of the Royal Statistical Society. Series B (Methodological)*, 43(1):97–99, 1981.

B. W. Silverman. *Density estimation for statistics and data analysis*, volume 26 of *Monographs on Statistics and Applied Probability*. Chapman & Hall/CRC, London, New York, 1986.

A Grid spacing for estimating critical bandwidths

We define the size of a mode as the area between the density curve and the horizontal line extending from the highest of the two neighboring local minima (valleys). We will consider grids with spacing ah, where h is the bandwidth. For the case $h = 1$ we will show that we cannot miss modes higher than $2a^2\sqrt{2/\pi}\exp(-3/2)$, measured from the highest adjacent valley. Since every kernel density estimate with bandwidth h is a rescaled version of a kernel density estimate with bandwidth 1, where the height changes by a factor $1/h$ and the width changes by a factor h, with bandwidth h we cannot miss modes higher than $2h^{-1}a^2\sqrt{2/\pi}\exp(-3/2)$. For $a = 0.05$ this gives a bound of $9 \cdot 10^{-4} \cdot h^{-1}$.

The key element of the proof is that, since we use Gaussian kernels, a kernel density estimate with bandwidth 1 has a second derivative which is absolutely bounded by $\sqrt{2/\pi}\exp(-3/2)$, which follows immediately from that the standard normal density has a second derivative bounded by this value.

Suppose that we compute the values of a kernel density estimate with bandwidth 1 at grid points spaced a apart and use these values to find local maxima, i.e. modes. If there is a mode at x' that is not found by this method, then there must be consecutive grid points x_1, x_2 and x_3 such that $x' \in [x_1, x_2]$ and $f(x_1) \leq f(x_2) \leq f(x_3)$ or $x' \in [x_2, x_3]$ and $f(x_1) \geq f(x_2) \geq f(x_3)$. By symmetry we can assume that $x' \in [x_1, x_2]$ and $f(x_1) \leq f(x_2) \leq f(x_3)$. Since $f'(x') = 0$, $f''(x') \leq 0$ and $f(x_3) \geq f(x_2)$ there must be a point $x'' \in [x', x_3]$ which is a neighboring local minimum (i.e. with $f'(x'') = 0$). The height of the mode will then be at most

$$\begin{aligned} f(x') - f(x'') &\leq 2a \cdot \max_{x \in [x', x'']} f'(x) \\ &\leq 2a^2 \cdot \max_{x \in [x', x'']} f''(x) \leq 2a^2\sqrt{2/\pi}\exp(-3/2), \end{aligned}$$

where the second inequality follows from that $f'(x') = f'(x'') = 0$.

B Calibration

The calibration constants λ_α are defined by

$$P_{\mathcal{X}}(P_{\mathcal{X}^*|\mathcal{X}}(S^*/S > \lambda_\alpha) < \alpha) = \alpha,$$

where S is the statistic under study, i.e. either the dip Δ or the critical bandwidth h_{crit}, with the underlying data $\mathcal{X}$ sampled from a reference distribution, and S^* is the statistic computed from the resampled data $\mathcal{X}^*$ (Cheng and Hall, 1999, Hall and York, 2001). Following Cheng and Hall (1999), we use as a reference the standard normal distribution for the class of strictly unimodal densities and the mixture

$$\frac{1}{17}N(-1.25, 0.25^2) + \frac{16}{17}N(0, 1). \tag{1}$$

for the class of unimodal densities with a shoulder.

To determine λ_α we use either a probabilistic bisection search strategy or—when this is infeasible—an adaptive version that takes into account how hard it is to distinguish between different possible values of λ_α.

B.1 Strict unimodality or shoulder?

Cheng and Hall (1998b) showed that $n^{3/5}\Delta \to C_1R_1$ as $n \to \infty$ when the density belongs to the class of strictly unimodal distribution functions, where C_1 is a constant which cancels when dividing by the resampled statistic Δ^* and R_1 is a random variable that does not depend on the density as long as it belongs to the same class. For the class of unimodal distributions with a shoulder the corresponding limit is instead $n^{4/7}\Delta \to C_2R_2$ (Cheng and Hall, 1999). Since $4/7 \approx 0.57 < 3/5 = 0.6$, the dips from densities with shoulders will be larger than the dips from strictly unimodal densities. For the bandwidth test, $n^{1/5}h_{\text{crit}} \to C_3R_3$ for strictly unimodal densities (Mammen et al., 1992) and $n^{1/7}h_{\text{crit}} \to C_4R_4$ for densities with a shoulder (Cheng and Hall, 1999).

Thus unimodal distributions with a shoulder will be rejected for data sets with many data points if calibration is done with strictly unimodal distributions. Since distributions with a shoulder are by definition unimodal, but closer to bimodal than strictly unimodal distributions, from a statistical viewpoint it is more accurate to use shoulder distributions as a reference. However, since 0.57 is not far from 0.6, for the dip test it is reasonable to assume that the calibration will not make a large difference, as is verified by the experiments in the article.

It might be confusing that there are different convergence rate for the two classes of unimodal densities since strictly unimodal densities can come arbitrarily close to densities with a shoulder. For a strictly unimodal density that "almost" has a shoulder, it will behave as a density with a shoulder as long as there are not sufficiently many data points to distinguish it from a density with a shoulder. However, as $n \to \infty$ it will eventually get the convergence rate of a strictly unimodal density.

B.2 Probabilistic bisection search

For the probabilistic bisection search for λ_α we need to evaluate whether a proposed value t is larger or smaller than λ_α, i.e. we need to test the hypothesis $H_0 : \lambda_\alpha = t$ versus the alternative hypotheses $H_1 : \lambda_\alpha < t$ and $H_2 : \lambda_\alpha > t$. Have we decided that $\lambda_\alpha < t$ we set t as a new upper bound for λ_α; lower bounds are set in an analogous manner. Given upper and lower bounds for λ_α, the next proposal t is the mean of these two.

If we let

$$P_{\mathcal{X}}(P_{\mathcal{X}^*|\mathcal{X}}(S^*/S > t) < \alpha) = \beta,$$

H_0 translates to $\beta = \alpha$, H_1 translates to $\beta > \alpha$ and H_2 translates to $\beta < \alpha$.

If $\{\mathcal{X}, \dots, \mathcal{X}_N\}$ are iid samples from the reference distribution we get that

$$Z_N = \sum_{i=1}^{N} I(P_{\mathcal{X}^*|\mathcal{X}_i}(S^*/S > t) < \alpha) \sim \mathrm{Bin}(N, \beta).$$

We reject H_0 in favor of H_1 if we observe a value z_n such that $P(Z_N \geq z_N) < 0.01$ under the null hypothesis $\alpha = \beta$. Similarly we reject H_0 in favor of H_2 if $P(Z_N \leq z_N) < 0.01$ under the null hypothesis. If we cannot reject the null hypothesis, we increase N and redo the test. We stop the search when we have an upper and lower bound for λ_α within 0.01 of each other.

In order to compute z_N, we need to evaluate whether $P_{\mathcal{X}^*|\mathcal{X}_i}(S^*/S > \lambda_\alpha) < \alpha$ or not. Letting $P_{\mathcal{X}^*|\mathcal{X}_i}(S^*/S > \lambda_\alpha) = \gamma$ we assert that this is true if we can reject the null hypothesis $H_0' : \alpha = \gamma$ in favor of $H_1' : \alpha > \gamma$ at significance level 0.01 and we assert that this is false if we can reject H_0' in favor of $H_2' : \alpha < \gamma$ at significance level 0.01. The tests are performed in an analogous way to the tests between H_0, H_1 and H_2, with independent samples $\mathcal{X}_{ij}^*$ from the critically smoothed density of $\mathcal{X}_i$ in case of the bandwidth test and from the closest unimodal distribution in case of the dip test. Since this test has to be performed very many times, to gain speedup we put a bound $N_{\max}$ on N, and if we do not have confidence for H_1 or H_2 at level 0.01 when we have $N_{\max}$ samples, we randomly select, with equal probabilities, if we should assert H_1 or H_2. This happens in approximately 1% of the tests.

B.3 Adaptive probabilistic bisection search

When testing H_0 versus H_1 or H_2, a very large number of samples $\mathcal{X}$ might be required. For example, if $\alpha = 0.01$ and $\beta = 0.009$, we will most likely need a value of N around 50,000 before we can say that $\beta < \alpha$. In practice, this amount of precision will rarely be noticed. Hence we construct an interval $[\underline{\alpha}, \overline{\alpha}]$ such that it would require around 5000 tests to distinguish between $\underline{\alpha}$ or $\overline{\alpha}$ and α. For this we use quantiles of $\mathrm{Bin}(5000, \alpha)$; we set $\underline{\alpha} = q^{\alpha}_{0.05}$ and $\overline{\alpha} = q^{\alpha}_{0.95}$.

We stop the bisection search when we are sufficiently sure that $\beta \in [\underline{\alpha}, \overline{\alpha}]$. 'Sufficiently sure' is here defined as: We are confident at a 0.01 level that $\beta < \overline{\alpha}$ and also confident at a 0.05 level that $\beta > \underline{\alpha}$ or vice versa. We have a less strict confidence level for the second bound, since this is not subject to multiple testing during the bisection search.

If we have determined at level 0.01 that $\beta < \overline{\alpha}$, but we cannot say at level 0.05 that $\beta > \underline{\alpha}$, we increase N if with the current estimated $\hat{\beta} = Z_N/N$ we would get significance with fewer than 10,000 tests. If this is not the case we set the current value of t as a lower bound for λ_α and continue the bisection search.

C Blurring

To test the effect of different variants of blurring, we used the shoulder density (1), since this is precisely on the boundary between unimodal and bimodal and should be more severely affected by blurring than other densities. Apart from standard blurring, i.e. adding uniform noise, and frequency polygon (FP) blurring (Minnotte, 1993), which resamples the data in each bin following a density estimated by linear interpolation of the data in the bin and its neighboring bins, we also use a deterministic variant of FP blurring that spreads data evenly according to the frequency polygon density estimate. We call this variant deterministic FP blurring.

The dynamic range of the data as compared to the truncation, and the number of data points are the two variables that we believe will influence the effect of blurring most. The dynamic range was varied by linearly scaling the data so that the interval $[-3, 3]$ (within which most data lies) was transported to $[0, R]$, where the range R took the values 10, 100, 1000 and 10,000. Then data was truncated to integer values before blurring was applied.

The test results for the dip and the bandwidth tests on the original data was compared to test results after truncation and blurring, on 200 generated data

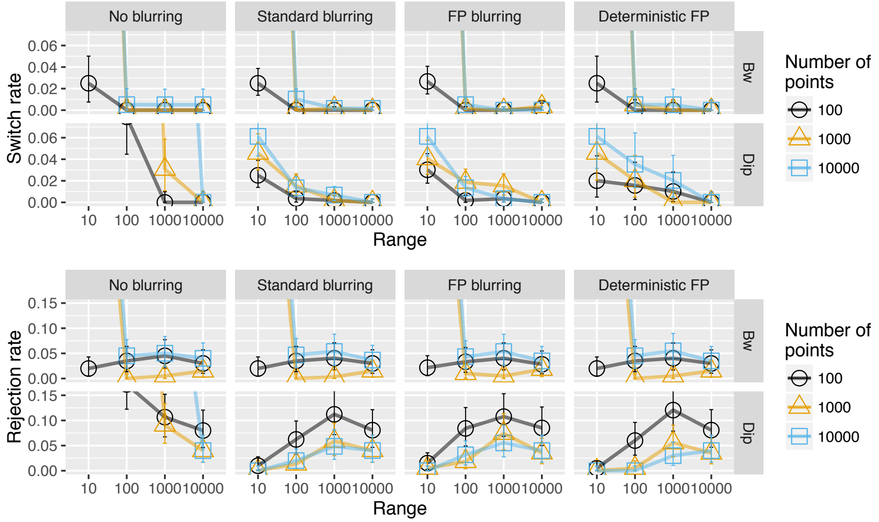


Figure S1: Top: Proportion of switches from rejection to acceptance of unimodality and vice versa by truncation and blurring, for the calibrated dip and bandwidth tests (shoulder reference). Bottom: Rejection rates for truncated and blurred data, to be compared with Fig. S3 which shows the same result for non-truncated data in the 'Shoulder' panel. Error bars show the Jeffreys 95% confidence interval (Brown et al., 2001) computed using the R package binom (Dorai-Raj, 2014).

sets with three replicates of blurring. The adaptive resampling was done three consecutive times on the original data, and data sets which were not decided after the maximum number of samples, or which had differing results in the three rounds were discarded (1.1% of the data sets). Also for tests on blurred data, only tests result that were 'accept' or 'reject' were considered.

Fig. S1 shows that blurring is essential for the dip test when the range is less than 10,000 for 10,000 data points or less than 1000 for 100 or 1000 data points. Without it the test result switches in a very large number of cases. It can also be seen that no blurring variant works for the bandwidth test if the range is only 10 and the number of data points is 100 or 1000. The three blurring methods perform similarly and it not clear from the experiments that it is advantageous to use one instead of another.

The FP blurring algorithm is based on a density estimate which is smoother than the histogram density estimate on which standard blurring is based (Minnotte, 1997). Therefore, it should in general yield a better approximation of the original data than standard blurring. Deterministic FP blurring approximates data as well as the non-deterministic variant, but it changes the distribution radically in a qualitative manner, hence we recommend to use it with caution. The advantage of using the deterministic variant is of course that no additional randomness is introduced into the testing procedure.

D Data

D.1 Synthetic data

All synthetic data are generated from mixtures of the from

$$f(x; a, w_1, w_2, \sigma) = w_1 N(x; 0, 1) + w_2 N(x; a, \sigma^2),$$

where $w_1 + w_2 = 1$.

To find shoulder distributions we fix the ratio w_1/w_2 and σ and solve numerically the system of equations

$$\begin{cases} f'(x_0; a, w_1, w_2, \sigma) = 0 \\ f''(x_0; a, w_1, w_2, \sigma) = 0 \end{cases}$$

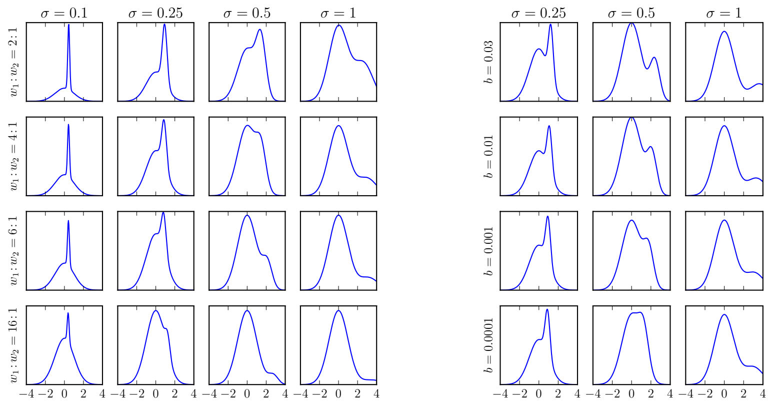
for x_0 and a so that $f(x; a, w_1, w_2, \sigma)$ has a shoulder point at x_0. The resulting distributions are shown in Fig. S2 (a). To find distributions with bumps of given sizes we define a function $B(f)$ giving the size of the minor mode for a density f and solve numerically $B(f) = b$, where b is the wanted bump size. Fig. S2 (b) shows the bump distributions used.

For the bandwidth test we set the interval where to look for modes to $[-1.5, a+1]$.

E Synthetic data results

E.1 Reference distributions

In Fig. S3, rejection rates for the distributions used for calibration are shown. The calibration is done with data sets with 10,000 points, and for these data sets the rejection rate is 0.05 for the tests with the corresponding calibration.



(a) (b)

Figure S2: Mixtures of the form $f(x) = w_1 N(x; 0, 1) + w_2 N(x; a, \sigma^2)$. (a) Densities with shoulders. (b) Densities with bumps (minor modes) of size b and shoulder ratio $w_1 : w_2 = 4 : 1$.

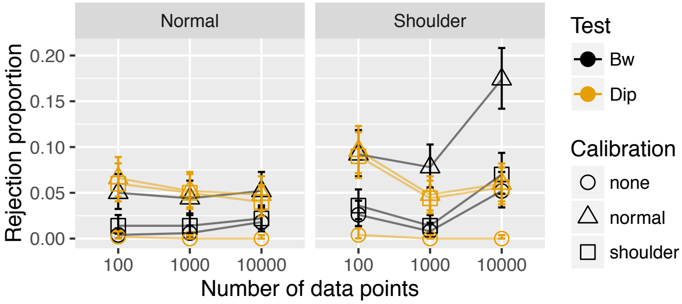


Figure S3: Rejection ratios for reference distributions: the standard normal distribution for strictly unimodal null hypothesis, and the mixture 1 for shoulder null hypothesis. For each number of data points, 500 data sets are drawn from each reference distribution. The error bars show Jeffrey's confidence interval (Brown et al., 2001) computed with the R package binom (Dorai-Raj, 2014).

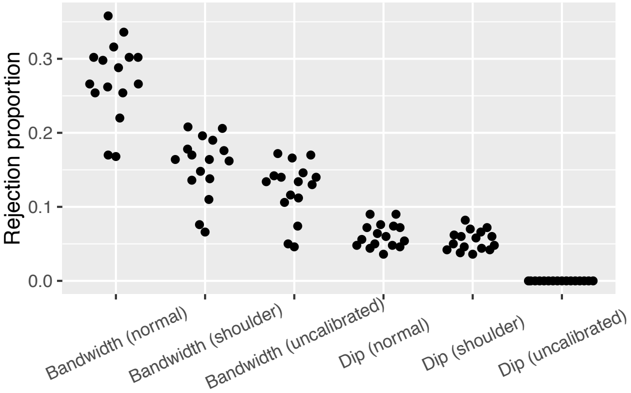


Figure S4: Sensitivity of rejection rate to shoulder shape. Each dot shows the rejection rate across 500 data sets with 10,000 points each, from each of the 16 shoulder densities depicted in Fig. S2 (a).

E.2 Sensitivity to shoulder shape and number of points

Fig. S4 shows that the bandwidth test is more sensitive than dip test to shoulder shape. Fig. S5 shows how number of data points and shoulder shape interact.

E.3 Power

Fig. S6 shows how shoulder shape and number of points interact for the bimodal densities with a small second mode. Note that for $b = 10^{-4}$ the bump has on average only one point for 10,000 data points, and rejection rates are similar to those for the corresponding shoulder distributions.

F Flow cytometry data results

Fig. S7 compares test results for the calibrated dip and bandwidth tests and for the calibrated versus uncalibrated dip and bandwidth tests. The patterns seen are similar across data set sizes.

Fig. S8 shows for the calibrated and uncalibrated dip and bandwidth tests how rejection rates vary with gater concurrence.

Figs. S9 and S10 show examples of data distributions where the dip test and

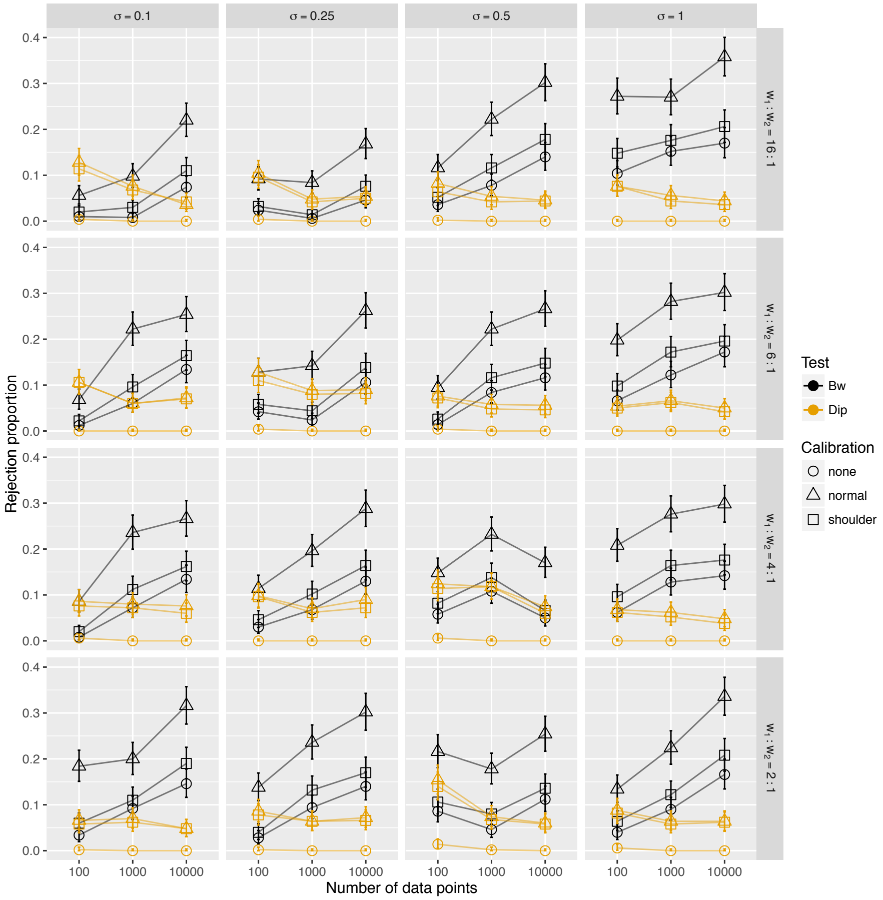


Figure S5: Rejection rates for data sets with 100, 1000 and 10,000 data points sampled from the 16 shoulder distributions depicted in Fig. S2 (a). Each rate is based on 500 data sets. The error bars show Jeffrey's confidence interval.

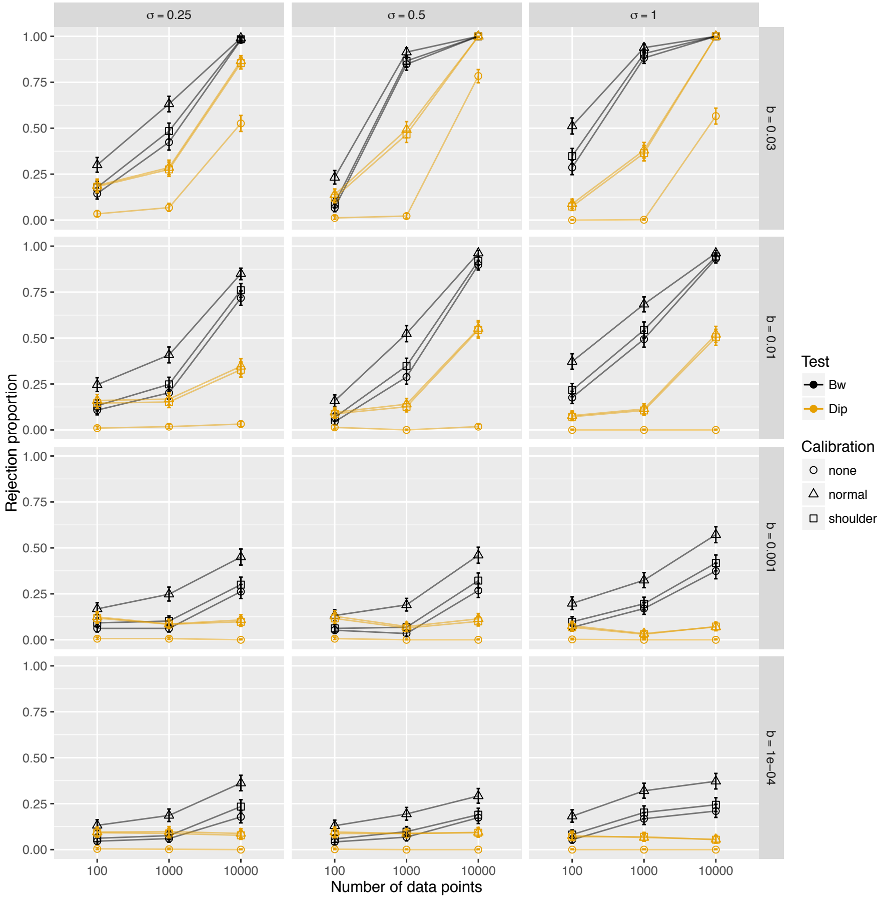


Figure S6: Rejection rates for data sets with 100, 1000 and 10,000 data points sampled from the 16 densities with a bump depicted in Fig. S2 (b). Each rate is based on 500 data sets. The error bars show Jeffrey's confidence interval. For definition of the bump size b, see Section 3.1.1.

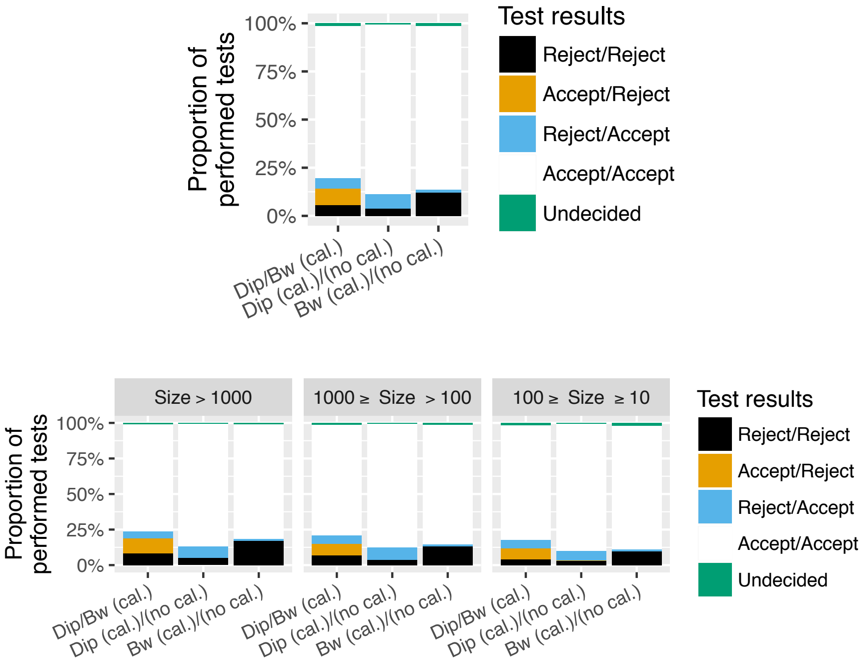


Figure S7: Agreement and disagreements between tests on FlowCAP I data. All tests are performed at significance level 0.05. Top: All cluster dimensions, irrespective of data set size. Bottom: Data sets partitioned by size.

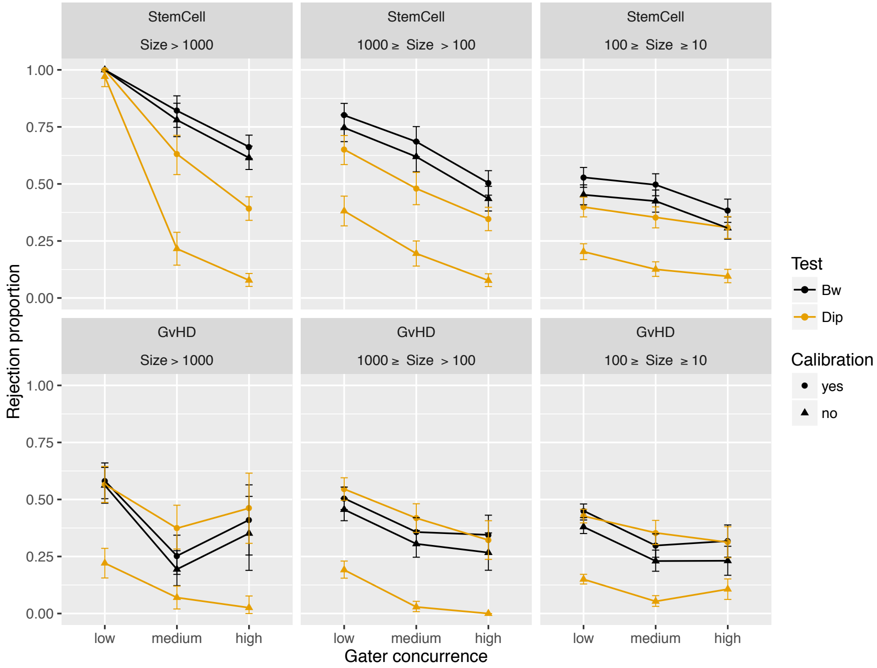


Figure S8: Rates of rejection calibrated and uncalibrated dip and bandwidth tests (shoulder reference) in at least one dimension, split by gater concurrence and cluster size. The gater concurrence categories are defined in Section 3.1.2. The error bars show Jeffrey's confidence interval.

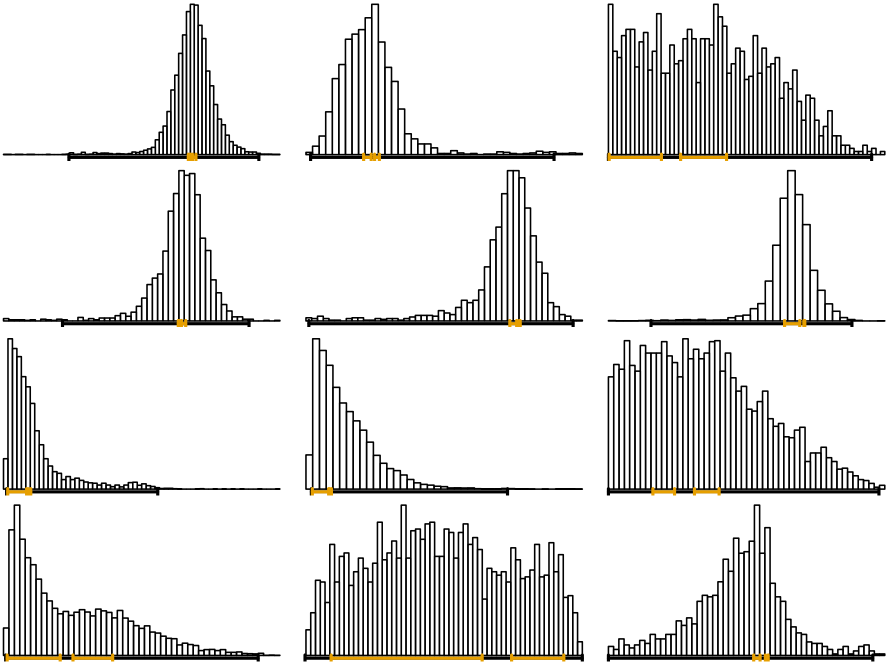


Figure S9: Histograms of data sets accepted by the dip test and rejected by the bandwidth test. The black interval shown at the bottom is the interval selected for the bandwidth test wherein to count for modes. The two yellow intervals show the two most significant modes for the dip test. The two top rows are the six first such data sets in the StemCell data set with at least 1000 points, the bottom two rows are the first such sets in the GvHD data set with at least 1000 points.

bandwidth test disagree. In Fig. S9, which shows data sets which are accepted by the dip test but rejected by the bandwidth test, it is common with a flat region.

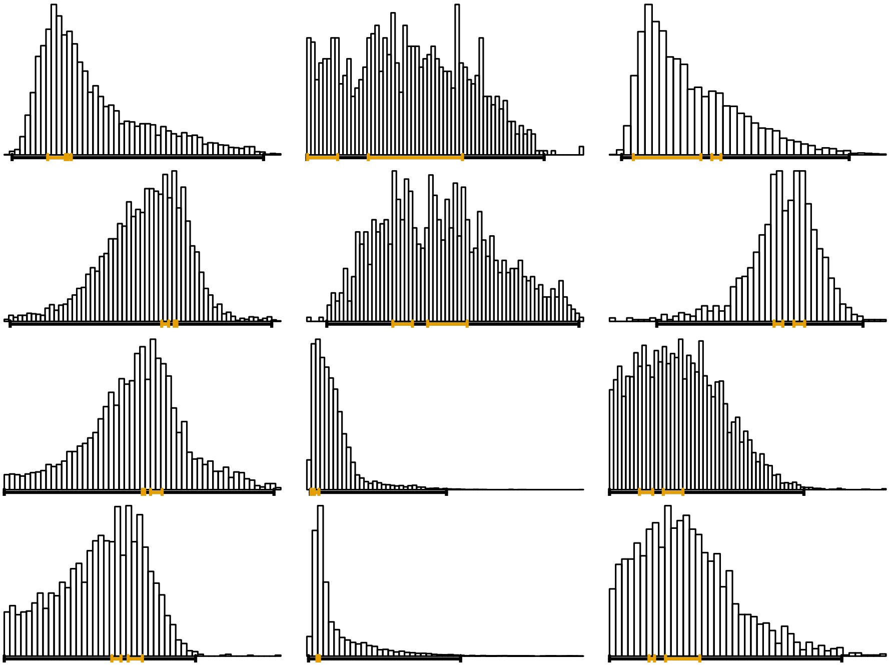


Figure S10: Histograms of data sets accepted by the bandwidth test and rejected by the dip test. The black interval shown at the bottom is the interval selected for the bandwidth test wherein to count for modes. The two yellow intervals show the two most significant modes for the dip test. The two top rows are the six first such data sets in the StemCell data set with at least 1000 points, the bottom two rows are the first such sets in the GvHD data set with at least 1000 points.