Machine Learning Techniques for the Analysis of High Throughput DNA and RNA Sequencing Data

Dissertation
zur Erlangung des akademischen Grades
Doktor
im Doktoratsstudium der
Naturwissenschaften

Eingereicht von:
Günter Klambauer

Angefertigt am:
Institut für Bioinformatik

Beurteilung:
Dr. Sepp Hochreiter (Betreuung)
Dr. Andreas Bender

Linz, März 2014
Acknowledgements

I am thanking my supervisor Sepp Hochreiter for giving me the opportunity of this PhD study, for his support, his patience, for the chance to visit conferences all over the world, for talks about bioinformatics, machine learning, and soccer, and for my first Bavarian breakfast.

My sincere thanks also go to the people of the QSTAR Consortium and at Janssen Pharmaceutical, especially to Hinrich Göhlmann, Willem Talloen, Ziv Shkedy, Jörg Wegner, and to Andreas Bender for fruitful collaboration.

Thanks to all my colleagues for numerous things and friendship. To Andreas Mayr and Andreas Mitterecker for helping me to get into scientific work and for teamwork throughout the last years. To Karin Schwarzbauer for great discussions, to Ulrich Bodenhofer for giving me help any time I needed and for rocking the christmas parties, to Thomas Unterthiner for productive cooperation on the DEXUS paper and for taking all my jokes on Tyroleans and Italians with a smile, to Gundula Povysil for being an awesome screen-to-screen mate, to Birgit Hauer for laughing with me about my ignorance of administrative matters, and to Herbert Zaunmair for doing the magic tricks with the servers. To my colleagues at Berlin, Martin Heusel and Okko Clevert, for discussions, advice and inspiration.

Most of all I want to thank my family who was always there for me and encouraged me: my parents Gabi and Joe, my brother Markus and my sister Carina with her family Andreas and my godson Emil, my girlfriend Julia, my aunt Elli, who had to leave far too soon, and her husband Konrad, my grandparents Anna, Werner, Margarete, and Josef, and my great-grandmother Anna. I can never thank you enough – you are the best family in the world!
Abstract

The identification of copy number variations in high-throughput DNA sequencing data and the detection of differential expression in RNA sequencing data are central topics in genetics and molecular biology. In these fields, new analysis methods should either enable researchers to investigate the data in a novel way that provides biologically relevant information, or have higher performance than previous methods, e.g. by yielding a lower false discovery rate and a lower false negative rate.

This thesis describes two new methods, called “cn.MOPS” and “DEXUS”, for copy number detection in DNA sequencing data and identification of differentially expressed genes in RNA sequencing data, respectively. cn.MOPS outperformed all other methods with respect to false discovery rate and recall and is currently developing into a standard analysis tool for both genome and exome sequencing data. DEXUS enabled researchers for the first time to analyze RNA sequencing data even if the sample conditions are unknown, which is the case for many study designs. For study designs in which sample conditions are known, DEXUS outperformed in almost all settings all other methods with respect to the area under the ROC curve.

Both methods are based on a probabilistic latent variable model. Model selection is done by maximizing the posterior with an expectation maximization (EM) algorithm. The EM algorithm makes model selection computationally efficient, such that the methods are fast enough to analyze huge amounts of data, which is an important criterion for bioinformatics methods. cn.MOPS and DEXUS are tested on a large number of benchmarking data sets and on many data sets with highly-relevant biological research questions, and there both algorithms provide excellent results.
ABSTRACT
Kurzfassung

Die Identifikation von Kopienzahlvariationen in Hochdurchsatz-DNA-Sequenzierungsdaten und die Detektion von differenzieller Expression in RNA-Sequenzierungsdaten sind zentrale Themen der Genetik und Molekularbiologie. In diesen Gebieten sollten neue Analysemethoden entweder den ForscherInnen die Möglichkeit geben die Daten auf eine neue Weise zu untersuchen, die auch relevante biologische Informationen liefert, oder sollten eine bessere Performance als alle anderen Methoden haben, zum Beispiel indem sie weniger falsche Detektionen oder weniger Falsch-Negative liefern.

In dieser Arbeit werden zwei neue Methoden, genannt “cn.MOPS” und “DEXUS”, beschrieben. Die erste identifiziert Kopienzahlvariationen in DNA Sequenzierungsdaten und die zweite detektiert differenziell exprimierte Gene in RNA Sequenzierungsdaten. cn.MOPS übertraf alle anderen Methoden bezüglich des positive Vorhersagewerts und der Sensitivität und entwickelt sich momentan zu einem Standard-Analyseverfahren für Genom und Exom Sequenzierungsdaten. DEXUS ermöglichte ForscherInnen zum ersten Mal RNA Sequenzierungsdaten auch ohne bekannte Probandengruppen zu analysieren, was für viele Studientypen der Fall ist. Außerdem übertraf DEXUS in fast allen Fällen alle anderen Methoden hinsichtlich der Fläche unter der ROC Kurve für Studientypen bei denen die Probandengruppen bekannt sind.

Beide Methoden basieren auf einem probabilistischem Modell. Das Modell wird durch deinen “expectation maximization” (EM) Algorithmus selektiert, der die Maximum-A-Posteriori Wahrscheinlichkeit maximiert. Der EM Algorithmus macht die Modellselektion rechnerisch effizient, so dass die Methoden große Mengen an Daten verarbeiten können, was in der Bioinformatik ein wichtiges Kriterium ist. cn.MOPS und DEXUS wurden auf einer großen Zahl von Benchmark-Datensätzen und vielen Datensätzen mit hochrelevanten biologischen Fragestellungen getestet und lieferten darauf hervorragende Ergebnisse.

Engl.: “precision” und “recall”.
Engl.: “area under the ROC curve”
Selected Publications


Awards

2012 Austrian Life Science Award 2012.

Conference Talks


2012 CNV detection from exome sequencing data using a generative probabilistic model, 13th International Meeting on Human Genome Variation and Complex Genome Analysis (HGV 2012), Shanghai, China, Sept 6-8 2012.

2012 cn.MOPS: mixture of Poissons for discovering copy number variations in next generation sequencing data with a low false discovery rate, 20th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB), Long Beach, CA USA, July 15-17 2012.

Posters

2013 Detecting differentially expressed genes in RNA-Seq data with unknown conditions, Annual Meeting of the American Society of Human Genetics (ASHG2013), Boston, USA, October 22-26, 2013.

2011 cn.MOPS: mixture of Poissons for discovering copy number variations in next generation sequencing data. International Meeting on Human Genome Variation and Complex Genome Analysis (HGV 2011), San Francisco, CA USA, Sept 8-10 2011.


2010 Identifying copy number variations based on next generation sequencing data by a mixture of Poisson model, 18th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB), Boston, MA USA, July 11-13 2010, doi:10.1038/npre.2010.4716.1.

Education

2009-present PhD Study Bioinformatics, Johannes Kepler University, Linz, Supervisor: Sepp Hochreiter.
Areas of Study:
- Quantitative analysis of next generation sequencing data
- Quantitative structure-activity relationship of compounds for drug discovery
- Microarray data analysis and biclustering of gene expression data

2007-2009 Master Study Bioinformatics, Johannes Kepler University, Linz.


Areas of study: Functional Analysis, Algebra, and Numerical Calculus

2001-2007 Mathematics and Biology Diploma-Studies, University of Vienna.
Areas of study: Biomathematics, Ecophysiology of Plants, Population genetics, Complex Analysis


A-levels passed with distinction


Teaching Experience

2014 Special Topics on Bioinformatics: Population genetics, Lecturer, Johannes Kepler University, Linz, Austria.

2013/2014 Sequence Analysis and Phylogenetics, Lecturer, Johannes Kepler University, Linz, Austria.

2013 Special Topics on Bioinformatics: Population genetics, Lecturer, Johannes Kepler University, Linz, Austria.

2012/2013 Sequence Analysis and Phylogenetics, Lecturer, Johannes Kepler University, Linz, Austria.

2012 Special Topics on Bioinformatics: Population genetics, Lecturer, Johannes Kepler University, Linz, Austria.
2011/2012 **Exercises in Bioinformatics I: Sequence Analysis and Phylogenetics**, Lecturer, Johannes Kepler University, Linz, Austria.

2011 **Special Topics on Bioinformatics: Population genetics**, Lecturer, Johannes Kepler University, Linz, Austria.

2010/2011 **Exercises in Bioinformatics I: Sequence Analysis and Phylogenetics**, Lecturer, Johannes Kepler University, Linz, Austria.

2010 **Special Topics on Bioinformatics: Population genetics**, Lecturer, Johannes Kepler University, Linz, Austria.

2009/2010 **Exercises in Bioinformatics I: Sequence Analysis and Phylogenetics**, Lecturer, Johannes Kepler University, Linz, Austria.


### Computer skills
- **Programming Languages**: R, Perl, Java, Pascal, C++
- **Markup Languages**: \TeX, \LaTeX, Html

### Languages
- **German**: Native language
- **English**: Fluent
- **Italian**: Intermediate

Linz, March 24, 2014

*Klambauer Günter*
Affidavit

I declare under penalty of perjury that I have written this thesis independently and without assistance, that I have not used other than the stated sources and aids and that I have designated literally or analogously excerpted locations as such. This thesis is identical to the text document transmitted electronically.
Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe. Die vorliegende Dissertation ist mit dem elektronisch übermittelten Textdokument identisch.
Contents

Acknowledgements iii

Abstract v

Curriculum Vitae xii

Affidavit xiii

1 Introduction 1
   1.1 Overview ................................................. 1
   1.2 Background ............................................. 1
   1.3 High throughput sequencing – next generation sequencing . . 2
   1.4 Detection of CNVs from NGS data .......................... 2
   1.5 Detection of differential expression from NGS ............... 4

2 cn.MOPS 7

3 DEXUS 79

4 Discussion 169
   4.1 The german enterohemorrhagic Escherichia coli O104:H4 out-
        break ..................................................... 169
   4.2 Copy number detection in exome sequencing data ............. 170
   4.3 Research suggested by cn.MOPS .............................. 172
   4.4 The unknown conditions of DEXUS .......................... 174
   4.5 Research suggested by DEXUS .............................. 174
List of Figures

1.1 Schematic representation of a copy number variation . . . . . . 3
1.2 The principle of copy number detection methods . . . . . . . . 4
1.3 The difference between previous RNA-Seq methods and DEXUS 6

4.1 Copy numbers variations of O104:H4 . . . . . . . . . . . . . . 170
4.2 Publications per year with research based on exome sequenc-
ing data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 171
4.3 The distribution of read counts for whole genome sequencing
and whole exome sequencing . . . . . . . . . . . . . . . . . . . . . 172
LIST OF FIGURES
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNV</td>
<td>copy number variation</td>
</tr>
<tr>
<td>DNA</td>
<td>deoxyribonucleic acid</td>
</tr>
<tr>
<td>EM</td>
<td>expectation maximization</td>
</tr>
<tr>
<td>eQTL</td>
<td>expression quantitative trait locus</td>
</tr>
<tr>
<td>FDR</td>
<td>false discovery rate</td>
</tr>
<tr>
<td>GWAS</td>
<td>genome wide association study</td>
</tr>
<tr>
<td>NGS</td>
<td>next generation sequencing</td>
</tr>
<tr>
<td>RNA</td>
<td>ribonucleic acid</td>
</tr>
<tr>
<td>RNA-Seq</td>
<td>ribonucleic acid sequencing</td>
</tr>
<tr>
<td>ROC</td>
<td>receiver operating characteristic</td>
</tr>
<tr>
<td>RPKM</td>
<td>reads per kilo base of exonic sequence per million mapped reads</td>
</tr>
<tr>
<td>SNV</td>
<td>single nucleotide variant</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Overview

This document contains the scientific work I have contributed to the field of bioinformatics. The first chapter explains the background and gives an overview of high throughput DNA and RNA sequencing techniques from the bioinformatics point of view. The following chapters hold two articles that I published as first author in “Nucleic Acids Research”, a peer-reviewed journal, that covers research on DNA and RNA. Both articles are accompanied by “Supplementary Data” documents that are also included here, because they present the full derivative of the new machine learning algorithms.

The papers are:


In chapter 4 I revisit the papers, summarize their results, and present an outlook to research that they suggest.

1.2 Background

Bioinformatics is the research discipline that develops computational methods to acquire biological, or medical, knowledge. This discipline thereby assists
CHAPTER 1. INTRODUCTION

researchers of life sciences at their scientific work. To solve problems posed by life sciences, bioinformatics applies methods of mathematics, statistics, and computer science. Recent breakthroughs in life sciences, such as the sequencing of the human genome, mapping genetic variation, and cellular reprogramming, were made possible by bioinformatics methods.

1.3 High throughput sequencing – next generation sequencing

The plan of a whole organism is stored in a simple molecule, the DNA. Therefore, reading the DNA provides immense knowledge about life. The process of reading the DNA is called “sequencing”. However, this process was very costly until a few years ago. Fortunately, the cost of sequencing of DNA and RNA has rapidly declined since the advent of high throughput sequencing technologies, nowadays called “next generation sequencing” (NGS) (Metzker 2009). Whereas traditional sequencing techniques read one DNA fragment at a time, new technologies can read the sequence of millions of fragments simultaneously. On the one hand, this offers a great opportunity for biologists to obtain insights into a large number of genomes and transcriptomes in a very short time (The 1000 Genomes Project Consortium 2010). On the other hand, a number of new challenges are posed by the high throughput sequencing technologies: how to handle the amount of data produced, how to assemble a genome or transcriptome based on millions of short fragments, how to call variants, i.e. the differences between genomes, and how to quantify the amount of DNA or RNA (Treangen and Salzberg 2011; DePristo et al. 2011; Labaj et al. 2011)? In my scientific work I have addressed the problem of DNA and RNA quantification, which resulted in the development of two new methods and software tools. First, a method for identification of copy number variations from next generation DNA sequencing data. Second, a method for detection of differential expression from next generation RNA sequencing data.

1.4 Detection of copy number variations from next generation DNA sequencing data

The next generation DNA sequencing technology can sequence millions of short DNA fragments in parallel and thereby provide researchers with information about the sequenced genome. To identify genetic variants of the
1.4. DETECTION OF CNVS FROM NGS DATA

A B C
BA B B B C

Copy number variation

Figure 1.1: Schematic representation of a copy number variation. The majority of individuals has one copy of segment “B” (“Normal”), whereas some individuals show an abnormal, in this case four, number of copies of this segment “B” (“Copy number variation”).

sequenced organism these sequence reads are usually mapped to the human reference genome (Li et al. 2008; Li and Durbin 2009; Langmead et al. 2009). Qualitative analysis of this sequence map results in the detection of single nucleotide variants (SNVs) and short InDels (Shendure and Ji 2008; DePristo et al. 2011). Beside SNVs and InDels, another type of variants contribute largely to genetic variation: copy number variations (CNVs) (Redon et al. 2006; International HapMap 3 Consortium 2010). CNVs are DNA segments with a length of at least 1,000 bases that occur in variable number among individuals, see Figure 1.1. Quantifying the amount of a DNA segment can be used to identify copy number variations (Shendure and Ji 2008; Alkan et al. 2009; Chiang et al. 2008; Yoon et al. 2009). If a DNA segment occurs in the sequenced individual more or less often than in the reference sequence, the number of reads mapping to the DNA segment will be higher or lower, respectively, see Figure 1.2. Computational methods can utilize these changes of the read count signal\(^1\) to identify CNVs (Alkan et al. 2009; Chiang et al. 2008; Yoon et al. 2009; Magi et al. 2011; Boeva et al. 2011).

I developed a method that identifies copy number variations in next generation sequencing data of multiple individuals. The algorithm is based on a machine learning technique called EM-algorithm and outperforms other algorithms with respect to the false discovery rate (FDR).

\(^1\)Some methods use a depth of coverage signal for detection of CNVs, which is essentially the same as the read count signal, because there is a simple linear dependency between depth of coverage and read count.
Figure 1.2: The principle of detection of copy number variations by quantitative analysis of next generation sequencing data. In the bottom row the genome of and individual is depicted. Sequence reads are shown as black dashes (second row from bottom). The sequence reads are mapped to a reference genome by a read mapping algorithm (fourth row from below). The number of reads of segment “B” are elevated (top row), because it is present in the individual genome four times, but only one time in the reference genome.

1.5 Detection of differential expression from next generation RNA sequencing data

Next generation RNA sequencing (RNA-Seq) allows to characterize transcriptomes of cells, tissues and organisms (Blow 2009; Shendure and Ji 2008; Jiang and Wong 2009; Trapnell et al. 2010). The transcriptome is the set of all RNA molecules of a cell and thereby reflects the genes that are active, i.e. expressed, in a cell at a certain time. The first studies that used the RNA-Seq technology analyzed the complex transcriptomes of human embryonic kidney and B-cells (Sultan et al. 2008), mouse stem cells (Cloonan et al. 2008), and mouse tissues (Mortazavi et al. 2008). Besides these qualitative analyses of transcriptomes, also quantitative analysis of RNA-Seq data provides relevant biological information (Blow 2009; Trapnell et al. 2010): The more mRNA of a certain transcript is present, the more reads of this transcript will be sequenced, which offers the opportunity to detect differential expression. The number of reads per kilo base of exonic sequence per million
mapped reads (RPKM) highly correlates with gene expression measures of microarrays (Mortazavi et al. 2008; Pickrell et al. 2010). Comparing transcript expression measurements of different samples enables researchers to find differentially expressed transcripts (Anders and Huber 2010; Robinson et al. 2010). Differential expression plays a key role in identifying causes of human diseases, cancer classification, drug development, and gaining insights into the function of genes and proteins.

I developed an algorithm that detects differentially expressed transcripts in RNA-Seq experiments that could not be analyzed for differential expression before. For detection of differential expression from RNA-Seq data several methods had already been suggested (Anders and Huber 2010; Robinson et al. 2010; Li and Tibshirani 2013; Wu et al. 2013; Hardcastle and Kelly 2010). All of these algorithms were only able to identify differential expression between a priori known sample groups, e.g. in a classical case-control study. However, for many study designs, such as cohort or cross-sectional studies, the groups are not known a priori. I proposed a new algorithm, DEXUS, that is able to identify differentially expressed genes in these studies. On benchmarking data sets with unknown conditions DEXUS yields excellent performances with respect to sensitivity, specificity, precision and recall. On real-world data sets from different areas, such as eQTL studies (Montgomery et al. 2010; Pickrell et al. 2010), plant physiology (Li et al. 2010), and anthropology (Blekhman et al. 2010), the algorithm provided novel biological insights. Even on data sets with known conditions DEXUS outperforms previously published algorithms, when the sample size is medium or large.\footnote{with “medium sample size” we mean that there are at least 6 samples per group. For much smaller sample sizes the new algorithm has comparable performance to the best performing competitors.}

As stated above, all state-of-the art methods were tailored to identify transcriptional effects that were indicative for a priori fixed conditions and thereby neglected much information that was present in the data. Figure 1.3 shows an example for this situation – a study, in which the mRNA of lung cells is sequenced and the difference in expression between males and females should be analyzed. Suppose a transcript C is differentially expressed between males and females. However, another transcript E is present whose expression is correlated with smoking activity of the samples. Transcript A is influenced by a single nucleotide variant – individuals with the nucleotide A at a certain position in the genome express this gene, others do not express it.\footnote{Such variants are called “expression quantitative trait loci” (eQTLs).} Traditional RNA-Seq methods would only detect transcript C, because it is differentially expressed between the given condition (males, females). A lot of valuable information, e.g. transcripts whose expression is correlated with
CHAPTER 1. INTRODUCTION

Figure 1.3: Illustration of the difference between previous RNA-Seq methods and DEXUS. The upper panel displays samples and their associated conditions. The top row depicts the known condition, i.e. sex. In the second row an unknown or hidden condition of the samples is smoking activity and in the third row the nucleotide of a certain position in the genome is shown (either A or C). In the lower panel five transcripts are shown. Dashes symbolize that the transcript has a normal expression level (compared to controls). Arrows denote differential expression, either over- or under-expression of a transcript in a sample. Previous methods would only detect transcript C as differentially expressed, because its expression correlates with the known condition (sex). DEXUS would also identify transcript A, and transcript E, because they are differentially expressed due to genetic factors or smoking activity, respectively.

smoking activity or differential expression due to genetic variability beside sex, remains undisclosed. Although transcript C might be the only relevant transcript for the study mentioned above, the data itself contain much more information that is biologically relevant.
Chapter 2

cn.MOPS: mixture of Poissons for discovering copy number variations in next-generation sequencing data with a low false discovery rate.

In the following, I present my publication of a method for copy number detection in next generation sequencing data that was published in the journal *Nucleic Acids Research* in February 2012.
cn.MOPS: mixture of Poissons for discovering copy number variations in next-generation sequencing data with a low false discovery rate

Günter Klambauer, Karin Schwarzbauer, Andreas Mayr, Djork-Arné Clevert, Andreas Mitterecker, Ulrich Bodenhofer and Sepp Hochreiter*

Institute of Bioinformatics, Johannes Kepler University, A-4040 Linz, Austria

Received August 31, 2011; Revised December 1, 2011; Accepted December 28, 2011

ABSTRACT
Quantitative analyses of next-generation sequencing (NGS) data, such as the detection of copy number variations (CNVs), remain challenging. Current methods detect CNVs as changes in the depth of coverage along chromosomes. Technological or genomic variations in the depth of coverage thus lead to a high false discovery rate (FDR), even upon correction for GC content. In the context of association studies between CNVs and disease, a high FDR means many false CNVs, thereby decreasing the discovery power of the study after correction for multiple testing. We propose ‘Copy Number estimation by a Mixture Of Poissons’ (cn.MOPS), a data processing pipeline for CNV detection in NGS data. In contrast to previous approaches, cn.MOPS incorporates modeling of depths of coverage across samples at each genomic position. Therefore, cn.MOPS is not affected by read count variations along chromosomes. Using a Bayesian approach, cn.MOPS decomposes variations in the depth of coverage across samples into integer copy numbers and noise by means of its mixture components and Poisson distributions, respectively. The noise estimate allows for reducing the FDR by filtering out detections having high noise that are likely to be false detections. We compared cn.MOPS with the five most popular methods for CNV detection in NGS data using four benchmark datasets: (i) simulated data, (ii) NGS data from a male HapMap individual with implanted CNVs from the X chromosome, (iii) data from HapMap individuals with known CNVs, (iv) high coverage data from the 1000 Genomes Project. cn.MOPS outperformed its five competitors in terms of precision (1–FDR) and recall for both gains and losses in all benchmark data sets. The software cn.MOPS is publicly available as an R package at http://www.bioinf.jku.at/software/cnmops/ and at Bioconductor.

INTRODUCTION
Next-generation sequencing (NGS) has evolved into an important technology for genotyping (1) and genome assembly (2). NGS has also been applied to transcriptomics (mRNA-Seq), where it revealed new splice variants and new transcripts (3). Despite these successes, quantitative analyses of NGS data, for instance, determination of the expression levels of genes, are still challenging (4,5). Estimation of DNA copy numbers is another important kind of quantitative analysis, in which local depths of coverage must be mapped to integer copy numbers. Copy number analysis by NGS has the following potential advantages compared with array-based techniques: (i) estimation of integer copy numbers from NGS data is more accurate for large copy numbers, since depths of coverage scale linearly with copy numbers (6). (ii) Breakpoints of copy number regions can be determined more precisely (7) because they do not rely on predefined probes. (iii) Allele-specific copy numbers may be estimated for observed alleles, while array-based techniques are restricted to predefined alleles. Allele-specific copy numbers are of interest because they allow for determining whether an allele is fully functional, which is important, for example, for the identification of mutations leading to cancer development (8).

In the following, we review existing methods for estimating DNA copy numbers in NGS data. These methods represent the depth of coverage either as read counts or as log read counts in an interval and can be

*To whom correspondence should be addressed. Tel: +43 732 2468 8880; Fax: +43 732 2468 9511; Email: hochreit@bioinf.jku.at

© The Author(s) 2012. Published by Oxford University Press. This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
classified into (a) approaches detecting read count deviations and (b) reference-based approaches. Class (a) methods detect CNVs as deviations of (log) read counts from an average (log) read count of a chromosome. Class (b) methods detect CNVs as intervals for which (log) read count ratios between a sample and a reference deviate from 1 (10). This reference can either be a designated control sample or a constructed average sample.

MOFDOC ('MOdel Free Depth Of Coverage') belongs to class (a) and has been used by Alkan et al. (6), Campbell et al. (9), Wang et al. (10), Bentley et al. (11) and Wheeler et al. (12) for NGS copy number detection and earlier by Bailey et al. (13) for whole-genome shotgun sequencing. The variant of MOFDOC, as introduced by Alkan et al. (6), first divides the genome into non-overlapping segments of equal length in which reads are counted. Note that some variants of MOFDOC are based on log read counts. Using the overall mean and standard deviation of those segment read counts, each segment is characterized by the multiple of the standard deviation by which its read count differs from the overall mean; analogously, it combines segments into a loss segment if they have read counts larger than three times the standard deviation above the mean, analogously, it combines segments into a loss segment if they have read counts smaller than two times the standard deviation below the mean (see blue boxes in Figure 2). GC correction is crucial for proper performance of MOFDOC because of the GC content bias of NGS (14). However, MOFDOC, like most class (a) methods, has a high false discovery rate (FDR), even upon GC correction. The reason is that mean segment read counts for copy number two may vary along the chromosome due to technological biases or local genomic characteristics. These read count variations along the chromosome appear consistently across samples, for example, all samples tend to have either larger or smaller read counts (see Supplementary Figure S11 and the third bar in Figure 2). Class (a) methods confound these variations with copy number changes leading to false discoveries.

EWT ('Event-Wise Testing'), introduced by Yoon et al. (15), is identical to MOFDOC except for the final segmentation algorithm. EWT uses a probabilistic approach to join consecutive segments that, under a Gaussian assumption, show either significantly larger or significantly smaller read counts than the overall mean (see blue boxes in Figure 2). As for MOFDOC, read count variations along the chromosome lead to false CNVs (see Supplementary Figure S11).

JointSLM (16) also belongs to class (a), and extends the idea of EWT to a simultaneous segmentation of multiple samples. Again, the genome is divided into equally sized, non-overlapping segments for which the logarithm of GC-corrected and normalized (divided by the median per sample) read counts is computed. A hidden Markov model (HMM) slides along the chromosome and simultaneously scans the log-normalized read counts of all samples. The more samples show large or small read counts, the more likely a segment is detected (see blue boxes in Figure 2). JointSLM hardly detects CNVs that occur only in a few samples, because its HMM uses a single state variable for simultaneously explaining the copy numbers of all individuals (see Supplementary Figure S9). Furthermore, CNV regions may contain both gains and losses (17), which impedes JointSLM in detecting such regions, since samples have propensities to transit to different HMM hidden states. Like other class (a) methods, JointSLM detects spurious regions if they contain read counts that, due to genomic and technical biases, are smaller or larger than the chromosome average (see Supplementary Figure S11). Note that we consider JointSLM as a class (a) method because it detects simultaneous deviations of log read counts from an average log read count.

SeqSeg (7) is a class (b) method that was designed to identify copy number aberrations (CNAs) in tumor samples by comparing them to references, that is, their matched controls. SeqSeg evaluates the likelihood of each tumor read being a CNA breakpoint and keeps the most likely ones, thereby segmenting the chromosome. For each segment, the ratio between sample read counts and reference read counts is computed. Segments are called gains if their ratios are above 1.5, which corresponds to a copy number of at least 3, or losses if their ratios are below 0.5, which corresponds to a copy number of at most 1. Read count variations along the chromosome stemming, for instance, from the GC bias are implicitly corrected because these variations affect both the tumor sample and the reference in a similar way. SeqSeg relies on a single reference and does not consider local read count variations across replicates or multiple samples. Consequently, SeqSeg falsely detects CNVs in genomic regions where read counts of replicates are highly variable (see Supplementary Figure S12).

rSW-seq (18) improves SeqSeg with respect to breakpoint identification, but the local read count variability remains disregarded. Note that both methods, SeqSeg and rSW-seq, were designed for CNA detection in tumor samples, especially at the breakpoint identification step.

CNAseg (19) was also designed to detect CNAs in tumor samples, using an approach similar to that of SeqSeg. CNAseg, like JointSLM for a single sample, employs an HMM for joining equally sized, non-overlapping segments using the difference in segment read counts between tumor and reference. The resulting segments are joined on the basis of a $\gamma$ statistic.

CNV-Seq (20) is another class (b) method. It also divides the genome into equally sized, non-overlapping segments for which read count ratios are computed using a reference sample. The read counts are assumed to follow a Poisson distribution, which is approximated by a Gaussian distribution. Subsequently, the Geary-Hinkley transformation is applied to the ratios of Gaussians to produce an approximately Gaussian output. In a final step, a segmentation algorithm joins consecutive segments with log ratios above or below a certain threshold. Like all other methods, CNV-Seq is prone to falsely detecting CNVs since it does not take local read count variability into account (see Supplementary Figure S12).
FREEC ('control-FREE Copy number calling') is a class (b) method suggested by Boeva et al. (21). FREEC also counts reads in equally sized, non-overlapping segments and computes read count ratios per segment using a reference sample. Hypothetical read counts estimated by a polynomial function of the segment's GC content can be used instead of a reference. In the segmentation step, the breakpoints are determined by LASSO ('Least Absolute Shrinkage eStimatOr') regression (22). FREEC does not consider local read count variability either, which makes it susceptible to falsely discovered CNVs (see Supplementary Figure S12).

In summary, existing methods suffer from a high FDR that results (i) from read count variations along the chromosome, especially if no references are used, and (ii) from variations in read counts (noisy counts) across samples that may occur even for constant copy numbers. The high FDR can be moderated for paired-end reads by confirming CNVs by means of clusters of discordant read pairs—incorrect orientation, order or distance (23). However, this approach may considerably decrease the discovery power since clusters may be missed, especially in cases of low coverage.

Below we introduce cn.MOPS, a CNV detection method together with a data processing pipeline which, in contrast to most previous methods, (i) provides integer copy numbers, (ii) estimates variations in read counts across samples and (iii) uses these estimates for CNV calling, thereby keeping the FDR low. A high FDR is particularly critical in association studies between CNVs and diseases: a high FDR implies many false CNVs. Correction for multiple testing must then consider these false discoveries, which increases the correction step, the breakpoints are determined by LASSO ('Least Absolute Shrinkage eStimatOr') regression (22). FREEC does not consider local read count variability either, which makes it susceptible to falsely discovered CNVs (see Supplementary Figure S12).

In summary, existing methods suffer from a high FDR that results (i) from read count variations along the chromosome, especially if no references are used, and (ii) from variations in read counts (noisy counts) across samples that may occur even for constant copy numbers. The high FDR can be moderated for paired-end reads by confirming CNVs by means of clusters of discordant read pairs—incorrect orientation, order or distance (23). However, this approach may considerably decrease the discovery power since clusters may be missed, especially in cases of low coverage.

Below we introduce cn.MOPS, a CNV detection method together with a data processing pipeline which, in contrast to most previous methods, (i) provides integer copy numbers, (ii) estimates variations in read counts across samples and (iii) uses these estimates for CNV calling, thereby keeping the FDR low. A high FDR is particularly critical in association studies between CNVs and diseases: a high FDR implies many false CNVs. Correction for multiple testing must then consider these false discoveries, which increases the corrected P-values and reduces the discovery power of a study. The novelty of cn.MOPS is modeling across samples, which improves the performance considerably, as technical and biological variations are estimated and taken into account. By 'modeling across samples' we mean the construction of a generative model that explains how the data have been produced. This model decomposes the read count of each sample into signal and noise. The idea of modeling across samples has already improved CNV detection in microarray data by reducing the FDR, as the recent cn.FARMs method (24) demonstrates.

**METHODS**

The cn.MOPS processing pipeline is depicted in Figure 1. The left column shows modeling across samples and integer copy number estimation that are unique to the cn.MOPS pipeline. On the right-hand side, GC correction is unique to some previous analysis pipelines; however, this step is not necessary for cn.MOPS, as the local model automatically captures GC content effects. The steps of the cn.MOPS processing pipeline and the central cn.MOPS model are described in the following subsections.

**Read mapping and segment read counts**

After quality control, read mapping is the first step in analyzing NGS data with respect to CNVs (Figure 1). Depending on the technology, the read length and whether the reads are paired or single end, the FDR of the mapping method should be adjusted to minimize the number of false positives without generating too many false negatives. The most important mapping parameters are the number of mismatches allowed and the gap parameters of the employed alignment algorithm. These parameters depend on the expected sequencing errors [see (14) for a statistical analysis of sequencing errors]. In general, the best mapping positions are found for a read, then the read can be randomly assigned to one of them, to all of them, or can be discarded. In our experiments, we mapped the reads by Bowtie (25) for paired reads, allowing mismatches and mapped to one random best mapping position.

After read mapping, segments must be defined in which the reads are counted (Figure 1). For cn.MOPS, as for all other approaches except SeqSeg and rSW-seq, the genome is first divided into non-overlapping segments in which reads are counted. Although cn.MOPS also uses equally sized segments by default, equal size is not strictly required, since a separate model is generated for each segment. The later segmentation along the chromosome is based on the expected copy number, which is independent of the segment length. If the sizes of segments in which reads are counted are variable, then the resolution can be traded off against the confidence in the estimated copy numbers.

**Sample normalization and GC correction**

GC correction is a crucial first step for proper performance of class (a) methods, such as MOFDOC (Figure 1). These methods subsequently apply a segmentation algorithm to (log) read counts along the chromosome. Therefore, the (log) read counts must be normalized to be comparable between different genomic loci. Since the numbers of reads within segments depend on their GC content (14), the segments' (log) read counts must be normalized for their GC content.

**Sample normalization** is important for modeling across samples because the reads of all samples are assumed to be caused by the same model (Figure 1). Sample normalization corrects the read counts of one sample by the number of mappable reads of the sample to make read counts comparable across samples. Using data of HapMap individuals from the 1000 Genomes Project (26), we tested read counts of 25 kbp segments for being Poisson distributed with and without sample normalization. Without sample normalization, the Poisson assumption was rejected by a Poisson test (27) for 92% of the segments. Using normalization, however, the Poisson assumption was rejected for only 2% of the segments. Segments that were rejected coincide significantly with known CNV regions according to Fisher’s exact test.
with \( p < 2.2 \times 10^{-16} \) (for details see Supplementary Table S1). Thus, our model assumption that, for a constant copy number, the read counts are Poisson distributed is justified with sample normalization. This assumption is also in concordance with the findings of Sathirapongsasuti et al. (28).

The mixture of Poissons model

Our main contribution and novelty is modeling of read count variations across samples in order to separate variations caused by copy numbers from local variations caused by technical or biological noise (Figure 1). For CNV detection, we use a mixture of Poissons model that is not affected by read count variations along the chromosome, because a separate model is constructed at each DNA locus. The model incorporates the linear dependency between average read counts in segments and copy numbers (6,7). In contrast to existing methods, cn.MOPS provides integer copy numbers together with their confidence intervals. Model selection in a Bayesian framework is based on maximizing the posterior by an expectation maximization (EM) algorithm. Most importantly, a Dirichlet prior on the mixture components prefers a constant copy number of 2 for all samples. Only if the data drive the posterior away from this prior, the segment receives a high informative/non-informative call (I/NI call), that is, the part of the CNV call which detects variation across samples.
The Model. cn.MOPS is a generative probabilistic model that explains the observed read counts by copy numbers and by measurement variations. Consequently, the model assumes that the read counts $x$ in a segment are distributed across samples according to a mixture of Poissons, in which each mixture component corresponds to specific copy number and the Poisson parameter reflects the noise:

$$p(x) = \sum_{i=0}^{n} a_i P(x; \lambda_i) .$$

(1)

In this model, $a_i$ is the percentage of samples with copy number $i$ for $0 \leq i \leq n$, and $\lambda$ is the mean read count for copy number $2$. $P$ is the Poisson distribution:

$$P(x; \beta) = \frac{1}{x!} e^{-\beta} \beta^x .$$

(2)

The model integrates the assumption that the read counts are linearly related to the number of copies. For copy number $i \geq 1$, the mean read count is thus $\beta = \frac{2}{i}$. For copy number $i = 0$, we assume a Poisson distribution with parameter $\beta = \frac{2}{i}$, which accounts for background noise stemming from wrongly or ambiguously mapped reads and for sample contamination by other DNA. See Supplementary Section S2.5, for a justification of the noise model. Note, that the results of cn.MOPS are robust against the choice of the hyperparameter $\epsilon$ (see Supplementary Section S3.8). The robustness is due to the fact that copy number zero can be detected with a broad range of $\epsilon$ values.

The model in Equation (1) allows estimation of integer copy numbers with fixed model parameters $a_i$ and $\lambda$. The prior probability that a read count stems from copy number $i$ is $p(i) = a_i$. The likelihood that a read count $x$ is produced by the $i$-th mixture component is
Then Bayes’ formula can be used to compute the posterior \( p(i \mid x) \), that is, the probability that read count \( x \) is caused by the \( i \)-th component corresponding to copy number \( i \). Consequently, a read count is assigned the integer copy number with the largest posterior probability.

**Model Selection by an EM Algorithm and Dirichlet Prior.** Suppose that read counts \( \{x_1, \ldots, x_N\} \) have been observed for \( N \) samples in a given segment. Model selection is concerned with fitting a model that best explains the training data \( \{x_1, \ldots, x_N\} \). In a Bayesian framework, \( \alpha \) and \( \lambda \) are considered as random variables; thus, \( p(x \mid \alpha, \lambda) \) in Equation (1) becomes a conditional probability \( p(x \mid \alpha, \lambda) \), i.e. the likelihood that read count \( x \) has been produced by the model with parameters \( \alpha \) and \( \lambda \). If we assume that, for the prior distribution, the parameters \( \alpha \) and \( \lambda \) are independent \( p(\alpha, \lambda) = p(\alpha)p(\lambda) \), then the parameter posterior is

\[
p(\alpha, \lambda \mid x) = \frac{p(x \mid \alpha, \lambda) p(\alpha) p(\lambda)}{\int p(x \mid \alpha, \lambda) p(\alpha) p(\lambda) \, d\alpha \, d\lambda}.
\]

(3)

We introduce a Dirichlet prior \( p(\alpha) \) on \( \alpha = (\alpha_0, \alpha_1, \ldots, \alpha_N) \) to include the prior knowledge that almost all locations have copy number 2 for all samples, which is the null hypothesis of constant copy number 2. The Dirichlet prior

\[
p(\alpha) = \frac{1}{B(\gamma_0)^N} \prod_{i=0}^{N} \alpha_i^{\gamma_i - 1}
\]

(4)

with parameter vector \( \gamma = (\gamma_0, \gamma_1, \ldots, \gamma_N) \) is well suited to express our prior assumptions about \( \alpha \). By setting \( \gamma_2 > \gamma_i \geq 1 \) for \( i \neq 2 \), we ensure that vectors \( \alpha \) with a large value for \( \gamma_2 \)—the percentage of samples having copy number 2—are the most likely to be drawn. Each component \( i \) of the Dirichlet distribution \( p(\alpha) \) is distributed according to a beta distribution with the parameters \( (\gamma_i - 1)/(\gamma_i - \gamma_0) \), where \( \gamma_i = \sum_{i=0}^{N} \gamma_i \).

For the prior on \( \lambda \), we simply choose a uniform distribution on a sufficiently large interval with left endpoint 0.

An EM algorithm minimizes an upper bound on the negative log-posterior of the parameters \( \alpha \) and \( \lambda \) by the following update rules (for details see Supplementary Section S2.2):

\[
\bar{a}_k = \frac{\alpha_k^{old} P(x_k \mid \lambda^{old})}{P(x_k \mid \lambda^{old})},
\]

(5)

\[
a_k^{new} = \frac{\frac{1}{N} \sum_{i=1}^{N} \bar{a}_i + \frac{1}{2}(\gamma_i - 1)}{1 + \frac{1}{2}(\gamma_i - n)}.
\]

(6)

\[
\lambda^{new} = \frac{\frac{1}{N} \sum_{i=1}^{N} \bar{a}_i}{1 + \frac{1}{2} \sum_{i=1}^{N} \bar{a}_i}.
\]

(7)

Here, \( \bar{a}_k \) is an estimate of the E-step of the EM algorithm of the posterior \( a_k = p(i \mid x_k, \alpha, \lambda) \) using current estimations \( \alpha^{old} \) and \( \lambda^{old} \) of the parameters. We simplify the hyperparameter vector \( \gamma \) to one intuitively interpretable hyperparameter \( G \) by setting \( \gamma_i = 1 \) for \( i \neq 2 \) and \( \gamma_2 = 1 + G \). This setting ensures that \( \gamma_2 \) update Equation (6). Thus, a minimum percentage of individuals that have copy number 2 can be ensured by the hyperparameter \( G \) (for details see Supplementary Section S2.2).

**I/NI Call: Information Gain of Posterior over Prior.** Based on the Bayesian framework, we define an informative/ non-informative (I/NI) call analogous to that of the FARMS algorithm, which excelled at summarization and gene filtering of microarray data (29-31). In contrast to \( \lambda \), which captures noise variation, \( \alpha \) captures variation arising from CNVs; therefore, its posterior indicates CNVs in the data. The I/NI call measures the information gain of the posterior compared to its prior distribution \( p(\alpha) \), which represents the null hypothesis that all samples have copy number 2. Therefore, the I/NI call measures the tendency to reject the null hypothesis based on the observed data.

We define the I/NI call as a weighted distance between the posterior’s mode and the prior’s mode \((0, 0, 1, 0, \ldots, 0)\), which results in the expected absolute log fold change relative to copy number 2 (for details see Supplementary Section S2.3):

\[
I/NI(\alpha) = \sum_{i=0}^{N} a_i |\log(i/2)| = \sum_{i=0}^{N} \frac{1}{N} \sum_{a_k} a_k |\log(i/2)|.
\]

(8)

with \( a_i = \frac{1}{N} \sum_{a_k} a_k \) [this equation is derived in Supplementary Equation (S25)]. For notational convenience, we did not distinguish between \( i = 0 \) and \( i = 1 \) in the above formula. \( |\log(0/2)| \) must be understood as \( \log(1/2) \) in accordance with the fact that read counts for copy number 0 are Poisson distributed with parameter \( \lambda/2 \) (see ‘The Model’ section). For \( \alpha = (0, 0, 1, 0, \ldots, 0) \), the I/NI call is zero, whereas any other \( \alpha \) gives a positive I/NI call. The more copy numbers differ from 2, the higher is the I/NI call, where gains and losses are treated on the same level by the absolute value of the logarithm. The rightmost term in Equation (8) makes clear that the I/NI call can be understood as the sum of individual I/NI calls, \( I/NI(\alpha_k) \), that is, the contribution of the \( k \)-th sample to the I/NI call:

\[
I/NI(\alpha) = \sum_{i=0}^{N} \sum_{a_k} a_k |\log(i/2)| = \sum_{i=0}^{N} \frac{1}{N} \sum_{\alpha_k} I/NI(\alpha_k).
\]

(9)

where \( \alpha_k = (\alpha_{0k}, \alpha_{1k}, \ldots, \alpha_{Nk}) \) is the vector of posteriors for the read count \( x_k \).

**Segmentation and CNV call**

Segmentation is an important step in CNV detection as it determines the length and position of a CNV (Figure 1). Class (a) methods perform segmentation on the GC-corrected (log) read counts, while ratio-based methods perform segmentation on the (log) ratios. Some methods, such as JointSLM, apply an HMM for segmentation and CNV detection.

In the cn.MOPS pipeline, segmentation is based on the results of the modeling step. More specifically, cn.MOPS detects CNVs by segmenting the chromosomes of
individuals based on their individual I/NI calls, joining genomically adjacent I/NI calls that show the same copy numbers. Note, however, that the I/NI call defined in Equation (9) does not distinguish between losses and gains with the same fold change. To avoid joining losses and gains, we define the signed individual I/NI call as the expected log fold change:

$$s_{\text{I/NI}(\alpha_i)} = \sum_{k=0}^{10} \alpha_k \log(2)^k.$$  

The absolute value of the signed I/NI call $|s_{\text{I/NI}(\alpha_i)}|$ is not exactly the I/NI call $s_{\text{I/NI}(\alpha_i)}$, but the two values are always very close (see Supplementary Section S2.4, for detailed mathematical analysis and experimental evaluations).

cn.MOPS applies either its own algorithm ‘fastseg’ or, alternatively, the circular binary segmentation algorithm [DNAcopy (32)] to $s_{\text{I/NI}(\alpha_i)}$ along the chromosome. The segmentation algorithm joins consecutive segments with large or small expected fold changes to make a candidate segment. It then supplies candidate segments that show variations along the chromosome and also across samples indicated by the signed individual I/NI calls.

All CNV detection methods except those based on HMMs decide on the basis of a threshold on the average/median (log) read count or (log) ratio over the segments whether the candidate segments are CNVs. In the cn.MOPS pipeline, a candidate segment is called a CNV segment if the median of the signed individual I/NI call $s_{\text{I/NI}(\alpha_i)}$ over the segment is at least 0.6 $\log(3/2)$ for gains or at most $-1 - \log(1/2)$ for losses. This $CNV$ call incorporates two calls: (i) an I/NI call across samples and (ii) a segment call along the chromosome. Only if consecutive segments obtain an I/NI call, they are joined by the segmentation algorithm (see second bar and third sample in Figure 2). This idea of calling a CNV by detecting both variation across samples and variation along a chromosome has already led to improvements in CNV detection based on DNA microarray data using the cn.FARMS method (24).

**Integer copy number estimation**

The final step is concerned with estimating the integer copy numbers of the CNVs (Figure 1). Methods based on HMMs, such as JointSLM, automatically obtain integer copy numbers by means of their hidden states. However, most existing methods do not estimate the integer copy numbers of the CNVs.

cn.MOPS automatically supplies posterior estimates of the integer copy numbers for each segment (Figure 1). The estimated copy number of a CNV is the most probable posterior copy number, where the segment posterior within the CNV are assumed to be independent.

**RESULTS**

In order to compare methods that detect copy number variations in NGS data, we first specify the evaluation procedure. Subsequently, we provide an overview of the methods compared and finally we present results on four benchmark data sets.

**Evaluation of CNV detection results**

We assume that the true CNVs are known and to be rediscovered. Each chromosome is split into equally large evaluation segments the size of which is chosen to accommodate the shortest known CNV. An evaluation segment is called a true positive (TP) if it is entirely contained in a true CNV and in a detected CNV segment. It is called a false negative (FN) if it is entirely contained in a true CNV but does not overlap with any predicted CNV segment. An evaluation segment is called a false positive (FP) if it is entirely detected as a CNV segment but does not overlap with any true CNV. Finally, it is called a true negative (TN) if it overlaps neither with a true CNV nor with a detected CNV segment.

These definitions imply that all evaluation segments that partially overlap with true CNVs or detected CNV segments remain ignored, as the copy numbers in these segments are ambiguous. Figure 3 illustrates the definitions of the four categories of evaluation segments. The two measures we employ hereafter are recall $\frac{\#TP}{\#TP + \#FN}$ and precision $\frac{\#TP}{\#TP + \#FP}$.

Note that precision is 1-FDR, in which we are especially interested. A CNV calling threshold governs the trade-off between recall and precision or, in other words, the trade-off between FNs and FPs, because more detected CNVs lead to more FPs but fewer FNs, and vice versa. To assess the performance of methods at different CNV
calling thresholds, we use precision-recall curves. Precision-recall curves are independent of the number of TNs, which makes them an ideal tool for our evaluation, as the majority of samples are negatives (non-CNVs).

Methods compared

We compared the following methods (see the ‘Introduction’ section for an overview):

1. cn.MOPS: our new model and pipeline,
2. MOFDOC: according to the variant of Alkan et al. (6),
3. EWT: Yoon et al. (15),
4. JointSLM: Magi et al. (16),
5. CNV-Seq: Xie and Tammi (20),
6. FREEC: Boeva et al. (21).

In this subsection, we provide an overview of the parameter settings, initializations and how these methods were employed.

cn.MOPS: we initialized the parameter λ with the median read count \( \tilde{x} \). We set \( \varepsilon = 0.05 \) and assumed nine possible copy numbers \( 0 \leq i \leq n = 8 \), which covers previously observed copy numbers in the HapMap individuals (17). The parameter \( \alpha \) should be initialized close to the location of the prior’s mode \((0, 0, 1, 0, \ldots, 0)\), which are the optimal parameters if all samples have copy number 2. However, initializing \( \alpha \) with this vector would clamp all \( z_{ik} \) and \( \sigma_{ik}^{\text{new}} \) to zero according to Equation (6) and Equation (7).

Therefore, we initialized \( \alpha \) with \((0.05, 0.05, 0.6, 0.05, \ldots, 0.05)\).

MOFDOC: we implemented MOFDOC using the CNV calling criterion of Alkan et al. (6). A CNV region is called if a set of \( b \) consecutive segments show a read count with a z-score below or above thresholds specified to call a loss or gain segment (default values \( a \sim 6 \) and \( b \sim 7 \)).

We generalized this ‘a-b-smoother’ to a smoothing algorithm that is not only able to smooth logical, but also real values arising from CNV calls, which improved MOFDOC’s results.

EWT: we reimplemented event-wise testing as described by Yoon et al. (15), but improved the GC correction by using all samples to estimate the GC effect. Further, we restricted the minimum ‘event size’, a parameter that prevents EWT from testing for CNVs that are too short. We generalized EWT to a variety of segment sizes that are also apt for low coverage CNV detection. Our modifications to EWT improved its results.

JointSLM: we applied version 0.1 of the R package jointSLM adjusting the package to GC content correction for a variety of segment sizes.

CNV-Seq: we used the authors’ implementation (http://tiger.dbs.nus.edu.sg/cnv-seq/), taking the median of the samples’ read counts as reference read count.

FREEC: we used version 3.2 (http://bioinfo-out.curie.fr/projects/freec/), taking, analogously to CNV-Seq, the median of the samples’ read counts as reference. Although FREEC can perform the analysis without a reference, we used it in ‘reference mode’ because of the improved performance.

We did not include SeqSeg (7) because we were not able to find suitable parameters, not even after an extensive search. The problem was that SeqSeg either did not detect any breakpoints or the thresholds for the \( p \)-values were not determined. However, the performance of SeqSeg can be estimated via CNV-Seq that is very similar. We also omitted CNAseg (19) from the comparison as its developers state that this method is specifically tailored to CNA detection in tumor samples.

To ensure a fair comparison, the parameters of the methods were optimized on simulated data sets similar to the one we used in our first experiment. More details, for instance, on the parameters that were used and computation time, can be found in the ‘Discussion’ section.

Simulated data with constructed CNVs

We constructed 100 artificial benchmark data sets, assuming an artificial genome to consist of a single chromosome of 125 Mb length, divided into 5000 segments of length 25 kb. We created 40 samples by sampling read counts for all segments and samples according to a Poisson process. The overall number of evaluation segments was therefore \( 40 \times 5000 \sim 200000 \).

The Poisson parameters \( \lambda \) of the Poisson process for simulating the read counts in the evaluation segments were drawn from a distribution of \( \lambda \) values that was estimated using median GC-corrected read counts of HapMap individuals from the 1000 Genomes Project. Therefore, the simulated Poisson distributions are similar to those found in real sequencing experiments. We scaled these \( \lambda \) values by a random number between 0.3 and 1 in order to simulate different coverages. This read count simulation yielded 290 000–770 000 reads, corresponding to a coverage of 0.18–0.46 and 0.08–0.22 for 75 and 36 bp reads, respectively.

Note that we consider low-coverage sequencing data here, because methods for analyzing SNPs and CNVs in low-coverage data will continue to be relevant in the future. Le and Durbin (33) showed that low-coverage data will remain important in the context of single nucleotide polymorphism (SNP) data analysis: in terms of a study’s discovery power, where a fixed number of reads should rather be used for sequencing more samples with lower coverage than for sequencing fewer samples with higher coverage. The relationship between discovery power and coverage is similar for CNV data. In the ‘High Coverage Data Sets’ section below, we also simulate high coverage data sets.

On the basis of HapMap data (17), we determined CNV region characteristics and how copy numbers are distributed. We implanted 20 CNV regions into each of the benchmark chromosomes. The lengths of the CNV regions were chosen randomly from the interval 75–200 kb, which, according to Xie and Tammi (20), is the range of accurate detection for the given coverage. The 20 starting points of the CNV regions were chosen randomly along the chromosome. After having determined the 20 CNV regions, we had to decide how CNVs are implanted into the single samples. To this end, we first had to take into account that CNVs of
different individuals cluster at specific regions of the DNA called ‘CNV regions’, of which many contain only losses or only gains. Based on characteristics of HapMap individuals, we assigned CNV region types such that 80% are of type ‘loss region’ (containing only losses), 15% of type ‘gain region’ (containing only gains), and 5% of type ‘mixed region’ (containing both losses and gains). Then the actual copy number for each sample was drawn according to the copy numbers observed for HapMap individuals (17): For a CNV region of type ‘loss region’, a sample has probabilities of 0.8, 0.15 and 0.05 of having copy numbers 2, 1 and 0, respectively. For a CNV region of type ‘gain region’, a sample has probabilities of 0.85, 0.08, 0.06 and 0.01 of having copy numbers 2, 3, 4 and 5, respectively. For a CNV region of type ‘mixed region’, a sample has probabilities 0.04, 0.16, 0.67, 0.11 and 0.02 of having copy numbers 0, 1, 2, 3 and 4, respectively. Of the 200000 evaluation segments, on average 101 (± 56) are gains and 612 (±104) are losses. The CNV lengths range from 75006 bp to 199848 bp with an average of 136921 bp.

Table 1 reports the performance of the compared copy number detection methods separately for gains and losses. As evaluation measures, we use the area under the precision-recall curve and the recall for a fixed FDR of 0.05. All methods perform better at detecting losses because of the low number 1 (see Supplementary Section S3.3). JoinSLM performs worse than other methods because of the low percentage of samples showing an abnormal copy number (the rare events). cn.MOPS yielded the largest average area under the precision-recall curve. The

Table 1. Performance of the compared copy number detection methods on the artificial benchmark data set

<table>
<thead>
<tr>
<th></th>
<th>PR AUC</th>
<th>P-value</th>
<th>Recall</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gains</td>
<td>cn.MOPS</td>
<td>0.94</td>
<td>-</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td>MOFDOC</td>
<td>0.83</td>
<td>1.14e-13</td>
<td>0.76</td>
</tr>
<tr>
<td></td>
<td>EWT</td>
<td>0.79</td>
<td>5.95e-14</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>JointSLM</td>
<td>0.25</td>
<td>4.23e-18</td>
<td>0.22</td>
</tr>
<tr>
<td></td>
<td>CNV-Seq</td>
<td>0.35</td>
<td>4.23e-18</td>
<td>0.35</td>
</tr>
<tr>
<td></td>
<td>FREEC</td>
<td>0.65</td>
<td>1.95e-17</td>
<td>0.53</td>
</tr>
<tr>
<td>Losses</td>
<td>cn.MOPS</td>
<td>0.96</td>
<td>-</td>
<td>0.96</td>
</tr>
<tr>
<td></td>
<td>MOFDOC</td>
<td>0.92</td>
<td>3.50e-17</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td>EWT</td>
<td>0.91</td>
<td>3.20e-18</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td>JointSLM</td>
<td>0.34</td>
<td>1.98e-18</td>
<td>0.28</td>
</tr>
<tr>
<td></td>
<td>CNV-Seq</td>
<td>0.81</td>
<td>1.98e-18</td>
<td>0.31</td>
</tr>
<tr>
<td></td>
<td>FREEC</td>
<td>0.73</td>
<td>1.98e-18</td>
<td>0.72</td>
</tr>
</tbody>
</table>

‘PR AUC’ gives the average area under the precision-recall curve of 100 experiments. The second column, ‘P-value’, reports the P-value of a Wilcoxon signed-rank test (over the 100 experiments) with the null hypothesis that cn.MOPS (in bold) and another method have the same area under the curve. ‘Recall’ reports the recall at a precision of 0.95, that is, an FDR of 0.05. The last column, ‘P-value’, gives the P-value of an analogous Wilcoxon test for the recall with an FDR of 0.05. cn.MOPS performed significantly better than all other methods.

Real sequencing data with implanted CNVs from the X chromosome
In contrast to simulated read counts, we next considered real reads obtained from the sequencing of a single male HapMap individual (NA20755). This man’s genome was sequenced 17 times by the Solexa Genome Analyzer II at the Wellcome Trust Sanger Institute [(26) see Supplementary Table S5]. These 17 samples ensure a constant copy number, as they stem from the same individual. We mapped the reads with Bowtie (25) for paired reads, allowing two mismatches. The numbers of reads range from 12069755 to 18101212, of which between 10419510 and 16041464 could be mapped, which corresponds to coverages between 0.13 and 0.21 (see Supplementary Section S3.3, for details on read mapping and the number of reads).

We created 110 benchmark data sets by choosing each of human chromosomes 1–22 five times, implanting 20 random CNV regions in each chromosome data set. The lengths of these implanted CNV regions were chosen to be 75, 100, 150, and 200 kb (5 each), and, for each of the regions, a random segment on the X chromosome which supplied reads for the region was selected. CNV region types and individual copy numbers were determined according to the procedure and distributions described in the first experiment except that we only considered CNV copy numbers 1 and 3 since they are the most difficult to distinguish from copy number 2. We assigned CNV region types such that 80% are of ‘loss region’ type, 15% of ‘gain region’ type, and 5% of ‘mixed region’ type. For a CNV region of ‘loss region’ type, a sample has probabilities 0.8 and 0.2 of having copy number 2 and 1, respectively, for a CNV region of ‘gain region’ type, 0.85 and 0.15 of having copy numbers 2 and 3, respectively, and for a CNV region of ‘mixed region’ type, 0.2, 0.67 and 0.13 of having copy numbers 1, 2 and 3, respectively. Finally, the read counts of the 17 samples were computed in the following way: outside CNVs the original reads counts were used; within CNVs, we added as many read counts as there are copies from the corresponding segment on the X chromosome, taking independent read counts from the considered sample and other random samples.

The CNV detection results were evaluated as described in the ‘Evaluation of CNV Detection Results’ section. The number of evaluation segments ranges from around 32000 for chromosome 21 to around 168000 for chromosome 1. On average, 0.1% of the evaluation segments are gains and 0.4% are losses.

Table 2 reports the performance of the compared copy number detection methods separately for gains and losses. As before, we use the area under the precision-recall curve and the recall for the FDR fixed at 0.05. Again, all
Tables show the performance of the six compared methods at rediscovering known CNVs for the 18 HapMap individuals, where the average area under the precision-recall curve is used as evaluation criterion. As found in previous experiments, all methods perform better at detecting losses. cn.MOPS performs significantly better than its competitors in terms of both the PR AUC and the recall for an FDR of 0.05, although FREEC performs equally well for gains.

So far we have considered CNV detection as a classification task whose goal was to detect CNVs in individual samples. In order to assess the quality of the CNV calling across HapMap samples, we also investigated the performance of each method at a different task—detecting segments in which at least one CNV occurs in one sample. For this task, we did not obtain segments from a segmentation algorithm, but we computed a CNV call for each evaluation segment. The CNV calls must be defined depending on the method. For cn.MOPS, we utilized the I/NI call. For z-score based methods, namely MOFDOC, EWT and JointSLM, we used the mean of the z-score on the evaluation segment. For the ratio-based methods, CNV-Seq and FREEC, we took the mean log-ratios of the evaluation segments. The area under precision-recall curve was 0.18 for the I/NI call, 0.02 for the mean z-score, and 0.14 for the mean log-ratio. The area under curve values are lower than in the other experiments because outliers were not filtered out by a segmentation algorithm. Alternative CNV calls such as variance and maximum-based values are reported in Supplementary Section S3.5.

Figure 4 visualizes the results of this comparison in the form of whole-genome CNV calling plots along a evaluation segments. cn.MOPS separates true CNVs (indicated by red dots) from non-CNV segments (blue dots) more clearly than the other methods. Furthermore, cn.MOPS has lower FDRs for different calling thresholds, as can be seen from the lower variance of the blue dots at the bottom. The superior performance of cn.MOPS at CNV calling across samples is the reason why cn.MOPS outperformed the other methods in previous experiments.

High coverage data sets

Finally, we compared the performance of CNV detection methods on two high coverage data sets. The first data set is simulated analogously to our previous simulated data but now with high coverage. On this data set we first show that, if we fix the resolution, higher coverage leads to better performance in terms of precision and recall. Next we show that, if we fix the performance in terms of precision and recall, higher coverage allows for higher resolution. The second data set consists of six high coverage methods performed better at detecting losses. If we adhere to the Poisson assumption, the reasons for this performance difference are the same as those stated in the first experiment. cn.MOPS significantly outperformed all other methods with respect to both the area under the precision-recall curve (PR AUC) and the recall for FDR at 0.05. No method considers the variation of the read counts across samples, except cn.MOPS, which estimates this variation via its Poisson parameter, thus achieving superior performance.

Rediscovery of known CNVs in HapMap sequencing data

Next, we compared how well the methods are able to rediscover known CNVs of HapMap individuals whose DNA was sequenced by the Solexa Genome Analyzer II at the Wellcome Trust Sanger Institute (26). We focused on 18 individuals for each of whom the reads were produced on one lane (one sequencing run contains seven lanes). The reads were mapped by Bowtie (25) for paired reads, allowing three mismatches. The numbers of reads range from 12 442 124 to 31 977 690, of which 7 498 420–22 217 020 could be mapped, which lead to a coverage between 0.20 and 0.60 (see Supplementary Section S3.4, for details on individuals, read mapping and the number of reads). We considered the CNVs of these 18 individuals, determined previously by means of microarrays (17), to be the true CNVs. They were detected by the Affymetrix Human SNP array 6.0 and reconfirmed with the Illumina HumanM–single BeadChip. After filtering for CNVs larger than 75 kb, we obtained 170 CNVs, of which 66 are gains and 104 are losses, with lengths ranging from 76 kb to 457 kb. Though some of these CNVs might still be false positives, the double confirmation and considering only CNVs of vast lengths approaches a golden standard. The CNV detection results were evaluated as described in the ‘Evaluation of CNV Detection Results’ section with evaluations segments of length 25 kb. In total, we have 2064906 evaluation segments, of which 450 are labeled as losses, as they lie within one of the 104 loss CNVs, and 469 are labeled as gains, as they lie within one of the 66 gain CNVs.

Table 3 shows the performance of the six compared methods at rediscovering known CNVs for the 18 HapMap individuals, where the average area under the precision-recall curve is used as evaluation criterion. As found in previous experiments, all methods perform better at detecting losses. cn.MOPS performs significantly better than its competitors in terms of both the PR AUC and the recall for an FDR of 0.05, although FREEC performs equally well for gains.

So far we have considered CNV detection as a classification task whose goal was to detect CNVs in individual samples. In order to assess the quality of the CNV calling across HapMap samples, we also investigated the performance of each method at a different task—detecting segments in which at least one CNV occurs in one sample. For this task, we did not obtain segments from a segmentation algorithm, but we computed a CNV call for each evaluation segment. The CNV calls must be defined depending on the method. For cn.MOPS, we utilized the I/NI call. For z-score based methods, namely MOFDOC, EWT and JointSLM, we used the mean of the z-score on the evaluation segment. For the ratio-based methods, CNV-Seq and FREEC, we took the mean log-ratios of the evaluation segments. The area under precision-recall curve was 0.18 for the I/NI call, 0.02 for the mean z-score, and 0.14 for the mean log-ratio. The area under curve values are lower than in the other experiments because outliers were not filtered out by a segmentation algorithm. Alternative CNV calls such as variance and maximum-based values are reported in Supplementary Section S3.5.

Figure 4 visualizes the results of this comparison in the form of whole-genome CNV calling plots along a evaluation segments. cn.MOPS separates true CNVs (indicated by red dots) from non-CNV segments (blue dots) more clearly than the other methods. Furthermore, cn.MOPS has lower FDRs for different calling thresholds, as can be seen from the lower variance of the blue dots at the bottom. The superior performance of cn.MOPS at CNV calling across samples is the reason why cn.MOPS outperformed the other methods in previous experiments.

High coverage data sets

Finally, we compared the performance of CNV detection methods on two high coverage data sets. The first data set is simulated analogously to our previous simulated data but now with high coverage. On this data set we first show that, if we fix the resolution, higher coverage leads to better performance in terms of precision and recall. Next we show that, if we fix the performance in terms of precision and recall, higher coverage allows for higher resolution. The second data set consists of six high coverage
samples from the 1000 Genomes Project on which we show that cn.MOPS is well suited for high coverage data sets (for details see Supplementary Section S3.6). If the majority of samples has a copy number different from 2, then the cn.MOPS model regards this copy number as copy number 2. If the majority of samples has a copy number different from 0, then the cn.MOPS model regards this copy number as copy number 1.

Second, we analyzed the dependencies between coverage and resolution for cn.MOPS. We implanted CNVs of different ranges of lengths 1–5 kb, 5–25 kb, 25–75 kb and 100–125 kb into chromosomes of lengths 25, 125, 250 and 250 Mb, respectively. For each of the coverages 1 kb, 5 x, 10 x, 25 x or 50 x and each range of CNV lengths, cn.MOPS is evaluated on 10 simulated data sets. In each run, we chose the segment size as a fifth of the minimal CNV length. Table 4 shows the recall of cn.MOPS for different coverages, again at a fixed FDR of 0.05. Obviously, for a given performance threshold of 0.95, higher coverage allows for higher resolution.

### Table 3. Performance of the compared copy number detection methods on HapMap individuals, where known CNVs should be rediscovered

<table>
<thead>
<tr>
<th></th>
<th>PR AUC</th>
<th>P-value</th>
<th>Recall</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gains</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cn.MOPS</td>
<td>0.35</td>
<td>–</td>
<td>0.24</td>
<td>–</td>
</tr>
<tr>
<td>MOFDOC</td>
<td>0.13</td>
<td>1.17e-03</td>
<td>0.06</td>
<td>1.95e-03</td>
</tr>
<tr>
<td>EWT</td>
<td>0.16</td>
<td>5.34e-04</td>
<td>0.10</td>
<td>1.86e-02</td>
</tr>
<tr>
<td>JointSLM</td>
<td>0.08</td>
<td>3.81e-05</td>
<td>0.05</td>
<td>7.81e-03</td>
</tr>
<tr>
<td>CNV-Seq</td>
<td>0.22</td>
<td>1.74e-02</td>
<td>0.21</td>
<td>3.61e-01</td>
</tr>
<tr>
<td>FREEC</td>
<td>0.35</td>
<td>8.68e-01</td>
<td>0.17</td>
<td>2.36e-01</td>
</tr>
<tr>
<td><strong>Losses</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cn.MOPS</td>
<td>0.55</td>
<td>–</td>
<td>0.45</td>
<td>–</td>
</tr>
<tr>
<td>MOFDOC</td>
<td>0.40</td>
<td>2.67e-04</td>
<td>0.33</td>
<td>3.42e-03</td>
</tr>
<tr>
<td>EWT</td>
<td>0.36</td>
<td>7.63e-06</td>
<td>0.23</td>
<td>6.10e-05</td>
</tr>
<tr>
<td>JointSLM</td>
<td>0.15</td>
<td>3.81e-06</td>
<td>0.06</td>
<td>1.55e-05</td>
</tr>
<tr>
<td>CNV-Seq</td>
<td>0.32</td>
<td>7.63e-05</td>
<td>0.27</td>
<td>3.66e-04</td>
</tr>
<tr>
<td>FREEC</td>
<td>0.42</td>
<td>2.37e-03</td>
<td>0.26</td>
<td>1.01e-02</td>
</tr>
</tbody>
</table>

PR AUC gives the average area under the precision-recall curve of 18 samples. The second column, ‘P-value’, reports the P-value of a Wilcoxon signed-rank test (over the 18 samples) with the null hypothesis that cn.MOPS (in bold) and another method have the same area under the curve. ‘Recall’ reports the recall at a precision of 0.95, that is, an FDR of 0.05. The last column, ‘P-value’, gives the P-value of an analogous Wilcoxon test for the recall with an FDR of 0.05. cn.MOPS rediscovered known CNVs most reliably. Only for gains the performance of FREEC is similar to that of cn.MOPS, whereas cn.MOPS rediscovered known CNVs most reliably. Only for gains the performance of FREEC is similar to that of cn.MOPS, whereas cn.MOPS performs significantly better than all its competitors at losses.

DISCUSSION

Data Access. The data of the second, third and fourth experiment are part of the 1000 Genomes Project. For the second experiment we used one chromosome from a Tuscan sample (NA20755), for the third experiment 18 samples from Pilot Phase 1, and for the fourth experiment chromosome 1 of 6 high coverage samples in order to comply with the Ft. Lauderdale principle for use of unpublished data for method development.

Limitations. cn.MOPS cannot be applied to a single sample because it decomposes variations along samples into those stemming from copy numbers and those from noise. The quality of this decomposition increases with the number of samples. We recommend to use at least 6 samples for proper parameter estimation (see Supplementary Section S3.9). If the majority of samples has a copy number different from 2, then the cn.MOPS model regards this copy number as copy number 2. However, this incorrect assignment of components to copy numbers can readily be corrected by comparing the expected read counts (the parameter \( \lambda \)) along the chromosome.
Computational Costs. With massively growing amounts of NGS data, computation time is becoming an increasingly important, and possibly limiting, factor in CNV analysis. To create an impression of the computational cost of cn.MOPS, we report the computation times for a medium coverage data set (see Supplementary Section S3.7 and Supplementary Table S17). The data set consists of chromosome 20 of 38 samples from the 1000 Genomes Project with coverages ranging from 2.5× to 8×. Not surprisingly, the model-free approaches MOFDOC (110s), EWT (239s) and CNV-Seq (96s) were faster than the model-based approaches cn.MOPS (250s), JointSLM...
Table 4. Average recall values of cn.MOPS at an FDR of 0.05 for different levels of coverage and different CNV lengths, along with their standard deviations over the 10 runs.

<table>
<thead>
<tr>
<th>Coverage</th>
<th>Gains of length [kbp]</th>
<th>Losses of length [kbp]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1x</td>
<td>5x</td>
</tr>
<tr>
<td>Gains</td>
<td>1-5</td>
<td>0.00±0.00</td>
</tr>
<tr>
<td></td>
<td>5-25</td>
<td>0.28±0.16</td>
</tr>
<tr>
<td></td>
<td>25-75</td>
<td>0.80±0.30</td>
</tr>
<tr>
<td></td>
<td>100-125</td>
<td>&gt;0.95</td>
</tr>
<tr>
<td>Losses</td>
<td>1-5</td>
<td>0.00±0.00</td>
</tr>
<tr>
<td></td>
<td>5-25</td>
<td>0.24±0.02</td>
</tr>
<tr>
<td></td>
<td>25-50</td>
<td>&gt;0.95</td>
</tr>
<tr>
<td></td>
<td>100-125</td>
<td>&gt;0.95</td>
</tr>
</tbody>
</table>

Table 5. Performance of the compared copy number detection methods using six high coverage samples from the 1000 Genomes Project.

<table>
<thead>
<tr>
<th></th>
<th>PR</th>
<th>AUC</th>
<th>Recall</th>
<th>PR</th>
<th>AUC</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn.MOPS</td>
<td>0.34</td>
<td>0.92</td>
<td>0.33</td>
<td>0.59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODiLOC</td>
<td>0.00</td>
<td>0.00</td>
<td>0.21</td>
<td>0.28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EWT</td>
<td>0.00</td>
<td>0.00</td>
<td>0.30</td>
<td>0.37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JointSLM</td>
<td>0.01</td>
<td>0.00</td>
<td>0.26</td>
<td>0.26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNV-Seq</td>
<td>0.14</td>
<td>0.78</td>
<td>0.26</td>
<td>0.38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FREEC</td>
<td>0.26</td>
<td>0.92</td>
<td>0.22</td>
<td>0.25</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PR AUC gives the area under precision-recall curve. Recall reports the recall at a precision of 0.1. cn.MOPS performs best, where FREEC performs equally well for gains in terms of recall.

DNA fragments are first captured by hybridization to probes attached to baits and then sequenced. For exon sequencing, the read counts show higher variation along the chromosome because hybridization and cross-hybridization effects are introduced via the baits. Thus, cn.MOPS is even better suited to this task than other methods. First results are very promising.

CONCLUSION

We have introduced cn.MOPS—a novel method and pipeline for the detection of copy number variations in NGS data. cn.MOPS incorporates a probabilistic model that decomposes read variations across samples into integer copy numbers and noise by means of its mixture components and its Poisson distributions, respectively. cn.MOPS is able to control the FDR for CNV detection via a Dirichlet prior on the model’s mixture components. The Dirichlet prior prefers a constant copy number of 2 for all samples, which corresponds to the null hypothesis. The more the data drag the posterior away from the Dirichlet prior, the more likely a CNV is present in the data.

We compared cn.MOPS with the five most popular CNV detection methods using four benchmark data sets. For all benchmarks, cn.MOPS outperformed its competitors, especially in terms of FDR.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online: Supplementary Tables S1–S17, Supplementary Figures S1–S14 and Supplementary Sections S1–S4.

FUNDING

Funding for open access charge: Funds from the Institute of Bioinformatics, Johannes Kepler University Linz.

Conflict of interest statement. None declared.

REFERENCES


cn.MOPS: mixture of Poissons for discovering copy number variations in next generation sequencing data with a low false discovery rate

— Supplementary Information —

Günter Klambauer, Karin Schwarzbauer, Andreas Mayr, Djork-Arné Clevert, Andreas Mitterecker, Ulrich Bodenhofer, and Sepp Hochreiter

Institute of Bioinformatics, Johannes Kepler University, Linz, Austria
### Contents

**S1 Introduction**  
Introduction to the study and its objectives

**S2 The Mixture of Poissons Model**  
S2.1 The Model  
S2.1.1 The Mixture of Poissons  
S2.1.2 Estimation of Integer Copy Numbers  
S2.1.3 The Poisson assumption is justified after normalization  
S2.2 Model Selection: EM Algorithm  
S2.2.1 Dirichlet Prior on Alpha  
S2.2.2 Uniform Prior on \( \lambda \)  
S2.2.3 Upper Bound on the Negative Log Posterior  
S2.2.4 E-step  
S2.2.5 M-step: Alpha Optimization  
S2.2.6 M-step: Lambda Optimization  
S2.2.7 Update Rules  
S2.2.8 Parameter Setting for the Dirichlet Prior in the Update Rule  
S2.2.9 Three Posteriors in Our Framework  
S2.3 I/NI Call: Information Gain of Posterior over Prior  
S2.4 Segmentation and CNV Call  
S2.4.1 Segmentation  
S2.4.2 CNV Call of cn.MOPS  
S2.5 Noise Model Variants  
S2.5.1 Variant (a): Additive Poisson Noise for Each Segment  
S2.5.2 Variant (b): Additive Poisson Noise Scales with Average Read Count  
S2.5.3 Variant (c): Poisson Noise Level as Model Parameter  
S2.5.4 cn.MOPS Variants Tested on Simulated Data  
S2.5.5 cn.MOPS Variants Tested on Real Sequencing Data with Implanted CNVs from the X Chromosome  
S2.5.6 Conclusion  

**S3 Experiments**  
S3.1 Evaluation of CNV Detection Results  
S3.2 Compared CNV Detection Methods  
S3.2.1 cn.MOPS  
S3.2.2 Class (a) Methods: MOFDOC, EWT, and JointSLM  
S3.2.3 Class (b) Methods: SeqSeg, CNV-Seq, and FREEC  
S3.3 Simulated Sequencing Data with Constructed CNVs  
S3.3.1 Distribution of CNV Types and Copy Numbers for Data Generation  
S3.3.2 Results  
S3.3.3 Different Performance on Gains and Losses  
S3.4 Real Sequencing Data with Implanted CNVs From the X Chromosome  
S3.4.1 Data and Mapping  
S3.4.2 Results  
S3.4.3 Estimation of Integer Copy Numbers  
S3.5 Rediscovering of Known CNVs in HapMap Sequencing Data
List of Figures

S3.5.1 Data and Mapping ............................................. 32
S3.5.2 Results ......................................................... 33
S3.5.3 Evaluation based on a different criterion ................. 35
S3.5.4 CNV Calls of Different Methods ............................. 35
S3.6 High Coverage Real World Data Set .............................. 37
S3.7 Medium Coverage Data Set ....................................... 39
S3.8 Influence of the Hyperparameter $\epsilon$ on cn.MOPS Results 40
S3.8.1 Influence of the Hyperparameter $\epsilon$ tested on Simulated Data 40
S3.8.2 Influence of the Hyperparameter $\epsilon$ tested on Real Sequencing Data with Implanted CNVs .......................... 41
S3.9 Number of Samples vs. Performance for cn.MOPS .............. 41
S3.10 Exemplary DNA Locations With CNV Calls .................... 45
S3.10.1 CNV Calls at Exemplary DNA Locations With CNVs ..... 45
S3.10.2 CNV Calls at Exemplary DNA Locations Without CNVs .... 45

S4 Additional Information 51
S4.1 Distribution of Read Counts Along the Chromosome .......... 51
S4.2 Data Structure of Read Counts ................................ 51

List of Figures

S1 Illustration of the basic concept of cn.MOPS .......................... 17
S2 Definitions for the evaluation of copy number detection methods 22
S3 Histogram of CNV region types “loss region”, “gain region”, and “mixed region” 26
S4 Histograms of integer copy numbers ................................ 27
S5 Mean based whole genome CNV calling plots ....................... 37
S6 Variance based whole genome CNV calling plots ................... 38
S7 Maximum based whole genome CNV calling plots .................. 39
S8 Copy number call plot for CNV region HM_CNP_463 ............... 46
S9 Copy number call plot for CNV region HM_CNP_494 ............... 47
S10 Copy number call plot for CNV region HM_CNP_618 ............... 48
S11 Copy number call plot for a non-CNV region no. 1 ............... 49
S12 Copy number call plot for a non-CNV region no. 2 ............... 50
S13 Histogram for non-uniquestly mapped reads of GC corrected read counts 52
S14 Histogram for uniquely mapped reads of GC corrected read counts 53

List of Tables

S1 Contingency table of segments within known CNV regions and non-Poisson segments ........................................... 7
S2 Performance of cn.MOPS and variants on simulated data ........ 20
S3 Performance of cn.MOPS and variants on real sequencing data with implanted CNVs from the X chromosome ......................... 21
S4 Performance of the compared copy number detection methods on simulated ......................................................... 28
S5 Summary of the sequencing data for the implanted CNVs benchmark data set .......................................................... 31
S6 Performance of the compared copy number detection methods on real sequencing data with implanted CNVs ........................................... 32
S7 Summary of the sequencing data for the HapMap CNV reconfirmation .......... 33
S8 Performance of the compared copy number detection methods on HapMap individuals ................................................................. 34
S9 Recall of known copy number regions on HapMap individuals ................. 35
S10 Performance of different approaches without segmentation ..................... 36
S11 Information on the high coverage data set from the 1000 Genomes Project .... 38
S12 Performance of the compared copy number detection methods on the medium coverage data (gains only) ...................................... 40
S14 Performance of cn.MOPS on simulated data with different choices of the hyperparameter $\epsilon$ ...................................................... 41
S15 Performance of cn.MOPS on real world benchmarking data with different choices of the hyperparameter $\epsilon$ ................................. 41
S13 Overview of the medium coverage data set ........................................... 43
S16 Number of samples vs. performance for cn.MOPS .................................. 44
S1 Introduction

This report gives supplementary information to the manuscript “cn.MOPS: Mixture of Poissons for Discovering Copy Number Variations in Next Generation Sequencing Data”.

The supplementary information contain

- derivative of the algorithm, that is of the update rules of the EM algorithm and of the different I/NI calls,
- numerical tests of the approximations — especially of the signed I/NI call by the expected fold change,
- test for Poisson distribution with and without normalization of read counts,
- parameter settings for the compared CNV detection methods in the experiments,
- comparison of the likelihoods for copy number 2 if the true copy number is 3 or 1,
- additional experiments,
- additional information on the data used in the experiments.
- variants of the noise model of cn.MOPS
- investigations about the influence of the hyperparameter $\epsilon$
- exemplary DNA locations with CNV calls of different methods

S2 The Mixture of Poissons Model

Summary. To avoid the false discoveries induced by read count variations along the chromosome or across samples, we propose a “Mixture Of PoissonS model for CNV detection” (cn.MOPS). The cn.MOPS model is not affected by read count variations along the chromosome, because at each DNA position a local model is constructed. Read count variations across samples are decomposed by the cn.MOPS model into integer copy numbers and noise by its mixture components and Poisson distributions, respectively. In contrast to existing methods, cn.MOPS model’s posterior provides integer copy numbers together with their uncertainty. Model selection in a Bayesian framework is based on maximizing the posterior given the samples by an expectation maximization (EM) algorithm. The model incorporates the linear dependency between average read counts in a DNA segment and its copy number. Most importantly, a Dirichlet prior on the mixture components prefers constant copy number 2 for all samples. The more the data drives the posterior away from the Dirichlet prior corresponding to copy number two, the more likely the data is caused by a CNV, and, the higher is the informative/non-informative (I/NI) call. cn.MOPS detects a CNV in the DNA of an individual as a segment with high I/NI calls. I/NI call based CNV detection guarantees a low false discovery rate (FDR) because false detections are less likely for high I/NI calls.

We assume that the genome is partitioned into segments in which reads are counted but which need not be of constant length throughout the genome. For each of such an segment we build a
model. We consider the read counts $x$ at a certain segment of the genome, for which we construct a model across samples. The model incorporates both read count variations due to technical or biological noise and variations stemming from copy number variations.

S2.1 The Model

In this Subsection we introduce the cn.MOPS model, which models the read counts of the samples at a certain chromosome segment by copy numbers and noise due to technical or DNA variations.

S2.1.1 The Mixture of Poissons

The cn.MOPS model assumes that the read counts $x$ for a certain copy number $i$ are distributed across samples according to a Poisson. Assuming different copy numbers across samples, the cn.MOPS model is a mixture of Poissons:

$$p(x) = \sum_{i=0}^{n} \alpha_i P(x; \frac{i^2}{2}\lambda). \quad (S1)$$

In model Eq. (S1) $\alpha_i$ are the percentages of samples with copy numbers $0 \leq i \leq n$ and $\lambda$ is the mean as well as the variance of read counts for copy number 2, where $n$ is the number of different copy numbers. For copy number $i$, the Poisson parameter is $\frac{i^2}{2}\lambda$, by which we assume that the read counts are linearly related to the number of copies. $P$ is the density of the Poisson distribution:

$$P(x; \beta) = \frac{1}{x!} e^{-\beta} \beta^x. \quad (S2)$$

For notational convenience, we did not distinguish between $i = 0$ and $i \geq 1$ in the above formula. For copy number $i = 0$, we assume a Poisson distribution with parameter $\beta = \frac{\epsilon}{2}\lambda$ which accounts for background noise stemming from wrongly or ambiguously mapped reads as well as for sample contamination by other DNA.

S2.1.2 Estimation of Integer Copy Numbers

The model Eq. (S1) allows for estimating integer copy numbers with fixed model parameters $\alpha_i$ and $\lambda$. The prior probability that a read count stems from copy number $i$ is $p(i) = \alpha_i$. The likelihood that a read count $x$ is produced by the $i$-th mixture component is $p(x \mid i) = P(x; \frac{i^2}{2}\lambda)$. Then Bayes’ formula can be used to compute the posterior $p(i \mid x)$, that is, the probability that read count $x$ stems from the $i$-th component corresponding to copy number $i$. We estimate the copy number by the component that has the largest posterior probability. In Subsection S3.4.3 we found that 99.383% ($\pm$ 0.001%) of the integer copy numbers were correctly assigned using our posterior integer copy number estimate.

S2.1.3 The Poisson assumption is justified after normalization

The latent variable model of cn.MOPS assumes Poisson distributed read counts across samples for a segment with a constant copy number. This assumption is only justified if sample normalization is applied. Sample normalization corrects the read counts of one sample by the number
of mappable reads of the sample. We tested segments of constant size (25kbp) for being Poisson distributed with and without sample normalization. The data is from the Sanger sequencing center on HapMap phase 1 individuals (see Subsection S3.5). Without sample normalization the Poisson assumption was rejected for 92% of the genomic segments. With sample normalization the rejection rate has dropped to 2%. It is plausible, that two percent of segments were rejected by the Poisson test, if the occurrence of known CNV regions is considered. Table S1 shows the contingency table of segments within known CNV regions vs. non-CNV segments and Poisson vs. non-Poisson segments as determined by a test for Poisson suggested by Brown and Zhao (2002). Segments that were rejected by the Poisson test with sample normalization coincide significantly (p-value 2.2e-16) with segments within known CNV regions.

Table S1: Contingency table of segments within known CNV regions vs. non-CNV segments and Poisson vs. non-Poisson segments. The table gives counts of non-CNV segments (first column) and segments within known CNV regions (second column) and counts of segments not rejected by a Poisson test suggested by Brown and Zhao (2002) (first row) and segments rejected by the test (second row). Fisher’s exact test for coincidence of non-Poisson segments with segments within known CNV regions is highly significant with a p-value of 2.2e-16. Thus, segments within known CNV regions coincide with segments which are not Poisson distributed.

<table>
<thead>
<tr>
<th>Poisson assumption/segments</th>
<th>non-CNV</th>
<th>within known CNV region</th>
<th>sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>not rejected</td>
<td>111,876</td>
<td>145</td>
<td>112,021</td>
</tr>
<tr>
<td>rejected</td>
<td>2,573</td>
<td>123</td>
<td>2,697</td>
</tr>
<tr>
<td>sum</td>
<td>114,449</td>
<td>268</td>
<td>114,717</td>
</tr>
</tbody>
</table>

### S2.2 Model Selection: EM Algorithm

In a Bayes framework for model selection, $\alpha$ and $\lambda$ are considered as random variables, thus, $p(x)$ in Eq. (S1) becomes a conditional probability $p(x \mid \alpha, \lambda)$, i.e. the likelihood that read count $x$ has been produced by the model with parameters $\alpha$ and $\lambda$. The EM algorithm minimizes an upper bound on the negative log-posterior of the parameters. The parameter posterior of $\alpha$ and $\lambda$ is given by:

$$p(\alpha, \lambda \mid x) = \frac{p(x \mid \alpha, \lambda) \cdot p(\alpha) \cdot p(\lambda)}{\int p(x \mid \alpha, \lambda) \cdot p(\alpha) \cdot p(\lambda) \, d\alpha \, d\lambda} ,$$  

(S3)

where we assumed that the priors on $\alpha$ and $\lambda$ are independent of each other. This independence is justified because the copy number distribution $\alpha$ on the samples (determined by the cohort which is investigated) is independent of the expected read count $\lambda$ for copy number 2 (determined by DNA and biotechnological characteristics). We now introduce priors on $\alpha$ and $\lambda$. The parameter posterior $p(\alpha, \lambda \mid x)$ should not be confused with the posterior of latent variable $p(i \mid x)$, the probability that $x$ has been drawn from the $i$-th mixture component after having observed $x$.

#### S2.2.1 Dirichlet Prior on Alpha

In the cn.MOPS model, the prior $p(\alpha)$ on $\alpha$ should reflect the fact that predominantly locations with copy number 2 for all samples are present in the data set. Thus, the prior represents the null
hypothesis that at a location the copy number is the same across samples, i.e. no sample has a CNV. The Dirichlet prior is well suited to express our prior assumptions on $\alpha$. The Dirichlet prior with parameters $\gamma$ is:

$$p(\alpha) = D(\alpha^1; \gamma) = b(\gamma) \prod_{i=0}^{n} \alpha_i^\gamma - 1,$$  \hspace{1cm} (S4)

where $\alpha^1$ is the $n$-dimensional vector $(\alpha_1, \ldots, \alpha_n)$ while $\alpha_0$ is obtained via $\alpha_0 = 1 - \sum_{i=1}^{n} \alpha_i$. Each component $\alpha_i$ is distributed according to a beta distribution with mean

$$\text{mean}(\alpha_i) = \frac{\gamma_i}{\gamma_s},$$  \hspace{1cm} (S5)

mode

$$\text{mode}(\alpha_i) = \frac{\gamma_i - 1}{\gamma_s - n},$$  \hspace{1cm} (S6)

and variance

$$\text{var}(\alpha_i) = \frac{\gamma_i (\gamma_s - \gamma_i)}{\gamma_s (\gamma_s + 1)},$$  \hspace{1cm} (S7)

where we set

$$\gamma_s = \sum_{i=0}^{n} \gamma_i.$$  \hspace{1cm} (S8)

To express our prior knowledge that predominantly locations with copy number 2 for all samples are present, we set $\gamma_2 \gg \gamma_i$ for $i \neq 2$.

**S2.2.2 Uniform Prior on $\lambda$**

For the prior on $\lambda$ we use an uniform distribution on a sufficiently large interval $(0, 1/t]$ with left endpoint 0 and right endpoint $1/t$. Thus, the density in $(0, 1/t]$ is

$$p(\lambda) = \frac{t}{1/t}.$$  \hspace{1cm} (S9)

**S2.2.3 Upper Bound on the Negative Log Posterior**

According to Eq. (S3), the posterior of the model parameters is

$$p(\alpha, \lambda \mid x) = \frac{p(x \mid \alpha, \lambda) p(\alpha) p(\lambda)}{\int p(x \mid \alpha, \lambda) p(\alpha) p(\lambda) \, d\alpha \, d\lambda}$$

$$= \frac{1}{c(x)} p(x \mid \alpha, \lambda) p(\alpha),$$  \hspace{1cm} (S10)

where $c(x)$ is independent of the parameters $\alpha$ and $\lambda$. 

Deriving the upper bound. For deriving an upper bound on the log posterior needed by the EM algorithm, we deduce the following inequality for one sample $x$ by introducing variables $\hat{\alpha}_i$ with $\sum_{i=1}^n \hat{\alpha}_i = 1$:

$$ - \log p(\alpha, \lambda | x) = - \log (p(x | \alpha, \lambda) p(\alpha) / c(x)) $$

$$ = - \log \sum_{i=0}^n \alpha_i P(x; i \lambda) - \log p(\alpha) + \log(c(x)) $$

$$ = - \log \sum_{i=0}^n \frac{\hat{\alpha}_i}{\hat{\alpha}_i} \alpha_i P(x; i \lambda) - \log p(\alpha) + \log(c(x)) $$

$$ \leq - \sum_{i=0}^n \hat{\alpha}_i \log \frac{\alpha_i P(x; i \lambda)}{\hat{\alpha}_i} - \log p(\alpha) $$

$$ + \sum_{i=0}^n \hat{\alpha}_i \log \hat{\alpha}_i + \log(c(x)) , $$

where we applied Jensen’s inequality. Note that $c(x)$ is independent of $\alpha$ and that for

$$ \hat{\alpha}_i = p(i | x, \alpha, \lambda) = \frac{\alpha_i P(x; i \lambda)}{p(x | \alpha, \lambda)} $$

we have in the fifth line of Eq. (S11)

$$ \log \frac{\alpha_i P(x; i \lambda)}{\hat{\alpha}_i} = \log p(x | \alpha, \lambda) , $$

thus the inequality Eq. (S11) becomes an equality.

The data set. We assume that the data set $\{x_1, \ldots, x_N\}$ of the read counts across the samples is given, where the read count from the $k$-th sample is denoted by $x_k$. Model selection and therefore the EM algorithm is based on these samples. The posterior that $x_k$ is drawn from the $i$-th mixture component is

$$ \alpha_{ik} = p(i | x_k, \alpha, \lambda) = \frac{p(i) p(x_k | i, \alpha, \lambda)}{p(x_k | \alpha, \lambda)} = \frac{\alpha_i P(x_k; i \lambda)}{p(x_k | \alpha, \lambda)} , $$

where $\alpha_i$ is the prior of being drawn from the $i$-th mixture component.

S2.2.4 E-step

In analogy to the $\hat{\alpha}_i$ in Subsection S2.2.3, we introduce for each $x_k$ variables $\hat{\alpha}_{ik}$ with $\sum_{i=1}^n \hat{\alpha}_{ik} = 1$ which estimate $p(i | \alpha, x_k, \lambda)$ (see Eq. (S12)), are is formally independent of the parameters $\alpha$ and $\lambda$. For the E-step of the EM algorithm, we estimate the posterior $\alpha_{ik}$ by

$$ \hat{\alpha}_{ik} = \frac{\alpha_{ik}^{\text{old}} P(x_k; i \lambda^{\text{old}})}{p(x_k; \alpha^{\text{old}}, \lambda^{\text{old}})} , $$
where for the estimation the actual parameters $\alpha^{\text{old}}$ and $\lambda^{\text{old}}$ are used instead of the optimal parameters $\alpha$ and $\lambda$ in the expression for the posterior in Eq. (S14).

Based on inequality Eq. (S11) but with $\hat{\alpha}_{ik}$ instead of $\tilde{\alpha}_i$, we define an upper bound $B$ on the $\frac{1}{N}$ scaled negative log-posterior as

$$B = - \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \hat{\alpha}_{ik} \log \left( \alpha_i \mathcal{P}(x; \frac{i}{2}, \lambda) \right) - \frac{1}{N} \log p(\alpha)$$  \hspace{1cm} (S16)

$$+ \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \alpha_{ik} \log \hat{\alpha}_{ik} + \frac{1}{N} \sum_{k=1}^{N} \log c(x_k),$$

where we summed over all terms depending on $x_k$. Note, that according to Eq. (S12) and Eq. (S13) an exact estimate in the E-step Eq. (S15) (using the optimal parameters $\alpha$ and $\lambda$) make inequality Eq. (S11) to an equality, thus the upper bound $B$ would be equal to the negative log posterior. For notational convenience $\frac{\alpha^{\text{old}} i}{2 N \lambda}$ stands for $\frac{\alpha^{\text{old}} i}{2 N \lambda}$ according to the model defined in Eq. (S1).

### S2.2.5 M-step: Alpha Optimization

In the M-step, we minimize the upper bound $B$ on the negative log posterior with respect to $\alpha$ under the constraint that the $\alpha_i$ sum to 1. Only terms depending on $\alpha$ are considered:

$$\min_{\alpha} \quad - \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \hat{\alpha}_{ik} \log \alpha_i - \frac{1}{N} \log p(\alpha)$$ \hspace{1cm} (S17)

$$\text{s.t.} \quad \sum_{i=0}^{n} \alpha_i = 1 .$$

The Lagrangian with Lagrange parameter $\rho$ is

$$L = - \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \hat{\alpha}_{ik} \log \alpha_i - \frac{1}{N} \log p(\alpha)$$  

$$+ \rho \left( \sum_{i=0}^{n} \alpha_i - 1 \right)$$

$$= - \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \hat{\alpha}_{ik} \log \alpha_i - \frac{1}{N} \sum_{i=0}^{n} (\gamma_i - 1) \log \alpha_i$$

$$+ \rho \left( \sum_{i=0}^{n} \alpha_i - 1 \right).$$

The solution requires that, the derivative of $L$ with respect to $\alpha_i$ is zero:

$$\frac{\partial L}{\partial \alpha_i} = - \frac{1}{N} \sum_{k=1}^{N} \hat{\alpha}_{ik} \frac{1}{\alpha_i} - \frac{1}{N} \frac{1}{\alpha_i} (\gamma_i - 1) + \rho = 0 .$$  \hspace{1cm} (S19)
Multiplying this equation by $\alpha_i$ gives

$$- \frac{1}{N} \sum_{k=1}^{N} \hat{\alpha}_{ik} - \frac{1}{N} (\gamma_i - 1) + \rho \alpha_i = 0 .$$  \hfill (S20)

Summation over $i$ leads to

$$1 + \frac{1}{N} (\gamma_s - n) = \rho .$$  \hfill (S21)

Inserting this expression for $\rho$ in Eq. (S20) results in

$$- \frac{1}{N} \sum_{k=1}^{N} \hat{\alpha}_{ik} - \frac{1}{N} (\gamma_i - 1) + \left(1 + \frac{1}{N} (\gamma_s - n)\right) \alpha_i = 0 .$$  \hfill (S22)

Solving Eq. (S22) for $\alpha_i$ gives the update rule for $\alpha_i$:

$$\alpha_{i}^{\text{new}} = \hat{\alpha}_i + \frac{1}{N} (\gamma_i - 1) \frac{1}{1 + \frac{1}{N} (\gamma_s - n)} ,$$  \hfill (S23)

where we used

$$\hat{\alpha}_i = \frac{1}{N} \sum_{k=1}^{N} \hat{\alpha}_{ik} .$$  \hfill (S24)

We introduced $\hat{\alpha}_i$ which sums up the $\hat{\alpha}_{ik}$ and thereby approximates $\alpha_i$. This approximation is justified because $\alpha_i$ can be decomposed into $\alpha_{ik}$:

$$\alpha_i = p(i) = p(i | \alpha, \lambda) = \int p(i, x | \alpha, \lambda) \, dx$$  \hfill (S25)

$$= \int p(i | x, \alpha, \lambda) \, p(x | \alpha, \lambda) \, dx = \mathbb{E}_{p(x|\alpha, \lambda)} (p(i | x, \alpha, \lambda))$$

$$\approx \frac{1}{N} \sum_{k=1}^{N} p(i | x_k, \alpha, \lambda) = \frac{1}{N} \sum_{k=1}^{N} \alpha_{ik} .$$

### S2.2.6 M-Step: Lambda Optimization

In the M-step, $B$ need not only be minimized with respect to $\alpha$ but also with respect to $\lambda$ (only terms depending on $\lambda$ are considered):

$$\min_{\lambda} \left( - \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \hat{\alpha}_{ik} \log P(x_k; \frac{i}{2} \lambda) \right) .$$  \hfill (S26)

For the minimum, the derivative of the above objective with respect to $\lambda$ must be zero. Using

$$\log P(x_k; \frac{i}{2} \lambda) = - \log(x_k!) - \frac{i}{2} \lambda + x_k (\log(\lambda) + \log(i/2)) ,$$  \hfill (S27)
The derivative is
\[
- \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \left( - \frac{i}{2} + x_k \frac{1}{\lambda} \right) \hat{\alpha}_{ik}.
\]  

(S28)

Multiplying Eq. (S28) by \( \lambda \) and solving it for \( \lambda \) gives the update rule:
\[
\lambda^{\text{new}} = \frac{\sum_{k=1}^{N} \sum_{i=0}^{n} x_k \hat{\alpha}_{ik}}{\sum_{k=1}^{N} \sum_{i=0}^{n} \frac{i}{2} \hat{\alpha}_{ik}} = \frac{\sum_{k=1}^{N} x_k}{\sum_{k=1}^{N} \sum_{i=0}^{n} \frac{i}{2} \hat{\alpha}_{ik}}.
\]  

(S29)

where according to the model defined in Eq. (S1) for notational convenience \( \frac{\gamma}{2} \) stands for \( \frac{\gamma}{2} \).

**S2.2.7 Update Rules**

The update rules of previous subsections can be summarized as follows:

\[
\hat{\alpha}_{ik} = \frac{\alpha_{i,k}^{\text{old}} P(x_k; \frac{\gamma}{2}, \lambda^{\text{old}})}{p(x_k | \alpha^{\text{old}}, \lambda^{\text{old}})}, \]

(S30)

\[
\alpha_{i,k}^{\text{new}} = \frac{\frac{1}{N} \sum_{k=1}^{N} \hat{\alpha}_{ik} + \frac{1}{N} (\gamma_i - 1)}{1 + \frac{1}{N} (\gamma_i - n)},
\]

(S31)

\[
\lambda^{\text{new}} = \frac{\frac{1}{N} \sum_{k=1}^{N} x_k}{\sum_{i=0}^{n} \left( \frac{1}{N} \frac{\gamma}{2} \sum_{k=1}^{N} \hat{\alpha}_{ik} \right)}.
\]

(S32)

Concerning the EM algorithm the update rule Eq. (S30) is the E-step, the update rule Eq. (S31) is the M-step for \( \alpha \), and the update rule Eq. (S31) is the M-step for \( \lambda \).

The update rule Eq. (S31) can be obtained in an alternative way. The Dirichlet distribution is conjugate to the multinomial distribution, that is the posterior \( p(\alpha | \{ \alpha_1, \ldots, \alpha_k, \ldots, \alpha_N \} \) is a Dirichlet distribution as is the prior \( p(\alpha) \) with \( \alpha_k = p(\alpha | x_k) \). The Dirichlet prior \( p(\alpha) = D(\alpha^1; \gamma) \) with parameters \( \gamma \) leads to the conjugate posterior \( p(\alpha | \{ \alpha_1, \ldots, \alpha_k, \ldots, \alpha_N \} \) with parameters

\[
\hat{\gamma} = \gamma + \sum_{k=1}^{N} \alpha_k = \gamma + N \alpha,
\]

(S33)

where we used Eq. (S25). We obtain update rule Eq. (S31) from Eq. (S33) component-wise by first replacing the unknown values \( \alpha_{ik} \) by their estimates \( \hat{\alpha}_{ik} \) and then computing the posterior’s mode because we search for the maximum posterior.
S2.2.8 Parameter Setting for the Dirichlet Prior in the Update Rule Eq. (S31)

We set the parameter $\gamma$ of the Dirichlet prior $D(\alpha^1; \gamma)$ to $\gamma = (1, 1, 1 + G, 1, 1, \ldots, 1)$, where $G > 0$ is a hyperparameter that controls the prior’s impact during model selection. Note that $\gamma_s - n = I$ and therefore we obtain the mode $\alpha = (0, 0, 1, 0, \ldots, 0)$, since mode($\alpha_i$) = $\frac{\gamma_i - 1}{\gamma_s - n}$.

This mode corresponds to our null hypothesis that all samples have copy number 2. The mode is not affected by the choice of the hyperparameter $G$, however the variance decreases as we increase $G$:

$$\text{var}(\alpha_i) = \frac{\gamma_i (\gamma_s - \gamma_i)}{\gamma_s^2 (\gamma_s + 1)}.$$  

(S34)

The hyperparameter $G$ affects the EM algorithm via the update rule and can serve to keep percentage of samples having copy number 2 above a threshold. The smaller the variance, the less likely a deviation from copy number 2. The update rule is

$$\alpha_i^{\text{new}} = \frac{\frac{1}{N} \sum_{k=1}^{N} \hat{a}_{ik} + \frac{1}{N} (\gamma_i - 1)}{1 + \frac{1}{N} (\gamma_s - n)}.$$  

(S35)

Thus, the estimate for the percentage of samples having copy number 2 cannot fall below $\frac{G}{G+N}$ for $\gamma = (1, 1, 1 + G, 1, 1, \ldots, 1)$ because

$$\alpha_2^{\text{new}} \geq \frac{1}{N} (\gamma_2 - 1) + \frac{1}{N} \gamma_2 = \frac{G}{G+N}.$$  

(S36)

In our experiments we ensured that the estimate for the percentage of the samples having copy number 2 is always greater or equal to 50% by setting $G$ to $N$ ($G = N$) which leads to

$$\alpha_2^{\text{new}} = \frac{\hat{a}_2 + 1}{2} \geq \frac{1}{2},$$  

(S37)

$$\alpha_i^{\text{new}} = \frac{\hat{a}_i}{2} \text{ for } i \neq 2.$$  

(S38)

S2.2.9 Three Posteriors in Our Framework

In our Bayesian framework we introduced 3 different posterior distributions: (i) in Eq. (S14) the posterior $\alpha_{i,k} = p(i \mid x_k, \alpha, \lambda)$ of the data $x_k$ stemming from the $i$-th component with prior $\alpha_i = p(i)$ — this posterior is defined for fixed model parameters $(\alpha, \lambda)$; (ii) in Eq. (S3) the parameter posterior $p(\alpha, \lambda \mid x)$ with priors $p(\alpha)$ and $p(\lambda)$ — this posterior is the objective that we maximize during model selection; (iii) the posterior $p(\alpha \mid \{\alpha_1, \ldots, \alpha_k, \ldots, \alpha_N\})$ used in Eq. (S33) with prior $p(\alpha)$ — this posterior is used for the I/NI call (see Subsection S2.3), but in contrast to (ii) it is not the posterior for the full mixture of Poisson model but only for the multinomial distribution given by $\alpha$ where the posteriors $\alpha_{i,k} = p(i \mid x_k, \alpha, \lambda)$ from (i) serve as data. At (i) we consider the fixed parameter mixture model which can be combined with the parameter $\alpha$ multinomial model at (iii) to the full model at (ii) if the posterior on $\lambda$ analog to the posterior on $\alpha$ at (iii) is included.
S2.3 I/NI Call: Information Gain of Posterior over Prior

Based on cn.MOPS’ Bayesian approach to model selection, we define an informative/non-informative (I/NI) call analogous to the I/NI call obtained for the FARMS algorithm which excelled in summarization and gene filtering for microarray data (Hochreiter et al. 2006; Talloen et al. 2007, 2010).

In contrast to \( \lambda \), which captures noise variation, \( \alpha \) captures variation stemming from CNVs, therefore, its posterior indicates CNVs in the data. The I/NI call measures the information gain of the posterior compared to its prior distribution \( p(\alpha) \) which represents the null hypothesis that all samples have copy number 2. Therefore, the I/NI call measures the tendency to reject the null hypothesis based on the observed data.

The multidimensional distribution and the Dirichlet distribution are conjugate, therefore the posterior is also a Dirichlet distribution with parameters according to Eq. (S33). The I/NI call measures the information gain of the Dirichlet posterior \( p(\alpha|{\alpha_1, \ldots, \alpha_k, \ldots, \alpha_N}) \) over the Dirichlet prior \( p(\alpha) = D(\alpha_1^1; \gamma) \). The prior \( p(\alpha) \) represents the null hypothesis that all samples have copy number 2, therefore we set \( \gamma = (1, 1, 1 + G, 1, 1, \ldots, 1) \) which leads to the mode \( m = (0, 0, 1, 0, \ldots, 0) \). The I/NI call is the distance between the prior’s and the posterior’s mode.

We assess how much the prior assumption to see only copy number 2 has changed after having observed the data. Note, that we do not consider the variance because it is determined by the hyperparameter \( G \) and the number of samples. The difference between the prior’s mode and the posterior’s mode is component-wise

\[
\frac{\gamma_i - 1}{\gamma_s - n} - \frac{\gamma_i + N \alpha_i - 1}{\gamma_s + N - n} = \frac{N}{\gamma_s - n + N} \left( \frac{\gamma_i - 1}{\gamma_s - n} - \alpha_i \right). \quad (S39)
\]

The difference for copy number \( i \) is difference between the prior’s mode \( (\gamma_i - 1)/(\gamma_s - n) \) and the estimate \( \alpha_i \) of observing copy number \( i \) in the data set, where the difference is weighted by a factor.

If we assume a Poisson distribution for read counts of each copy number, then the read count distributions for copy numbers > 4 or 0 have less overlap with the copy number 2 read count distribution than copy number 1, 3, or 4 distributions (see Subsection S3.3.3). Further, the read count distribution for copy number 1 has less overlap with the copy number 2 read count distribution than the copy number 3 distribution as shown in Subsection S3.3.3. Summarizing, the more the copy number differs from 2, the less overlap has its read count distribution with those of copy number 2. Consequently, the more a read count differs from the average copy number 2 read count, the more likely a copy number different from copy number 2 is present.

We incorporate this fact of being more sure on read counts belonging to copy numbers which differ more from 2 into the I/NI call. We weight the difference between the prior’s mode \( m \) and the posterior’s mode per component by its absolute log fold change relative to copy number 2. Thus components 1 and 4, which half and double the read counts of copy number 2 respectively, are weighted equally. With \( m = (0, 0, 1, 0, \ldots, 0) \), we define the I/NI call as

\[
I/NI(\alpha) = \sum_{i=0}^{n} |m_i - \alpha_i| |\log(i/2)| = \sum_{i=0}^{n} \alpha_i |\log(i/2)|. \quad (S40)
\]

The I/NI call is the expected fold change given the data set. For notational convenience, we did not distinguish between \( i = 0 \) and \( i \geq 1 \) in the above formula. “\( \log(0/2) \)” must be understood as
\[ \log(\epsilon/2) \] — in accordance with the fact that read counts for copy number 0 are Poisson distributed with parameter \( \epsilon \lambda / 2 \). Note that the I/NI call does not depend on the value of \( \alpha_2 \) but it is a distance measure for vectors \( \| \alpha \|_1 = 1 \) and \( \alpha_i \geq 0 \) from \( \mathbf{m} \) as shown in the following.

Let \( \alpha_{-2} \) be the vector \( \alpha \) where the second component \( \alpha_2 \) is removed. We define the distance between two vectors \( \alpha_{1-2} \) and \( \alpha_{2-2} \), where from both the second component is removed, as

\[ \| L(\alpha_{1-2} - \alpha_{2-2}) \|_1 \] (S41)

with diagonal matrix \( L \) having diagonal elements \( L_{ii} = | \log(i/2) | \) for \( i > 0, i \neq 2 \) and \( L_{00} = | \log(\epsilon/2) | \). Eq. (S41) is a valid distance measure between vectors \( \| \alpha \|_1 = 1 \) where \( \alpha_i \geq 0 \) from \( \mathbf{m} \) and \( \alpha_i \geq 0 \). Therefore Eq. (S41) is a valid distance measure for \( \| \alpha \|_1 = 1 \) with \( \alpha_i \geq 0 \) between vectors \( \alpha^1 \) and \( \alpha^2 \). It follows that the I/NI call in Eq. (S40) is a valid distance measure of vectors \( \| \alpha \|_1 = 1 \) with \( \alpha_i \geq 0 \) between vectors \( \alpha_1 \) and \( \alpha_2 \). Consequently, \( \text{I/NI}(\mathbf{m}) = 0 \) and \( \text{I/NI}(\alpha) > 0 \) for \( \alpha \neq \mathbf{m} \). The more copy numbers differ from 2, the higher is the I/NI call, where gains and losses are treated on the same level by the absolute value of the logarithm.

Using Eq. (S25), the I/NI call can be decomposed into contributions from each sample \( k \):

\[
\text{I/NI}(\alpha) = \sum_{i=0}^{n} \alpha_i | \log(i/2) | = \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \alpha_{ik} | \log(i/2) | = \frac{1}{N} \sum_{k=1}^{N} \text{I/NI}(\alpha_k),
\] (S42)

with \( \alpha_k = (\alpha_{1k}, \ldots, \alpha_{ik}, \ldots, \alpha_{nk}) \). The individual I/NI call of sample \( k \) is \( \text{I/NI}(\alpha_k) = \sum_{i=0}^{n} \alpha_{ik} | \log(i/2) | \) which is the contribution of sample \( k \) to the I/NI call and the expected copy number fold change of sample \( k \).

### S2.4 Segmentation and CNV Call

#### S2.4.1 Segmentation

CNVs are detected by segmenting the chromosomes of individuals based on their individual I/NI calls, where genomic adjacent I/NI calls that show the same copy numbers are joined. Note, however, that the individual I/NI call defined Eq. (S42) does not allow for distinguishing losses and gains with the same fold change.
The signed individual I/NI call. To avoid joining losses and gains, we define the signed individual I/NI call as the expected log fold change:

\[ s_{I/NI}(\alpha_k) = \sum_{i=0}^{n} \alpha_{ik} \log(i/2) \]  

\[ \approx \text{sgn} \left( \sum_{i=0}^{n} \alpha_{ik} \log(i/2) \right) \sum_{i=0}^{n} \alpha_{ik} |\log(i/2)| \]

\[ = \text{sgn} \left( \sum_{i=0}^{n} \alpha_{ik} \log(i/2) \right) I/NI(\alpha_k) \]

The absolute value of the signed I/NI call \(|s_{I/NI}(\alpha_k)|\) is not exactly the individual I/NI call \(I/NI(\alpha_k)\), but the two values are always very close. They are close because for one sample the summands with largest \(\alpha_{ik}\) are either \(\geq 0\) or \(\leq 0\), that is the model assumes for one sample at one location either a loss \((i/2 \leq 1)\) or a gain \((i/2 \geq 1)\) if deviating from the prior of constant copy number 2.

Numerical investigation of the difference between sI/NI and expected log fold change. We investigated the numerical difference between the signed I/NI call and the expected log fold change, that is the quality of the approximation in Eq. (S43). Based on data from the Sanger sequencing center on HapMap phase 1 individuals, we calculated the difference of these values for more than 2 million data points and found that the median difference was zero, the third quartile was \(6.2e-17\), the maximum was \(1.4e-02\). Note, that for copy number 3 sI/NI values are in the range of \(0.6 \approx \log(3/2)\) and for copy number 1 in the range of \(-1 = \log_2(1/2)\), therefore, the difference between the signed I/NI call and the expected log fold change is negligible.

Segmentation algorithm. The circular binary segmentation algorithm (DNAcopy; Venkatraman and Olshen 2007) is applied to \(s_{I/NI}(\alpha_k)\) along the chromosome. DNAcopy joins consecutive segments with large or small expected fold changes to a candidate segment. Note, that other segmentation algorithms led to similar results as DNAcopy on the experimental data. The segments obtained by the segmentation algorithm are candidate segments as they show a variation along the chromosome indicated by the signed individual I/NI call.

S2.4.2 CNV Call of cn.MOPS

A candidate segment is called a CNV if the median of the signed individual I/NI call \(s_{I/NI}(\alpha_k)\) over the segment is at least 0.6 for gains and at most \(-1\) for losses. Thus, also a variation across samples is needed to call a CNV. The CNV call combines two calls: (1) an I/NI call across samples and (2) a segment call along the chromosome. Only if consecutive segments obtain an I/NI call, they are joined by the segmentation algorithm (see second bar and third sample in Fig. S1). This idea of calling a CNV by two calls, where one call is supplied by a model across samples, has already led to improvements of CNV detection based on DNA microarray data via the cn.FARMS method (Clevert et al. 2011).
Figure S1: Illustration of the basic concept of cn.MOPS: a CNV call incorporates the detection of variation across samples (I/NI call) and the detection of variation along a chromosome (segmentation). Curves depict read counts along one chromosome for five samples. I/NI calls (green) detect variation across samples (green vertical boxes). A CNV (red box) is called, if consecutive segments have high I/NI calls. Blue boxes mark segments that a segmentation algorithm of class (a) would combine into a CNV. First vertical bar (from the left) and first sample: the I/NI call indicates variation across samples (“I/NI call +”). However, too few adjacent segments show high I/NI calls. Second bar and third sample: The I/NI call indicates variation across samples (“I/NI call +”) and sufficient adjacent segments show high I/NI calls, which leads to a CNV call (red box). Third bar: the read counts drop consistently, thus would be detected by a segmentation algorithm of class (a) methods (blue boxes). However, the samples’ read counts do not vary, which does not lead to an I/NI call (“I/NI call -”). A CNV is not detected, which is correct as the copy number does not vary across samples. Fourth bar and samples no. two and four: I/NI call indicates variation across samples (“I/NI call +”). As in the first bar, too few adjacent segments show high I/NI calls. Fifth bar and second sample: a segmentation algorithm of class (a) methods would combine adjacent read counts that are consistently small (blue box) into a CNV. However, the read counts are within the variation of the constant copy number at this location. Therefore the I/NI call does not indicate variation across samples (“I/NI call -”).
S2.5 Noise Model Variants

In the following, we introduce some variants of cn.MOPS with different noise assumptions. These variants have in common that not only copy number 0, but also other copy number regions may have additional reads stemming from wrong mappings or sample contamination.

The main problem of these noise assumptions is that only an increase of read counts by noise is modeled, but no decrease. For copy number 0, this is correct, but it is hard to justify for other copy number regions. To allow negative $\epsilon$, that is, a loss of reads, leads to numerical problems at copy number 0 regions. This is the main reason why we included a noise term only for copy number 0 in cn.MOPS.

In the following two subsections, we consider two variants of cn.MOPS with alternative noise models. In a third subsection, we investigate another variant of cn.MOPS where the noise level $\epsilon$ is considered as a parameter which is optimized by the EM algorithm. As it will turn out later, the main problem of this approach is that the increased model complexity potentially leads to overfitting.

S2.5.1 Variant (a): Additive Poisson Noise for Each Segment

First, we introduce a variant where, in each segment, additional reads are modeled via additive Poisson-distributed reads with parameter $\epsilon$.

The objective Eq. (S26) for optimizing the average read count $\lambda$ now becomes:

$$\min_{\lambda} - \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \hat{\alpha}_{ik} \log P(x_k; \frac{i}{2} \lambda + \frac{\epsilon}{2}).$$

(S44)

Using

$$\log P(x_k; \frac{i}{2} \lambda + \frac{\epsilon}{2}) = -\log(x_k!) - \frac{i}{2} \lambda - \frac{\epsilon}{2} + x_k \log\left(\frac{i}{2} \lambda + \frac{\epsilon}{2}\right),$$

(S45)

the derivative of the objective is

$$- \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \left( - \frac{i}{2} + x_k \frac{i}{2} \lambda + \frac{\epsilon}{2} \right) \hat{\alpha}_{ik}.$$

(S46)

Setting this derivative to zero and solving the resulting equation with respect to $\lambda$ leads to the following alternative update rule:

$$\lambda_{\text{new}} = \frac{1}{N} \sum_{k=1}^{N} x_k - \frac{\epsilon}{2} \frac{1}{\sum_{i=0}^{n} \hat{\alpha}_{i} \frac{i}{2}}.$$

(S47)

We will refer to this approach as Variant (a) in the following. Recall that the $\lambda$ update rule of cn.MOPS is

$$\lambda_{\text{new}} = \frac{1}{N} \sum_{k=1}^{N} x_k = \frac{1}{N} \sum_{i=0}^{n} \hat{\alpha}_{i} \frac{i}{2}.$$

(S48)
S2.5.2 Variant (b): Additive Poisson Noise Scales with Average Read Count

Secondly, we introduce a variant where, in each segment, the additional noise reads scale with the average number of reads in this segment.

The objective Eq. (S26) for optimizing the average read count $\lambda$ now becomes:

$$
\min_{\lambda} - \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \hat{\alpha}_{ik} \log P(x; \frac{i + \epsilon}{2} \lambda).
$$

Using

$$
\log P(x_k; \frac{i + \epsilon}{2} \lambda) = - \log(x_k!) - \frac{i + \epsilon}{2} \lambda + x_k \left( \log(\lambda) + \log\left(\frac{i + \epsilon}{2}\right) \right),
$$

the derivative of the objective is

$$
- \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \left( - \frac{i + \epsilon}{2} + x_k \frac{1}{\lambda} \right) \hat{\alpha}_{ik}.
$$

We will refer to this approach as Variant (b) in the following. Setting this derivative to zero and solving the resulting equation with respect to $\lambda$ leads to the following alternative update rule:

$$
\lambda_{\text{new}} = \frac{1}{N} \sum_{k=1}^{N} x_k \frac{1}{\sum_{i=1}^{n} \hat{\alpha}_{i i} + \frac{\epsilon}{2}}.
$$

S2.5.3 Variant (c): Poisson Noise Level as Model Parameter

As a third variant, we consider the noise level $\epsilon$ as a model parameter which is adjusted by the EM algorithm.

The objective for optimizing $\epsilon$ can be derived analogously to Eq. (S26), where the terms containing $\epsilon$ are:

$$
\min_{\epsilon} - \frac{1}{N} \sum_{k=1}^{N} \hat{\alpha}_{0k} \log P(x; \frac{\epsilon}{2} \lambda).
$$

Using

$$
\log P(x_k; \frac{\epsilon}{2} \lambda) = - \log(x_k!) - \frac{\epsilon}{2} \lambda + x_k \left( \log(\lambda) + \log\left(\frac{\epsilon}{2}\right) \right),
$$

the derivative of the objective is

$$
- \frac{1}{N} \sum_{k=1}^{N} \hat{\alpha}_{0k} \left( - \frac{1}{2} \lambda + \frac{x_k}{\epsilon} \right)
$$

We will refer to this approach as Variant (c) in the following. Setting this derivative to zero and solving the resulting equation with respect to $\epsilon$ leads to the following update rule for $\epsilon$:

$$
\epsilon_{\text{new}} = 2 \frac{1}{N} \sum_{k=1}^{N} \hat{\alpha}_{0k} x_k \frac{1}{\sum_{k=1}^{N} \hat{\alpha}_{0k}}.
$$
The problem with variant (c) is that the model complexity increases by introducing a second parameter; therefore, it is prone to overfitting.

In experiments (see below), we observed that $\epsilon$ codes for the average read counts of copy number 2 for data with large copy numbers, while the copy number 2 component is used for large copy numbers. In this case, the model can model the large fold changes in the data, although the assignment of integer copy numbers can be incorrect.

These two drawbacks of adjusting the noise via a model parameter are the main reasons why we decided to consider $\epsilon$ as a hyperparameter in the cn.MOPS model.

### S2.5.4 cn.MOPS Variants Tested on Simulated Data

First, we compared cn.MOPS with variants (a)–(c) on the simulated data used in the experiments described in Section “Simulated Data with Constructed CNVs” of the main manuscript. Table S2 shows the results. The low performance of variant (c) is caused by overfitting via an overly complex model class. Variant (b) performs best for gains, while cn.MOPS performs best for losses. The differences, however, are only marginal.

<table>
<thead>
<tr>
<th>Variant</th>
<th>Gains</th>
<th></th>
<th>Losses</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PR AUC</td>
<td>Recall</td>
<td>PR AUC</td>
<td>Recall</td>
</tr>
<tr>
<td>cn.MOPS</td>
<td>0.935</td>
<td>0.883</td>
<td>0.963</td>
<td>0.961</td>
</tr>
<tr>
<td>(a) additive $\epsilon$</td>
<td>0.931</td>
<td>0.860</td>
<td>0.961</td>
<td>0.959</td>
</tr>
<tr>
<td>(b) additive scaled $\epsilon$</td>
<td>0.941</td>
<td>0.889</td>
<td>0.962</td>
<td>0.959</td>
</tr>
<tr>
<td>(c) $\epsilon$ as parameter</td>
<td>0.904</td>
<td>0.821</td>
<td>0.938</td>
<td>0.933</td>
</tr>
</tbody>
</table>

### S2.5.5 cn.MOPS Variants Tested on Real Sequencing Data with Implanted CNVs from the X Chromosome

Next, we compared cn.MOPS with variants (a)–(c) on the data used in the experiments described in Section “Real Sequencing Data with Implanted CNVs from the X Chromosome” of the main manuscript. Table S3 shows the results. The low performance of variant (c) is again caused by overfitting as in the previous experiment. Variant (b) performs worse in this experiment, while cn.MOPS performs generally best.
Table S3: Performance of cn.MOPS and variants on real sequencing data with implanted CNVs from the X chromosome. “PR AUC” gives the area under the precision-recall curve. “Recall” reports the recall at a precision of 0.95. For gains, Variant (b) performs best in terms of PR AUC, but only marginally. For losses and in terms of recall for gains, cn.MOPS performs best.

<table>
<thead>
<tr>
<th>Variant</th>
<th>Gains PR AUC</th>
<th>Gains Recall</th>
<th>Losses PR AUC</th>
<th>Losses Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn.MOPS</td>
<td>0.703</td>
<td>0.652</td>
<td>0.888</td>
<td>0.878</td>
</tr>
<tr>
<td>(a) additive $\epsilon$</td>
<td>0.705</td>
<td>0.640</td>
<td>0.886</td>
<td>0.876</td>
</tr>
<tr>
<td>(b) additive scaled $\epsilon$</td>
<td>0.712</td>
<td>0.544</td>
<td>0.844</td>
<td>0.831</td>
</tr>
<tr>
<td>(c) $\epsilon$ as parameter</td>
<td>0.680</td>
<td>0.590</td>
<td>0.864</td>
<td>0.853</td>
</tr>
</tbody>
</table>

S2.5.6 Conclusion

In summary, other variants of cn.MOPS show inferior performance. For variants (a) and (b) with additive noise for all copy numbers, only additional reads are explained, but potentially missing reads are not modeled. The value of $\lambda$ is therefore systematically underestimated, which leads to a decrease in performance. Variant (c) considers $\epsilon$ as a model parameter and thus uses a model of higher complexity, which leads to overfitting and, consequently, to a decrease in performance. We conclude that cn.MOPS is the best choice compared to the variants considered here.
S3.1 Evaluation of CNV Detection Results

In order to compare methods which detect copy number variations in next generation sequencing data, we need an evaluation criterion. We assume that the true CNVs are known and to be rediscovered. Each chromosome is split into equally large evaluation segments the size of which is chosen to accommodate the shortest known CNV. An evaluation segment is called a true positive (TP) if it is entirely contained both in a true CNV and in a detected CNV segment. It is called a false negative (FN) if it is entirely contained in a true CNV but does not overlap with any predicted CNV segment. An evaluation segment is called a false positive (FP) if it is entirely detected as a CNV segment but does not overlap with any true CNV. Finally, it is called a true negative (TN) if it overlaps neither with a true CNV nor with a detected CNV segment. These definitions imply that all evaluation segments that partly overlap with true CNVs or detected CNV segments remain ignored, as the copy numbers in these segments are ambiguous. Figure S2 illustrates the definitions of the four categories of evaluation segments. The two measures we employ hereafter are recall ($\frac{\#TP}{\#TP + \#FN}$) and precision ($\frac{\#TP}{\#TP + \#FP}$). Note that precision is one minus the false discovery rate, in which we are especially interested. A CNV calling threshold governs the trade-off between recall and precision or, in other words, the trade-off between FNs and FPs, because more detected CNVs lead to more FPs but fewer FNs, and vice versa. To assess the performance of methods at different CNV calling thresholds, we use precision-recall curves. Precision-recall curves are independent of the number of TNs, which makes them an ideal tool for our evaluation, as the majority of samples are negatives (non-CNVs).

We also considered using “receiver-operator characteristic (ROC) curves” or the Matthews correlation coefficient as evaluating criterion for the performance of different methods. However, we decided to use the area under the precision recall curve as evaluation criterion because it is independent of the number of true negatives. Our data sets contain many negatives (segments with constant copy number 2) and most methods classify the majority of them correctly which leads to a large number of true negatives. By using precision-recall curves we avoid that methods which tend to classify most segments as negatives, that are methods with a low discovery power (low
S3.2 Compared CNV Detection Methods

We compared following methods:

1. cn.MOPS our new model and pipeline,
2. MOFDOC according to the variant described in (Alkan et al. 2009),
3. EWT (Yoon et al. 2009) event-wise testing,
4. JointSLM (Magi et al. 2011),
5. CNV-Seq (Xie and Tammi 2009),
6. FREEC (Boeva et al. 2011).

For a fair comparison, the parameters of the methods were optimized on simulated data sets similar to the one we used in our first experiment in Section S3.3.

S3.2.1 cn.MOPS

For cn.MOPS model we initialized the parameter $\lambda$ by the median read count of the segment across samples $\lambda_{\text{median}}$ from Eq. (S1) is set to $\epsilon = 0.05$, which is our estimate for the percentage of wrongly mapped reads. We set $n = 8$ which leads to nine possible copy numbers $0 \leq i \leq 8$. The parameter $\alpha$ should be initialized close to the location of the prior’s mode $(0, 0, 1, 0, \ldots, 0)$, which are the optimal parameters if all samples have copy number 2. However, initializing $\alpha$ by $(0, 0, 1, 0, \ldots, 0)$ would clamp all $\hat{\alpha}_{ik}$ and $\alpha_{ik}^{\text{new}}$ to zero according to Eq. (S31) and Eq. (S32). Therefore we initialized $\alpha$ by $\alpha = (0.05, 0.05, 0.6, 0.05, \ldots, 0.05)$.

S3.2.2 Class (a) Methods: MOFDOC, EWT, and JointSLM

The methods MOFDOC (“model free depth of coverage”) according to Alkan et al. (2009), EWT (“event-wise testing”) according to Yoon et al. (2009), and JointSLM (Magi et al. 2011) are all based on detecting deviations of read counts from an average read count which can be measured by $z$-scores or log $z$-scores, i.e. the multiple in standard deviations the read count differs from the mean.

MOFDOC: We implemented MOFDOC using the CNV calling criterion from Alkan et al. (2009). A CNV region is called if $a$ out of $b$ consecutive segments show a read count with a $z$-score beyond a threshold (abnormal large or small read counts) to call a segment (default $a = 6$ and $b = 7$). We generalized this “$a$-$b$-smoother” to a smoothing algorithm, that is not only able to smooth logical, but also real values stemming from CNV calls (see Section S3.5 and Fig. S5), which improved MOFDOC’s results.

All parameters are optimized on artificial test data sets similar to one used in Section S3.3, which resulted in following parameter settings:

- $a=4$
b=4
GCcorrection=TRUE

The parameter WL was set to 25000/2500/500 for the low coverage, medium coverage and high coverage data set, respectively.

**EWT**: We reimplemented EWT as described in Yoon et al. (2009) but improved the GC correction by using all samples for estimating the GC effect. Further we restricted the parameter “event size” to an upper bound maximumEventSize and a lower bound minimumEventSize. “event size” is a parameter that prevents EWT from testing too short CNVs. We generalized EWT to variable segment sizes like 10kbp, 25kbp and 50kbp that are apt for low coverage CNV detection. Our modifications of EWT improved its results. EWT adjusts a threshold on the p-values of joined segments (called “false positive rate”) assuming independent Gaussian read counts per segment. This threshold governs the number of detections. To compute the precision-recall curve we considered all possible thresholds of the log “false positive rate”. All parameters are optimized on artificial test data sets similar to one used in Section S3.3, which resulted in following parameter settings:

- minimumEventSize = 4
- maximumEventSize = 8
- GCcorrection = TRUE

The parameter WL was set to 25000/2500/500 for the low, medium and high coverage data set, respectively.

**JointSLM**: We applied the R -package (version 0.1) to 25kbp/2500bp/500bp (low/medium/high coverage) segments, for which the GC content was computed. In contrast to the original implementation, we did not round the scores per segment to integer copy numbers to allow for thresholding and to compute the area under precision-recall curve. All parameters are optimized on artificial test data sets similar to one used in Section S3.3, which resulted in following parameter settings:

- omega = 0.1
- eta = 1e-06
- K0 = 20
- baseCopy = 2

**S3.2.3 Class (b) Methods: SeqSeg, CNV-Seq, and FREEC**

The methods SeqSeg (Chiang et al. 2008), CNV-Seq (Xie and Tammi 2009), and FREEC (Boeva et al. 2011) require a reference genome for copy number detection. Most of these methods are designed for and applied to studies with tumor samples and matched normal samples, which are often blood cells of the same individual. For CNV detection, matched normals are in general...
not available. Therefore, reference read counts per segment are build as the median of read counts over all samples. Additionally, the median is more robust than using a matched sample, because both the reference and the analyzed genome are subject to random read count variations (note, the read count variation is not estimated across samples). For the method SeqSeg, which requires read positions on the genome, we generated a reference genome in the following way: we pooled the read positions of all samples and then sorted them according to their genomic position. We then used the median of \( n \) (number of samples) consecutive reads as read position for the reference genome.

**SeqSeg:** We did not include SeqSeg (Chiang *et al.* 2008) in the comparisons because we were not able to find suitable parameters, not even after an extensive search. The problem was that SeqSeg either did not detect any breakpoints or the thresholds for the \( p \)-values were not determined. However, the performance of SeqSeg can be estimated via CNV-Seq which is very similar to SeqSeg.

**CNASeg:** We also omitted CNAseg (Ivakhno *et al.* 2010) from the comparison, as its developers state that this method is specifically tailored to CNA detection in tumor samples.

**CNV-Seq:** We used the authors’ implementation\(^1\), where the median of the samples’ read counts served as reference read count. All parameters are optimized on artificial test data sets similar to one used in Section S3.3, which resulted in following parameter settings:

- \( p \text{Value} = 0.001 \)
- \( \log_{2} \text{Threshold} = 0.6 \)

All other parameters were set to their default values. The parameter \( \text{windowLength} \) was set to 25000/2500/500 for the low, medium and high coverage data set, respectively.

**FRECC:** We used Version 3.2\(^2\), where analogously to CNV-Seq, the median of the samples’ read counts was used as reference. For computing the precision-recall curves, all possible thresholds for the returned median ratio per segment are used. All parameters are optimized on artificial test data sets similar to one used in Section S3.3, which resulted in following parameter settings:

- \( \text{breakPointThreshold} = -0.001 \)
- \( \text{ploidy} = 2 \)
- \( \text{minCNAlength} = 4 \)
- \( \text{step} = 10000 \)
- \( \text{mode} = \text{reference} \)

All other parameters were set to their default values. The parameter \( \text{window} \) was set to 25000/2500/500 for the low, medium and high coverage data set, respectively.

\(^1\)http://tiger.dbs.nus.edu.sg/cnv-seq/ (version as of 2010/07/16)
\(^2\)http://bioinfo-out.curie.fr/projects/frecc/ (version as of 2011/04/04)
S3.3 Simulated Sequencing Data with Constructed CNVs

We constructed 100 artificial benchmark data sets. We assume an artificial genome to consist of a single chromosome of 125Mbps length which is divided into 5,000 segments of length 25kbp. We created 40 samples by sampling read counts for all segments and all samples according to a Poisson process. The overall number of evaluation segments was, therefore, \(40 \times 5,000 = 200,000\).

S3.3.1 Distribution of CNV Types and Copy Numbers for Data Generation

We determined characteristics of CNV regions and how copy numbers are distributed from the HapMap individuals (The International HapMap 3 Consortium 2010). CNVs of different individuals cluster at certain regions of the DNA, the “CNV regions”, of which many contain only losses or only gains. CNV regions can be divided into 3 types: CNV regions of the type “loss region” contain only losses, of type “gain region” contain only gains, and of type “mixed region” contain both losses and gains. As depicted in Fig. S3, the CNV region type “loss region” was observed in 80%, “gain region” in 15%, and “mixed region” in 5%.

![Histogram of CNV region type](image_url)

Figure S3: Histogram of CNV region types “loss region”, “gain region”, and “mixed region” according to The International HapMap 3 Consortium (2010).

We implanted 20 CNV regions into each of these benchmark chromosomes. The CNV regions’ lengths were chosen randomly from the interval 75–200kbp, which is the range of accurate detection for the given coverage according to Xie and Tammi (2009). The 20 starting points of the CNV regions are randomly chosen along the chromosome. After having determined the 20 CNV regions, we have to decide how CNVs are implanted into the single samples. According to the HapMap individuals, we assign CNV region types such that 80% are “loss region” (contain only losses), 15% “gain region” (contain only gains), and 5% “mixed region” (contain both losses and gains). Then the actual copy number for each sample is drawn according to the copy numbers observed for HapMap individuals (The International HapMap 3 Consortium 2010): For a loss region, a sample has probabilities of 0.8, 0.15, and 0.05 of having copy numbers 2, 1, and 0, respectively. For a gain region, a sample has probabilities of 0.85, 0.08, 0.06, and 0.01 of having copy numbers 2, 3, 4 and 5, respectively. For a mixed region, a sample has probabilities 0.04, 0.16, 0.67, 0.11, and 0.02 of having copy numbers 0, 1, 2, 3 and 4, respectively. Of the 200,000 evaluation segments, on average 101(±56) are gains and 612 (±104) are losses. The CNVs’ lengths range from 75,006bp to 199,848bp with an average of 136,921bp.
Figure S4: Histograms of integer copy numbers for CNV region types “loss region”, “gain region”, and “mixed region” according to The International HapMap 3 Consortium (2010).
S3.3.2 Results

Table S4 reports the performance of the compared copy number detection methods separately for gains and losses. As evaluation measures, we use the area under the precision-recall curve and the recall for an FDR fixed to 0.05.

Table S4: Performance of the compared copy number detection methods on the artificial benchmark data set. “PR AUC” gives the average area under the precision-recall curve of 100 experiments. The second column “p-value” reports the p-value of a Wilcoxon signed-rank test (over the 100 experiments) with null hypothesis that cn.MOPS and another method have the same area under the curve. “Recall” reports the recall at a precision of 0.95, that is, an FDR of 0.05. The last column “p-value” gives the p-value of an analogous Wilcoxon test for the recall with an FDR of 0.05. cn.MOPS had significantly higher performance than all other methods.

<table>
<thead>
<tr>
<th></th>
<th>Gains</th>
<th>PR AUC</th>
<th>p-value</th>
<th>Recall</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn.MOPS</td>
<td>0.94</td>
<td>—</td>
<td>0.88</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>MOFDOC</td>
<td>0.81</td>
<td>1.14e-13</td>
<td>0.76</td>
<td>9.75e-12</td>
<td></td>
</tr>
<tr>
<td>EWT</td>
<td>0.79</td>
<td>5.95e-14</td>
<td>0.74</td>
<td>1.34e-12</td>
<td></td>
</tr>
<tr>
<td>JointSLM</td>
<td>0.25</td>
<td>4.23e-18</td>
<td>0.22</td>
<td>2.80e-17</td>
<td></td>
</tr>
<tr>
<td>CNV-Seq</td>
<td>0.35</td>
<td>4.23e-18</td>
<td>0.35</td>
<td>3.98e-17</td>
<td></td>
</tr>
<tr>
<td>FREEC</td>
<td>0.65</td>
<td>1.95e-17</td>
<td>0.53</td>
<td>3.42e-14</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Losses</th>
<th>PR AUC</th>
<th>p-value</th>
<th>Recall</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn.MOPS</td>
<td>0.96</td>
<td>—</td>
<td>0.96</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>MOFDOC</td>
<td>0.92</td>
<td>3.50e-17</td>
<td>0.90</td>
<td>9.22e-17</td>
<td></td>
</tr>
<tr>
<td>EWT</td>
<td>0.91</td>
<td>3.20e-18</td>
<td>0.90</td>
<td>8.44e-17</td>
<td></td>
</tr>
<tr>
<td>JointSLM</td>
<td>0.34</td>
<td>1.98e-18</td>
<td>0.28</td>
<td>1.98e-18</td>
<td></td>
</tr>
<tr>
<td>CNV-Seq</td>
<td>0.81</td>
<td>1.98e-18</td>
<td>0.81</td>
<td>3.84e-17</td>
<td></td>
</tr>
<tr>
<td>FREEC</td>
<td>0.73</td>
<td>1.98e-18</td>
<td>0.72</td>
<td>3.32e-17</td>
<td></td>
</tr>
</tbody>
</table>

S3.3.3 Different Performance on Gains and Losses

Table S4 shows that all methods perform better at detecting losses. The superior performance at losses can be explained by the fact that copy number 3 can be more likely be confused with copy number 2 than copy number 1. We show that under a Poisson assumption, a typical read for copy number 2, that is $\lambda$, is more likely to come from copy number 3 than from copy number 1.
Assuming $\lambda > 0$, we obtain:

$$0 < \log(3) - 1$$  \hspace{1cm} (S57)

$$\iff -\frac{1}{2} < \log(3) - \frac{3}{2}$$

$$\iff \log \left( \frac{1}{2} \right) - \frac{1}{2} < \log \left( \frac{3}{2} \right) - \frac{3}{2}$$

$$\iff \lambda \log \left( \frac{1}{2} \right) + \lambda \log(\lambda) - \frac{1}{2} \lambda < \lambda \log \left( \frac{3}{2} \right) + \lambda \log(\lambda) - \frac{3}{2} \lambda$$

$$\iff \left( \frac{1}{2} \lambda \right)^\lambda e^{-\frac{1}{2}\lambda} < \left( \frac{3}{2} \lambda \right)^\lambda e^{-\frac{3}{2}\lambda}$$

$$\iff P \left( \lambda ; \frac{1}{2} \lambda \right) < P \left( \lambda ; \frac{3}{2} \lambda \right)$$,

where $P(x; \beta)$ is the Poisson distribution with parameter $\beta$ evaluated at $x$. The inequality shows that the average read count $\lambda$ for copy number 2 has higher probability to be drawn from a copy number 3 than from a copy number 1 distribution.

### S3.4 Real Sequencing Data with Implanted CNVs From the X Chromosome

In contrast to the previous benchmark for which the read counts were simulated, we now consider real reads stemming from sequencing of a single male HapMap individual (NA20755). This man’s genome was sequenced 17 times by the Solexa Genome Analyzer II at the Sanger Sequencing Center (see Table S5). These 17 samples ensure a constant copy number, as they stem from the same individual. The reads were mapped by Bowtie (Langmead et al. 2009) for paired reads. We allowed for two mismatches. The numbers of reads range from 12,069,758 to 18,810,212 of which between 10,419,510 and 16,041,464 could be mapped, which corresponds to coverages between 0.13 and 0.21.

We created 110 benchmark data sets by choosing each human chromosome 1–22 five times, where in each chromosome data set 20 random CNV regions were implanted. The lengths of these implanted CNV regions were chosen to be 75kbp, 100kbp, 150kbp and 200kbp (5 each), and for each of the regions a random segment on the X chromosome was selected which supplied reads for the region. CNV region types and individual copy numbers were determined according to the same procedure and distributions as described in Subsection S3.3 except that we only consider CNV copy numbers 1 and 3, since they are most difficult to distinguish from copy number 2. We chose 80% of the CNV regions as loss regions, 15% as gain regions, and 5% as mixed regions. For a loss region, a sample has probabilities 0.8 and 0.2 of having copy number 2 and 1, respectively, for a gain region, 0.85 and 0.15 of having copy numbers 2 and 3, respectively, and for a mixed region, 0.2, 0.67, and 0.13 of having copy numbers 1, 2, and 3, respectively. Finally, the read counts of the 17 samples are computed in the following way: outside CNVs, for constant copy numbers, the original reads counts are used; within CNVs we added as many read counts as there are copies from the corresponding segment on the X chromosome, where read counts are obtained from the considered sample and other random samples.

The CNV detection results were evaluated as described in Subsection S3.1. The number of evaluation segments ranges from around 32,000 for chromosome 21 to around 168,000 for chromosome 1. On average, 0.1% of the evaluation segments are gains and 0.4% are losses.
S3.4.1 Data and Mapping

The sequencing reads of the male sample NA20755 were obtained from the 1000 genomes project (The 1000 Genomes Project Consortium 2010) web page (http://www.1000genomes.org). Table S5 lists the unique names of the sequencing read files. We applied the Bowtie software (Langmead et al. 2009) to map the reads against the human reference genome 18 (build 36). The Bowtie parameters were set as follows:

- `-q` $\Rightarrow$ Input files are fastq files.
- `-v 2` $\Rightarrow$ Two mismatches are allowed.
- `-M 1` $\Rightarrow$ M-alignment mode. Reports at most one valid alignment. If more than one best mapping position is available then the read is randomly assigned to one of them.
- `--best` $\Rightarrow$ The alignment is the best matching position.
- `--sam` $\Rightarrow$ Output format is SAM.

Table S5 further lists the number of sequenced reads, the number of mapped reads, the number of used reads, and the ratio of mapped reads.

Mapping reads to only unique positions by Bowtie parameter `-m 1` leads to many segments with low read counts as the histogram in Fig. S14 in Subsection S4.1 shows. Compared to the histogram in Fig. S13 in Subsection S4.1 for non-uniquely mapped reads, we observe a shift of the density toward lower read counts (left) because some segments systematically lose reads due to ambiguous mappings. Because there are too many low read counts, class (a) methods like MOFOG, EWT, and JointSLM are not suited for this kind of read mapping. However, methods based on ratios perform well on data from unique position mapping, since they use reference read counts which have similar read counts as the samples for the same copy number. cn.MOPS is also suited to handle data from unique position mapping as it builds a local model, which regards read count characteristics.

S3.4.2 Results

Table S6 reports the performance of the compared copy number detection methods separately for gains and losses. As before, we use area under the precision-recall curve and the recall for an FDR fixed to 0.05.

S3.4.3 Estimation of Integer Copy Numbers

For this experiment we were able to evaluated the model’s performance of assigning an integer copy number to each sample in each genomic location. Evaluation segments that contained a CNV breakpoint were excluded from the further analysis since they have no unique copy number. Inside CNVs 92.293% ($\pm$ 0.026%) and on the whole genome, including constant copy number two, 99.383% ($\pm$ 0.001%) of the integer copy numbers were correctly assigned.
Table S5: Summary of the sequencing data for the implanted CNVs benchmark data set. Column “individual” reports the ID number of the HapMap sample. Column “base name” shows the base name of the files containing the sequence reads. Mate 1 has the filename “base name_1.filt.fastq.gz” and mate 2 “base name_2.filt.fastq.gz”. The following columns “sequenced reads”, “mapped reads” and “used reads” report the number of totally sequenced reads, the number of reads that were mapped to the reference genome, and the number of reads that were mapped to the reference genome, and the number of reads used for the analysis (after removing potential PCR duplicates). The last column “ratio mapped” gives the proportion of mapped reads to sequenced reads.

<table>
<thead>
<tr>
<th>individual</th>
<th>base name</th>
<th>sequenced reads</th>
<th>mapped reads</th>
<th>used reads</th>
<th>ratio mapped</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 NA20755</td>
<td>ERR003683</td>
<td>17,303,620</td>
<td>14,890,416</td>
<td>14,846,540</td>
<td>0.86</td>
</tr>
<tr>
<td>2 NA20755</td>
<td>ERR003775</td>
<td>16,453,240</td>
<td>14,017,334</td>
<td>13,974,864</td>
<td>0.85</td>
</tr>
<tr>
<td>3 NA20755</td>
<td>ERR003776</td>
<td>17,471,474</td>
<td>15,051,916</td>
<td>14,998,386</td>
<td>0.86</td>
</tr>
<tr>
<td>4 NA20755</td>
<td>ERR003777</td>
<td>18,275,368</td>
<td>15,763,846</td>
<td>15,702,220</td>
<td>0.86</td>
</tr>
<tr>
<td>5 NA20755</td>
<td>ERR003778</td>
<td>18,651,270</td>
<td>15,892,764</td>
<td>15,831,810</td>
<td>0.85</td>
</tr>
<tr>
<td>6 NA20755</td>
<td>ERR003779</td>
<td>18,566,172</td>
<td>15,881,196</td>
<td>15,816,916</td>
<td>0.86</td>
</tr>
<tr>
<td>7 NA20755</td>
<td>ERR003780</td>
<td>18,611,728</td>
<td>15,840,592</td>
<td>15,774,308</td>
<td>0.85</td>
</tr>
<tr>
<td>8 NA20755</td>
<td>ERR003781</td>
<td>17,713,468</td>
<td>14,974,160</td>
<td>14,918,680</td>
<td>0.85</td>
</tr>
<tr>
<td>9 NA20755</td>
<td>ERR003782</td>
<td>17,649,380</td>
<td>15,039,086</td>
<td>14,995,282</td>
<td>0.85</td>
</tr>
<tr>
<td>10 NA20755</td>
<td>ERR003783</td>
<td>18,810,212</td>
<td>16,041,464</td>
<td>15,991,994</td>
<td>0.85</td>
</tr>
<tr>
<td>11 NA20755</td>
<td>ERR003784</td>
<td>18,650,906</td>
<td>16,020,806</td>
<td>15,971,248</td>
<td>0.86</td>
</tr>
<tr>
<td>12 NA20755</td>
<td>ERR003785</td>
<td>16,960,214</td>
<td>14,052,008</td>
<td>14,012,730</td>
<td>0.83</td>
</tr>
<tr>
<td>13 NA20755</td>
<td>ERR003786</td>
<td>14,799,104</td>
<td>11,741,258</td>
<td>11,711,502</td>
<td>0.79</td>
</tr>
<tr>
<td>14 NA20755</td>
<td>ERR003787</td>
<td>12,888,982</td>
<td>10,419,510</td>
<td>10,395,492</td>
<td>0.81</td>
</tr>
<tr>
<td>15 NA20755</td>
<td>ERR003855</td>
<td>15,003,086</td>
<td>13,131,902</td>
<td>13,067,934</td>
<td>0.88</td>
</tr>
<tr>
<td>16 NA20755</td>
<td>ERR003867</td>
<td>17,359,258</td>
<td>15,090,776</td>
<td>15,044,648</td>
<td>0.87</td>
</tr>
<tr>
<td>17 NA20755</td>
<td>ERR003878</td>
<td>12,069,758</td>
<td>10,652,836</td>
<td>10,603,636</td>
<td>0.88</td>
</tr>
</tbody>
</table>

S3.5 Rediscovering of Known CNVs in HapMap Sequencing Data

Finally, we compare how well the methods are able to rediscover known CNVs of HapMap individuals whose DNA was sequenced by the Solexa Genome Analyzer II at the Sanger Sequencing Center. We focused on 18 individuals for each of which the reads were produced on one lane (one sequencing run contains 7 lanes). The reads were mapped by Bowtie (Langmead et al. 2009) for paired reads, again allowing three mismatches. The numbers of reads range from 12,442,124 to 31,977,690 of which 7,498,420 to 22,217,020 could be mapped, leading to a coverage between 0.20 and 0.60 (see Supplement for details on read mapping and the number of reads).

The CNVs of these 18 individuals have been determined previously using microarrays (The International HapMap 3 Consortium 2010) which we consider as the true CNVs in the following. These true CNVs were detected by the Affymetrix Human SNP array 6.0 and reconfirmed with the Illumina Human1M-single beadchip. After filtering for CNVs larger than 75kbp, we obtained 170 CNVs, of which 66 are gains and 104 are losses, with lengths ranging from 76kbp to 457kbp. The CNV detection results are evaluated as described in Subsection S3.1 with evaluation segments of length 25kbp. In total, we have 2,064,906 evaluation segments of which 450 are labeled as losses as they lie within one of the 104 loss CNVs and 469 are labeled as gains as they lie within one of...
Table S6: Performance of the compared copy number detection methods on real sequencing data with implanted CNVs from the X chromosome. “PR AUC” gives the average area under the precision-recall curve of 100 experiments. The second column “p-value” reports the p-value of a Wilcoxon signed-rank test (over the 100 experiments) with null hypothesis that cn.MOPS and another method have the same area under the curve. “Recall” reports the recall at a precision of 0.95, that is, an FDR of 0.05. The last column “p-value” gives the p-value of an analogous Wilcoxon test for the recall with an FDR of 0.05. cn.MOPS significantly outperformed all other methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>PR AUC</th>
<th>p-value</th>
<th>Recall</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gains</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cn.MOPS</td>
<td>0.70</td>
<td>—</td>
<td>0.65</td>
<td>—</td>
</tr>
<tr>
<td>MOFDOC</td>
<td>0.20</td>
<td>1.12e-17</td>
<td>0.10</td>
<td>2.31e-17</td>
</tr>
<tr>
<td>EWT</td>
<td>0.22</td>
<td>1.95e-16</td>
<td>0.13</td>
<td>8.70e-17</td>
</tr>
<tr>
<td>JointSLM</td>
<td>0.06</td>
<td>1.94e-19</td>
<td>0.03</td>
<td>7.00e-18</td>
</tr>
<tr>
<td>CNV-Seq</td>
<td>0.13</td>
<td>1.74e-19</td>
<td>0.13</td>
<td>5.75e-18</td>
</tr>
<tr>
<td>FREEC</td>
<td>0.49</td>
<td>1.22e-12</td>
<td>0.30</td>
<td>4.41e-15</td>
</tr>
<tr>
<td>Losses</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cn.MOPS</td>
<td>0.89</td>
<td>—</td>
<td>0.88</td>
<td>—</td>
</tr>
<tr>
<td>MOFDOC</td>
<td>0.57</td>
<td>3.78e-15</td>
<td>0.21</td>
<td>2.48e-18</td>
</tr>
<tr>
<td>EWT</td>
<td>0.62</td>
<td>1.77e-12</td>
<td>0.34</td>
<td>2.02e-17</td>
</tr>
<tr>
<td>JointSLM</td>
<td>0.17</td>
<td>4.43e-20</td>
<td>0.08</td>
<td>4.43e-20</td>
</tr>
<tr>
<td>CNV-Seq</td>
<td>0.50</td>
<td>4.43e-20</td>
<td>0.50</td>
<td>4.43e-20</td>
</tr>
<tr>
<td>FREEC</td>
<td>0.52</td>
<td>7.05e-17</td>
<td>0.36</td>
<td>4.56e-20</td>
</tr>
</tbody>
</table>

S3.5.1 Data and Mapping

The sequence reads of 18 different HapMap samples were obtained from the 1000 genomes project (The 1000 Genomes Project Consortium 2010) web page (http://www.1000genomes.org). Table S7 lists the unique names of the sequence read files. We used the Bowtie software (Langmead et al. 2009) to map the reads against the human reference genome 18 (build 36). The Bowtie parameters were set as follows:

- `-q` → Input files are fastq files.
- `-v 3` → Three mismatches are allowed.
- `-k 1` → M-alignment mode. Reports at most one valid alignment. If more than one best mapping position is available then the read is randomly assigned to one of them.
- `--best` → The alignment is the best matching position.
- `--sam` → Output format is SAM.
Table S7 further lists the number of sequenced reads, the number of mapped reads, the number of used reads, and the ratio of mapped reads.

Table S7: Summary of the sequencing data for the HapMap CNV reconfirmation benchmark data set. Column “individual” reports the ID number of the HapMap sample. Column “base name” displays the base name of the files containing the sequence reads. Mate 1 has the filename “base name_1.filt.fastq.gz” and mate 2 “base name_2.filt.fastq.gz”. The following columns “sequenced reads”, “mapped reads” and “used reads” report the number of totally sequenced reads, the number of reads that were mapped to the reference genome, and the number of reads used for the analysis (after removing potential PCR duplicates). The last column “ratio mapped” gives the proportion of mapped reads to sequenced reads.

<table>
<thead>
<tr>
<th>individual</th>
<th>base name</th>
<th>sequenced reads</th>
<th>mapped reads</th>
<th>used reads</th>
<th>ratio mapped</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NA11832</td>
<td>SRR023299</td>
<td>31,977,690</td>
<td>22,217,020</td>
<td>22,056,918</td>
</tr>
<tr>
<td>2</td>
<td>NA11920</td>
<td>SRR024102</td>
<td>24,558,838</td>
<td>16,315,660</td>
<td>16,177,166</td>
</tr>
<tr>
<td>3</td>
<td>NA12003</td>
<td>SRR020472</td>
<td>19,089,406</td>
<td>15,043,846</td>
<td>14,956,114</td>
</tr>
<tr>
<td>4</td>
<td>NA12045</td>
<td>SRR020475</td>
<td>22,117,122</td>
<td>15,081,430</td>
<td>15,023,326</td>
</tr>
<tr>
<td>5</td>
<td>NA12154</td>
<td>SRR023306</td>
<td>17,922,674</td>
<td>14,292,494</td>
<td>14,236,294</td>
</tr>
<tr>
<td>6</td>
<td>NA07051</td>
<td>SRR023301</td>
<td>12,442,124</td>
<td>7,498,420</td>
<td>7,472,714</td>
</tr>
<tr>
<td>7</td>
<td>NA07347</td>
<td>SRR029852</td>
<td>26,281,042</td>
<td>10,691,556</td>
<td>10,649,784</td>
</tr>
<tr>
<td>8</td>
<td>NA11831</td>
<td>SRR027529</td>
<td>25,276,660</td>
<td>19,211,670</td>
<td>19,067,842</td>
</tr>
<tr>
<td>9</td>
<td>NA18486</td>
<td>SRR027528</td>
<td>20,145,972</td>
<td>11,397,190</td>
<td>11,367,976</td>
</tr>
<tr>
<td>10</td>
<td>NA18499</td>
<td>SRR011011</td>
<td>24,605,896</td>
<td>16,997,702</td>
<td>16,839,102</td>
</tr>
<tr>
<td>11</td>
<td>NA18510</td>
<td>SRR024103</td>
<td>25,813,190</td>
<td>18,631,922</td>
<td>18,541,544</td>
</tr>
<tr>
<td>12</td>
<td>NA18516</td>
<td>SRR020479</td>
<td>22,861,892</td>
<td>17,427,526</td>
<td>17,398,112</td>
</tr>
<tr>
<td>13</td>
<td>NA18519</td>
<td>SRR018113</td>
<td>17,574,956</td>
<td>10,444,800</td>
<td>10,420,156</td>
</tr>
<tr>
<td>14</td>
<td>NA18871</td>
<td>SRR020470</td>
<td>19,613,172</td>
<td>15,084,744</td>
<td>15,022,780</td>
</tr>
<tr>
<td>15</td>
<td>NA18959</td>
<td>SRR023789</td>
<td>27,674,990</td>
<td>9,137,412</td>
<td>9,123,166</td>
</tr>
<tr>
<td>16</td>
<td>NA18960</td>
<td>SRR029849</td>
<td>16,607,250</td>
<td>9,384,384</td>
<td>9,245,626</td>
</tr>
<tr>
<td>17</td>
<td>NA18964</td>
<td>SRR022591</td>
<td>23,445,028</td>
<td>18,035,194</td>
<td>17,979,096</td>
</tr>
<tr>
<td>18</td>
<td>NA19102</td>
<td>SRR023300</td>
<td>25,236,608</td>
<td>18,016,936</td>
<td>17,987,502</td>
</tr>
</tbody>
</table>

**S3.5.2 Results**

Table S8 shows the performance of the six compared methods in rediscovering known CNVs for the 18 HapMap individuals, where the average area under the precision-recall curve is used as evaluation criterion. All methods perform better at detecting losses as already seen in previous experiments. cn.MOPS yields a significantly higher performance than its competitors both in terms of the AUC as well as in terms of the recall for FDR set to 0.05, except that FREEC performs equally well for gains.
Table S8: Performance of the compared copy number detection methods on HapMap individuals, where known CNVs should be rediscovered. “PR AUC” gives the average area under the precision-recall curve of 18 samples. “p-value” reports the p-value of a Wilcoxon signed-rank test (over the 18 samples) with null hypothesis that cn.MOPS and another method have the same area under the curve. “Recall” reports the recall at a precision of 0.95, that is, an FDR of 0.05. The last column “p-value” gives the p-value of an analogous Wilcoxon test for the recall with an FDR of 0.05. cn.MOPS could most reliably reconfirm known CNVs. Only for gains, FREEC and cn.MOPS have similar performance, whereas cn.MOPS has significantly higher performance than its competitors at losses.

<table>
<thead>
<tr>
<th>Method</th>
<th>Gains PR AUC</th>
<th>p-value</th>
<th>Recall p-value</th>
<th>Losses PR AUC</th>
<th>p-value</th>
<th>Recall p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cn.MOPS</td>
<td>0.35</td>
<td>—</td>
<td>0.53</td>
<td>—</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td>MOFDOC</td>
<td>0.13</td>
<td>1.17e-03</td>
<td>0.40</td>
<td>2.67e-04</td>
<td>0.38</td>
</tr>
<tr>
<td></td>
<td>EWT</td>
<td>0.16</td>
<td>5.34e-04</td>
<td>0.36</td>
<td>7.63e-06</td>
<td>0.23</td>
</tr>
<tr>
<td></td>
<td>JointSLM</td>
<td>0.08</td>
<td>3.81e-05</td>
<td>0.15</td>
<td>3.81e-06</td>
<td>0.27</td>
</tr>
<tr>
<td></td>
<td>CNV-Seq</td>
<td>0.22</td>
<td>1.74e-02</td>
<td>0.32</td>
<td>7.63e-05</td>
<td>0.27</td>
</tr>
<tr>
<td></td>
<td>FREEC</td>
<td>0.35</td>
<td>8.68e-01</td>
<td>0.42</td>
<td>2.37e-03</td>
<td>0.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Only for gains, FREEC and cn.MOPS have similar performance, whereas cn.MOPS has significantly higher performance than its competitors at losses.
S3.5.3 Evaluation based on a different criterion

In an additional assessment we determine how many known CNVs are rediscovered in NGS data, but now we do not regard the precision of the detection in terms of CNV length and position. In previous experiments we used evaluation segments and assessed not only whether known CNVs are rediscovered but also how precisely. However, array techniques were not able to precisely determine the known CNVs’ breakpoints, thus the ground truth is not reliable concerning length and position of known CNVs.

We are interested in the recall, the true positive rate \( \frac{\#TP}{\#TP + \#FN} \), where the positives are the 170 known CNVs (66 gains and 104 losses). As we do not regard CNV length and position, we redefine true positives: a known CNV is a true positive of a method’s result if at least one of its detected CNVs overlaps with a known CNV. A method should not be able to improve its performance by calling more CNVs, because the increased recall comes at the cost of more false positives and hence a reduced precision \( \frac{\#TP}{\#TP + \#FP} \). To trade true positives off against false positives, we limit the number of detections \( \#TP + \#FP \) for each method. As detections we select the 66 top ranked gain segments and the 104 top ranked loss segments in accordance with the known CNVs. Note that there is bias toward methods that detect longer segments, because they are more likely to overlap with known CNVs (we avoided this bias with the precision-recall curves used above).

Table S9 shows the recall results without regarding the precision in terms of CNV length and position. \texttt{cn.MOPS} had significantly (McNemar’s test) larger recall values.

Table S9: Recall of known copy number regions by detection methods on HapMap individuals without regarding the precision in terms of CNV length and position. “recall” is the recall (true positive rate) and “p-value” gives the p-values of McNemar’s test which indicates whether \texttt{cn.MOPS} has a larger recall than its competitors. Recall values in boldface indicate methods that have significantly larger recall than all other methods. \texttt{cn.MOPS} has significantly larger recall values than other methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Gains</th>
<th></th>
<th>Losses</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall</td>
<td>p-value</td>
<td>Recall</td>
<td>p-value</td>
</tr>
<tr>
<td>\texttt{cn.MOPS}</td>
<td>0.58</td>
<td>—</td>
<td>0.75</td>
<td>—</td>
</tr>
<tr>
<td>\texttt{MOFDOC}</td>
<td>0.00</td>
<td>1.95e-09</td>
<td>0.26</td>
<td>2.53e-12</td>
</tr>
<tr>
<td>\texttt{EWT}</td>
<td>0.12</td>
<td>1.19e-07</td>
<td>0.16</td>
<td>1.56e-14</td>
</tr>
<tr>
<td>\texttt{JointSLM}</td>
<td>0.00</td>
<td>1.95e-09</td>
<td>0.18</td>
<td>4.32e-14</td>
</tr>
<tr>
<td>\texttt{CNV-Seq}</td>
<td>0.45</td>
<td>1.33e-02</td>
<td>0.68</td>
<td>1.46e-01</td>
</tr>
<tr>
<td>\texttt{FREEC}</td>
<td>0.08</td>
<td>2.54e-08</td>
<td>0.51</td>
<td>1.59e-06</td>
</tr>
</tbody>
</table>

S3.5.4 CNV Calls of Different Methods

So far we have considered CNV detection as a classification task whose goal was to detect CNVs in individual samples. Next we assess the quality of the CNV calling across HapMap samples for detecting CNV regions. In contrast to the previous task, we consider a CNV call for a genomic segment across samples but not individual CNV calls. The task is to classify the 114,717 evaluation segments from Subsection S3.1 into segments within a CNV region or non-CN V segments.
The CNV calls have to be defined depending on the method. For cn.MOPS we can readily use the I/NI call. The hyperparameter $G$ of cn.MOPS model was set to $100 \cdot N$. For class (a) methods, namely MOFDOC, EWT, and JointSLM, we use the mean of the $z$-score on the evaluation segment. For the class (b) methods, CNV-Seq and FREEC, we use the mean log-ratios of the evaluation segments. Log-ratios per segment were computed as the log of the read count divided by the segment’s median read count. Note, that the calls shown in the following plots are not the final calls, since all methods suggest a segmentation algorithm that joins initial segments to larger segments for the final CNV call.

Fig. S5 visualizes the results of this task by whole genome CNV calling plots along all evaluation segments. cn.MOPS separates segments within true CNV regions (indicated by red dots) from normal segments (blue dots) better than the other methods. Furthermore, cn.MOPS has lower FDRs for different calling thresholds, as can be seen from the lower variance of the blue dots at the bottom. cn.MOPS’s superior performance at CNV calling across samples is the reason that cn.MOPS has outperformed the other methods in previous experiments.

In Fig. S6 the CNV call is based on variances. For cn.MOPS the variance of the individual I/NI call is used. For both the $z$-score and the log-ratio based CNV calls their variances are used. Also for the variance-based criterion cn.MOPS separates segments within true CNV regions better from non-CNV segments than the other methods.

Finally, in Fig. S7 the CNV call is based on maximal values across samples. For cn.MOPS the maximum of the individual I/NI call is used. For both the $z$-score and the log-ratio based CNV calls their maxima are used. Also for this maximum criterion cn.MOPS separates segments within true CNV regions better from non-CNV segments than the other methods.

The superiority of the I/NI call over $z$-score or log-ratio based methods can not only be deduced from the visualizations in Fig. S5, Fig. S6, and Fig. S7 but also from the area under the precision-recall curve (PR AUC). We compared the performance of the mean, variance and maximum of the I/NI call, $z$-scores and log-ratios. Table S10 reports the area under the precision-recall curve of different methods, where the task was to classify a genomic segment into segments within CNV regions or non-CNV segments. The classification thresholds were the mean, variance and maximum of the individual I/NI call, $z$-scores and log-ratios. Note that there was no segmentation algorithm applied.

Table S10: Performance of different approaches for CNV calling. The task was to classify genomic segments into segments within CNV regions and non-CNV segments. The mean, variance and maximum of the individual I/NI call, $z$-scores and log-ratios served as classification criteria and allowed to compute the area under the precision-recall curve (PR AUC). cn.MOPS outperformed the other methods in all three CNV calling approaches.

<table>
<thead>
<tr>
<th></th>
<th>PR AUC</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>variance</td>
<td>maximum</td>
</tr>
<tr>
<td>individual I/NI call</td>
<td>0.18</td>
<td>0.21</td>
<td>0.21</td>
</tr>
<tr>
<td>$z$-Score</td>
<td>0.02</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>log-ratio</td>
<td>0.14</td>
<td>0.14</td>
<td>0.13</td>
</tr>
</tbody>
</table>
Figure S5: Whole genome CNV calling plots that visualize the performance of cn.MOPS, MOF-DOC, EWT, JointSLM, CNV-Seq, and FREEC in rediscovering known CNVs of HapMap individuals. The plots visualize CNV calling values (vertical axis) along chromosomes 1–22 of the human genome without segmentation. The first panel shows the I/NI call of cn.MOPS. The second panel provides mean z-scores used by MOFDOC, EWT, JointSLM, while the last panel depicts mean log-ratios used by CNV-Seq and FREEC. We called the largest 0.5% of the CNV calling values (blue dots) and scaled them to maximum one. Darker shades of blue indicate a high density of calling values. True CNV regions are displayed as light red bars, and the corresponding CNV calls are indicated by red dots. Segments without calling values (white segments) correspond to assembly gaps in the reference genome. A perfect calling method would call all segments in true CNV regions (red dots) at maximum 1 and would call others (blue dots) at minimum 0. Arrows indicate segments in true CNV regions that are called by one method group, but not by the other method groups. cn.MOPS separates segments in true CNV regions from non-CNV segments better than the other methods, as indicated by the lower variance of I/NI values (see blue area at the bottom of the first panel). The better separation by cn.MOPS results in lower FDRs than those of other methods, regardless of the calling thresholds.

S3.6 High Coverage Real World Data Set

This subsection supplies additional information on the data used in the experiments described in Section “High Coverage Real World Data Set” of the main manuscript. The sequencing files were downloaded on October 25, 2011, from the 1000 Genomes Project Web page. Table S11 provides information on filenames, mapped and used reads of the data.

http://www.1000genomes.org
Figure S6: Whole genome CNV calling plots that visualize the performance of cn.MOPS, MOF-DOC, EWT, JointSLM, CNV-Seq, and FREEC in rediscovering known CNVs of HapMap individuals. The plots visualize CNV calling values (vertical axis) along chromosomes 1–22 of the human genome without segmentation. The first panel shows the variance of the signed I/NI call of cn.MOPS. The second panel provides variance of the z-scores used by MOFDOC, EWT, JointSLM, while the last panel depicts variance of the log-ratios used by CNV-Seq and FREEC. We called the largest 0.5% of the CNV calling values (blue dots) and scaled them to maximum one. Darker shades of blue indicate a high density of calling values. True CNV regions are displayed as light red bars, and the corresponding CNV calls are indicated by red dots. Segments without calling values (white segments) correspond to assembly gaps in the reference genome. A perfect calling method would call all segments in true CNV regions (red dots) at maximum 1 and would call others (blue dots) at minimum 0. Arrows indicate segments in true CNV regions that are called by one method group, but not by the other method groups. cn.MOPS separates segments in true CNV regions from non-CNV segments better than the other methods, as indicated by the lower variance of I/NI values (see blue area at the bottom of the first panel). The better separation by cn.MOPS results in lower FDRs than those of other methods, regardless of the calling thresholds.

Table S11: Information on the high coverage data set from the 1000 Genomes Project. Column “individual” provides the individual’s identifier, “mapped reads” the number of mapped reads, “used reads” the number of reads that were used, and “filename” the reads’ file name. The numbers in the second and third column differ because the sequence library files contain both single and paired end reads.

<table>
<thead>
<tr>
<th>individual</th>
<th>mapped reads</th>
<th>used reads</th>
<th>filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>NA12878</td>
<td>258,212,016</td>
<td>178,971,764</td>
<td>NA12878.chrom1.ILLUMINA.bwa.CEU.high_coverage.20100311.bam</td>
</tr>
<tr>
<td>NA12891</td>
<td>190,462,551</td>
<td>120,566,718</td>
<td>NA12891.chrom1.ILLUMINA.bwa.CEU.high_coverage.20100517.bam</td>
</tr>
<tr>
<td>NA12892</td>
<td>165,437,943</td>
<td>86,444,082</td>
<td>NA12892.chrom1.ILLUMINA.bwa.CEU.high_coverage.20100617.bam</td>
</tr>
<tr>
<td>NA19238</td>
<td>127,491,937</td>
<td>72,871,012</td>
<td>NA19238.chrom1.ILLUMINA.bwa.YRI.high_coverage.20100311.bam</td>
</tr>
<tr>
<td>NA19239</td>
<td>173,589,216</td>
<td>123,354,710</td>
<td>NA19239.chrom1.ILLUMINA.bwa.YRI.high_coverage.20100311.bam</td>
</tr>
<tr>
<td>NA19240</td>
<td>216,431,838</td>
<td>173,422,058</td>
<td>NA19240.chrom1.ILLUMINA.bwa.YRI.high_coverage.20100311.bam</td>
</tr>
</tbody>
</table>
Figure S7: Whole genome CNV calling plots that visualize the performance of cn.MOPS, MOF-DOC, EWT, JointSLM, CNV-Seq, and FREEC in rediscovering known CNVs of HapMap individuals. The plots visualize CNV calling values (vertical axis) along chromosomes 1–22 of the human genome without segmentation. The first panel shows the maximum of the absolute value of the signed I/NI call of cn.MOPS. The second panel provides maximum absolute value of the z-scores used by MOF-DOC, EWT, JointSLM, while the last panel depicts maximum of the absolute value of the log-ratios used by CNV-Seq and FREEC. We called the largest 0.5% of the CNV calling values (blue dots) and scaled them to maximum one. Darker shades of blue indicate a high density of calling values. True CNV regions are displayed as light red bars, and the corresponding CNV calls are indicated by red dots. Segments without calling values (white segments) correspond to assembly gaps in the reference genome. A perfect calling method would call all segments in true CNV regions (red dots) at maximum 1 and would call others (blue dots) at minimum 0. Arrows indicate segments in true CNV regions that are called by one method group, but not by the other method groups. The discrete nature of the cn.MOPS model which is caused by calls of copy number 0, 1, 3 or larger is revealed in this plot. cn.MOPS separates segments in true CNV regions from non-CNV segments better than the other methods, as indicated by the lower variance of I/NI values (see blue area at the bottom of the first panel). The better separation by cn.MOPS results in lower FDRs than those of other methods, regardless of the calling thresholds.

S3.7 Medium Coverage Data Set

In an additional experiment, we investigated the performance of cn.MOPS for medium coverage data. The data set consists of 58 samples of the 1000 Genomes Project that were sequenced at coverages ranging from 2.5X to 8X. Table S13 provides information on filenames, mapped and used reads of the data.

Fixing a segment length of 2,500bp resulted in 25,211 segments on chromosome 20. The International HapMap 3 Consortium identified two CNVs of type “loss” and 21 of type “gain” after filtering for CNVs longer than 10kbp Xie and Tammi Xie (2010). This 10kbp range is the limit for accurate detection for the given coverage according to Xie and Tammi Xie.
and Tammi (2009). The final data set consisted of 1,462,238 evaluation segments of which 18 are losses and 294 are gains.

We decided to analyze gains only, as there were too few losses for a reliable evaluation. Table S12 presents the results. The recall is low since a lot of newly detected CNVs are ranked higher than the confirmed CNVs. Note, however, that the newly detected CNVs highly overlap between CNVSeq, FREEC and cn.MOPS. CNV-Seq and cn.MOPS performed best for gains. The resulting copy number table is available as a separate file (Supplementary Table S17).

Table S12: Performance of the compared copy number detection methods on the medium coverage data (gains only). “PR AUC” gives the area under the precision-recall curve. “Recall” reports the recall at a precision of 0.95. cn.MOPS and CNV-Seq perform equally well in terms of PR AUC, while cn.MOPS performs best in terms of recall. The performance, however, is generally low.

<table>
<thead>
<tr>
<th>Method</th>
<th>PR AUC</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn.MOPS</td>
<td>0.48</td>
<td>0.07</td>
</tr>
<tr>
<td>MOFDOC</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>EWT</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>JointSLM</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>CNV-Seq</td>
<td>0.48</td>
<td>0.03</td>
</tr>
<tr>
<td>FREEC</td>
<td>0.41</td>
<td>0.00</td>
</tr>
</tbody>
</table>

S3.8 Influence of the Hyperparameter $\epsilon$ on cn.MOPS Results

Using two data sets, we investigated the influence of the choice of $\epsilon$ on the performance of cn.MOPS.

S3.8.1 Influence of the Hyperparameter $\epsilon$ tested on Simulated Data

The first data set again consists of the simulated data described in Section “Simulated Data with Constructed CNVs” of the main manuscript. Noisy reads for copy number 0 were generated via a Poisson distribution with parameter $\epsilon = 0.05$. Table S14 shows results obtained by cn.MOPS for different choices of $\epsilon$. Different $\epsilon$ values only lead to minor changes of the performance, thus the cn.MOPS results are robust against the choice of the hyperparameter $\epsilon$. 
Table S14: Performance of cn.MOPS on simulated data with different choices of the hyperparameter $\epsilon$. Different $\epsilon$ values only lead to minor changes of the performance, thus the cn.MOPS results are robust against the choice of the hyperparameter $\epsilon$.

<table>
<thead>
<tr>
<th>$\epsilon$</th>
<th>Gains PR AUC</th>
<th>Gains Recall</th>
<th>Losses PR AUC</th>
<th>Losses Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00001</td>
<td>0.94</td>
<td>0.89</td>
<td>0.96</td>
<td>0.96</td>
</tr>
<tr>
<td>0.001</td>
<td>0.94</td>
<td>0.89</td>
<td>0.98</td>
<td>0.97</td>
</tr>
<tr>
<td>0.01</td>
<td>0.94</td>
<td>0.89</td>
<td>0.97</td>
<td>0.97</td>
</tr>
<tr>
<td>0.02</td>
<td>0.94</td>
<td>0.89</td>
<td>0.97</td>
<td>0.97</td>
</tr>
<tr>
<td>0.05</td>
<td>0.94</td>
<td>0.88</td>
<td>0.96</td>
<td>0.96</td>
</tr>
<tr>
<td>0.1</td>
<td>0.93</td>
<td>0.88</td>
<td>0.96</td>
<td>0.95</td>
</tr>
<tr>
<td>0.2</td>
<td>0.93</td>
<td>0.87</td>
<td>0.96</td>
<td>0.95</td>
</tr>
</tbody>
</table>

S3.8.2 Influence of the Hyperparameter $\epsilon$ tested on Real Sequencing Data with Implanted CNVs

As a second benchmark, we use the data set described in section “Real Sequencing Data with Implanted CNVs from the X Chromosome” of the main manuscript. Table S15 shows results obtained by cn.MOPS for different choices of $\epsilon$. In this case, the performance does not even depend on $\epsilon$ which can be explained easily by the fact that there are no copy number 0 segments in this data set. So we again confirmed that the results of cn.MOPS are robust against the choice of the hyperparameter $\epsilon$.

Table S15: Performance of cn.MOPS on real world benchmarking data with different choices of the hyperparameter $\epsilon$. In this case, $\epsilon$ does not influence the results of cn.MOPS at all.

<table>
<thead>
<tr>
<th>$\epsilon$</th>
<th>Gains PR AUC</th>
<th>Gains Recall</th>
<th>Losses PR AUC</th>
<th>Losses Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>0.70</td>
<td>0.65</td>
<td>0.89</td>
<td>0.88</td>
</tr>
<tr>
<td>0.01</td>
<td>0.70</td>
<td>0.65</td>
<td>0.89</td>
<td>0.88</td>
</tr>
<tr>
<td>0.02</td>
<td>0.70</td>
<td>0.65</td>
<td>0.89</td>
<td>0.88</td>
</tr>
<tr>
<td>0.05</td>
<td>0.70</td>
<td>0.65</td>
<td>0.89</td>
<td>0.88</td>
</tr>
<tr>
<td>0.1</td>
<td>0.70</td>
<td>0.65</td>
<td>0.89</td>
<td>0.88</td>
</tr>
<tr>
<td>0.2</td>
<td>0.70</td>
<td>0.65</td>
<td>0.89</td>
<td>0.88</td>
</tr>
</tbody>
</table>

S3.9 Number of Samples vs. Performance for cn.MOPS

In order to study the influence of the number of samples on the performance of cn.MOPS, we again generated simulated data as described in the section S3.3, but with varying numbers of samples. Table S16 shows the performance of cn.MOPS for different numbers of samples. At least 6 samples seem to be necessary to ensure sufficient performance for detecting gains, while
losses are also detected with fewer samples. For sample numbers larger than 15, the performance saturates.
Table S13: Overview of the medium coverage data set. Column “individual” gives the identifier (the file names are [identifier].chrom20.ILLUMINA.bwa.CEU.low_coverage.20101123.bam), “mapped reads” the number of mapped reads contained in the BAM file, and “used reads” the number of reads that were used. Mapped and used reads differ because the sequence library files contained both single and paired end reads.

<table>
<thead>
<tr>
<th>Individual</th>
<th>Mapped Reads</th>
<th>Used Reads</th>
<th>Individual</th>
<th>Mapped Reads</th>
<th>Used Reads</th>
</tr>
</thead>
<tbody>
<tr>
<td>NA06984</td>
<td>6,964,852</td>
<td>5,684,482</td>
<td>NA11993</td>
<td>5,981,863</td>
<td>5,162,066</td>
</tr>
<tr>
<td>NA06986</td>
<td>8,039,090</td>
<td>6,883,008</td>
<td>NA11994</td>
<td>4,098,534</td>
<td>3,782,234</td>
</tr>
<tr>
<td>NA06989</td>
<td>4,091,782</td>
<td>3,268,000</td>
<td>NA11995</td>
<td>3,757,432</td>
<td>3,416,176</td>
</tr>
<tr>
<td>NA09994</td>
<td>4,874,326</td>
<td>4,520,672</td>
<td>NA12003</td>
<td>3,474,069</td>
<td>3,180,788</td>
</tr>
<tr>
<td>NA07000</td>
<td>8,645,550</td>
<td>8,086,286</td>
<td>NA12004</td>
<td>5,908,548</td>
<td>4,956,778</td>
</tr>
<tr>
<td>NA07037</td>
<td>4,139,953</td>
<td>3,385,920</td>
<td>NA12006</td>
<td>6,922,378</td>
<td>5,092,864</td>
</tr>
<tr>
<td>NA07048</td>
<td>10,981,715</td>
<td>8,253,422</td>
<td>NA12043</td>
<td>6,060,516</td>
<td>5,164,746</td>
</tr>
<tr>
<td>NA07051</td>
<td>4,964,792</td>
<td>4,337,588</td>
<td>NA12044</td>
<td>5,395,490</td>
<td>4,863,338</td>
</tr>
<tr>
<td>NA07056</td>
<td>6,155,442</td>
<td>5,194,792</td>
<td>NA12045</td>
<td>7,799,642</td>
<td>6,532,574</td>
</tr>
<tr>
<td>NA07346</td>
<td>6,100,820</td>
<td>4,857,210</td>
<td>NA12046</td>
<td>4,193,666</td>
<td>4,042,864</td>
</tr>
<tr>
<td>NA07347</td>
<td>7,261,061</td>
<td>6,804,926</td>
<td>NA12058</td>
<td>3,392,356</td>
<td>3,269,980</td>
</tr>
<tr>
<td>NA07357</td>
<td>10,972,910</td>
<td>10,244,264</td>
<td>NA1214</td>
<td>4,214,321</td>
<td>3,899,976</td>
</tr>
<tr>
<td>NA10847</td>
<td>5,766,593</td>
<td>4,709,486</td>
<td>NA12154</td>
<td>6,558,385</td>
<td>5,993,336</td>
</tr>
<tr>
<td>NA10851</td>
<td>6,302,624</td>
<td>4,650,842</td>
<td>NA12155</td>
<td>11,425,911</td>
<td>10,337,646</td>
</tr>
<tr>
<td>NA11829</td>
<td>6,942,857</td>
<td>5,142,718</td>
<td>NA12249</td>
<td>5,607,636</td>
<td>4,764,802</td>
</tr>
<tr>
<td>NA11830</td>
<td>4,812,141</td>
<td>4,137,246</td>
<td>NA12227</td>
<td>5,277,117</td>
<td>4,954,532</td>
</tr>
<tr>
<td>NA11831</td>
<td>5,337,376</td>
<td>4,737,914</td>
<td>NA12273</td>
<td>4,904,914</td>
<td>4,168,754</td>
</tr>
<tr>
<td>NA11843</td>
<td>4,695,305</td>
<td>4,098,448</td>
<td>NA12275</td>
<td>5,248,711</td>
<td>3,861,836</td>
</tr>
<tr>
<td>NA11892</td>
<td>4,711,946</td>
<td>4,240,890</td>
<td>NA12282</td>
<td>5,843,315</td>
<td>4,318,018</td>
</tr>
<tr>
<td>NA11893</td>
<td>6,593,431</td>
<td>5,118,210</td>
<td>NA12283</td>
<td>6,361,286</td>
<td>5,411,708</td>
</tr>
<tr>
<td>NA11894</td>
<td>6,341,119</td>
<td>3,997,266</td>
<td>NA12286</td>
<td>3,616,759</td>
<td>3,314,524</td>
</tr>
<tr>
<td>NA11918</td>
<td>6,346,989</td>
<td>5,514,434</td>
<td>NA12287</td>
<td>5,880,103</td>
<td>4,890,098</td>
</tr>
<tr>
<td>NA11919</td>
<td>8,961,941</td>
<td>8,327,522</td>
<td>NA12340</td>
<td>4,029,639</td>
<td>3,543,810</td>
</tr>
<tr>
<td>NA11920</td>
<td>4,243,883</td>
<td>3,905,198</td>
<td>NA12341</td>
<td>6,851,344</td>
<td>5,359,532</td>
</tr>
<tr>
<td>NA11930</td>
<td>6,415,281</td>
<td>5,654,630</td>
<td>NA12342</td>
<td>7,817,761</td>
<td>4,174,086</td>
</tr>
<tr>
<td>NA11931</td>
<td>5,243,455</td>
<td>4,030,984</td>
<td>NA12347</td>
<td>5,184,621</td>
<td>4,591,614</td>
</tr>
<tr>
<td>NA11932</td>
<td>4,303,807</td>
<td>3,959,018</td>
<td>NA12348</td>
<td>6,175,829</td>
<td>5,053,302</td>
</tr>
<tr>
<td>NA11933</td>
<td>11,876,139</td>
<td>7,682,740</td>
<td>NA12383</td>
<td>5,182,815</td>
<td>4,307,720</td>
</tr>
<tr>
<td>NA11992</td>
<td>4,517,788</td>
<td>4,141,492</td>
<td>NA12399</td>
<td>9,099,699</td>
<td>5,821,360</td>
</tr>
</tbody>
</table>
Table S16: Number of samples vs. performance for cn.MOPS. At least 6 samples seem to be necessary to ensure sufficient performance for detecting gains, while losses are also detected with fewer samples. For sample numbers larger than 15, the performance saturates.

<table>
<thead>
<tr>
<th>samples</th>
<th>Gains PR AUC</th>
<th>Gains Recall</th>
<th>Losses PR AUC</th>
<th>Losses Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.38</td>
<td>0.08</td>
<td>0.69</td>
<td>0.53</td>
</tr>
<tr>
<td>4</td>
<td>0.59</td>
<td>0.21</td>
<td>0.78</td>
<td>0.66</td>
</tr>
<tr>
<td>6</td>
<td>0.74</td>
<td>0.35</td>
<td>0.87</td>
<td>0.76</td>
</tr>
<tr>
<td>8</td>
<td>0.85</td>
<td>0.48</td>
<td>0.92</td>
<td>0.84</td>
</tr>
<tr>
<td>15</td>
<td>0.90</td>
<td>0.72</td>
<td>0.95</td>
<td>0.94</td>
</tr>
<tr>
<td>25</td>
<td>0.93</td>
<td>0.84</td>
<td>0.96</td>
<td>0.96</td>
</tr>
<tr>
<td>40</td>
<td>0.94</td>
<td>0.88</td>
<td>0.96</td>
<td>0.96</td>
</tr>
</tbody>
</table>
S3.10 Exemplary DNA Locations With CNV Calls

In this subsection, we exemplify CNV calls of different methods. Each compared method supplies a CNV call value (signed I/NI call, \(z\)-scores, or log-ratios) at each evaluation segment. We visualize these CNV calls for different methods at exemplary DNA locations.

S3.10.1 CNV Calls at Exemplary DNA Locations With CNVs

First we visualize CNV calls at exemplary DNA locations with CNVs that were previously found and confirmed by the International HapMap 3 Consortium The International HapMap 3 Consortium (2010). Figures S8, S9, and S10 show CNV calls along with GC-corrected read counts. Each line represents the read counts or CNV calls of a sample across consecutive genomic segments, where green lines indicate losses and red lines indicate gains. For \(\text{cn.MOPS}\), the mean signed I/NI call per segment is plotted; for \(\text{MOFDOC}\), the return values of the segmentation algorithm are plotted; for \(\text{EWT}\), the scaled and signed (positive for gains, negative for losses) log-\(p\)-values (transformed \(z\)-scores) are plotted; for \(\text{FREEC}\) and \(\text{CNV-Seq}\), the log-ratios per segment are plotted; for \(\text{JointSLM}\), the median normalized read counts per segment are plotted. Note that even if methods use the same approach (\(z\)-scores or log-ratios), the calling values can be different, because of the different segmentation algorithms they apply.

Figure S8 shows a CNV region that is detected by all methods. Figure S9 shows a CNV region in which only one sample has a deletion that is detected by all methods except \(\text{JointSLM}\). The reason for this is that \(\text{JointSLM}\) only detects variations that appear consistently in the majority of samples. Figure S10 shows a CNV region only detected by \(\text{cn.MOPS}\), \(\text{MOFDOC}\), and \(\text{EWT}\). Note that \(\text{MOFDOC}\) and \(\text{EWT}\) would have a high false discovery rate if the detection threshold was chosen low enough to detect the gain. To conclude, the figures show that \(\text{cn.MOPS}\) produces the most robust and reliable CNV calls.

S3.10.2 CNV Calls at Exemplary DNA Locations Without CNVs

Next, we visualize CNV calls at exemplary DNA locations in which no CNVs have been reported. Figures S11 and S12 again show read counts and CNV calls as in previous figures (see description above). In contrast to previous figures, line colors now represent individual samples.

Figure S11 shows that the class (a) methods \(\text{MOFDOC}\), \(\text{EWT}\), and \(\text{JointSLM}\) falsely detect a CNV at this genomic region. The detection is caused by technical or genomic biases in the center of the region shown, in which read counts are consistently larger. \(\text{FREEC}\) and \(\text{CNV-Seq}\) are based on ratios, where the bias is removed by normalization using a reference read count. Therefore, they correctly do not detect a CNV. \(\text{cn.MOPS}\) does not detect a CNV either at this region because the variation across the samples is too low.

The class (b) methods \(\text{FREEC}\) and \(\text{CNV-Seq}\) are prone to false detections in regions of low coverage, which is exemplified by Figure S12. \(\text{cn.MOPS}\) avoids the low coverage problem by fitting a Poisson distribution across samples.
Figure S8: Copy number call plot for CNV region HM3_CNP_463. Top middle: read counts of each sample around the CNV region (vertical lines); middle left: cn.MOPS’ mean signed I/N1 call.; middle: MOFDOC’s smoothed z-scores; middle right: EWT’s scaled and signed log-p-values (transformed z-scores); lower left: JointSLM’s median normalized read count; lower middle: CNV-Seq’s median log-ratio; lower right: FREEC’s median log-ratio. Each line represents read counts or CNV calls of a sample across consecutive genomic segments; green lines indicate a loss and red lines a gain. All methods detected this loss region.
Figure S9: Copy number call plot for CNV region HM3_CNP_494. Top middle: read counts of each sample around the CNV region (vertical lines); middle left: cn.MOPS’ mean signed I/NI call.; middle: MOFDOC’s smoothed z-scores; middle right: EWT’s scaled and signed log-p-values (transformed z-scores); lower left: JointSLM’s median normalized read count; lower middle: CNV-Seq’s median log-ratio; lower right: FREEC’s median log-ratio. Each line represents read counts or CNV calls of a sample across consecutive genomic segments; green lines indicate a loss and red lines a gain. JointSLM did not detect the CNV segment, as it is only able to detect only variations that appear consistently in the majority of samples.
Figure S10: Copy number call plot for CNV region HM3_CNP_618. Top middle: read counts of each sample around the CNV region (vertical lines); middle left: cn.MOPS' mean signed I/NI call.; middle: MOFDOC's smoothed z-scores; middle right: EWT's scaled and signed log-p-values (transformed z-scores); lower left: JointSLM's median normalized read count; lower middle: CNV-Seq's median log-ratio; lower right: FREEC's median log-ratio. Each line represents read counts or CNV calls of a sample across consecutive genomic segments; green lines indicate a loss and red lines a gain. Only cn.MOPS and maybe MOFDOC detect this gain in one sample. Note that MOFDOC would have a high false discovery rate if the detection threshold was chosen low enough to detect the gain.
Figure S11: Copy number call plot for the region 56.81Mbp–58.01Mbp on chromosome 6 of the human reference genome 18 (build 36). Different colors represent different samples. Top middle: read counts of each sample around the CNV region (vertical lines); middle left: cn.MOPS’ mean signed I/NI call.; middle: MOFDOC’s smoothed $z$-scores; middle right: EWT’s scaled and signed log-$p$-values (transformed $z$-scores); lower left: JointSLM’s median normalized read count; lower middle: CNV-Seq’s median log-ratio; lower right: FREEC’s median log-ratio. The class (a) methods MOFDOC, EWT, and JointSLM falsely detect a CNV in this genomic region. cn.MOPS and the class (b) methods FREEC and CNV-Seq do not detect a CNV in this region.
Figure S12: Copy number call plot for the region 28.91Mbp–9.95Mbp on chromosome 6 of the human reference genome 18 (build 36). Different colors represent different samples. Top middle: read counts of each sample around the CNV region (vertical lines); middle left: cn.MOPS’ mean signed I/NI call.; middle: MOFDOC’s smoothed z-scores; middle right: EWT’s scaled and signed log-p-values (transformed z-scores); lower left: JointSLM’s median normalized read count; lower middle: CNV-Seq’s median log-ratio; lower right: FREEC’s median log-ratio. The class (b) methods FREEC and CNV-Seq falsely detect CNVs in this CNV-free region. cn.MOPS avoids the low coverage problem by fitting a Poisson distribution across samples.
This section is divided into two subsections. The first subsection investigates the read count distribution along a chromosome. The second subsection gives information on how read counts are summarized in a data structure for further processing by CNV detection methods.

### S4.1 Distribution of Read Counts Along the Chromosome

The distribution of read counts of equally sized segments along the chromosome is not Poisson distributed even upon GC correction (Dohm et al. 2008). We confirmed the result in (Dohm et al. 2008).

We found that for segments with a length of 10kbp, 25kbp and 50kbp, the GC corrected read counts have a variance-to-mean ratio larger than 1. For example, on data from the Sanger sequencing center on HapMap phase 1 individuals for segments of 25kbp the variance-to-mean ratio of GC corrected reads was 2.11, after removing sequencing gaps and outliers along the chromosome (read counts larger than two times the median read count). Note, that outliers would even increase the ratio. This ratio larger than 1 contradicts the assumption of a Poisson distribution which would lead to a ratio of 1. Actually, the read counts approximately follow a Gaussian distribution.

The histogram in Fig. S13 for non-uniquely read mapping (see Subsection S3.4.1) shows that the GC corrected read counts are not Poisson distributed. Thus, effects other than the GC bias lead to different average read counts at different genomic segments. The biases cannot be avoided by different mapping strategies like mapping only unique positions (see Subsection S3.4.1) as shown in Fig. S14. Compared to the histogram in Fig. S13, we observe a shift of the density toward lower read counts (left) in the histogram in Fig. S14, because some segments systematically loose reads due to ambiguous mapping.

### S4.2 Data Structure of Read Counts

Next generation sequencing (NGS) data for copy number detection or estimation is in most cases represented as a read count matrix $Z \in \mathbb{N}^{L \times N}$, where the genome is partitioned into $L$ segments of not necessarily equal length for $N$ samples. Therefore $z_{lk} \in \mathbb{N}$ represents the number of reads of sample $k$ that are mapped to the $l$-th segment. Note, that in previous sections we considered only one segment $l$ with read counts $x_k = z_{lk}$. Copy number detection methods applied to such a read count matrix $Z$ are often called “depth of coverage”-based methods. $z^k$ is the $k$-th column of $Z$, which is the read count vector of sample $k$. $z_l$ is $l$-th row of the read count matrix, which is the vector containing read counts for the $l$-th genomic segment for all samples $k$ with $1 \leq k \leq N$. Note, that here we see a substantial novel approach of the cn.MOPS model which uses $z_l = (z_{l1}, \ldots, z_{lk}, \ldots, z_{lN}) = (x_1, \ldots, x_k, \ldots, x_N)$ for modeling across samples, while other methods use $z^k$ to find variations along the chromosome. Fig. S1 depicts entries of the matrix $Z$ by connected by lines and shows modeling along the chromosome and modeling across samples (vertical green boxes).
Figure S13: Histogram for non-uniquely mapped reads of GC corrected read counts from 18 HapMap samples sequenced at the Sanger sequencing center (see Subsection S3.5). Reads are mapped to a random position if more than one best matching position is available.
Figure S14: Histogram for uniquely mapped reads (see Subsection S3.4.1) of the GC corrected read counts from 18 HapMap samples sequenced at the Sanger sequencing center (see Subsection S3.5). Reads with multiple maps to the genome are not regarded. Compared to the histogram in Fig. S13, we observe a shift of the density toward lower read counts (left) because some segments systematically lose reads due to ambiguous mapping.
References


Chapter 3

DEXUS: identifying differential expression in RNA-Seq studies with unknown conditions.

On the following pages I present my publication of a method for identifying differential expression in RNA sequencing data that was published in the journal *Nucleic Acids Research* in September 2013.
DEXUS: identifying differential expression in RNA-Seq studies with unknown conditions

Günter Klambauer, Thomas Unterthiner and Sepp Hochreiter*

Institute of Bioinformatics, Johannes Kepler University, A-4040 Linz, Austria

ABSTRACT

Detection of differential expression in RNA-Seq data is currently limited to studies in which two or more sample conditions are known a priori. However, these biological conditions are typically unknown in cohort, cross-sectional and nonrandomized controlled studies such as the HapMap, the ENCODE or the 1000 Genomes project. We present DEXUS for detecting differential expression in RNA-Seq data for which the sample conditions are unknown. DEXUS models read counts as a finite mixture of negative binomial distributions in which each mixture component corresponds to a condition. A transcript is considered differentially expressed if modeling of its read counts requires more than one condition. DEXUS decomposes read count variation into variation due to noise and variation due to differential expression. Evidence of differential expression is measured by the informative/noninformative (I/NI) value, which allows differentially expressed transcripts to be extracted at a desired specificity (significance level) or sensitivity (power). DEXUS performed excellently in identifying differentially expressed transcripts in data with unknown conditions. On 2400 simulated data sets, I/NI value thresholds of 0.025, 0.05 and 0.1 yielded average specificities of 92, 97 and 99% at sensitivities of 76, 61 and 38%, respectively. On real-world data sets, DEXUS was able to detect differentially expressed transcripts related to sex, species, tissue, structural variants or quantitative trait loci. The DEXUS R package is publicly available from Bioconductor and the scripts for all experiments are available at http://www.bioinf.jku.at/software/dexus/.

INTRODUCTION

The advent of next-generation sequencing has greatly expanded our knowledge about transcriptomes. New transcripts and splice variants have been found and break points of known transcripts determined more accurately (1–6). However, in RNA-Seq experiments, quantification of the expression of transcripts can be difficult (7). Without biological variability, transcripts that are differentially expressed between two conditions can be detected reliably (8). In studies with biological variability, however, detection of differential expression between two conditions remains challenging (9). A transcript that is differentially expressed between many conditions is hard to detect because read count variation due to differential expression and due to high overdispersion can only be distinguished with many samples and high coverage. See Supplementary Section S2 for more details. To detect differentially expressed transcripts, we therefore assume that the number of conditions is small compared with the number of samples.

Identifying differential expression is currently limited to particular study designs

Current methods for analyzing RNA-Seq data can identify differential expression between two conditions. For example, in a case-control study, only transcripts that are differentially expressed between cases and controls can be identified. Similarly, in a randomized controlled study, differential expression between treated and untreated subjects can be detected. These study designs can be generalized to more case groups or more treatments, which leads to multiple (more than two) known conditions. For example, multiple conditions may be due to different tissue types, as in the ‘Allen Brain Atlas’ (10), the ‘Gene Expression Nervous System Atlas’ (11), and the ‘BioGPS’ (12).

Identification of differential expression in RNA-Seq data requires a priori known conditions. In cohort, cross-sectional and nonrandomized controlled studies, the biological conditions are unknown or only partially known. Cohort and cross-sectional studies are observational studies in which the conditions of the subjects are unknown. Examples of observational studies include the HapMap (13), ENCODE (6) and the 1000 Genomes (14) project, for which RNA-Seq data are available (15,16). Nonrandomized controlled studies are treatment studies in which conditions such as genetic, environmental...
or treatment effects are not completely known. In nonrandomized controlled studies, unknown genetic variations such as single-nucleotide polymorphisms (SNPs), copy number variations and unknown environmental factors may result in differential expression between treated subjects. Furthermore, individual unknown treatment effects may cause variation in gene expression, for instance, responses of cell lines to the addition of compounds (17). Other examples are found in oncology, where unknown cancer subtypes or unknown cancer stages are characterized by a particular gene expression profile (18,19).

In nonclinical studies, the conditions are also often unknown. During development, the transcriptome regulates and controls cell growth, differentiation, movement and morphogenesis. Genes are differentially expressed between different time points and between different tissues; even within one tissue, gene expression may vary spatially. For two samples taken at different times or from different locations it is often unknown whether the conditions differ. Another example is in vivo or in vitro gene expression in mice treated with drug candidates (20,21).

Unknown factors such as individual responses or side effects lead to differentially expressed transcripts between the samples.

The detection of differential expression in RNA-Seq studies with unknown conditions is important to obtain new biological knowledge. Current RNA-Seq methods, however, require the conditions to be known. For microarray data, a method for identifying unknown conditions in gene expression has been suggested (22). However, this method cannot be applied to RNA-Seq data with unknown conditions because a primary modeled factor is required and the noise is assumed to be Gaussian, which is not appropriate for RNA-Seq count data (23).

We therefore present DEXUS, a method capable of detecting differential expression in RNA-Seq studies with unknown conditions.

A summary of study designs and methods that can detect differential expression in them is shown in Table 1.

**Existing methods for detecting differential expression in RNA-Seq data**

Methods that detect differential expression in RNA-Seq data are usually based on read counts, i.e. the number of reads mapping to a DNA region that is transcribed, such as a gene or an exon (32). These methods compare read counts for two conditions. If read counts show a large and consistent difference between the conditions, then the according transcript is differentially expressed. In this subsection, we review methods that detect differential expression in RNA-Seq data. Many methods model read counts by a negative binomial distribution because even after normalization the read counts have high variance. Therefore, we divide methods into two classes: those which do not use negative binomials (class A) and those which do (class B).

The following methods belong to class A.

DESeq (26) assumes that the log fold change of mean read counts between the two conditions follows a normal distribution given the log average expression. A differentially expressed gene is identified by a small P-value by means of this distribution.

NOISeq (30) also considers the log fold change of read counts between two given conditions together with their absolute difference. Empirical distributions are calculated using all pairs of replicates from different conditions. NOISeq identifies a gene as differentially expressed if the log fold change of read counts and the absolute difference of read counts between the two conditions have both a small P-value for the empirical distributions.

SAMSeq (27) performs a Wilcoxon test for each transcript testing the counts of one condition against the counts of the other. Because standard normalization techniques are not applicable, subsampling is used to normalize the read counts. SAMSeq requires a relatively high number of samplings per condition to obtain significance for differential expression.

PoissonSeq (29) fits a Poisson log-linear model to the read counts after transforming them. A score statistic on the model parameters determines the significance for differential expression.

The following class B methods use negative binomial distributions to model the read counts.

edgeR (25) uses a quantile-adjusted conditional maximum likelihood estimator for the overdispersion parameter of the negative binomial distribution. This estimator is more accurate than the standard maximum likelihood estimator when only few replicates per condition are available (33). Borrowing information across transcripts allows the dispersion parameter to be adjusted toward a consensus value using an empirical Bayes procedure (34). Finally, edgeR uses an exact test
to determine whether the counts of the two conditions come from the same negative binomial distribution.

DESeq (24) pools together transcripts with similar expressions values to improve the estimate of the overdispersion parameter. The overdispersion is assumed to be a function of the mean read count and is therefore estimated per condition. To determine whether a transcript is differentially expressed, the distribution parameters of the two conditions are tested by an exact test for equality of means.

baySeq (26) determines the distribution of the overdispersion parameter by applying a quasi-likelihood method to the read counts of one condition. The resulting distribution is used as prior for estimating the overdispersion parameter when fitting the model to the read count data.

DSS (31) is similar to baySeq. A negative binomial distribution is fitted to the read count data using a prior on the overdispersion parameter. This prior is a log-normal distribution, whose parameters are optimized using the dispersion parameters of each condition. Finally, a Wald test is used to determine differential expression.

In summary, the class B methods, which use negative binomial distributions, i.e. DESeq, baySeq, DSS and edgeR, mainly differ in the way they estimate the overdispersion parameter. Estimating the overdispersion parameter is crucial for the performance and not trivial because the maximum likelihood estimator is biased and has high variance if the sample size is small (33). The subsequent statistical test has a smaller effect on the results than the parameter estimates (23,31).

Extensions to multiple known conditions
McCarthy et al. (32) extended the R package edgeR to more than two conditions. A generalized linear model is fitted to the data, and then coefficients are tested for being different from zero, which leads to the final P-values. Again, the estimation of the overdispersion parameter for a transcript borrows information from other transcripts. DESeq, baySeq and SAMSeq have also been extended to more than two conditions.

MATERIALS AND METHODS
Method overview
Our goal is to identify differentially expressed transcripts in studies with unknown conditions. A transcript is differentially expressed if the mean expression levels for different conditions are different and read counts are observed under more than one condition. Therefore we assume a small number of conditions because, as mentioned above, the detection of differential expression for many conditions is difficult. RNA-Seq expression data are usually represented as read counts per transcript, or alternatively by exon or gene. It was observed that read counts from a single condition follow a negative binomial distribution (24–26,31). DEXUS therefore models read counts as a finite mixture of negative binomial distributions.

The model
Read count x per transcript is explained by a mixture of n negative binomial distributions:

\[ p(x) = \sum_{i=1}^{n} \alpha_i \text{NB}(x ; \mu_i, r) \]  

where \( \alpha_i \) is the probability of being in condition i out of n possible conditions. In condition i, read counts are drawn from a negative binomial distribution with mean \( \mu_i \) and size \( r_i \), where the size parameter \( r_i \) is the inverse of the overdispersion \( \phi \). Note that we use the \( (\mu, r) \) instead of the usual \( (\mu, \phi) \) parameterization to locally accumulate parameters that are associated with large overdispersions. This accumulation is essential to define a prior within a Bayesian framework.

A nondegenerate DEXUS model is identifiable (see Supplementary Section S3.1.3), as required for the maximum likelihood and the maximum a posteriori estimator to be consistent. Consistency means that the estimator converges to the true parameter values with more data points, which is important for identifying differential expression. If the mean read count exceeds the variance, the maximum likelihood estimate of \( r \) tends to \( \infty \) and the negative binomial converges to a Poisson distribution (see Supplementary Section S3.2.2).

Model selection
We perform model selection in a Bayesian framework by maximizing the posterior, i.e. by a MAP approach (35–37). Therefore, the parameters \( \pi = (\pi_1, \ldots, \pi_n) \), \( \mu = (\mu_1, \ldots, \mu_n) \) and \( r = (r_1, \ldots, r_n) \) are considered to be random variables, and the likelihood \( p(x) \) in Equation (1) becomes the conditional probability
\[ p(x|\alpha,\mu,r) \hspace{1cm} \text{The objective of the model selection is to maximize the posterior of the parameters:} \]
\[ p(\mu,\alpha,r|x) = \frac{p(x|\mu,\alpha,\gamma) p(\mu) p(\alpha) p(\gamma)}{\int p(x|\mu,\alpha,\gamma) p(\mu) p(\alpha) p(\gamma) \, d\alpha \, d\mu} \]
\[ = \frac{1}{\ell(x)} p(x|\mu,\alpha,\gamma) p(\mu) p(\alpha) p(\gamma) \hspace{1cm} (2) \]
where the priors on \( x, \mu \) and \( r \) are assumed to be independent of each other, and are defined in the following.

\textbf{Dirichlet prior for probabilities of conditions}

First we choose the prior \( p(\alpha) \) on the probabilities of the conditions. Since the majority of transcripts in a data set are usually not differentially expressed, the model should favor explaining the read counts for a transcript with a single condition. The null hypothesis of one condition should only be rejected if the data contain strong evidence for more than one condition. The prior reduces the number of falsely discovered differentially expressed transcripts and therefore keeps the false discovery rate low. DEXUS uses a Dirichlet prior \( p(\alpha) \) on \( \alpha \) with parameters \( \gamma \) to incorporate the preference for only one condition:
\[ p(\alpha) = \frac{1}{\Gamma(\gamma)} \prod_{i=1}^{n} \alpha_i^{\gamma_i - 1} \hspace{1cm} (3) \]
where \( \alpha_i \) is an \( n \)-dimensional probability vector. Each component \( \alpha_i \) is distributed according to a beta distribution with \( \text{mode}(\alpha_i) = (\gamma_i - 1)/\sum_{j=1}^{n} (\gamma_j - n) \).

To express the prior knowledge that most transcripts are not differentially expressed and are generated under only one condition, we set \( \gamma_i > \gamma_j \) (for \( i > 1 \)). This setting assumes that most read counts are generated under condition \( i = 1 \), which we call the major condition, while conditions \( i > 1 \) are called minor conditions. The vector of hyperparameters \( (\gamma_1, \gamma_2, \ldots , \gamma_n) \) can be simplified to one hyperparameter \( G \) (Supplementary Section S3.2.1). In Supplementary Section S3.4 we show that DEXUS is not sensitive to the choice of the hyperparameter \( G \).

Therefore DEXUS is easy to use as good results are obtained with the default setting of \( G = 1 \) (see Supplementary Section S3.4). Without having seen the data, we assume that only the major condition is present, which means that the transcript is not differentially expressed. Only when the data show strong evidence also for minor conditions, does the posterior assign nonzero probabilities to minor conditions and the transcript is called differentially expressed.

\textbf{Truncated exponential priors for overdispersions}

In DEXUS model selection, the second prior is on the size parameter \( r \) of the negative binomial distribution, which determines the overdispersion. A prior on \( r \) improves the estimation of \( r \) if the number of samples is small. The maximum likelihood estimator of \( r \) is biased for few samples and overestimates the true size parameter (38,39), as confirmed in Supplementary Section S3.2.5. In a Bayesian approach, the influence of the prior decreases with an increasing number of samples, and therefore the MAP estimator is asymptotically (number of samples tending to infinity) unbiased.

To keep the estimate of \( r \) small, the prior pushes \( r \) toward zero. We choose an exponential distribution as prior:
\[ p(r) = \eta \exp(-\eta r) \hspace{1cm} (4) \]
where \( \eta \) is a hyperparameter.

Like DESeq (24), we truncate the size parameter at the right-hand tail by using the constraint \( r \leq r_{\text{max}} \). The upper bound \( r_{\text{max}} \) on the size parameter is equivalent to a lower bound on the overdispersion and ensures minimal overdispersion for the read counts of each transcript. Further, this bound makes the parameter space compact, which is required for a consistent estimator.

The same exponential prior is used for each component of \( r \). The hyperparameter \( \eta \) for the exponential prior on \( r \) is transformed to a hyperparameter \( \theta \) (see Supplementary Section S3.2.5). Like the hyperparameter \( G \), also \( \theta \) is robust and supplies good results with \( \theta = 2.5 \).

\textbf{Uniform priors for means}

Finally, DEXUS model selection uses a prior on the mean \( \mu \) of the negative binomial distribution. If \( \mu \) is the same in all conditions, all read counts were close to zero (transcripts are not present), the estimate of the mean of the negative binomial would not converge. Therefore, \( \mu_i \) is lower bounded by \( \mu_{\text{min}} \). To ensure a compact parameter space as required for a consistent estimator, we use a uniform prior on \( \mu_i \) on the compact interval \( [\mu_{\text{min}}, \mu_{\text{max}}] \), where \( \mu_{\text{max}} \) can be set to the largest observed read count.

In summary, DEXUS has only few parameters which in most applications need not be adjusted by the user, as their default values give good results.

\textbf{EM algorithm}

With the priors defined, the model with maximum parameter posterior can be selected. The EM algorithm (40) is used to minimize an upper bound on the negative log-posterior of the parameters. The E-step of the EM algorithm estimates the probability that read count \( x_k \) is generated under condition \( i \) (the probability of condition \( i \) after observing data \( x_k \)):
\[ a_{ik} = \frac{a_i \cdot NB(x_k; \mu_i, r)}{\sum_j a_j \cdot NB(x_k; \mu_j, r)} \hspace{1cm} (5) \]

This equation is the E-step (expectation step) of the EM algorithm. Using the posterior estimates \( a_{ik} \), we obtain following update rules for the M-step (maximization step):

- estimate for \( a_i \):
\[ a_i = \frac{1}{N} \sum_{k=1}^{N} a_{ik} \hspace{1cm} (6) \]
\[ \mu_i = \frac{1}{\lambda} \sum_{k=1}^{n} \alpha_k x_k / \alpha_i, \] 

(7)

**r** update:

The new \( r_i \) is obtained by solving the following equation for \( r_i \):

\[ \sum_{k=1}^{n} \alpha_k \psi(x_k + r_i) - N(\hat{\alpha}_i) \psi(r_i) + N(\hat{\alpha}_i) \frac{r_i}{\sum_{k=1}^{n} \alpha_k (x_k + r_i)} - \eta = 0, \]

where \( \psi \) is the digamma function. The equation is solved numerically for \( r_i \) by means of a ‘bisection’ procedure.

\[ \alpha_i = \hat{\alpha}_i + \frac{1}{\lambda} \frac{(y_i - 1) - \frac{1}{\lambda} (\sum_{k=1}^{n} y_k - n)}{1 + \frac{1}{\lambda} (\sum_{k=1}^{n} y_k - n)}, \]

(9)

The complete derivation of the EM algorithm can be found in the Supplementary Section S3.2.1. \( \mu_i \) and \( r_i \) are initialized by using the results of k-means clustering (see Supplementary Section S3.2.4). The values \( \alpha_i \) are simply initialized with the maximum entropy setting \( \alpha_i = 1/n \).

**I/NI value: evidence of differential expression**

The Bayesian framework allows definition of an I/NI call (36,37,41,42). The I/NI call serves to extract differentially expressed transcripts. The Bayesian framework allows definition of an I/NI call (36,37,41,42). The I/NI call serves to extract differentially expressed transcripts.

The evidence of differential expression (the I/NI value) should consider two factors: (i) \( \psi \), as the evidence for the presence of the minor condition \( i > 1 \); (ii) the difference between the means of the major and minor conditions as evidence that they are indeed different.

The difference between the means is expressed by the log difference \( \log(\mu_i) - \log(\mu_i) \). Factor (i) is incorporated into the I/NI value by weighting these differences by \( \hat{\alpha}_i \), which yields

\[ \text{I/NI}(\hat{\alpha}, \mu) = \sum_{i=1}^{n} \hat{\alpha}_i \left[ \log(\frac{\mu_i}{\mu_i}) \right] \]

(10)

The I/NI value is the expected log fold change of read counts with respect to the mean read count of the major condition given a noise-free model. ‘Noise-free’ refers to the assumption that under each condition, only the mean read count is observed. For a mathematical interpretation of the I/NI value see Supplementary Section S3.5.2.

**Experiments**

We evaluated DEXUS on simulated and real-world data sets. The simulated data sets had various library sizes, different numbers of replicates and different ratios between mean read counts under the different conditions. DEXUS was tested on the following real-world RNA-Seq data sets: (i) ‘Nigerian HapMap’, based on 69 Nigerian HapMap individuals, (ii) ‘European HapMap’, based on 60 European HapMap individuals, (iii) ‘Primate Liver’, based on liver tissue samples from humans, chimpanzees and rhesus macaques, (iv) ‘Maize Leaves’, using samples from different locations of maize plant leaves, and (v) ‘Mice Strains’, based on different strains of mice (Supplementary Section S4.2.4).

First we report the performance of DEXUS on 2400 simulated data sets for which the conditions were known but withheld from DEXUS. We then present tests on real-world data sets with either unknown conditions (‘Nigerian HapMap’, ‘European HapMap’) or partially known conditions (‘Primate Liver’, ‘Maize Leaves’). In the latter case the conditions were withheld from DEXUS to ascertain whether it can identify them.

**DEXUS for known conditions**

Before we tested DEXUS on data with unknown conditions, we assessed how well it performs if the conditions of interest are known. For known conditions, DEXUS estimates only the parameters of a negative binomial for each condition. Therefore, we compared the parameter estimates of DEXUS to previously suggested methods in terms of detecting differentially expressed transcripts, namely the following eight state-of-the-art methods: DSS (31), DESeq (24), baySeq (26), edgeR (25), DEGseq (28), NOISeq (30), PoissonSeq (29) and SAMseq (27).

If only few samples per condition are available, the performance of DEXUS is below the best performing other methods. For medium and large sample numbers and large
library sizes (1e7 and 1e8) DEXUS outperforms all other methods. The experiments and the respective results are described in detail in Supplementary Section S4.2.

Simulated RNA-Seq data

Generating simulated RNA-Seq data

We simulated data sets from different experimental settings following the suggestions of Robinson et al. (34), Hardcastle and Kelly (26) and Wu et al. (31). For all samples of a data set, the library size was 1e6, 1e7 or 1e8 to cover a wide range of applications. Keeping the library size and the read quality constant for each sample in a data set avoids the need for normalization of the read counts, i.e. it avoids normalization biases. For each experiment, we choose a particular number of replicates per condition to evaluate DEXUS for different sample sizes and for unbalanced data. In case of two conditions, the numbers of replicates were 6/6, 9/3, 10/2, 11/1, 12/6, 18/6, 20/4, 22/2 (condition1/condition2). Each experiment consisted of 100 data sets with 1e00 transcripts each. The conditions were known and used for evaluation but withheld from DEXUS.

For the simulation we assumed that under condition $i$ the reads $x$ for a transcript are distributed according to a negative binomial $NB(x; \mu_i, \phi_i)$. After Wu et al. (31), we took the mean $\mu_i$ and the size $r_i$ from the ‘Mice Strains’ benchmark RNA-Seq data set (43) using only data from one particular biological condition. For a randomly selected transcript, the value $\mu_i$ was obtained as the median read count of the condition.

The overdispersion $\phi_i = 1/r_i$ tends to decrease with increasing mean read counts (see Supplementary Figure S15). Therefore, we fitted a regression line to overdispersion values by least squares. After sampling $\mu_i$ values, the corresponding $\phi_i$ values were obtained by means of the regression line to which zero-one normally distributed noise was added. Thirty percent of the genes were chosen to be differentially expressed. Differential expression was modeled by adjusting the means of the negative binomials to obtain log fold changes of 0.5, 1 and 1.5 between the mean of the major and the minor condition. The fold change values are randomly chosen with equal probability, such that all 3-fold change conditions that were found by DEXUS are related to the sex. For four of the 12 top-ranked genes, the identified conditions are visualized as a heatmap in Figure 1.

Evaluation criteria for simulated RNA-Seq data

We formulate the detection of differential expression as a classification task: DEXUS must decide whether a transcript is differentially expressed (positive prediction) or not (negative prediction). For the simulated data, we knew which transcripts were differentially expressed (the positives) and which were not (the negatives). DEXUS ranks the transcripts by the I/NI value from Equation (10). For a given I/NI threshold (the I/NI call), we can determine true positives, false positives, true negatives and false negatives. Using these numbers, we computed the specificity and the sensitivity of DEXUS. The specificity corresponds to ‘1 – significance level’ or ‘1 – type I error rate’. The type I error rate is the ratio between false detections and all negatives. The sensitivity corresponds to the ‘power’ or ‘1 – type II error rate’. The type II error rate is the ratio between missed positives and all positives.

Results on simulated RNA-Seq data

We tested DEXUS on the simulated RNA-Seq data using its default parameters. Table 2 shows the results in terms of sensitivity and specificity for library size 1e8 at different thresholds for the I/NI value. Transcripts with an I/NI value above the threshold are called informative or (equivalently) differentially expressed. Results for other library sizes are presented in Supplementary Tables S12 and S13. The specificity of DEXUS is high across various numbers of replicates, whereas the sensitivity varies considerably. High specificity means that few transcripts are falsely identified as being differentially expressed. In highly unbalanced experiments, i.e. 11/1 and 22/2 replicates, differentially expressed transcripts are detected only at low I/NI thresholds of 0.025 and 0.05. Note that the minor condition i = 2 (smaller subgroup) leads to a small $\alpha_2$ and therefore to a small I/NI value. For unbalanced data, the few minor condition samples must be distinguished from random outliers of the major condition.

Real-world RNA-Seq data

‘Nigerian HapMap’

Pickrell et al. (16) sequenced RNA from 69 Nigerian HapMap individuals to study expression quantitative trait loci (eQTLs). The read count data were provided by the ReCount repository (44). As in previous experiments, DEXUS was applied to these data with its default parameters and ranked genes according to the I/NI value. The read counts of top-ranked genes and the conditions identified by DEXUS are visualized as a heatmap in Figure 1.

Five out of the 12 top-ranked genes are located on the Y chromosome (RPS4Y1, CYorf15A, EIF1AY, TMSB4Y, RPS4Y2). For these genes, the identified conditions are related to the sex. For four of the 12 top-ranked genes, at least one eQTL is known. For ZFP57, the associated eQTL is the SNP rs1736924 with a minor allele frequency (MAF) of 0.14 (16). CDH1 has 6 eQTLs, one of which is SNP rs7196495 with a MAF of 0.22 (45). CLLU1OS possesses the eQTL SNP rs12580153 with a MAF of 0.19 (46). LITD1 has 2 eQTLs, one of which is SNP rs12137088 with a MAF 0.30 (47). Because the MAFs are high, it is plausible that the minor alleles are observed in the HapMap data set and that they lead to differential expressions of the associated genes. The conditions that were found by DEXUS are related to the alleles of corresponding SNPs. Because the HapMap samples are lymphoblastoid cells, we confirmed that the genes detected by DEXUS are indeed expressed in lymphoblastoid cell lines. The gene NLRP2, ranked 11th by DEXUS, is expressed in lymphoblastoid cells but with large variability (48), as shown in Supplementary Figure S17. NLRP2 is expressed in the HapMap individuals, but in some at a low level.
Schlattl et al. (49) identified a copy number variable region that partially covers NLRP2 and causes its differential expression. Therefore, the conditions that DEXUS identified for NLRP2 may be related to copy number states of the samples. Copy number states may also be explained by sex-related transcripts. An analogous test for the Y chromosome was ranked 12th by DEXUS. Pinto et al. (51) identified a copy number variable region covering MKRN3. However, interpreting the MKRN3 conditions is difficult because only the paternal copy of MKRN3 is expressed.

We analyzed the I/NI value ranking of transcripts: genes on the X chromosome were ranked significantly higher than other genes ($P = 3.0e^{-12}$), which can be explained by sex-related transcripts. An analogous test for the Y chromosome was not significant because too few genes were expressed. However, as already mentioned, five out of the 12 top-ranked genes are located on the Y chromosome. At an I/NI threshold of 0.1, DEXUS called 366 differentially expressed genes. Gene set enrichment analysis showed that the called genes are associated with the extracellular region and the plasma membrane. In total, 20 significant GO terms were found, including ‘extracellular space’, ‘extracellular region part’ and ‘plasma membrane part’ with $P = 2.5e^{-5}$, $P = 8.8e^{-5}$ and $P = 0.01$, respectively. ‘Cell–cell signaling’, ‘chemokine receptor binding’ and ‘chemokine activity’ were also significant at $P = 4.0e^{-3}$, $P = 8.0e^{-4}$ and $P = 9.8e^{-4}$ ($P$-values were corrected for multiple testing by means of the Benjamini–Hochberg procedure). These GO terms are in agreement with characteristics of lymphoblastoid cells. Supplementary Table S18 provides a complete list of all significant GO terms.

**European HapMap**

We analyzed the RNA-Seq data of 60 individuals from the HapMap cohort from Montgomery et al. (15), which were

---

### Table 2. The performance of DEXUS in terms of sensitivity and specificity in detecting differential expression with unknown conditions

<table>
<thead>
<tr>
<th>I/NI threshold</th>
<th>0.025</th>
<th>0.05</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1/C2</td>
<td>Specificity</td>
<td>Sensitivity</td>
<td>Specificity</td>
</tr>
<tr>
<td>6/6</td>
<td>0.893 ± 0.001</td>
<td>0.775 ± 0.009</td>
<td>0.951 ± 0.002</td>
</tr>
<tr>
<td>9/3</td>
<td>0.893 ± 0.004</td>
<td>0.827 ± 0.006</td>
<td>0.951 ± 0.002</td>
</tr>
<tr>
<td>10/2</td>
<td>0.893 ± 0.003</td>
<td>0.819 ± 0.008</td>
<td>0.950 ± 0.002</td>
</tr>
<tr>
<td>11/1</td>
<td>0.893 ± 0.003</td>
<td>0.677 ± 0.009</td>
<td>0.951 ± 0.002</td>
</tr>
<tr>
<td>12/12</td>
<td>0.945 ± 0.002</td>
<td>0.735 ± 0.008</td>
<td>0.982 ± 0.001</td>
</tr>
<tr>
<td>18/6</td>
<td>0.945 ± 0.001</td>
<td>0.816 ± 0.008</td>
<td>0.950 ± 0.002</td>
</tr>
<tr>
<td>20/4</td>
<td>0.945 ± 0.003</td>
<td>0.810 ± 0.008</td>
<td>0.982 ± 0.002</td>
</tr>
<tr>
<td>22/2</td>
<td>0.946 ± 0.002</td>
<td>0.650 ± 0.009</td>
<td>0.982 ± 0.001</td>
</tr>
<tr>
<td>Mean</td>
<td>0.919 ± 0.028</td>
<td>0.764 ± 0.069</td>
<td>0.966 ± 0.017</td>
</tr>
</tbody>
</table>

The first column 'C1/C2' contains the numbers of replicates for the first and second condition. The other columns list sensitivity and specificity (with standard deviations) of DEXUS at different I/NI thresholds as the average across 100 data sets. The last row ('Mean') gives the average of the results in the columns. The library size was 18f for all experiments.

![Figure 1. Heatmap of the normalized read counts of the 12 genes with the largest I/NI values for the 'European HapMap' data set. Colors range from white for low expression to blue for high expression. Different individuals are denoted along the x-axis, while the top-ranked genes are denoted by their gene symbols along the y-axis. Red crosses indicate samples that belong to the minor condition. At the right side of the heatmap, each gene is annotated by the minimum (‘<’), the median of two conditions ('m1' and 'm2') and the maximum (‘>’).](image-url)
Figure 2. Heatmap of the normalized read counts of the 12 genes with the largest I/NI values for the ‘European HapMap’ data set. Colors range from white for low expression to blue for high expression. Different individuals are denoted along the x-axis, while the top-ranked genes are denoted by their gene symbols along the y-axis. Red crosses indicate samples that belong to the minor condition. At the right hand side of the heatmap, each gene is annotated by the minimum (‘<’), the median of two conditions (‘=’), and the maximum (‘>’) read count.

provided by the ReCount repository (44). Again, DEXUS was applied to these data with its default parameters and ranked genes according to the I/NI value. The read counts of top-ranked genes and the identified conditions are visualized as a heatmap in Figure 2.

RPS4Y1 is the gene with the largest I/NI value, differentially expressed between males and females, and located on the Y chromosome. The genes CYorf15A and TMSB4Y, ranked fourth and fifth according to the I/NI value, are also located on the Y chromosome. As in the ‘Nigerian HapMap’ data set, ZFPS57 was detected as being differentially expressed. In addition to ZFP57, two other of the 12 top-ranked genes have eQTLs. CLLL1OS has as eQTL the SNP rs12580153 with a MAF of 0.19 (46). For the genes T, PRSS21 and RASSF10, DEXUS differentially expressed owing to variable copy numbers (49). As in the ‘Nigerian HapMap’ data set, some top-ranked genes, such as NLRP2 (again rank 1), were differentially expressed owing to variable copy numbers (49). For the genes T, PRSS21 and RASSF10 are visualized in Supplementary Figures S19, S20 and S21.

Analyzing the I/NI value ranking, we found that genes on the X chromosome are ranked significantly higher ($P = 0.0c-6$, Wilcoxon test). The analog test for the Y chromosome yielded no significant results, as too few genes were expressed. However, three out of the 12 top-ranked genes with the largest I/NI value are located on the Y chromosome.

At an I/NI threshold of 0.1, DEXUS called 680 differentially expressed genes. Gene set enrichment analysis showed that the called genes are associated with ion transport. Significant Gene Ontology (GO) terms were ‘ion transport’, ‘potassium ion transport’ with $P = 0.04$ and $P = 4.3e-03$, respectively. Again ‘plasma membrane part’ was significant at $P = 0.027$. Although 36 of the 680 genes were related to ‘cell-cell signaling’ and 6 to ‘chemokine activity’, these GO terms were not significant in this data set after correction for multiple testing by means of the Benjamini-Hochberg procedure. A table of all significant GO terms can be found in Supplementary Table S19.

‘Primate Liver’

Blekhman et al. (53) investigated the differences in alternative splicing in liver tissue between humans, chimpanzees and rhesus macaques. For this purpose, they performed RNA-Seq on three male and three female liver samples from each species. They focused on the expression values of exons that had reliably determined orthologs in all species. Read counts for exons were provided by Blekhman et al. (53), who used gene models from Ensemble (Release 50). After pooling technical replicates, DEXUS ranked genes according to the I/NI value using its default parameters. The 10 top-ranked genes are visualized in Figure 3, which shows clear differential expression between the species. For these genes, and without having been provided with this information, DEXUS determined one of the three species as minor condition. Interestingly, out of the 10 top-ranked genes, six are human pseudogenes, AC010591.10, AC105383.3, AC093874.3-1, AC105383.3, AL132855.4 and UOX, which are inactive in humans because of recent structural
rearrangements (54). Because the rearrangements are recent, their orthologs can be identified reliably in other primates. Differential expression is detected because these orthologs are still transcribed in Pan troglodytes and in Macaca mulatta.

Several of the 10 top-ranked genes are associated with liver pathways and are therefore expressed in liver tissues. Differential expression of these genes between species may have arisen from different diets. Examples of such genes are the human pseudogene UOX, which catalyzes the oxidation of uric acid to allantoin in M. mulatta, ABP1 and GSTM5, which participate in degradation and detoxification pathways, VNN3, which helps to recycle vitamin B5, and CHFR2, which is associated with lipoproteins.

Thresholding the I/NI call at 0.1, DEXUS called 3384 genes (16% of all genes) as differentially expressed. A gene set enrichment analysis found the GO terms ‘intrinsically to plasma membrane’ (P = 7.9e−7) and ‘integral to plasma membrane’ (P = 4.0e−6) to be significant. Thus, genes that encode membrane proteins seem to be differentially expressed between species more often than other genes. Interestingly, also ‘response to extracellular stimulus’, ‘response to nutrient’ and ‘response to nutrient levels’ were significant (all P-values <7.6e−5), which supports the hypothesis that some genes are differentially expressed owing to the different diets of the species. All P-values were corrected by means of the Benjamini–Hochberg procedure.

Figure 3. Heatmap of the normalized read counts of the 10 genes with the largest I/NI values for the ‘Primatc Liver’ data set. Colors range from white for low expression to blue for high expression. The y-axis shows female and male individuals from the three species human Homo sapiens (HS), chimpanzee Pan troglodytes (PT) and rhesus macaques Macaca mulatta (MM). The x-axis displays top-ranked genes indicated by their gene symbols. Red crosses mark samples that were assigned to the minor condition. At the right side of the heatmap, each gene is annotated by the minimum (‘<’), the median of two conditions (‘m1’ and ‘m2’) and the maximum (‘>’) read count.

Figure 4. Heatmap of the normalized read counts of the 10 genes with the largest I/NI values for the ‘Maize Leaves’ data set. Colors range from white for low expression to blue for high expression. The y-axis shows samples from different locations on the maize plant leaf. The x-axis displays different genes denoted by their gene symbols. Red crosses indicate that the according samples belong to the minor condition. At the right hand side of the heatmap, each gene is annotated by the minimum (‘<’), the median of two conditions (‘m1’ and ‘m2’) and the maximum (‘>’) read count.

‘Maize Leaves’

Using RNA-Seq data from various locations on maize plant leaves, Li et al. (55) studied the developmental dynamics of the maize transcriptome. For each location, two biological replicates were sequenced with Illumina’s Genome Analyzer II. The reads were mapped to the TE-masked Zea mays ZmBT3 reference genome version 2 (AGPv2), release 5a, using the GSNAP splicing short read mapper (56). We counted the overlaps between mapped reads and the Z. mays gene definitions from the Ensemble Plants database (Release 14). Reads that have multiple possible alignments or that overlap with more than one gene were discarded. DEXUS was applied to this data with its default parameters.

Figure 4 shows the genes with the largest I/NI value and the conditions that were identified by DEXUS. DEXUS found differences in gene expressions between different locations on the leaf despite this information being withheld. Further, it almost always assigned the two replicates to the same condition without knowledge of replicates or leaf locations. Thus, DEXUS assigns conditions reliably.

Eight of the 10 top-ranked genes were also measured by means of microarrays across different leaf locations of Z. mays (57). In this microarray experiment, all eight genes show an absolute log fold change of at least 1 between base and tip. Six of these eight genes show an absolute log fold change greater than four.

The two remaining genes, GRMZM2G331518 and AC213612.3_FG001, were not annotated on the microarray. The function of the top-ranked gene GRMZM2G331518 is not known. However, the
CONCLUSION
We have introduced DEXUS, an algorithm that identifies differentially expressed transcripts in RNA-Seq data with unknown conditions. DEXUS is appropriate for use with data from cohort, cross-sectional and nonrandomized controlled studies, where conditions are often unknown. In experiments with simulated and real-world data with known conditions, DEXUS successfully found differential expressed transcripts and conditions, although the conditions were corrected by means of the Benjamini–Hochberg procedure. It is plausible that that chloroplast also differs at different locations on the maize plant leaf. The GO term ‘cell wall’ was highly significant (\(P = 1.9e^{-30}\)) were significant. All \(P\)-values were corrected by means of the Benjamini–Hochberg procedure.

SUPPLEMENTARY DATA
Supplementary Data are available at NAR Online.

FUNDING
Funding for open access charge: Funds from the Institute of Bioinformatics, Johannes Kepler University Linz.

References
DEXUS: Identifying Differential Expression in RNA-Seq Studies with Unknown Conditions
— Supplementary Information —

Günter Klambauer, Thomas Unterthiner, and Sepp Hochreiter
Institute of Bioinformatics, Johannes Kepler University, Linz, Austria
CHAPTER 3. DEXUS

Contents

S1 Introduction 5
S2 Detection of Differential Expression With Many Conditions is Difficult 5
S3 The DEXUS Method 14
  S3.1 The Mixture of Negative Binomials Model 14
    S3.1.1 The Model 15
    S3.1.2 Parametrization of the Negative Binomial Distribution 15
    S3.1.3 Identifiability of the Model 17
  S3.2 Model Selection 17
    S3.2.1 Derivation of the EM Algorithm 17
    S3.2.2 Variance-To-Mean Ratio Approaching One 24
    S3.2.3 Update Rules 25
    S3.2.4 Initialization 26
    S3.2.5 Adjusting the Hyperparameter for the Size Parameter Prior 26
  S3.3 Calling Differentially Expressed Transcripts and I/NI Call 29
    S3.3.1 Data with Known Conditions 30
    S3.3.2 Data with Unknown Conditions: I/NI Call 31
  S3.4 Sensitivity Analysis of the Hyperparameter for the Dirichlet Prior 32
S4 Experiments 38
  S4.1 Evaluation Criteria for Simulated Data Sets 38
  S4.2 RNA-Seq Data with Known Conditions 38
    S4.2.1 Methods Compared 38
    S4.2.2 Simulated Data With Two Known Conditions 39
    S4.2.3 Simulated Data With Multiple Known Conditions 44
    S4.2.4 Real World Data with Two Known Conditions 46
  S4.3 RNA-Seq Data with Unknown Conditions 46
    S4.3.1 Methods compared 46
    S4.3.2 Simulated Data Sets with Unknown Conditions 47
    S4.3.3 The “Nigerian HapMap” data set 53
    S4.3.4 The “European HapMap” data set 54
    S4.3.5 The “Primate Liver” data set 56
    S4.3.6 The “Maize Leaves” data set 59
  S4.4 RNA-Seq Data with Subconditions 61
S5 Additional Information 64
  S5.1 Data set overview 64
  S5.2 Alternative Way to Derive the Update Rule for Mixture Weights 64
  S5.3 Posteriors in Our Framework 65
  S5.4 Maximum A Posterior for the Size Parameter of a Negative Binomial 65
  S5.5 Summary of the parameters and input values of DEXUS 66
    S5.5.1 Unknown Conditions 66
    S5.5.2 Known Conditions 67
  S5.6 Software Details of DEXUS and Experiments 67
List of Figures

S6 Performance Tables

S1 Density of the variance-to-mean ratio .......................... 6
S2 Probability mass of a negative binomial distribution and a BNB: example1 .... 9
S3 Probability mass of a negative binomial distribution and a BNB: example2 .... 10
S4 Probability mass of a negative binomial distribution and a BNB: example3 .... 11
S5 Probability mass of a negative binomial distribution and a BNB: example4 .... 12
S6 Probability mass of a negative binomial distribution and a mixture of 20 negative binomials: example5 ........................................ 13
S7 Probability mass of a negative binomial distribution and a mixture of two negative binomials: example6 ...................................... 14
S8 Graphical representation of the DEXUS model ................................ 20
S9 The bias of the maximum likelihood estimator of the size parameter ........... 27
S10 MSE of the maximum a posteriori estimator for different $\eta$ ................... 28
S11 MSE of the maximum likelihood estimator $r_{ML}$ for different means .......... 29
S12 The performance of DEXUS in terms of $AUC_{ROC}$ and $AUC_{PR}$ for different values of the hyperparameter $G$ for library size $10^6$ .................... 35
S13 The performance of DEXUS in terms of $AUC_{ROC}$ and $AUC_{PR}$ for different values of the hyperparameter $G$ for library size $10^7$ .................... 36
S14 The performance of DEXUS in terms of $AUC_{ROC}$ and $AUC_{PR}$ for different values of the hyperparameter $G$ for library size $10^8$ .................... 37
S15 Scatterplots of overdispersion and median read counts ........................... 40
S16 Heatmap of the normalized read counts of the twelve genes with the largest I/NI values for the “Nigerian HapMap” data set .......................... 53
S17 Expression values of the gene NLRP2 ...................................... 55
S18 Heatmap of the normalized read counts of the twelve genes with the largest I/NI values for the “European HapMap” data set ......................... 56
S19 Expression values of the gene T .............................................. 57
S20 Expression values of the gene PRSS21 ...................................... 57
S21 Expression values of the gene RASSF10 .................................... 59
S22 Heatmap of the normalized read counts of the ten genes with the largest I/NI values for the “Primate Liver” data set .......................... 60
S23 Heatmap of the normalized read counts of the ten genes with the largest DEXUS I/NI values for the “Maize Leafs” data set ........................... 61
S24 Heatmap of the normalized read counts of four exemplar genes of the “Primate Liver” data set that contain subconditions ............................. 63

List of Tables

S1 The performance in terms of area under ROC curve ($AUC_{ROC}$) for three different library sizes and different choices of the hyperparameter $G$. The displayed values are the means over 800 data sets, that is 100 data sets for each of the eight different settings for the number of replicates in the conditions .......................... 33
4 List of Tables

S2 The performance in terms of area under precision-recall curve (AUC
PR) for three different library sizes and different choices of the hyperparameter $G$. The displayed values are the means over 800 data sets, that is 100 data sets for each of the eight different settings for the number of replicates in the conditions. 34
S3 Performance of methods for two known conditions with 2 replicates and a library size of $10^6$, $10^7$, and $10^8$. 41
S4 Performance of methods for two known conditions with 6 replicates and a library size of $10^6$, $10^7$, and $10^8$. 42
S5 Performance of methods for two known conditions with 15 replicates and a library size of $10^6$, $10^7$, and $10^8$. 43
S6 Performance of methods for three known conditions with 2 replicates and a library size of $10^6$, $10^7$, and $10^8$. 44
S7 Performance of methods for three known conditions with 6 replicates and a library size of $10^6$, $10^7$, and $10^8$. 45
S8 Performance of methods for three known conditions with 15 replicates and a library size of $10^6$. 45
S9 Performance of methods for two known conditions with 2 replicates and a library size of $10^6$. 48
S10 Performance of methods for two known conditions with 6 replicates and a library size of $10^6$. 48
S11 Performance of methods for two known conditions with 15 replicates and a library size of $10^6$. 49
S12 Performance of DEXUS in terms of sensitivity and specificity with a library size of $10^6$. 50
S13 Performance of DEXUS in terms of sensitivity and specificity with a library size of $10^7$. 50
S14 Performance of DEXUS in terms of sensitivity and specificity with a library size of $10^8$. 51
S15 The performance of DEXUS in terms of sensitivity and specificity in detecting differential expression with unknown conditions for different fold change categories with library size of $10^6$. 51
S16 The performance of DEXUS in terms of sensitivity and specificity in detecting differential expression with unknown conditions for different fold change categories with library size of $10^7$. 52
S17 The performance of DEXUS in terms of sensitivity and specificity in detecting differential expression with unknown conditions for different fold change categories with library size of $10^8$. 52
S18 Significant GO terms of the differentially expressed genes of the “Nigerian HapMap” data set. 54
S19 Significant GO terms of the differentially expressed genes of the “European HapMap” data set. 58
S20 Overview of the data sets used in the manuscript. 64
S21 Results of DEXUS for unknown conditions with a library size of $10^6$. 69
S22 Results of DEXUS for unknown conditions with a library size of $10^7$. 70
S23 Results of DEXUS for unknown conditions with a library size of $10^8$. 71
S1 Introduction

This report gives supplementary information to the manuscript “DEXUS: Identifying Differential Expression in RNA-Seq Studies with Unknown Conditions”. The supplementary informations contain

- a result that with many conditions the detection of differential expression is only possible with a large number of samples and high coverage;
- a description and a motivation of the DEXUS model;
- a derivation of the DEXUS model selection algorithm;
- information on initialization and hyperparameter adjustment;
- results of additional experiments;
- methods for calling differentially expressed transcripts;
- further details on experiments;
- additional information on the DEXUS software, methods compared, data sets, and evaluation criteria.

Summary. Differentially expressed transcripts in RNA-Seq experiments with known conditions can be detected by current RNA-Seq methods. These methods test differential expression between two or more known conditions based on read counts per transcript. However in more general study designs some conditions are usually unknown, though genes may be differentially expressed between them. Current RNA-Seq methods cannot identify differentially expressed transcripts in data with unknown conditions. We suggest DEXUS, a statistical model based on finite mixture of negative binomial distributions to detect differential expression in studies with unknown conditions.

S2 Detection of Differential Expression With Many Conditions is Difficult

RNA-Seq data are usually represented as read counts per transcript. Read count data from technical replicates follow a Poisson distribution (Marioni et al. 2008). However read counts from biological replicates follow a negative binomial distribution (Anders and Huber 2010; Robinson et al. 2010; Hardcastle and Kelly 2010; Li and Tibshirani 2011; Wu et al. 2013), because the biological variation leads to overdispersion (Hansen et al. 2011). To confirm these findings, we analyzed RNA-Seq data sets using different normalizations. An example is the “European HapMap” data set (Montgomery et al. 2010), which contains RNA-Seq data of 60 individuals. After upper quartile normalization of these RNA-Seq data, we calculated the variance-to-mean ratio for each transcript. The density of this ratio is shown in Fig. S1. The vast majority of transcripts has a variance-to-mean ratio greater than one and, therefore, is in accordance with the negative binomial distribution. The value of this ratio is 1 for the Poisson distribution and smaller than 1 for the binomial distribution. The density of the variance-to-mean ratio for other RNA-Seq data sets
Figure S1: Density of the variance-to-mean ratio from RNA-Seq data of 60 individuals (Montgomery et al. 2010). Variance-to-mean ratios around one are characteristic for a Poisson distribution (blue), those smaller than one for a binomial distribution (purple), those larger than one for a negative binomial distribution (red). Thresholds are set by the test statistics of a Poisson test. The majority of transcripts has read counts that are in accordance with the negative binomial distribution.

Due to these findings, we assume that read counts of a set of biological replicates follows a **negative binomial distribution** with **mean expression level** \( \mu \). The negative binomial distribution with mean \( \mu \) and size \( r \) (representing the variance) is given by

\[
\text{NB}(x; \mu, r) = \frac{\Gamma(x + r)}{\Gamma(x + 1)\Gamma(r)} \left( \frac{\mu}{\mu + r} \right)^x \left( \frac{r}{\mu + r} \right)^r.
\]

\( \text{(S1)} \)

Other parametrizations and properties of the negative binomial distribution are discussed in Subsection S3.1.2.

We assume that biological replicates are generated under a particular **condition**, therefore a set of biological replicates corresponds to a particular condition. We define differential expression as different expression level between conditions (sets of replicates), where under each condition read counts are generated by a particular negative binomial distribution. A transcript is **differentially expressed if** (1) the mean expression levels \( \mu \) for different conditions are different and (2) samples are observed under at least two different conditions.

If the read count data of a transcript can be explained by one condition, i.e. one negative
binomial

\[ p(x) = \text{NB}(x; \mu, r) , \quad (S2) \]

then this transcript is not differentially expressed. If the read counts follow a mixture of negative binomials

\[ p(x) = \int p(\mu) \text{NB}(x; \mu, r) d\mu , \quad (S3) \]

then the transcript is differentially expressed. Here \( p(\mu) \) is the distribution of expression levels, each of which corresponds to a condition. Differential expression is identified by distinguishing a negative binomial from a mixture of negative binomials using the read count data.

An analytically tractable mixture of negative binomials is the beta negative binomial distribution (BNB):

\[ \text{BNB}(x; \iota, \kappa, r) = \int_0^\infty \frac{r}{(\mu + r)^2} B\left(\frac{\mu}{\mu + r}; \iota, \kappa\right) \text{NB}(x; \mu, r) d\mu , \quad (S4) \]

where \( B(\alpha; \iota, \kappa) \) is the density of the beta distribution with parameters \( \iota \) and \( \kappa \). Differential expression is identified by distinguishing a negative binomial from a beta negative binomial distribution (BNB) using the read count data.

We want to determine how many read counts and which coverages (given by the \( \mu \)'s) are necessary to distinguish a negative binomial from a BNB, i.e. to identify differential expression. Whether the read count data is better represented by a more complex than by a simpler model can be decided by means of the Bayesian Information Criterion (BIC):

\[ \text{BIC}_M = -2 \log L_M + l_M \log N , \quad (S5) \]

where \( L_M \) is the likelihood of the model \( M \), \( l_M \) is the number of parameters of the model \( M \), and \( N \) is the number of samples. The model with smaller BIC is more appropriate to represent the data.

If the difference between the BIC of the BNB and the BIC of the negative binomial model \( \text{BIC}_{\text{BNB}} - \text{BIC}_{\text{NB}} \) is negative, then BNB better represents the read counts and hints at differential expression. The number of parameters are \( l_{\text{BNB}} = 3 \) and \( l_{\text{NB}} = 2 \) for these models which should represent the read counts \( \{x_1, \ldots, x_N\} \) for \( N \) samples. Therefore detecting differential expression requires

\[ \text{BIC}_{\text{BNB}} - \text{BIC}_{\text{NB}} < 0 \quad (S6) \]

\[ \Leftrightarrow -2 \sum_{k=1}^N (\log \text{BNB}(x_k)) + 3 \log N + 2 \sum_{k=1}^N (\log \text{NB}(x_k)) - 2 \log N < 0 \]

\[ \Leftrightarrow 2 \frac{1}{N} \sum_{k=1}^N \log \left( \frac{\text{NB}(x_k)}{\text{BNB}(x_k)} \right) + \frac{\log N}{N} < 0 . \]
If averaging over data sets \( \{x_1, \ldots, x_N\} \) drawn from the BNB, the following equation holds:

\[
\frac{1}{N} \sum_{k=1}^{N} \log \left( \frac{\text{NB}(x_k)}{\text{BNB}(x_k)} \right) = \frac{1}{N} \sum_{k=1}^{N} \log \left( \frac{\text{NB}(x_k)}{\text{BNB}(x_k)} \right)
\]

\[= E \left( \log \left( \frac{\text{NB}(x_k)}{\text{BNB}(x_k)} \right) \right), \tag{S7} \]

where \( E \) is the expectation under the beta negative binomial distribution BNB. We have to use the expectation under the BNB, because the read counts are assumed to arise from a transcript that is differentially expressed. Note that \( E \left( \log \left( \frac{\text{NB}(x_k)}{\text{BNB}(x_k)} \right) \right) = -\text{KL}(\text{BNB}||\text{NB}) \), where KL is the Kullback-Leibler divergence. If averaging over data sets with \( N \) read counts, for differential expression following criterion is required:

\[
2 E \left( \log \left( \frac{\text{NB}(x_k)}{\text{BNB}(x_k)} \right) \right) + \log \frac{N}{N} < 0 \tag{S8}
\]

\[\iff \frac{N}{\log N} > \frac{1}{2 \text{KL}(\text{BNB}||\text{NB})}. \]

The Kullback-Leibler divergence is an asymmetric distance between two distributions. Thus, the more similar the distributions are to each other, the smaller is the Kullback-Leibler divergence, the more samples \( N \) are required to detect differential expression.

In the following, we compute the number of read counts that are required to detect differential expression if using the BIC criterion to discriminate between the negative binomial and the BNB. Fig. S2 shows the first example of a BNB with parameters \( \iota = 204, \kappa = 400, \) and \( r = 40 \) vs. a negative binomial with parameters \( \mu = 78.78 \) and \( r = 30.81 \). The Kullback-Leibler divergence of these two distributions is 0.0002. According to the BIC criterion \( N = 27,700 \) samples are required to identify differential expression, that is to detect that the read counts are from the BNB and not from the negative binomial. Fig. S3 shows a second example a BNB with parameters \( \iota = 40, \kappa = 40, \) and \( r = 40 \) vs. a negative binomial with parameters \( \mu = 40.89 \) and \( r = 13.07 \). In this case the Kullback-Leibler divergence is 0.02 and, therefore, \( N = 1,800 \) samples are required to identify differential expression. Fig. S4 shows a third example of a BNB with parameters \( \iota = 40, \kappa = 20, \) and \( r = 400 \) vs. a negative binomial with parameters \( \mu = 204.48 \) and \( r = 12.7 \). In this case the Kullback-Leibler divergence is 0.003 and, therefore, \( N = 1,300 \) samples are necessary to identify differential expression. Fig. S5 shows a fourth example of a BNB with parameters \( \iota = 10, \kappa = 10, \) and \( r = 400 \) vs. a negative binomial with parameters \( \mu = 435.56 \) and \( r = 4.43 \). In this case the Kullback-Leibler divergence is 0.03. Only \( N = 89 \) samples are required to identify differential expression, because the mean read counts are large. Large mean read counts means that the coverage is high.
Figure S2: Probability mass function of a negative binomial distribution with $\mu = 78.78$ and $r = 30.81$ and beta negative binomial distribution with $\iota = 204$, $\kappa = 400$, and $r = 40$. The inlay figure shows the distribution $p(\mu)$ of the mean read count $\mu$ which generates the BNB distribution. The Kullback-Leibler divergence of these two distributions is 0.0001. According to the BIC criterion $N=27,700$ samples are required to identify differential expression.
Figure S3: Probability mass function of a negative binomial distribution with $\mu = 40.89$ and $r = 13.07$ and beta negative binomial distribution with $\iota = 40$, $\kappa = 40$, and $r = 40$. The inlay figure shows the distribution $p(\mu)$ of the mean read count $\mu$ which generates the BNB distribution. The Kullback-Leibler divergence of these two distributions is 0.002. According to the BIC criterion $N=1,800$ samples are required to identify differential expression.
Figure S4: Probability mass function of a negative binomial distribution with $\mu = 204.48$ and $r = 12.7$ and beta negative binomial distribution with $\iota = 40$, $\kappa = 20$, and $\eta = 400$. The inlay figure shows the distribution $p(\mu)$ of the mean read count $\mu$ which generates the BNB distribution. The Kullback-Leibler divergence of these two distributions is 0.003. According to the BIC criterion $N=1,300$ samples are required to identify differential expression.
Figure S5: Probability mass function of a negative binomial distribution with $\mu = 435.56$ and $r = 4.43$ and beta negative binomial distribution with $\iota = 10$, $\kappa = 10$, and $r = 400$. The inlay figure shows the distribution $p(\mu)$ of the mean read count $\mu$ which generates the BNB distribution. The Kullback-Leibler divergence of these two distributions is 0.02. According to the BIC criterion $N=87$ samples are required to identify differential expression. The large mean read counts mean that the coverage is high. High coverage helps to detect differential expression.

If we want to determine whether a transcript is differentially expressed, we have to decide whether read counts are generated from a negative binomial or from a BNB. As shown above, this requires either a large number of samples or very high coverage (large $\mu$).

For generating the BNB we used a distribution $p(\mu)$ as mixing distribution which is unimodal (see inlay figures in Fig. S2 to Fig. S5). Other unimodal distributions $p(\mu)$ lead to similar results: it is difficult to distinguish a mixture distribution from a negative binomial. If $p(\mu)$ is a weighted sum of delta distributions, then we obtain a finite mixture of negative binomials, that is a finite number of conditions:

$$p(x) = \sum_{i=1}^{n} \alpha_i \text{NB}(x; \mu_i, r_i).$$

(S9)

However, if the number of conditions is large compared to the number of samples, then it is still difficult to distinguish between a mixture model and a negative binomial using only the read counts. Thus, it is difficult to detect differential expression if the number of conditions is large compared to the number of samples.

In the following, we investigate whether a finite mixture of negative binomials can be distinguished from a single negative binomial distribution. Trivially, a mixture with only one component cannot be distinguished from a negative binomial. If one component of the mixture is dominant, then the mixture is close to a negative binomial. To avoid these trivial cases, we require that, under each condition, not more than half of the read counts are generated. The second trivial case is that
all mixture components are identical, where the mixture is a negative binomial. If the means of the component distributions are identical or are locally accumulated, then the mixture is close to a negative binomial. To avoid these cases, we require that the means of the component distributions are placed equidistantly within a particular range of read counts. Concluding, we require $\alpha_i \leq 0.5$ and equidistantly distributed $\mu_i$ in a range to avoid trivial degenerated cases for which the mixture is close to a negative binomial. We will later use the $\alpha_i$ and $\mu_i$ to define an informative/non-informative call for a transcript, see Subsection S3.3. We choose the range of read counts between 0 and 150 similar to the example depicted in Fig. S2. First we construct a mixture of 10 negative binomials, where $r_i$ are set to 50 and the mixture weights $\alpha_i$ follow a unimodal distribution. As shown in Fig. S6, the mixture and the negative binomial are hard to distinguish. Next we construct a mixture of two negative binomials. Now the mixture can be distinguished from the negative binomial as shown in Fig. S7. Note that all parameters are optimized to make the mixture as close as possible to the negative binomial. In contrast to the negative binomial, the mixture is a binomial distribution. Further the probability mass at the tails of the negative binomial is smaller than the mass of the mixture. This example shows that a mixture can be distinguished from a single negative binomial, if there are few conditions compared to the number of read counts.

Figure S6: Probability mass function of a negative binomial distribution with $\mu = 75.22$ and $r = 8.87$ and a mixture of negative binomial distributions with mean parameters $\mu_i$ equidistantly in the range $[0, 150]$, $r_i$ set to 50, and non-zero mixture weights $\alpha_i$. The inlay figure shows the probability mass functions of the 10 negative binomial distributions of the mixture. The Kullback-Leibler divergence of these two distributions is 0.007.
Figure S7: Probability mass function of a negative binomial distribution with $\mu = 75.07$ and $r = 15.08$ and a mixture of two negative binomial distributions with mean parameters equidistantly in $[0, 150]$ that is $(\mu_1, \mu_2) = (50, 100)$, with $(r_1, r_2) = (50, 50)$, and with mixture weights $(\alpha_1, \alpha_2) = (0.5, 0.5)$. The inlay figure shows the probability mass functions of the two negative binomial distributions of the mixture. The Kullback-Leibler divergence of these two distributions is 0.49.

Therefore we assume that the number of conditions (the number of sets of replicates) is small compared to the number of samples. We will demonstrate that under this assumption differentially expressed transcripts can be detected. As we show in the experiments, the detection of differential expression is more reliable with more samples in each of the conditions and with larger differences of the mean expression levels of the conditions.

S3 The DEXUS Method

In the first subsection, we introduce and motivate the DEXUS model, which is a finite mixture of negative binomials (as motivated in previous section). In the second subsection, we explain how DEXUS selects a model using read count data. Model selection is based on a Bayesian framework for maximizing the parameter posterior via an expectation maximization (EM) algorithm. The next subsection focuses on how to determine whether transcripts are differentially expressed. In the last subsection we investigate the sensitivity of a hyperparameter of DEXUS.

S3.1 The Mixture of Negative Binomials Model

In this subsection, we introduce and motivate the DEXUS model. Read counts per sample are modeled across samples for a gene, an exon, or a transcript. The following Subsection S3.1.1 introduces the finite mixture of negative binomial distributions model as motivated in previous
Section S2. Subsection S3.1.2 explains why we have chosen a particular parametrization of the negative binomial distribution. In the next Subsection S3.1.3 we show that the model is identifiable which is essential to infer parameters and to detect differential expression.

S3.1.1 The Model

As motivated in Section S2, read counts are distributed according to a finite mixture negative binomial distributions. Each mixture component corresponds to a condition, that is, the read counts are generated under this condition. A transcript is differentially expressed if read counts are generated under at least two different conditions with different expression levels $\mu$.

If $\alpha_i$ is the probability of being in condition $i$, then read count $x$ is distributed according to:

$$ p(x) = \sum_{i=1}^{n} \alpha_i \text{NB}(x; \mu_i, r_i), $$

where $\text{NB}(x; \mu_i, r_i)$ is probability mass function of the negative binomial distribution with mean $\mu_i$ and size $r_i$. The $\alpha_i \geq 0$ are the non-negative mixture weights of the mixture model and sum to one: $\sum_{i=1}^{n} \alpha_i = 1$. The DEXUS model is a finite mixture of negative binomial distributions. As motivated in Section S2, we assume that the number of conditions $n$ is smaller than the number of samples $N$: $n < N$.

S3.1.2 Parametrization of the Negative Binomial Distribution

We use the $(\mu, \phi)$-parametrization of the negative binomial distribution (also called the mean-size-parametrization). The $(p, r)$-parametrization is the standard way to parametrize the negative binomial distribution. In the standard interpretation, $p$ is the probability of success with $p \in [0, 1]$ and $r$ is the number of failures with $r \in \mathbb{N}$. The probability mass function of the negative binomial in the $(p, r)$-parametrization is

$$ \text{NB}_{pr}(x; p, r) = \frac{(x + r - 1)!}{x! (r - 1)!} (1 - p)^{r - 1} p^x, $$

where $x$ is the number of successes until $r$ failures are observed (note that the last observation must be a failure). The variance of the negative binomial is

$$ \sigma^2 = \frac{pr}{(1 - p)^2}. $$

We chose a parametrization that includes $\mu$ because for RNA-Seq data with read counts per transcript, the mean read count is an important information. The mean is the expected or noise-free read count for a given condition and allows to determine whether transcripts are differentially expressed between conditions. Therefore in RNA-Seq applications, the negative binomial is re-parametrized using the mean parameter $\mu \in \mathbb{R}^+$ instead of the probability $p$. Furthermore in RNA-Seq applications, the overdispersion parameter $\phi \in \mathbb{R}^+$ is of interest to capture technical and biological variance which allows assessing the data quality. The overdispersion parameter measures how far the variance of the negative binomial exceeds the variance of a Poisson distribution, for which the variance is equal to the mean. These two parameters lead to the $(\mu, \phi)$-parametrization
of the negative binomial. The overdispersion parameter $\phi$ and the size parameter $r$ have a very simple relationship: $\phi = 1/r$, i.e. the overdispersion is the inverse of the size parameter. The relationship between the parametrization $(\mu, \phi)$ and $(p, r)$ is:

$$\mu = \frac{pr}{1 - p} \Rightarrow p = \frac{\mu}{\mu + r} \quad \phi = \frac{1}{r} \Rightarrow r = \phi^{-1} \quad (S13)$$

We will choose $r$ instead of $\phi$ in order to define a prior on $r$ in a Bayesian framework. This prior gives larger overdispersions higher probabilities, which is essential to improve the parameter estimator for small sample sizes (see Subsection S3.2.1).

To represent all overdispersions $\phi$ and to perform model selection in a continuous space, real positive values of $r$ are required. The definition Eq. (S11) can be generalized to $r \in \mathbb{R}^+$ by using the $\Gamma$-function instead of the factorial. Using positive real $r$, the probability mass function of the negative binomial for the $(\mu, r)$-parametrization is

$$\text{NB}(x; \mu, r) = \frac{\Gamma(x + r)}{\Gamma(x + 1)\Gamma(r)} \left( \frac{\mu}{\mu + r} \right)^x \left( \frac{1}{\mu + r} \right)^r . \quad (S14)$$

The variance of the negative binomial with the $(\mu, \phi)$-parametrization or the $(\mu, r)$-parametrization is

$$\sigma^2 = \frac{\mu + r}{\mu + \frac{\mu + r}{2}} = \mu + \frac{1}{r} \mu^2 = \mu + \phi \mu^2 . \quad (S15)$$

If DEXUS is applied to data with known conditions, we require an estimator of $(\mu, r)$ of a negative binomial distribution. In particular we require this estimator for the initialization of the EM algorithm in Subsection S3.2.4. We use the maximum likelihood estimator. Given a data set $x = (x_1, \ldots, x_N)$ of counts of $N$ samples, the maximum likelihood estimators for the $(\mu, r)$-parametrization are as follows:

- The maximum likelihood estimator $\mu_{ML}$ for $\mu$ is

$$\mu_{ML} = \frac{1}{N} \sum_{k=1}^{N} x_k . \quad (S16)$$

- A closed form for the maximum likelihood estimator $r_{ML}$ for $r$ does not exist. However following equation can be solved for $r_{ML}$:

$$\sum_{k=1}^{N} \psi(x_k + r_{ML}) - N \psi(r_{ML}) + N \log \left( \frac{r_{ML}}{r_{ML} + \frac{r_{ML}}{N} \sum_{k=1}^{N} x_k} \right) = 0 , \quad (S17)$$

where $\psi$ is the digamma function. The solution can be obtained numerically.

The estimator $(\mu_{ML}, r_{ML})$ is asymptotically unbiased and efficient. For finite sample size, however, neither the bias nor the variance of the estimator $r_{ML}$ exists, because for data whose mean exceeds the variance $r$ tends to infinity (Anscombe 1950).
S3.1.3 Identifiability of the Model

Finite mixtures of non-degenerate negative binomial distributions are identifiable (Yakowitz and Spragins 1968). For identifiable mixtures, Eq. (3) in (Yakowitz and Spragins 1968) and the text thereafter states that from

\[ \sum_{i=1}^{n} \alpha_i \text{NB}(x; \mu_i, r_i) = \sum_{i=1}^{n} \alpha'_i \text{NB}(x; \mu'_i, r'_i). \]  

(S18)

follows

\[ \alpha_i = \alpha'_i \]  
\[ \mu_i = \mu'_i \]  
\[ r_i = r'_i. \]  

(S19)

We assumed that the components are sorted (avoids ambiguities through permutations of the components) and that the distributions of the components are mutually different (avoids ambiguous splitting of one component into more).

Identifiability is required for the maximum likelihood estimator to be consistent. Consistency means that a parameter estimator converges with more data points to the true parameter values. Since the parameter space will be made compact, the mixture model is continuous in its parameters, and the log mixture distribution can be bounded, the maximum likelihood estimator for the mixture of negative binomials is consistent. Note that below we will introduce an upper bound \( r_{\text{max}} \) for the size parameter \( r \) and a lower bound \( \mu_{\text{min}} \) for the mean parameter of the negative binomial distributions.

More importantly, identifiability of the mixture of negative binomials guarantees that differentially expressed transcripts can be detected if sufficiently many read counts are available.

S3.2 Model Selection

In the next Subsection S3.2.1, DEXUS’ expectation maximization (EM) algorithm for model selection is derived. We first define the Bayesian framework, then chose appropriate priors for the parameters, then derive a bound on the parameter posterior using the chosen priors, and then use this bound to derive the E-step and the M-step of the EM algorithm. This subsection is one of the central parts of this supplementary. The following Subsection S3.2.2 describes the case when the variance-to-mean ratio of one negative binomial is approaching one and converges toward a Poisson distribution. The next Subsection S3.2.3 summarizes DEXUS update rules for the iterative EM algorithm. Then Subsection S3.3 describes the initialization for the DEXUS model selection algorithm. The final Subsection S3.2.5 shows how the hyperparameter for the size parameter prior is adjusted depending on the number of samples.

S3.2.1 Derivation of the EM Algorithm

In a Bayes framework for model selection, \( \alpha = (\alpha_1, \ldots, \alpha_n) \), \( \mu = (\mu_1, \ldots, \mu_n) \), and \( r = (r_1, \ldots, r_n) \) are considered as random variables, thus, \( p(x) \) in Eq. (S10) becomes a conditional
probability \( p(x \mid \alpha, \mu, r) \), i.e. the likelihood that read count \( x \) has been produced by the model with parameters \( \alpha \), \( \mu \), and \( r \). The expectation maximization (EM) algorithm (Dempster et al. 1977) minimizes an upper bound on the negative log-posterior of the parameters. The parameter posterior of \( \alpha \), \( \mu \), and \( r \) is given by:

\[
p(\alpha, \mu, r \mid x) = \frac{p(x \mid \mu, r, \alpha) p(\alpha) p(r) p(\mu)}{\int p(x \mid \mu, r, \alpha) p(\alpha) p(r) d\alpha dr d\mu} = \frac{1}{c(x)} p(x \mid \mu, r, \alpha) p(\alpha) p(r) p(\mu),
\]

where we assumed that the priors for \( \alpha \), \( \mu \) and \( r \) are independent of each other.

For deriving an upper bound on the log posterior as required by the EM algorithm, we deduce the following inequality for one sample \( x \) by introducing variables \( \hat{\alpha}_i \) with \( \sum_{i=1}^{n} \hat{\alpha}_i = 1 \):

\[
- \log p(\mu, r, \alpha \mid x) = - \log \sum_{i=1}^{n} \frac{\hat{\alpha}_i}{\alpha_i} \alpha_i \text{NB}(x; \mu_i, r_i) - \log p(\alpha) - \log p(\mu) - \log p(r) + \log(c(x))
\]

\[
\leq - \sum_{i=1}^{n} \hat{\alpha}_i \log \left( \frac{\alpha_i}{\hat{\alpha}_i} \text{NB}(x; \mu_i, r_i) \right) - \log p(\alpha) - \log p(\mu) - \log p(r) + \log(c(x)) \quad (*)
\]

\[
= - \sum_{i=1}^{n} \hat{\alpha}_i \log (\alpha_i \text{NB}(x; \mu_i, r_i)) - \log p(\alpha) - \log p(\mu) - \log p(r)
\]

\[
+ \sum_{i=1}^{n} \hat{\alpha}_i \log \hat{\alpha}_i + \log(c(x))
\]

where \( c(x) \) is independent of the parameters \( \alpha \), \( \mu \) and \( r \). We applied Jensen’s inequality to obtain the line ending with the \((*)\)-sign.

To derive an EM algorithm, we have to choose appropriate priors for the mixture weights \( p(\alpha) \), the overdispersion parameters \( p(r) \), and the means \( p(\mu) \).

**Dirichlet Prior on Mixture Weights.** In the DEXUS model, the prior \( p(\alpha) \) on the mixture weights \( \alpha \) should incorporate the prior knowledge that most transcripts are not differentially expressed into the model. The prior should represent the null hypothesis that the read counts are generated under only one condition. Such a prior enforces a low false discovery rate at the detection of differentially expressed transcripts because, for ambiguous data, read counts are not explained by differential expression.

A Dirichlet prior with parameters \( \gamma \) is well suited to represent this null hypothesis:

\[
p(\alpha) = D(\alpha; \gamma) = b(\gamma) \prod_{i=1}^{n} \alpha_i^{\gamma_i - 1},
\]

where \( \alpha \) an the \( n \)-dimensional probability vector, i.e. \( \alpha_1, \ldots, \alpha_n \geq 0 \) and \( \sum_{i=1}^{n} \alpha_i = 1 \). Each component \( \alpha_i \) is distributed according to a beta distribution with the following properties:

\[
\text{mean}(\alpha_i) = \frac{\gamma_i}{\gamma_s},
\]

(23)
\[
\text{mode}(\alpha_i) = \frac{\gamma_i - 1}{\gamma_s - n}, \quad (S24)
\]
\[
\text{var}(\alpha_i) = \frac{\gamma_i (\gamma_s - \gamma_i)}{\gamma_s^2 (\gamma_s + 1)}, \quad (S25)
\]
where we set
\[
\gamma_s = \sum_{i=1}^{n} \gamma_i. \quad (S26)
\]

To express our prior knowledge that most genes are not differentially expressed, we set \(\gamma_1 \gg \gamma_i\) (for \(i > 1\)). This setting of the values ensures that the model tries to explain the data by one mixture component, that is a single negative binomial distribution. For \(i \neq 1\) we set \(\gamma_i = 1\) in order to enforce a mode at zero. Therefore for most drawn \(\alpha_i\), the component \(\alpha_i\) is zero for \(i \neq 1\). This reduces the number of hyperparameters to just one, which is \(\gamma_1\).

We set the parameter \(\gamma\) of the Dirichlet prior to
\[
\gamma = (1 + G, 1, ..., 1). \quad (S27)
\]
This simplifies the setting of the hyperparameters to one hyperparameter \(G\). In all experiments we set \(G = 1\).

**Truncated Exponential Priors on the Size Parameter.** The maximum likelihood solution \(r_{ML}\), given in Eq. (S17), for the negative binomial tends to overestimate the true size parameter \(r\) (Piegorsch 1990). Therefore, we introduce a prior \(p(r_i)\) on \(r_i\) for each condition \(i\), which prefers small \(r_i\)-values. As prior for \(r_i\) we use an exponential distribution with parameter \(\eta\):
\[
p(r_i) = \text{EXP}(r_i) = \eta e^{-\eta r_i}. \quad (S28)
\]
Thus, the prior of \(r\) is \(\text{EXP}(r) = \prod_{i=1}^{n} \text{EXP}(r_i)\).

We truncate this exponential distribution at \(r_{max}\) in order to enforce a lower bound of \(1/r_{max}\) on the overdispersion. This bound is important to make the parameter space compact and, therefore, to ensure that the maximum likelihood estimator is consistent. Thus our prior is actually a truncated exponential distribution.

Furthermore, the bound ensures a certain minimal overdispersion for each gene which is another prior knowledge that we include for model selection. We follow Anders and Huber (2010) in their implementation of \text{DESeq} and set \(r_{max} = 13.0\). Truncating the exponential distribution changes the distribution only by a normalizing constant. In order to keep the notation uncluttered, we derive the algorithm without denoting this normalizing constant. Note that nothing changes in the derivation except that we have the constraint that \(r_i < r_{max}\) for all \(i\).
**Uniform Prior for the Mean Parameter.** If, in one condition, all read counts are close to zero (the transcript is not present), the estimate of the mean of the negative binomial would not converge. The reason is that the parameter space is not compact as $\mu = 0$ is excluded. A compact parameter space is required to ensure consistency of the maximum likelihood estimator.

To make the parameter space compact, we introduce a lower bound $\mu_{\text{min}}$ on $\mu_i$. We implement this bound by a uniform prior on $\mu_i$ on the interval $[\mu_{\text{min}}, \max_k x_k]$. In all experiments we used $\mu_{\text{min}} = 0.5$.

**Graphical Representation of the Model** A graphical representation of the model including the parameters and hyperparameters is given in Figure S8.

![Graphical representation of the DEXUS model](image.png)

Figure S8: Graphical representation of the DEXUS model as directed acyclic graph. Squares represent hyperparameters, white circles model parameters and black circles given data.

**Bound on the Posterior Using Priors and All Data.** Using these priors, the upper bound in Eq. (S21) on the posterior becomes:

$$
- \sum_{i=1}^{n} \hat{\alpha}_i \log(\alpha_i \text{NB}(x; \mu_i, r_i)) - \log D(\alpha) - \log \text{EXP}(r) + \sum_{i=1}^{n} \hat{\alpha}_i \log \hat{\alpha}_i + \log(c(x)).
$$

(S29)

The prior on $\mu$ is constant and, therefore, is absorbed in $c(x)$. During the EM algorithm $\mu$ values smaller than $\mu_{\text{min}}$ are projected back to $\mu_{\text{min}}$, which corresponds to the maximum a posteriori value given the uniform prior.

The likelihood for the whole data set $\{x_1, \ldots, x_N\}$ of $N$ samples is the product of the likelihoods for data points $x_k$. Thus, the log likelihood is a sum over the log likelihoods for data points $x_k$. That means the inequality in Eq. (S21) can be applied to each single data point $x_k$, where $\hat{\alpha}_i$ is replaced by $\tilde{\alpha}_{ik}$. Therefore, the upper bound $B$ on the scaled (by $\frac{1}{N}$) negative log-posterior for
The DEXUS Method

A data set \( \{x_1, \ldots, x_N\} \) is:

\[
B = -\frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{n} \tilde{\alpha}_{ik} \log \left( \alpha_i \text{NB}(x_k; \mu_i, r_i) \right) - \frac{1}{N} \log D(\alpha) - \frac{1}{N} \log \text{EXP}(r) \\
+ \frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{n} \tilde{\alpha}_{ik} \log \tilde{\alpha}_{ik} + \frac{1}{N} \sum_{k=1}^{N} \log(c(x_k)) .
\] (S30)

In above formula, the log of the negative binomial probability mass function is

\[
\log \text{NB}(x; \mu_i, r_i) = \log((x + r_i - 1)! - \log(x!) - \log((r_i - 1)!)) + x \log \left( \frac{\mu_i}{\mu_i + r_i} \right) + r_i \log \left( \frac{r_i}{\mu_i + r_i} \right).
\] (S31)

**E-step: Optimization w.r.t. Posterior Estimates.** For the E-step the upper bound \( B \) Eq. (S30) on the negative log posterior must be minimized with respect to \( \tilde{\alpha}_{ik} \). The condition

\[
\sum_{i=1}^{n} \tilde{\alpha}_{ik} = 1
\] (S32)

must hold to ensure that the posterior mixture weights are an \( n \)-dimensional probability vector. The bound \( B \) ensures that the \( \tilde{\alpha}_{ik} \) are positive via the log.

The Lagrangian using only terms in \( \tilde{\alpha}_{ik} \) is

\[
L = -\frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{n} \tilde{\alpha}_{ik} \log \left( \alpha_i \text{NB}(x_k; \mu_i, r_i) \right) \\
+ \frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{n} \tilde{\alpha}_{ik} \log \tilde{\alpha}_{ik} - \lambda_k \left( \sum_{i=1}^{n} \tilde{\alpha}_{ik} = 1 \right),
\] (S33)

where \( \lambda_k \) is the Lagrange multiplier for the \( k \)-th constraint given by Eq. (S32).

For the optimal value, the derivative of the Lagrangian \( L \) with respect to to \( \tilde{\alpha}_{ik} \) must be zero:

\[
\frac{\partial L}{\partial \tilde{\alpha}_{ik}} = -\frac{1}{N} \log \left( \alpha_i \text{NB}(x_k; \mu_i, r_i) \right) + \frac{1}{N} \left( \log \tilde{\alpha}_{ik} + 1 \right) - \lambda_k = 0.
\] (S34)

This equation can be solved for \( \tilde{\alpha}_{ik} \):

\[
\tilde{\alpha}_{ik} = \alpha_i \text{NB}(x_k; \mu_i, r_i) e^{N\lambda_k-1}.
\] (S35)

Exponentiation during solving the equation ensures positive \( \tilde{\alpha}_{ik} \). Summing over \( i \) from 1 to \( n \) gives

\[
e^{N\lambda_k-1} = \frac{1}{\sum_{i=1}^{n} \alpha_i \text{NB}(x_k; \mu_i, r_i)}.
\] (S36)
Inserting this equation into Eq. (S35) gives:

\[ \tilde{\alpha}_{ik} = \frac{\alpha_i \text{NB}(x_k; \mu_i, r_i)}{\sum_{i=1}^{n} \alpha_i \text{NB}(x_k; \mu_i, r_i)}. \]  \hspace{1cm} (S37)

Note that the optimal \( \tilde{\alpha}_{ik} \) is the posterior of condition \( i \) in the mixture model given data point \( x_k \). \( \alpha_i = p(i) \) is the prior for condition \( i \) and \( p(x_k \mid i) = \text{NB}(x_k; \mu_i, r_i) \) is the likelihood for condition \( i \), and \( \tilde{\alpha}_{ik} = p(i \mid x_k) \) the posterior for condition \( i \). \( \alpha_i \) can be decomposed into \( \alpha_{ik} \):

\[ \alpha_i = p(i) = \sum_{x=0}^{\infty} p(i, x) = \sum_{x=0}^{\infty} p(i \mid x) p(x) = \mathbb{E}_{p(x)} (p(i \mid x)) \]  \hspace{1cm} (S38)

\[ \approx \frac{1}{N} \sum_{k=1}^{N} p(i \mid x_k) = \frac{1}{N} \sum_{k=1}^{N} \alpha_{ik}. \]

Therefore, we estimate \( \alpha_i \) by \( \hat{\alpha}_i \):

\[ \hat{\alpha}_i = \frac{1}{N} \sum_{k=1}^{N} \tilde{\alpha}_{ik}. \]  \hspace{1cm} (S39)

This estimate \( \hat{\alpha}_i \) is used in the update rules below.

**M-step: Optimization w.r.t. Mean.** In the M-step, we minimize the upper bound \( B \) Eq. (S30) on the negative log posterior with respect to all parameters \( \mu, r, \) and \( \alpha \).

First we minimize \( B \) with respect to \( \mu_i \), where only terms depending on \( \mu_i \) are considered:

\[ \min_{\mu_i} - \frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{n} \tilde{\alpha}_{ik} \log (\alpha_i \text{NB}(x_k; \mu_i, r_i)). \]  \hspace{1cm} (S40)

The derivative of the log negative binomial distribution Eq. (S31) with respect to \( \mu_i \) is

\[ \frac{\partial \log \text{NB}(x_k; \mu_i, r_i)}{\partial \mu_i} = x_k \left( \frac{\mu_i + r_i}{\mu_i} \right) \left( \frac{r_i}{(\mu_i + r_i)^2} \right) - r_i \left( \frac{\mu_i + r_i}{r_i} \right) \left( \frac{r_i}{(\mu_i + r_i)^2} \right) \]

\[ = x_k \left( \frac{r_i - \mu_i r_i}{\mu_i + r_i} \right) = \left( \frac{x_k - \mu_i}{\mu_i + r_i} \right) \left( \frac{1}{\mu_i + r_i} \right). \]  \hspace{1cm} (S41)

The derivative of the upper bound \( B \) Eq. (S30) with respect to \( \mu_i \) is

\[ \frac{\partial B}{\partial \mu_i} = - \frac{1}{N} \sum_{k=1}^{N} \tilde{\alpha}_{ik} \left( \frac{x_k - \mu_i}{\mu_i + r_i} \right) \left( \frac{1}{\mu_i + r_i} \right). \]  \hspace{1cm} (S42)

At the minimum, this derivative must be zero. Setting the derivative equal to zero and solving for \( \mu_i \) gives the update rule

\[ \mu_i = \frac{\sum_{k=1}^{N} \tilde{\alpha}_{ik} x_k}{N \tilde{\alpha}_i}. \]  \hspace{1cm} (S43)

The update is simply a weighted mean, where the weight \( \tilde{\alpha}_{ik} \) is the posterior of condition \( i \) for data \( x_k \). \( \tilde{\alpha}_{ik} \) reflects how likely \( x_k \) was generated under condition \( i \). Note that the update does not depend on other parameters. Since we have introduced a uniform prior of \( \mu_i \) on the compact interval \([\mu_{\text{min}}, \max x_k x_k]\), \( \mu_i \) that exceed this interval after being updated are projected back to it.
M-step: Optimization w.r.t. Size Parameter  Secondly we minimize $B$ with respect to $r_i$. Only terms of $B$ that depend on $r_i$ are considered:

$$
\min_{r_i} - \frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{n} \tilde{\alpha}_{ik} \log (\alpha_i \text{NB}(x_k; \mu_i, r_i)) - \frac{1}{N} \log \text{EXP}(r) \quad (S44)
$$

The derivative of the log negative binomial Eq. (S31) with respect to $r_i$ is:

$$
\frac{\partial}{\partial r_i} \log \text{NB}(x_i; \mu_i, r_i) = \psi(x + r_i) - \psi(r_i) - \frac{x - \mu_i}{\mu_i + r_i} - \log \left( \frac{r_i}{\mu_i + r_i} \right), \quad (S45)
$$

where $\psi(x)$ is the digamma function. The derivative of the log exponential with respect to $r_i$ is

$$
\frac{\partial}{\partial r_i} \log \text{EXP}(r) = \frac{\partial}{\partial r_i} \log \left( \prod_{i=1}^{n} \eta e^{-\eta r_i} \right) = - \eta. \quad (S46)
$$

The derivative of the upper bound $B$ Eq. (S30) with respect to $r_i$ is

$$
\frac{\partial B}{\partial r_i} = - \frac{1}{N} \sum_{k=1}^{N} \left[ \tilde{\alpha}_{ik} \psi(x_k + r_i) - \psi(r_i) \right] - \frac{\mu_i}{\mu_i + r_i} - \log \left( \frac{r_i}{\mu_i + r_i} \right) + \frac{1}{N} \eta. \quad (S47)
$$

This derivative depends on the parameter $\mu_i$, where we have to use the new value for $\mu_i$: $\mu_i = (\sum_{k=1}^{N} \tilde{\alpha}_{ik} x_k) / (N \hat{\alpha}_i)$. The term $\sum_{k=1}^{N} \tilde{\alpha}_{ik} (x_k - \mu_i) / (\mu_i + r_i)$ is zero, because $\sum_{k=1}^{N} \tilde{\alpha}_{ik} x_k = N \hat{\alpha}_i \mu_i$ according to the $\mu_i$ update Eq. (S43) and $\mu_i \sum_{k=1}^{N} \tilde{\alpha}_{ik} = N \hat{\alpha}_i \mu_i$.

At the minimum this derivative must be zero, which leads to

$$
\sum_{k=1}^{N} \tilde{\alpha}_{ik} \psi(x_k + r_i) - \psi(r_i) \sum_{k=1}^{N} \hat{\alpha}_{ik} + \log \left( \frac{r_i}{\mu_i + r_i} \right) \sum_{k=1}^{N} \hat{\alpha}_{ik} - \eta = 0. \quad (S48)
$$

Inserting the new value for $\mu_i$ into this equation results in

$$
\sum_{k=1}^{N} \tilde{\alpha}_{ik} \psi(x_k + r_i) - N \hat{\alpha}_i \psi(r_i) + N \hat{\alpha}_i \log \left( \frac{r_i \hat{\alpha}_i}{\frac{1}{N} \sum_{k=1}^{N} \tilde{\alpha}_{ik} x_k + r_i \hat{\alpha}_i} \right) - \eta = 0. \quad (S49)
$$

This equation cannot be solved for $r_i$ in a closed form. However, the parameter $r_i$ can be obtained by solving this equation numerically. Since it is an equation in one variable, we use a simple bisection procedure.

Without a prior on $r_i$, the term $- \frac{1}{N} \log \text{EXP}(r)$ vanishes in Eq. (S44) and the equation which must be solved for $r_i$ becomes

$$
\sum_{k=1}^{N} \tilde{\alpha}_{ik} \psi(x_k + r_i) - N \alpha_i \psi(r_i) + N \hat{\alpha}_i \log \left( \frac{r_i}{\mu_i + r_i} \right) = 0. \quad (S50)
$$

Only the term $- \eta$ vanishes in comparison to Eq. (S49), i.e., the equation obtained with the prior. Since we have introduced an exponential prior of $r_i$ on the compact interval $[0, r_{\text{max}}]$, $r_i$ for which $r_i > r_{\text{max}}$ holds after being updated are set equal to $r_{\text{max}}$. 
M-step: Optimization of Mixture Weights  

Thirdly we minimize $B$ with respect to $\alpha$ with the constraint that the $\alpha_i$ sum to 1. Only terms depending on $\alpha$ are considered:

$$
\begin{align*}
\min_\alpha & - \frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{n} \tilde{\alpha}_{ik} \log \alpha_i - \frac{1}{N} \log D(\alpha) \\
\text{s.t.} & \sum_{i=1}^{n} \alpha_i = 1 .
\end{align*}
$$

The objective ensures that $\alpha_i > 0$. The Lagrangian is

$$
L = - \frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{n} \tilde{\alpha}_{ik} \log \alpha_i - \frac{1}{N} \log p(\alpha) + \rho \left( \sum_{i=1}^{n} \alpha_i - 1 \right)
$$

where $\rho$ is the Lagrange multiplier for the constraint. The solution requires that the derivative of $L$ with respect to $\alpha_i$ is zero:

$$
\frac{\partial L}{\partial \alpha_i} = - \frac{1}{N} \sum_{k=1}^{N} \tilde{\alpha}_{ik} \frac{1}{\alpha_i} - \frac{1}{N} (\gamma_i - 1) \frac{1}{\alpha_i} + \rho = 0 .
$$

Multiplying this equation by $\alpha_i$ gives

$$
- \frac{1}{N} \sum_{k=1}^{N} \tilde{\alpha}_{ik} - \frac{1}{N} (\gamma_i - 1) + \rho \alpha_i = 0 .
$$

Summation over $i$ leads to

$$
1 + \frac{1}{N} (\gamma_s - n) = \rho ,
$$

where $\gamma_s = \sum \gamma_i$. Inserting this expression for $\rho$ into Eq. (S54) gives

$$
- \frac{1}{N} \sum_{k=1}^{N} \tilde{\alpha}_{ik} = \frac{1}{N} (\gamma_i - 1) + \left( 1 + \frac{1}{N} (\gamma_s - n) \right) \alpha_i = 0 .
$$

Solving this equation for $\alpha_i$ using $\hat{\alpha}_i = \frac{1}{N} \sum_{k=1}^{N} \tilde{\alpha}_{ik}$ leads to the update formula for $\alpha_i$:

$$
\alpha_i = \frac{\hat{\alpha}_i + \frac{1}{N} (\gamma_s - 1)}{1 + \frac{1}{N} (\gamma_s - n)} .
$$

**S3.2.2 Variance-To-Mean Ratio Approaching One**

The variance-to-mean ratio of negative binomials is bounded from below by one, since $\sigma^2/\mu = (\mu + \mu^2/r)/\mu = 1 + \mu/r$. For data with variance-to-mean ratio smaller than one, the size parameter $r$ increases continuously during the EM algorithm. For numerical stability of the algorithm,
we approximate the distribution of the negative binomial for large values of \( r \) with a Poisson distribution. As mentioned before we use a truncated exponential function as a prior on \( r \) for which \( r < r_{\text{max}} \) (default \( r_{\text{max}} = 13.0 \). If the \( r \)-update leads to an \( r \) larger equal \( r_{\text{max}} \) and \( r_{\text{max}} \) is set to a value higher than 10,000, then we switch to the Poisson distribution for the according condition.

For \( r \to \infty \), the negative binomial converges to a Poisson distribution:

\[
\lim_{r \to \infty} \text{NB}(x; \mu, r) = \lim_{r \to \infty} \frac{\mu x^r}{x!} e^{-\mu} = P(x; \mu),
\]

where \( P(x; \mu) \) is the Poisson probability mass function with parameter \( \mu \) evaluated at \( x \). Note, that \( \Gamma(x + 1) = x! \) for integer \( x \).

### S3.2.3 Update Rules

We summarize the update rules for the EM algorithm. The update rules are:

- **posterior estimate**
  \[
  \hat{\alpha}_i = \frac{1}{N} \sum_{k=1}^{N} \tilde{\alpha}_{ik},
  \]
  \( \text{(S60)} \)

- **\( \mu \) update**
  \[
  \mu_{i_{\text{temp}}} = \frac{\sum_{k=1}^{N} \tilde{\alpha}_{ik} x_k}{N \hat{\alpha}_i},
  \]
  \[
  \mu_i = \max\{\mu_{i_{\text{temp}}}, \mu_{\text{min}}\}.
  \]
  \( \text{(S61)} \)

- **\( r \) update**
  The numeric solution \( r_{i_{\text{temp}}} \) of the equation:
  \[
  \sum_{k=1}^{N} \tilde{\alpha}_{ik} \psi(x_k + r_{i_{\text{temp}}}) - N \hat{\alpha}_i \psi(r_{i_{\text{temp}}}) + \]
  \[
  + N \hat{\alpha}_i \log \left( \frac{1}{N} \sum_{k=1}^{N} \tilde{\alpha}_{ik} x_k + r_{i_{\text{temp}}} \hat{\alpha}_i \right) - \eta = 0,
  \]
  \( \text{(S62)} \)
  where \( \psi \) is the digamma function. We use a bisection procedure to find the \( r_{i_{\text{temp}}} \). We then have to map \( r_{i_{\text{temp}}} \) to the allowed parameter space:
  \[
  r_i = \min\{r_{i_{\text{temp}}}, r_{\text{max}}\}.
  \]
  \( \text{(S63)} \)
α update

\[ \alpha_i = \hat{\alpha}_i + \frac{1}{N} (\gamma_i - 1) \left( \frac{1}{\alpha_i} + \frac{1}{N} (\gamma_s - n) \right). \]  
\[ \text{(S64)} \]

S3.2.4 Initialization

First we apply a k-means clustering algorithm (Hartigan and Wong 1979) with \( n \) centers ten times to the log read count data (adding a pseudo-count of 0.01 to avoid undefined values).

From these ten results, we select the result with minimal ratio between within-cluster distances to between-cluster distances of the log read counts. For each cluster \( i = 1, \ldots, n \) we calculate the maximum likelihood estimators for \( \mu_i \) and \( r_i \) of a negative binomial distribution using only read counts that belong to this cluster. These estimators are given in Eq. (S16) and Eq. (S17). After this estimations, the overdispersion parameter \( r_i \) bounded by \( r_i \leq r_{\text{max}} \). The values \( \alpha_i \) are initialized by \( \alpha_i = 1/n \), which is \( n \)-dimensional probability vector with maximum entropy. This initialization does not prefer any condition. Note that initializing an \( \alpha_i \) close to zero would clamp condition \( i \) to zero.

S3.2.5 Adjusting the Hyperparameter for the Size Parameter Prior

As mentioned in Subsection S3.1.2, for finite sample size \( N \), neither the bias nor the variance of the maximum likelihood estimator \( r_{\text{ML}} \) exists, because for data whose mean exceeds the variance \( r \) tends to infinity (Anscombe 1950). We empirically calculate a conditional bias, which is the bias under the condition that the mean is larger than the variance for the data set. For 10,000 experiments, we draw counts from a negative binomial distribution, removed experiments with mean larger than the variance, and computed the estimator \( r_{\text{ML}} \) for each experiment. Fig. S9 shows that the maximum likelihood estimator \( r_{\text{ML}} \) overestimates the true \( r \) for a small number of samples. Further it is shown that our truncated exponential prior on \( r \) reduces the effect of the overestimation for the maximum a posterior estimator. Both the bias and the variance of the maximum a posterior estimator is smaller than for the maximum likelihood estimator. The data underlying Fig. S9 were generated by drawing each \( r \) a normal distribution with mean 1.0 and standard deviation 0.1. Using this \( r \) and \( \mu = 20 \) for the parameters of a negative binomial distribution, five data points were drawn. With these five data points, for \( r \) the maximum likelihood estimator and the maximum a posterior estimator with \( \eta = 0.8 \) were calculated.
In Subsection S3.2.1 we introduced a truncated exponential function with hyperparameter $\eta$ as a prior on the size parameter $r$. This prior prefers small $r$ and countermands the bias of the maximum likelihood estimator $r_{ML}$ if using the maximum a posterior estimate. Note, that smaller estimates of $r$ also reduce the variance of the estimator because $r$ is bounded from below by zero. Thus, by adjusting $\eta$ we can decrease both the bias and the variance of the maximum a posterior estimator, and hence the mean squared error (MSE). We analyzed the effect of different values for $\eta$ for a large variety of values for $\mu$, $r$, and number of samples $N$ on the MSE, bias, and variance of the maximum a posterior estimator. Fig. S10 presents for the maximum a posterior estimator the mean over 10,000 experiments of the MSE, bias, and variance for different values of $\eta$. A data set of $N = 10$ data points is drawn from a negative binomial with parameters $\mu = 50$ and $r = 0.8$. The variance of the estimator decreases with increasing $\eta$, because of the lower bound at zero. The squared bias is minimal at $\eta = 1.6$. The MSE is minimal at 2.9.
Figure S10: The mean squared error (MSE) of the maximum a posterior estimator for \( r \). The hyperparameter \( \eta \) (x-axis) is plotted against the MSE (y-axis). Each box shows means of 10,000 experiments. One experiment consists of \( N = 10 \) data points, that were drawn from a negative binomial distribution with \( r = 0.8 \) and \( \mu = 50 \). The variance of the estimator decreases with increasing \( \eta \), whereas the squared bias is minimal at \( \eta = 1.6 \). The MSE is minimal at \( \eta = 2.9 \).

With increasing mean \( \mu \), both the variance and the mean squared error (MSE) of the maximum likelihood estimator decrease, as can be seen in Fig S11 for a mean over 10,000 experiments. One experiment of \( N = 10 \) data points is drawn from a negative binomial with \( \mu = (4, 8, 16, \ldots, 40) \) and \( r = 0.8 \). Thus, larger \( \mu \) needs less regularization by the prior to obtain a minimal MSE for the maximum posterior estimate. Therefore, \( \eta \) is selected depending on the \( \mu \) of the particular gene or transcript that is analyzed. We compute an optimal \( \eta \) for each transcript using only a single hyperparameter \( \theta \) for a data set. \( \eta \) is computed for each transcript with read counts \( x_i \) from \( \theta \) by:

\[
\eta = \frac{\theta}{1 + \mu_{ML}},
\]

where \( \mu_{ML} = 1/N \sum_{i=1}^{N} x_i \) is the mean read count for the transcript that is analyzed. \( \mu_{ML} \) is the maximum likelihood estimator for \( \mu \) of a negative binomial according to Eq. (S16).
Figure S11: The mean squared error (MSE) of the maximum likelihood estimator $r_{ML}$ for different means $\mu$. The mean $\mu$ of the negative binomial distribution (x-axis) is plotted against the MSE (y-axis). Each box shows means of 10,000 experiments. One experiment consists of $N = 10$ data points, that were drawn from a negative binomial distribution with $r = 0.8$ and $\mu$ varying from 4 to 40. Both the mean squared error and the variance of the maximum likelihood estimator decrease with increasing $\mu$. Thus, larger $\mu$ need less regularization by the prior to obtain an optimal maximum a posterior estimate.

S3.3 Calling Differentially Expressed Transcripts and I/NI Call

We suggest DEXUS for identifying differentially expressed transcripts in RNA-Seq data with unknown condition. In the E-step of the EM algorithm, $\alpha_{ik}$ estimates under which condition $i$ read count $x_k$ of a particular transcript was obtained.

In Subsection S3.2.1 in paragraph “E-step: Optimization w.r.t. Posterior Estimates” we noted that $\tilde{\alpha}_{ik}$ is the posterior of condition $i$ in the mixture model given data point $x_k$. $\alpha_i = p(i)$ is the prior for condition $i$, $p(x_k \mid i) = \text{NB}(x_k; \mu_i, r_i)$ is the likelihood for condition $i$, and $\tilde{\alpha}_{ik} = p(i \mid x_k)$ the posterior for condition $i$. According to the Bayes formula the posterior is

$$
\alpha_{ik} = \frac{\alpha_i \text{NB}(x_k; \mu_i, r_i)}{\sum_{i=1}^{n} \alpha_i \text{NB}(x_k; \mu_i, r_i)}.
$$

(S66)

In the Bayes interpretation, the prior $\alpha_i = p(i)$ gives the probability of drawing from condition $i$ without seeing any data, while the posterior $\tilde{\alpha}_{ik} = p(i \mid x_k)$ is the probability of $x_k$ being drawn from condition $i$. This means that the prior probability of the condition under which a read count is drawn (without seeing the read count) changes to the posterior probability after having observed the read count.
The posteriors $\tilde{\alpha}_{ik}$ are important to decide whether any two read counts are generated under the same or under different conditions. If any two read counts are generated under different conditions with different read count distributions, the according transcript is differentially expressed.

In the following, we have to distinguish between two cases: (i) data with known conditions and (ii) data with unknown conditions. For (i) data with known conditions, the $\tilde{\alpha}_{ik}$ are given. However the transcript may have the same read count distribution under the different conditions. To decide whether a transcript is differentially expressed in different given conditions, we have to determine whether read counts of different conditions arise from the same or from a different distribution. For (ii) data with unknown conditions, the EM algorithm ensures that different conditions have different read count distributions. The likelihood that a transcript is differentially expressed increases both with the likelihood that at least two conditions are observed and with the distance between the read count distributions of the conditions.

In the following subsections we consider first the case (i) data with known conditions and then case (ii) data with unknown conditions.

### S3.3.1 Data with Known Conditions

For data with known conditions, the condition under which the read count $x_k$ was generated is known. Therefore the $\alpha_{ik}$ values are binary: $\alpha_{ik}$ is one if $x_k$ is generated under the $i$-th condition and zero otherwise.

$$
\alpha_{ik} = \begin{cases} 
1 & \text{if } x_k \text{ is drawn under condition } i \\
0 & \text{if } x_k \text{ is not drawn under condition } i . 
\end{cases}
$$

The update rules in Subsection S3.2.3 simplify to the maximum likelihood estimators Eq. (S16) and Eq. (S17) from Subsection S3.1.2 for each condition. The regularization parameter $\eta$ can be used to determine a maximum a posterior estimate for $r$.

Two conditions: An exact test for differential expression between two conditions. We use the test suggested by Robinson and Smyth (2008) and Anders and Huber (2010), which is implemented in the R package DESeq. We use the function `nbinomTestForMatrices` of the R package DESeq. It is a test of the null hypothesis that the means of read count distributions for the two conditions $1$ and $2$ are equal. Like Fisher’s exact test, this test is a conditional test with the condition that the sum of all read counts has a particular value. We have $N$ read counts $x_k$ of which the $N_1$ read counts $x_1, \ldots, x_{N_1}$ are generated under condition $1$ and the $N_2 = N - N_1$ read counts $x_{N_1+1}, \ldots, x_N$ are generated under condition $2$. The test assumes that read counts of condition $1$ are distributed according to a negative binomial $\text{NB}(x; \mu_1, r_1)$ and read counts in condition $2$ according to $\text{NB}(x; \mu_2, r_2)$. The sum $S_1 = \sum_{k=1}^{N_1} x_k$ of $N_1$ read counts drawn from $\text{NB}(x; \mu_1, r_1)$ is distributed according to $\text{NB}(S_1; N_1 \mu_1, N_1 r_1)$ (Bean 2001; Furman 2007). Analogously, the sum $S_2 = \sum_{k=N_1+1}^{N} x_k$ of $N_2$ read counts drawn from $\text{NB}(x; \mu_2, r_2)$ is distributed according to $\text{NB}(S_2; N_2 \mu_2, N_2 r_2)$. The null hypothesis is that the mean $\mu_1$ in the first condition is equal to the mean $\mu_2$ in the second condition: $\mu = \mu_1 = \mu_2$. Using all $N = N_1 + N_2$ read counts $x_k$, the mean $\mu$ is estimated by $\mu = \frac{1}{N} \sum_{k=1}^{N} x_k$. Next, assuming $\mu_1 = \mu_2 = \mu$, the
values for \( r_1 \) and \( r_2 \) are estimated. Under the null hypothesis and with mutually independent read counts the probability \( p_S \) of observing the pair of sums \((S_1, S_2)\) with \( N_1 \) summands in \( S_1 \) and \( N_2 \) summands in \( S_2 \) is:

\[
p_S(S_1, S_2) = \text{NB}(S_1; N_1\mu, N_1 r_1) \text{NB}(S_2; N_2\mu, N_2 r_2).
\] (S68)

Next we compute the probability of observing \((S_1, S_2)\) or more extreme sum pairs \((a, b)\) under the condition that \( a + b = S \) with \( N_1 \) summands in \( a \) and \( N_2 \) summands in \( b \). Further we assume that \( a \sim \text{NB}(x; N_1\mu, N_1 r_1) \) and that \( b \sim \text{NB}(x; N_2\mu, N_2 r_2) \). The probability of observing \((S_1, S_2)\) or more extreme sum pairs \((a, b)\) is:

\[
p((S_1, S_2) \lesssim (a, b) \mid a + b = S) = \frac{p((S_1, S_2) \lesssim (a, b), a + b = S)}{p(a + b = S)},
\] (S69)

where \((S_1, S_2) \lesssim (a, b)\) means that \((a, b)\) is equal or more extreme than \((S_1, S_2)\). If \((S_1, S_2) \lesssim (a, b) \iff p_\alpha(a, b) \leq p_\alpha(S_1, S_2)\) then the \( p \)-value can be calculated by:

\[
p = \frac{\sum_{a+b=S} p_S(a, b) \leq p_\alpha(S_1, S_2)}{\sum_{a+b=S} p_S(a, b)}.
\] (S70)

**Multiple conditions: Generalized Linear Model.** For multiple known conditions we follow McCarthy et al. (2012) and fit a generalized linear model (GLM, Nelder and Wedderburn (1972)) for a negative binomial response using the logarithm as link function and the estimated dispersion parameters.

The GLM allows specifying any design and test for the significance of covariates. Without specifying a particular design, DEXUS will use a design that includes a covariate for each specified condition and compare it with a null hypothesis model that only includes an intercept term. The \( p \)-value from this comparison is used to rank transcripts according to the evidence for differential expression.

**S3.3.2 Data with Unknown Conditions: I/NI Call**

The Bayesian framework allows defining an informative/non-informative (I/NI) call (Hochreiter et al. 2006; Talloen et al. 2007, 2010; Clevert et al. 2011; Klambauer et al. 2012). An I/NI call reduces the false discovery rate at detecting differentially expressed transcripts because only those transcripts are called for which the evidence of being differentially expressed is high. DEXUS first computes the I/NI value (an evidence value) for differential expression. Subsequently, transcripts are called informative if the I/NI value is beyond a threshold.

In contrast to \( \phi \), or \( r_i \), which capture noise variation, \( \alpha \) captures variation arising from differentially expressed transcripts. Therefore, the posterior \( \alpha \) of \( \alpha \) indicates differential expression in the data if at least two conditions have a probability larger than zero. First, we want to have evidence that at least two conditions are present. The larger the posterior value of a condition, the more read counts have this condition, the higher is the evidence that this condition was present for at least one read count. The model may explain one true condition by two model conditions and
differential expression would be falsely detected. Secondly, we want to have evidence that model conditions are different. The more the means $\mu_i$ of conditions differ (the means of the associated negative binomials), the higher is the evidence that these conditions are indeed different and the transcript is differentially expressed. Thus, the evidence on differential expression (the I/NI value) should consider two factors: (I) at least two non-zero posterior values for $\alpha_i$, where larger values have more evidence that the conditions were indeed present; (II) differences of the means $\mu_i$, where larger differences have more evidence that the conditions are indeed different.

We select the largest $\alpha_i$ (the condition with largest probability), and assume without loss of generality that this is the first condition ($i = 1$). Then we compare other conditions to the first condition. Factor (II), the differences between means, is expressed by the log differences $|\log (\mu_i) - \log (\mu_1)|$. Factor (I), two large non-zero posterior values, is included by weighting these differences by $\alpha_i$. Thus, the I/NI value is

$$I/NI(\alpha, \mu) = \sum_{i=1}^{n} \alpha_i \left| \log \left( \frac{\mu_i}{\mu_1} \right) \right| = \sum_{i=1}^{n} \alpha_i \left| \log (\mu_i) - \log (\mu_1) \right|. \quad (S71)$$

The I/NI value is the expected fold change of read counts relative to read counts of the most prominent condition given a noise-free model (all read counts are equal to the mean of the according condition). Another interpretation of the I/NI value is: “the information gain of the posterior $\hat{\alpha}$ compared to the prior distribution $p(\alpha)$”. The prior represents the null hypothesis that only one condition is present and the transcript is not differentially expressed. Therefore, the I/NI call measures the tendency to reject the null hypothesis based on the observed data.

The I/NI value can also be viewed as the distance between a multiple component model (differential expression) and the its closest single component model (no differential expression). The I/NI value is a distance measure between a model with parameters $M_1 = (\alpha_1^1, ..., \alpha_n^1, \mu_1^1, ..., \mu_n^1)$ and another model with $M_2 = (\alpha_1^2, ..., \alpha_n^2, \mu_1^2, ..., \mu_n^2)$:

$$d(M_1, M_2) = \sum_{i=1}^{n} |\alpha_i^1 - \alpha_i^2| \left| \log \left( \frac{\mu_i^1}{\mu_i^2} \right) \right|. \quad (S72)$$

In the DEXUS method, the I/NI value measures the distance between the selected multiple component model with parameters $M = (\alpha_1, ..., \alpha_n, \mu_1, ..., \mu_n)$ to the closest single component model with parameters $M_0 = (1, 0, ..., 0, \mu_1, \mu_1, ..., \mu_1)$:

$$d(M, M_0) = |\alpha_i - 1| \left| \log \left( \frac{\mu_i}{\mu_1} \right) \right| + \sum_{i=2}^{n} |\alpha_i - 0| \left| \log \left( \frac{\mu_i}{\mu_1} \right) \right| = I/NI(\alpha, \mu). \quad (S73)$$

Note that other models than $M_0$ lead to larger distances to $M$.

### S3.4 Sensitivity Analysis of the Hyperparameter for the Dirichlet Prior

For investigating the sensitivity of the hyperparameter $G$, that we introduced in Eq. (S27), we applied DEXUS to simulated data sets with unknown conditions (see Section S4.3). These data sets
were produced assuming three different library sizes ($10^6$, $10^7$, and $10^8$), eight different settings with respect to the unknown conditions (6/6, 12/12, 9/3, 18/6, 10/2, 20/4, 11/1, and 22/2). In total we had $3 \times 8 \times 100 = 2,400$ data sets and we used five different settings for $G$, that is $G = 0.1$, $G = 0.5$, $G = 1$, $G = 5$, and $G = 10$. We assess the average performance of DEXUS for different hyperparameters $G$ in terms of the area under ROC curve ($\text{AUC}_{\text{ROC}}$) and the area under precision-recall curve ($\text{AUC}_{\text{PR}}$). The $\text{AUC}_{\text{ROC}}$ is determined by the ranking of the I/NI values, therefore it measures implicitly how much the I/NI value ranking change if different values for the hyperparameter $G$ are used. Figures S12, S13, and S14 report the performance in terms of $\text{AUC}_{\text{ROC}}$ and $\text{AUC}_{\text{PR}}$ for the eight different settings and the three library sizes. The figures show that the performance (therefore the I/NI value ranking) is relatively insensitive to the setting of the hyperparameter $G$. For data sets with few samples in one of the conditions, $G > 1$ performs better with respect to $\text{AUC}_{\text{PR}}$. The improvement results from the reduced pressure on the minor condition towards zero condition weight — otherwise the condition would die out. Note, that in these settings only highly unbalanced number of samples in the conditions are present. However this is a quite unusual case in biological or medical studies. Only for very extreme cases, e.g. only one sample in one of the conditions, the performance improvement over $G = 1$ is notable. For data sets with the same number of samples in each condition, $G < 1$ performs better. However the performance improvement compared to $G = 1$ is minor. In conclusion, $G = 1$ has quite good performance for typical biological or medical studies.

Since the conditions and how many samples are in a condition are not known a priori, we average the performance over the different settings (number of samples in a condition). Tables S1 and S2 show the average performance of the model for different hyperparameters $G$ in terms of the area under ROC curve ($\text{AUC}_{\text{ROC}}$) and the area under precision-recall curve ($\text{AUC}_{\text{PR}}$). The default value $G = 1$ is the best compromise for the different settings and gives for most library sizes the best average performance. The tables show that the performance is not very sensitive with respect to the value of $G$. A larger or smaller value of $G$ than 1, leads for some settings to an performance increase but for other settings to a decrease which average out. This averaging out is also to be expected for real data sets, in which different settings are assumed to be present simultaneously.

<table>
<thead>
<tr>
<th>Library size</th>
<th>$G = 0.1$</th>
<th>$G = 0.5$</th>
<th>$G = 1$</th>
<th>$G = 5$</th>
<th>$G = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^6$</td>
<td>0.74 ± 0.03</td>
<td>0.75 ± 0.02</td>
<td>0.75 ± 0.01</td>
<td>0.74 ± 0.02</td>
<td>0.73 ± 0.02</td>
</tr>
<tr>
<td>$10^7$</td>
<td>0.79 ± 0.05</td>
<td>0.80 ± 0.03</td>
<td>0.83 ± 0.02</td>
<td>0.77 ± 0.01</td>
<td>0.76 ± 0.01</td>
</tr>
<tr>
<td>$10^8$</td>
<td>0.89 ± 0.06</td>
<td>0.90 ± 0.03</td>
<td>0.91 ± 0.02</td>
<td>0.86 ± 0.02</td>
<td>0.85 ± 0.02</td>
</tr>
</tbody>
</table>

Table S1: The performance in terms of area under ROC curve ($\text{AUC}_{\text{ROC}}$) for three different library sizes and different choices of the hyperparameter $G$. The displayed values are the means over 800 data sets, that is 100 data sets for each of the eight different settings for the number of replicates in the conditions.

Though $G = 1$ is our recommendation, we still offer rules to set $G$ for users who want to find better values of $G$:

- If conditions with few samples should be detected, large values of $G$, like $G = 5$ or $G = 10$, improve the $\text{AUC}_{\text{PR}}$ of DEXUS.
Table S2: The performance in terms of area under precision-recall curve (AUC_{PR}) for three different library sizes and different choices of the hyperparameter $G$. The displayed values are the means over 800 data sets, that is 100 data sets for each of the eight different settings for the number of replicates in the conditions.

- For condition with equal number of samples, small $G$ like $G = 0.1$ gives slightly better results than $G = 1$.
- If the number of samples in the conditions are unbalanced but not very extreme, values of $G$ around one (the default) supply good performance.

<table>
<thead>
<tr>
<th>library size</th>
<th>$G = 0.1$</th>
<th>$G = 0.5$</th>
<th>$G = 1$</th>
<th>$G = 5$</th>
<th>$G = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^6$</td>
<td>0.51 ± 0.08</td>
<td>0.54 ± 0.06</td>
<td>0.55 ± 0.05</td>
<td>0.56 ± 0.03</td>
<td>0.55 ± 0.02</td>
</tr>
<tr>
<td>$10^7$</td>
<td>0.62 ± 0.12</td>
<td>0.67 ± 0.09</td>
<td>0.70 ± 0.07</td>
<td>0.69 ± 0.03</td>
<td>0.68 ± 0.03</td>
</tr>
<tr>
<td>$10^8$</td>
<td>0.74 ± 0.15</td>
<td>0.80 ± 0.11</td>
<td>0.82 ± 0.08</td>
<td>0.81 ± 0.04</td>
<td>0.80 ± 0.03</td>
</tr>
</tbody>
</table>
Figure S12: The performance of DEXUS in terms of AUC_{ROC} and AUC_{PR} for different values of the hyperparameter $G$. The library size is $10^6$. The panels show the results for different number of replicates in the conditions displayed above the panel: 6/6, 12/12, 9/3, 18/6, 10/2, 20/4, 11/1, and 22/2. The AUC_{ROC} and AUC_{PR} are plotted against $G$ values ($x$-axis). Each data point has an error bar that represents the standard deviation of the performance on 100 data sets.
Figure S13: The performance of DEXUS in terms of AUC\textsubscript{ROC} and AUC\textsubscript{PR} for different values of the hyperparameter $G$. The library size is $10^7$. The panels show the results for different number of replicates in the conditions displayed above the panel: 6/6, 12/12, 9/3, 18/6, 10/2, 20/4, 11/1, and 22/2. The AUC\textsubscript{ROC} and AUC\textsubscript{PR} are plotted against $G$ values (x-axis). Each data point has an error bar that represents the standard deviation of the performance on 100 data sets.
Figure S14: The performance of DEXUS in terms of AUC_{ROC} and AUC_{PR} for different values of the hyperparameter $G$. The library size is $10^8$. The panels show the results for different number of replicates in the conditions displayed above the panel: 6/6, 12/12, 9/3, 18/6, 10/2, 20/4, 11/1, and 22/2. The AUC_{ROC} and AUC_{PR} are plotted against $G$ values ($x$-axis). Each data point has an error bar that represents the standard deviation of the performance on 100 data sets.
S4 Experiments

S4.1 Evaluation Criteria for Simulated Data Sets

For simulated data sets with known ground truth, we formulate the detection of differential expression as a classification task. A method has to decide whether a gene or a transcript is differentially expressed (positive prediction) or not (negative prediction). For simulated data we know which genes are differentially expressed (the positives) and which are not (the negatives). Therefore, we can determine true positives, false positives, true negatives, and false negatives. The methods return a continuous value, like a $p$-value obtained from a test or the I/NI values with DEXUS, together with a threshold for deciding whether the gene is differentially expressed or not. This value allows to rank genes and to compute the receiver-operator characteristics (ROC), a standard measure to evaluate classification results. The area under the ROC curve ($\text{AUC}_{\text{ROC}}$) is a well-known classification performance criterion and is equivalent to a Mann-Whitney-Wilcoxon test of ranks.

Usually the number of differentially expressed genes is much smaller than the number of nondifferentially expressed genes. For these unbalanced classes, i.e. one class is much larger then the other, the area under the precision recall curve ($\text{AUC}_{\text{PR}}$) is more appropriate as performance criterion, because it is independent of the true negatives. We report both the $\text{AUC}_{\text{ROC}}$ and the $\text{AUC}_{\text{PR}}$.

S4.2 RNA-Seq Data with Known Conditions

S4.2.1 Methods Compared

We compare the following methods (available as R packages) for differential expression in RNA-Seq data:

- DEXUS (our novel method using known conditions, see Section S3.3.1)
- DSS 1.0.0 (Wu et al. 2013)
- DESeq 1.8.1 (Anders and Huber 2010)
- baySeq 1.10.0 (Hardcastle and Kelly 2010)
- edgeR 2.6.0 (Robinson et al. 2010)
- DEGseq 1.10.0 (Wang et al. 2010)
- NOISeq 29-IV-2011 (Tarazona et al. 2011)
- PoissonSeq (Li et al. 2012)
- SAMseq samr 2.0 (Li and Tibshirani 2011)
- QuasiSeq 1.0-2 (Lund et al. 2012)
- NBPSeq 0.1.8 (Cumbie et al. 2011)
We used the default settings of all methods. All methods supply a ranking criterion like a $p$-value.

### S4.2.2 Simulated Data With Two Known Conditions

**Data Simulation.** We simulated datasets with $10^6$, $10^7$, or $10^8$ reads per sample (the library size). For each library size 2, 6 or 15 replicates per condition were simulated. For each of these nine combinations we generated 100 datasets with 10,000 transcripts each. Under condition $i$ the reads for a transcript distributed according to $\text{NB}(x; \mu_i, r_i)$. For the selection of the mean $\mu_i$ and the size $r_i$ ($r_i = \phi_i^{-1}$ with overdispersion $\phi_i$) we sampled values from the from the “Mice Strains” RNA-Seq dataset (Bottomly et al. 2011), where we used only one biological condition. Following Wu et al. (2013), we sampled $\mu_i$ values from the median read counts of the transcripts. The overdispersion $\phi$ tends to decrease with increasing mean read counts as shown in Fig. S15. Therefore we fitted a regression line to overdispersions by least squares. After sampling the log $\mu_i$ values, we calculated the corresponding log $\phi_i$ values according to the regression line, added Gaussian noise ($\sigma^2 = 1$) to the log $\phi_i$ values and transformed the overdispersion into the size parameter $r_i = 1/\phi_i$. 30% of the genes were chosen to be differentially expressed. Differential expression was expressed by adjusting the means of the negative binomials to obtain fold changes of 0.5, 1 and 1.5 (randomly chosen) between these means.

**Results.** We first compared the methods on simulated data for two condition. Tabs. S3, S4, and S5 report the results for a library size of $10^6$, $10^7$, and $10^8$. DEXUS estimates the dispersion parameter with comparable performance to other methods when the sample size is low. DEXUS outperforms the other methods when the sample size is medium, i.e. six replicates, or large, i.e. fifteen replicates.
Figure S15: Scatterplots of overdispersion and median read counts of various datasets (see Tab. S20 in Subsection S5.1). For each transcript the median count was computed and the overdispersion estimated by maximum likelihood.
Table S3: Performance of methods for two known conditions with 2 replicates and a library size of $10^6$, $10^7$, and $10^8$. The best methods with respect to $\text{AUC}_{\text{ROC}}$ are DSS, baySeq, and DEXUS. With respect to $\text{AUC}_{\text{PR}}$ DSS, baySeq and edgeR perform best.

<table>
<thead>
<tr>
<th>Libsize</th>
<th>Method</th>
<th>$\text{AUC}_{\text{ROC}}$</th>
<th>$\text{AUC}_{\text{PR}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^6$</td>
<td>DEXUS</td>
<td>0.765±0.0019</td>
<td>0.669±0.0025</td>
</tr>
<tr>
<td></td>
<td>DESeq</td>
<td>0.752±0.0022</td>
<td>0.687±0.0025</td>
</tr>
<tr>
<td></td>
<td>DESeq2</td>
<td>0.774±0.0056</td>
<td>0.693±0.0069</td>
</tr>
<tr>
<td></td>
<td>edgeR</td>
<td>0.753±0.0023</td>
<td>0.687±0.0026</td>
</tr>
<tr>
<td></td>
<td>baySeq</td>
<td><strong>0.800±0.0017</strong></td>
<td><strong>0.700±0.0023</strong></td>
</tr>
<tr>
<td></td>
<td>NOISeq</td>
<td>0.750±0.0021</td>
<td>0.677±0.0026</td>
</tr>
<tr>
<td></td>
<td>SAMseq</td>
<td>0.735±0.0024</td>
<td>0.572±0.0053</td>
</tr>
<tr>
<td></td>
<td>DSS</td>
<td>0.773±0.0021</td>
<td>0.695±0.0025</td>
</tr>
<tr>
<td></td>
<td>PoissonSeq</td>
<td>0.700±0.0024</td>
<td>0.597±0.0042</td>
</tr>
<tr>
<td></td>
<td>NBPSeg</td>
<td>0.777±0.0056</td>
<td>0.698±0.0074</td>
</tr>
<tr>
<td></td>
<td>QuasiSeq</td>
<td>0.713±0.0069</td>
<td>0.670±0.0075</td>
</tr>
<tr>
<td></td>
<td>TSPM</td>
<td>0.628±0.0096</td>
<td>0.515±0.0074</td>
</tr>
<tr>
<td></td>
<td>TweeDEseq</td>
<td>0.733±0.0062</td>
<td>0.649±0.0071</td>
</tr>
<tr>
<td>$10^7$</td>
<td>DEXUS</td>
<td>0.896±0.0014</td>
<td>0.839±0.0021</td>
</tr>
<tr>
<td></td>
<td>DESeq</td>
<td>0.882±0.0016</td>
<td>0.841±0.0020</td>
</tr>
<tr>
<td></td>
<td>DESeq2</td>
<td>0.887±0.0042</td>
<td>0.843±0.0055</td>
</tr>
<tr>
<td></td>
<td>edgeR</td>
<td>0.894±0.0015</td>
<td>0.853±0.0020</td>
</tr>
<tr>
<td></td>
<td>baySeq</td>
<td>0.896±0.0014</td>
<td>0.849±0.0020</td>
</tr>
<tr>
<td></td>
<td>NOISeq</td>
<td>0.851±0.0017</td>
<td>0.794±0.0021</td>
</tr>
<tr>
<td></td>
<td>SAMseq</td>
<td>0.836±0.0018</td>
<td>0.663±0.0043</td>
</tr>
<tr>
<td></td>
<td>DSS</td>
<td><strong>0.897±0.0015</strong></td>
<td><strong>0.856±0.0019</strong></td>
</tr>
<tr>
<td></td>
<td>PoissonSeq</td>
<td>0.821±0.0025</td>
<td>0.711±0.0056</td>
</tr>
<tr>
<td></td>
<td>NBPSeg</td>
<td>0.893±0.0043</td>
<td>0.845±0.0061</td>
</tr>
<tr>
<td></td>
<td>QuasiSeq</td>
<td>0.889±0.0045</td>
<td>0.852±0.0054</td>
</tr>
<tr>
<td></td>
<td>TSPM</td>
<td>0.783±0.0051</td>
<td>0.619±0.0087</td>
</tr>
<tr>
<td></td>
<td>TweeDEseq</td>
<td>0.839±0.0052</td>
<td>0.778±0.0060</td>
</tr>
<tr>
<td>$10^8$</td>
<td>DEXUS</td>
<td>0.961±0.0008</td>
<td>0.930±0.0014</td>
</tr>
<tr>
<td></td>
<td>DESeq</td>
<td>0.955±0.0009</td>
<td>0.934±0.0011</td>
</tr>
<tr>
<td></td>
<td>DESeq2</td>
<td>0.956±0.0025</td>
<td>0.935±0.0033</td>
</tr>
<tr>
<td></td>
<td>edgeR</td>
<td>0.962±0.0009</td>
<td>0.942±0.0011</td>
</tr>
<tr>
<td></td>
<td>baySeq</td>
<td>0.954±0.0010</td>
<td>0.933±0.0011</td>
</tr>
<tr>
<td></td>
<td>NOISeq</td>
<td>0.908±0.0011</td>
<td>0.860±0.0016</td>
</tr>
<tr>
<td></td>
<td>SAMseq</td>
<td>0.888±0.0012</td>
<td>0.705±0.0037</td>
</tr>
<tr>
<td></td>
<td>DSS</td>
<td><strong>0.965±0.0008</strong></td>
<td><strong>0.945±0.0010</strong></td>
</tr>
<tr>
<td></td>
<td>PoissonSeq</td>
<td>0.884±0.0018</td>
<td>0.787±0.0059</td>
</tr>
<tr>
<td></td>
<td>NBPSeg</td>
<td>0.960±0.0026</td>
<td>0.936±0.0037</td>
</tr>
<tr>
<td></td>
<td>QuasiSeq</td>
<td>0.959±0.0024</td>
<td>0.941±0.0030</td>
</tr>
<tr>
<td></td>
<td>TSPM</td>
<td>0.871±0.0041</td>
<td>0.686±0.0142</td>
</tr>
<tr>
<td></td>
<td>TweeDEseq</td>
<td>0.897±0.0045</td>
<td>0.847±0.0057</td>
</tr>
</tbody>
</table>
Table S4: Performance of methods for two known conditions with 6 replicates and a library size of $10^6$, $10^7$, and $10^8$. The best methods with respect to $\text{AUC}_{\text{ROC}}$ are DEXUS and DSS. With respect to $\text{AUC}_{\text{PR}}$ DEXUS, baySeq and DSS perform best.

<table>
<thead>
<tr>
<th>Libsize</th>
<th>Method</th>
<th>$\text{AUC}_{\text{ROC}}$</th>
<th>$\text{AUC}_{\text{PR}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^6$</td>
<td>DEXUS</td>
<td>0.865 ± 0.0017</td>
<td>0.825 ± 0.0019</td>
</tr>
<tr>
<td>$10^6$</td>
<td>DESeq</td>
<td>0.855 ± 0.0018</td>
<td>0.828 ± 0.0020</td>
</tr>
<tr>
<td>$10^6$</td>
<td>DESeq2</td>
<td>0.864 ± 0.0045</td>
<td>0.831 ± 0.0052</td>
</tr>
<tr>
<td>$10^6$</td>
<td>edgeR</td>
<td>0.856 ± 0.0019</td>
<td>0.830 ± 0.0020</td>
</tr>
<tr>
<td>$10^6$</td>
<td>baySeq</td>
<td>0.892 ± 0.0013</td>
<td>0.845 ± 0.0018</td>
</tr>
<tr>
<td>$10^6$</td>
<td>NOISeq</td>
<td>0.812 ± 0.0020</td>
<td>0.783 ± 0.0023</td>
</tr>
<tr>
<td>$10^6$</td>
<td>SAMseq</td>
<td>0.847 ± 0.0019</td>
<td>0.812 ± 0.0020</td>
</tr>
<tr>
<td>$10^6$</td>
<td>DSS</td>
<td>0.868 ± 0.0016</td>
<td>0.875 ± 0.0019</td>
</tr>
<tr>
<td>$10^6$</td>
<td>PoissonSeq</td>
<td>0.803 ± 0.0023</td>
<td>0.723 ± 0.0040</td>
</tr>
<tr>
<td>$10^6$</td>
<td>NBPSseq</td>
<td>0.864 ± 0.0050</td>
<td>0.826 ± 0.0059</td>
</tr>
<tr>
<td>$10^6$</td>
<td>QuasiSeq</td>
<td>0.834 ± 0.0063</td>
<td>0.820 ± 0.0060</td>
</tr>
<tr>
<td>$10^6$</td>
<td>TSPM</td>
<td>0.828 ± 0.0061</td>
<td>0.782 ± 0.0077</td>
</tr>
<tr>
<td>$10^6$</td>
<td>TweeDEseq</td>
<td>0.852 ± 0.0048</td>
<td>0.810 ± 0.0056</td>
</tr>
<tr>
<td>$10^7$</td>
<td>DEXUS</td>
<td>0.964 ± 0.0008</td>
<td>0.949 ± 0.0010</td>
</tr>
<tr>
<td>$10^7$</td>
<td>DESeq</td>
<td>0.958 ± 0.0009</td>
<td>0.946 ± 0.0010</td>
</tr>
<tr>
<td>$10^7$</td>
<td>DESeq2</td>
<td>0.956 ± 0.0024</td>
<td>0.943 ± 0.0028</td>
</tr>
<tr>
<td>$10^7$</td>
<td>edgeR</td>
<td>0.961 ± 0.0010</td>
<td>0.949 ± 0.0011</td>
</tr>
<tr>
<td>$10^7$</td>
<td>baySeq</td>
<td>0.957 ± 0.0009</td>
<td>0.942 ± 0.0011</td>
</tr>
<tr>
<td>$10^7$</td>
<td>NOISeq</td>
<td>0.918 ± 0.0013</td>
<td>0.889 ± 0.0015</td>
</tr>
<tr>
<td>$10^7$</td>
<td>SAMseq</td>
<td>0.948 ± 0.0010</td>
<td>0.934 ± 0.0011</td>
</tr>
<tr>
<td>$10^7$</td>
<td>DSS</td>
<td>0.967 ± 0.0008</td>
<td>0.949 ± 0.0009</td>
</tr>
<tr>
<td>$10^7$</td>
<td>PoissonSeq</td>
<td>0.900 ± 0.0018</td>
<td>0.828 ± 0.0049</td>
</tr>
<tr>
<td>$10^7$</td>
<td>NBPSseq</td>
<td>0.959 ± 0.0028</td>
<td>0.941 ± 0.0036</td>
</tr>
<tr>
<td>$10^7$</td>
<td>QuasiSeq</td>
<td>0.959 ± 0.0024</td>
<td>0.946 ± 0.0028</td>
</tr>
<tr>
<td>$10^7$</td>
<td>TSPM</td>
<td>0.953 ± 0.0024</td>
<td>0.935 ± 0.0030</td>
</tr>
<tr>
<td>$10^7$</td>
<td>TweeDEseq</td>
<td>0.938 ± 0.0037</td>
<td>0.926 ± 0.0035</td>
</tr>
<tr>
<td>$10^8$</td>
<td>DEXUS</td>
<td>0.993 ± 0.0003</td>
<td>0.988 ± 0.0004</td>
</tr>
<tr>
<td>$10^8$</td>
<td>DESeq</td>
<td>0.990 ± 0.0004</td>
<td>0.986 ± 0.0005</td>
</tr>
<tr>
<td>$10^8$</td>
<td>DESeq2</td>
<td>0.989 ± 0.0013</td>
<td>0.985 ± 0.0015</td>
</tr>
<tr>
<td>$10^8$</td>
<td>edgeR</td>
<td>0.992 ± 0.0004</td>
<td>0.988 ± 0.0005</td>
</tr>
<tr>
<td>$10^8$</td>
<td>baySeq</td>
<td>0.985 ± 0.0007</td>
<td>0.981 ± 0.0008</td>
</tr>
<tr>
<td>$10^8$</td>
<td>NOISeq</td>
<td>0.958 ± 0.0007</td>
<td>0.934 ± 0.0010</td>
</tr>
<tr>
<td>$10^8$</td>
<td>SAMseq</td>
<td>0.986 ± 0.0005</td>
<td>0.980 ± 0.0007</td>
</tr>
<tr>
<td>$10^8$</td>
<td>DSS</td>
<td>0.992 ± 0.0004</td>
<td>0.988 ± 0.0004</td>
</tr>
<tr>
<td>$10^8$</td>
<td>PoissonSeq</td>
<td>0.941 ± 0.0013</td>
<td>0.881 ± 0.0041</td>
</tr>
<tr>
<td>$10^8$</td>
<td>NBPSseq</td>
<td>0.991 ± 0.0012</td>
<td>0.984 ± 0.0016</td>
</tr>
<tr>
<td>$10^8$</td>
<td>QuasiSeq</td>
<td>0.989 ± 0.0014</td>
<td>0.985 ± 0.0017</td>
</tr>
<tr>
<td>$10^8$</td>
<td>TSPM</td>
<td>0.987 ± 0.0014</td>
<td>0.981 ± 0.0018</td>
</tr>
<tr>
<td>$10^8$</td>
<td>TweeDEseq</td>
<td>0.965 ± 0.0032</td>
<td>0.965 ± 0.0027</td>
</tr>
<tr>
<td>Libsize</td>
<td>Method</td>
<td>AUC\textsubscript{ROC}</td>
<td>AUC\textsubscript{PR}</td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
<td>--------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>$10^6$</td>
<td>DEXUS</td>
<td>0.928 ± 0.0014</td>
<td>0.910 ± 0.0014</td>
</tr>
<tr>
<td>$10^6$</td>
<td>DESeq</td>
<td>0.922 ± 0.0015</td>
<td>0.910 ± 0.0014</td>
</tr>
<tr>
<td>$10^6$</td>
<td>DESeq2</td>
<td>0.925 ± 0.0038</td>
<td>0.909 ± 0.0040</td>
</tr>
<tr>
<td>$10^6$</td>
<td>edgeR</td>
<td>0.921 ± 0.0017</td>
<td>0.908 ± 0.0018</td>
</tr>
<tr>
<td>$10^6$</td>
<td>baySeq</td>
<td>0.943 ± 0.0011</td>
<td>0.919 ± 0.0014</td>
</tr>
<tr>
<td>$10^6$</td>
<td>NOISeq</td>
<td>0.851 ± 0.0020</td>
<td>0.840 ± 0.0026</td>
</tr>
<tr>
<td>$10^6$</td>
<td>SAMseq</td>
<td>0.910 ± 0.0015</td>
<td>0.902 ± 0.0062</td>
</tr>
<tr>
<td>$10^6$</td>
<td>DSS</td>
<td>0.927 ± 0.0014</td>
<td>0.912 ± 0.0014</td>
</tr>
<tr>
<td>$10^6$</td>
<td>PoissonSeq</td>
<td>0.861 ± 0.0027</td>
<td>0.800 ± 0.0047</td>
</tr>
<tr>
<td>$10^6$</td>
<td>NBPSeq</td>
<td>0.922 ± 0.0049</td>
<td>0.903 ± 0.0056</td>
</tr>
<tr>
<td>$10^6$</td>
<td>QuasiSeq</td>
<td>0.910 ± 0.0048</td>
<td>0.894 ± 0.0045</td>
</tr>
<tr>
<td>$10^6$</td>
<td>TSPM</td>
<td>0.916 ± 0.0046</td>
<td>0.900 ± 0.0047</td>
</tr>
<tr>
<td>$10^6$</td>
<td>TweeDEseq</td>
<td>0.921 ± 0.0038</td>
<td>0.902 ± 0.0041</td>
</tr>
<tr>
<td>$10^7$</td>
<td>DEXUS</td>
<td>0.989 ± 0.0004</td>
<td>0.985 ± 0.0005</td>
</tr>
<tr>
<td>$10^7$</td>
<td>DESeq</td>
<td>0.986 ± 0.0005</td>
<td>0.982 ± 0.0006</td>
</tr>
<tr>
<td>$10^7$</td>
<td>DESeq2</td>
<td>0.984 ± 0.0016</td>
<td>0.979 ± 0.0018</td>
</tr>
<tr>
<td>$10^7$</td>
<td>edgeR</td>
<td>0.986 ± 0.0008</td>
<td>0.982 ± 0.0009</td>
</tr>
<tr>
<td>$10^7$</td>
<td>baySeq</td>
<td>0.982 ± 0.0008</td>
<td>0.977 ± 0.0010</td>
</tr>
<tr>
<td>$10^7$</td>
<td>NOISeq</td>
<td>0.954 ± 0.0008</td>
<td>0.938 ± 0.0009</td>
</tr>
<tr>
<td>$10^7$</td>
<td>SAMseq</td>
<td>0.981 ± 0.0006</td>
<td>0.978 ± 0.0048</td>
</tr>
<tr>
<td>$10^7$</td>
<td>DSS</td>
<td>0.987 ± 0.0005</td>
<td>0.982 ± 0.0006</td>
</tr>
<tr>
<td>$10^7$</td>
<td>PoissonSeq</td>
<td>0.942 ± 0.0013</td>
<td>0.886 ± 0.0041</td>
</tr>
<tr>
<td>$10^7$</td>
<td>NBPSeq</td>
<td>0.985 ± 0.0019</td>
<td>0.979 ± 0.0022</td>
</tr>
<tr>
<td>$10^7$</td>
<td>QuasiSeq</td>
<td>0.981 ± 0.0023</td>
<td>0.970 ± 0.0032</td>
</tr>
<tr>
<td>$10^7$</td>
<td>TSPM</td>
<td>0.985 ± 0.0015</td>
<td>0.980 ± 0.0018</td>
</tr>
<tr>
<td>$10^7$</td>
<td>TweeDEseq</td>
<td>0.976 ± 0.0025</td>
<td>0.973 ± 0.0024</td>
</tr>
<tr>
<td>$10^8$</td>
<td>DEXUS</td>
<td>0.999 ± 0.0001</td>
<td>0.999 ± 0.0001</td>
</tr>
<tr>
<td>$10^8$</td>
<td>DESeq</td>
<td>0.998 ± 0.0002</td>
<td>0.997 ± 0.0002</td>
</tr>
<tr>
<td>$10^8$</td>
<td>DESeq2</td>
<td>0.997 ± 0.0006</td>
<td>0.996 ± 0.0007</td>
</tr>
<tr>
<td>$10^8$</td>
<td>edgeR</td>
<td>0.998 ± 0.0003</td>
<td>0.997 ± 0.0004</td>
</tr>
<tr>
<td>$10^8$</td>
<td>baySeq</td>
<td>0.995 ± 0.0006</td>
<td>0.994 ± 0.0007</td>
</tr>
<tr>
<td>$10^8$</td>
<td>NOISeq</td>
<td>0.981 ± 0.0004</td>
<td>0.967 ± 0.0006</td>
</tr>
<tr>
<td>$10^8$</td>
<td>SAMseq</td>
<td>0.997 ± 0.0002</td>
<td>0.996 ± 0.0003</td>
</tr>
<tr>
<td>$10^8$</td>
<td>DSS</td>
<td>0.998 ± 0.0002</td>
<td>0.998 ± 0.0002</td>
</tr>
<tr>
<td>$10^8$</td>
<td>PoissonSeq</td>
<td>0.969 ± 0.0008</td>
<td>0.927 ± 0.0029</td>
</tr>
<tr>
<td>$10^8$</td>
<td>NBPSeq</td>
<td>0.998 ± 0.0007</td>
<td>0.997 ± 0.0009</td>
</tr>
<tr>
<td>$10^8$</td>
<td>QuasiSeq</td>
<td>0.996 ± 0.0011</td>
<td>0.991 ± 0.0020</td>
</tr>
<tr>
<td>$10^8$</td>
<td>TSPM</td>
<td>0.998 ± 0.0005</td>
<td>0.996 ± 0.0006</td>
</tr>
<tr>
<td>$10^8$</td>
<td>TweeDEseq</td>
<td>0.979 ± 0.0025</td>
<td>0.983 ± 0.0019</td>
</tr>
</tbody>
</table>

Table S5: Performance of methods for two known conditions with 15 replicates and a library size of $10^6$, $10^7$, and $10^8$. The best methods with respect to AUC\textsubscript{ROC} are DEXUS, baySeq, and DSS. With respect to AUC\textsubscript{PR} DEXUS and DSS perform best.
S4.2.3 Simulated Data With Multiple Known Conditions

Data Simulation. We simulated data for multi-class problems for three conditions with 2, 6 or 15 replicates each. The data was generated like for two known conditions as described in Subsection S4.2.2. If a transcript was selected to be differentially expressed, one group or two groups were given a log fold change of either 0.5, 1.0 or 1.5 (randomly chosen). We compared DEXUS to the multi-class versions of edgeR, baySeq, and SAMSeq.

Results. We first compared the methods on simulated data for two condition. Tabs. S6, S7, and S8 report the results for a library size of $10^6$, $10^7$, and $10^8$. DEXUS outperforms the other methods when the sample size is medium, i.e. six replicates, or large, i.e. fifteen replicates.

<table>
<thead>
<tr>
<th>Libsize</th>
<th>Method</th>
<th>$\text{AUC}_{\text{ROC}}$</th>
<th>$\text{AUC}_{\text{PR}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^6$</td>
<td>DEXUS</td>
<td>$0.830 \pm 0.0023$</td>
<td>$0.745 \pm 0.0034$</td>
</tr>
<tr>
<td>$10^6$</td>
<td>edgeR</td>
<td>$0.827 \pm 0.0025$</td>
<td>$0.755 \pm 0.0029$</td>
</tr>
<tr>
<td>$10^6$</td>
<td>baySeq</td>
<td>$0.833 \pm 0.0023$</td>
<td>$0.745 \pm 0.0042$</td>
</tr>
<tr>
<td>$10^6$</td>
<td>DESeq</td>
<td>$0.820 \pm 0.0024$</td>
<td>$0.753 \pm 0.0029$</td>
</tr>
<tr>
<td>$10^6$</td>
<td>SAMseq</td>
<td>$0.780 \pm 0.0026$</td>
<td>$0.678 \pm 0.0038$</td>
</tr>
<tr>
<td>$10^7$</td>
<td>DEXUS</td>
<td>$0.936 \pm 0.0013$</td>
<td>$0.896 \pm 0.0020$</td>
</tr>
<tr>
<td>$10^7$</td>
<td>edgeR</td>
<td>$0.931 \pm 0.0014$</td>
<td>$0.896 \pm 0.0020$</td>
</tr>
<tr>
<td>$10^7$</td>
<td>baySeq</td>
<td>$0.920 \pm 0.0016$</td>
<td>$0.874 \pm 0.0047$</td>
</tr>
<tr>
<td>$10^7$</td>
<td>DESeq</td>
<td>$0.921 \pm 0.0016$</td>
<td>$0.888 \pm 0.0021$</td>
</tr>
<tr>
<td>$10^7$</td>
<td>SAMseq</td>
<td>$0.862 \pm 0.0044$</td>
<td>$0.713 \pm 0.0249$</td>
</tr>
<tr>
<td>$10^8$</td>
<td>DEXUS</td>
<td>$0.979 \pm 0.0006$</td>
<td>$0.961 \pm 0.0009$</td>
</tr>
<tr>
<td>$10^8$</td>
<td>edgeR</td>
<td>$0.977 \pm 0.0007$</td>
<td>$0.962 \pm 0.0010$</td>
</tr>
<tr>
<td>$10^8$</td>
<td>baySeq</td>
<td>$0.965 \pm 0.0010$</td>
<td>$0.939 \pm 0.0040$</td>
</tr>
<tr>
<td>$10^8$</td>
<td>DESeq</td>
<td>$0.972 \pm 0.0008$</td>
<td>$0.957 \pm 0.0010$</td>
</tr>
<tr>
<td>$10^8$</td>
<td>SAMseq</td>
<td>$0.877 \pm 0.0012$</td>
<td>$0.668 \pm 0.0114$</td>
</tr>
</tbody>
</table>

Table S6: Performance of methods for three known conditions with 2 replicates and a library size of $10^6$, $10^7$, and $10^8$. The best methods with respect to $\text{AUC}_{\text{ROC}}$ are baySeq and DEXUS. With respect to $\text{AUC}_{\text{PR}}$ edgeR and DEXUS perform best.
<table>
<thead>
<tr>
<th>Libsize</th>
<th>Method</th>
<th>AUC_{ROC}</th>
<th>AUC_{PR}</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^6</td>
<td>DEXUS</td>
<td>0.913±0.0015</td>
<td>0.882±0.0018</td>
</tr>
<tr>
<td>10^6</td>
<td>edgeR</td>
<td>0.907±0.0017</td>
<td>0.889±0.0019</td>
</tr>
<tr>
<td>10^6</td>
<td>baySeq</td>
<td>0.915±0.0015</td>
<td>0.874±0.0035</td>
</tr>
<tr>
<td>10^6</td>
<td>DESeq</td>
<td>0.905±0.0016</td>
<td>0.877±0.0019</td>
</tr>
<tr>
<td>10^6</td>
<td>SAMseq</td>
<td>0.890±0.0017</td>
<td>0.877±0.0170</td>
</tr>
<tr>
<td>10^7</td>
<td>DEXUS</td>
<td>0.982±0.0006</td>
<td>0.973±0.0008</td>
</tr>
<tr>
<td>10^7</td>
<td>edgeR</td>
<td>0.977±0.0009</td>
<td>0.969±0.0011</td>
</tr>
<tr>
<td>10^7</td>
<td>baySeq</td>
<td>0.969±0.0011</td>
<td>0.951±0.0039</td>
</tr>
<tr>
<td>10^7</td>
<td>DESeq</td>
<td>0.975±0.0008</td>
<td>0.966±0.0010</td>
</tr>
<tr>
<td>10^7</td>
<td>SAMseq</td>
<td>0.967±0.0010</td>
<td>0.957±0.0034</td>
</tr>
<tr>
<td>10^8</td>
<td>DEXUS</td>
<td>0.997±0.0002</td>
<td>0.995±0.0002</td>
</tr>
<tr>
<td>10^8</td>
<td>edgeR</td>
<td>0.996±0.0003</td>
<td>0.994±0.0004</td>
</tr>
<tr>
<td>10^8</td>
<td>baySeq</td>
<td>0.988±0.0007</td>
<td>0.976±0.0040</td>
</tr>
<tr>
<td>10^8</td>
<td>DESeq</td>
<td>0.995±0.0003</td>
<td>0.992±0.0004</td>
</tr>
<tr>
<td>10^8</td>
<td>SAMseq</td>
<td>0.992±0.0005</td>
<td>0.989±0.0006</td>
</tr>
</tbody>
</table>

Table S7: Performance of methods for three known conditions with 6 replicates and a library size of 10^6, 10^7, and 10^8. The best methods with respect to AUC_{ROC} are baySeq, edgeR, and DEXUS. With respect to AUC_{PR}, DEXUS performs best.

<table>
<thead>
<tr>
<th>Libsize</th>
<th>Method</th>
<th>AUC_{ROC}</th>
<th>AUC_{PR}</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^6</td>
<td>DEXUS</td>
<td>0.958±0.0012</td>
<td>0.945±0.0013</td>
</tr>
<tr>
<td>10^6</td>
<td>edgeR</td>
<td>0.952±0.0015</td>
<td>0.939±0.0016</td>
</tr>
<tr>
<td>10^6</td>
<td>baySeq</td>
<td>0.956±0.0011</td>
<td>0.931±0.0041</td>
</tr>
<tr>
<td>10^6</td>
<td>DESeq</td>
<td>0.953±0.0013</td>
<td>0.944±0.0013</td>
</tr>
<tr>
<td>10^6</td>
<td>SAMseq</td>
<td>0.942±0.0013</td>
<td>0.931±0.0078</td>
</tr>
<tr>
<td>10^7</td>
<td>DEXUS</td>
<td>0.996±0.0003</td>
<td>0.993±0.0004</td>
</tr>
<tr>
<td>10^7</td>
<td>edgeR</td>
<td>0.992±0.0006</td>
<td>0.990±0.0006</td>
</tr>
<tr>
<td>10^7</td>
<td>baySeq</td>
<td>0.986±0.0008</td>
<td>0.973±0.0035</td>
</tr>
<tr>
<td>10^7</td>
<td>DESeq</td>
<td>0.993±0.0005</td>
<td>0.990±0.0005</td>
</tr>
<tr>
<td>10^7</td>
<td>SAMseq</td>
<td>0.990±0.0006</td>
<td>0.986±0.0007</td>
</tr>
<tr>
<td>10^8</td>
<td>DEXUS</td>
<td>1.000±0.0001</td>
<td>1.000±0.0001</td>
</tr>
<tr>
<td>10^8</td>
<td>edgeR</td>
<td>0.999±0.0002</td>
<td>0.999±0.0003</td>
</tr>
<tr>
<td>10^8</td>
<td>baySeq</td>
<td>0.994±0.0006</td>
<td>0.983±0.0043</td>
</tr>
<tr>
<td>10^8</td>
<td>DESeq</td>
<td>0.999±0.0001</td>
<td>0.999±0.0002</td>
</tr>
<tr>
<td>10^8</td>
<td>SAMseq</td>
<td>0.998±0.0002</td>
<td>0.998±0.0009</td>
</tr>
</tbody>
</table>

Table S8: Performance of methods for three known conditions with 15 replicates and a library size of 10^6, 10^7, and 10^8. The best method with respect to both AUC_{ROC} and AUC_{PR} is DEXUS.
S4.2.4 Real World Data with Two Known Conditions

We compare the methods on real-world data, the “Mice Strains” data set, which has already been used for benchmarking RNA-Seq methods. In Bottomly et al. (2011), two strains of mice, C57BL/6J (B6) and DBA/2J (D2), were compared using both RNA-Seq and microarrays. The dataset consists of 21 lanes from male mice (10 of the B6 strain and 11 of D2 strain), produced using an Illumina GAIIx sequencing machine. The dataset was provided by the ReCount repository (Frazee et al. 2011) that is based on Ensembl 61 gene definitions. DEXUS found 157 genes that were significant using the mentioned test for differentially expression after Bonferroni correction at a significance level of 0.01. Of these 157 genes 97.5% were confirmed by at least one of the eight other methods, 91% by at least two other methods, and 85% by at least three other methods. 8% were confirmed by all eight methods. To compare the result of DEXUS to the results of the original publication (Bottomly et al. 2011), we used the authors’ read count data that is based on an older version of the Ensembl gene definitions (Ensembl 59). DEXUS identified 258 genes as differentially expressed. Of these 258 genes 245 were also identified in the original publication and confirmed by both Affymetrix and Illumina microarrays. The gene sets extracted by DEXUS are analyzed by the DAVID annotation tool (Huang et al. 2009b, a) for gene enrichment using gene ontology (Ashburner et al. 2000) and the INTERPRO data base (Hunter et al. 2012). Significant terms were “antigen processing and presentation” ($p = 9.7e-6$), “antigen processing an presentation of peptide antigen” ($p = 1.1e-5$), “Immunoglobulin/major histocompatibility complex, conserved site” ($p = 4.2e-4$), and “Immunoglobulin-like” ($p = 3.2e-4$). This shows that many transcripts that are differentially expressed between the two mice strains are related to the immune system.

S4.3 RNA-Seq Data with Unknown Conditions

The idea of DEXUS is to estimate the conditions and read counts belonging to them by a mixture model. This can also be done by a mixture of Gaussians. We compare DEXUS which is a mixture of negative binomials to a mixture of Gaussians to assess whether negative binomials are indeed the appropriate mixture components to model RNA-Seq data. We select mclust 4.0.0 (Fraley et al. 2012) as baseline method. It is used to model RNA-Seq data by a mixture of Gaussians. For the baseline method gene ranking was performed according to DEXUS’ I/NI value.

S4.3.1 Methods compared

We compare the following methods for differential gene expression in RNA-Seq data with unknown conditions:

- DEXUS,
- baseline method: mclust (Fraley et al. 2012; Fraley and Raftery 2002).

The baseline method is mixture of Gaussians clustering algorithm. We model the data with one, two and three Gaussians. We use the DEXUS I/NI value (see Subsection S3.3) to rank the transcripts according to the evidence of being differentially expressed. The values $\alpha$, $\mu$, and $\alpha_{ik}$ are provided by the Gaussian mixture EM algorithm.
In principle it is possible to test all possible partitions of the samples into two or more conditions and then apply standard RNA-Seq methods. However, this approach is not feasible because the number of partitions increases combinatorially. The number of partitions of a set with \( N \) elements is given by the Bell number \( B_N \).

\[
B_N = \frac{1}{e} \sum_{k=0}^{\infty} \frac{k^N}{k!}
\]  

(S74)

For multiple conditions, the number of data sets is the number of partitions minus one \( B_N - 1 \) (the set of all data is not considered). For \( N = 10 \) samples, the Bell number is \( B_{10} = 115,975 \), therefore a method has to run 115,974 data sets. For \( N = 20 \) samples, a method has to run 51,724,158,235,371 data sets.

### S4.3.2 Simulated Data Sets with Unknown Conditions

**Data Simulation.** We simulated datasets analogously to data with known conditions (see Subsection S3.3.1), except that the conditions are withhold from the methods. Furthermore, the conditions can have different numbers of replicates as expected for general study designs.

**Results.** Tab. S9, Tab. S10, and Tab. S11 show the results in terms of AUC_{ROC} and AUC_{PR} for DEXUS and mclust for library sizes \( 10^6 \), \( 10^7 \), and \( 10^8 \), respectively. In all experiments DEXUS outperforms the baseline method. This is not surprising as modeling with a negative binomial is supposed to perform better than modeling with a Gaussian distribution.

**Performance at different I/NI thresholds** Tab. S12, Tab. S13, and Tab. S14 show the results of DEXUS in terms of different performance measures like sensitivity and specificity at different I/NI thresholds and for library sizes \( 10^6 \), \( 10^7 \), and \( 10^8 \), respectively. Additional performance measures are given in Section S6.

**Performance for different fold change categories** We investigated whether DEXUS has a different performance on differentially expressed genes belonging to different fold change categories. Each data set in the simulated data consists of around 7,000 non-differentially expressed genes, around 1,000 genes with a log fold change of 0.5, around 1,000 genes with a log fold change of 1, and around 1,000 genes with a log fold change of 1.5. We assess the performance of DEXUS in terms of specificity and sensitivity on a data set of 8,000 genes (7,000 negatives and 1,000 positives), one data set for each fold change. The results for these different fold change categories are displayed in Tables S15, S16, and S17 for library sizes of \( 10^6 \), \( 10^7 \), and \( 10^8 \), respectively.

For the different fold change categories, the set of negatives (the 7,000 non-differentially expressed genes) is the same and the number of the false positives is the same (as the I/NI threshold is the same), therefore, also the specificity is the same. The sensitivity values increase with the log fold change. The reason is that genes with larger log fold changes lead to higher I/NI values (larger distances between the read count means), thus, easier to be detected. The lower the number of samples of the smaller condition, the lower the sensitivity. The signal of the smaller condition is more likely to be confounded with outliers. Table S shows that at a threshold of 0.05
## Table S9: Results of DEXUS and the baseline method (mclust) for unknown conditions (two conditions).

<table>
<thead>
<tr>
<th>C1/C2</th>
<th>Method</th>
<th>AUC&lt;sub&gt;ROC&lt;/sub&gt;</th>
<th>AUC&lt;sub&gt;PR&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/6</td>
<td>DEXUS</td>
<td>0.772±0.0036</td>
<td>0.580±0.0073</td>
</tr>
<tr>
<td>6/6</td>
<td>baseline method</td>
<td>0.648±0.0042</td>
<td>0.478±0.0077</td>
</tr>
<tr>
<td>9/3</td>
<td>DEXUS</td>
<td>0.773±0.0034</td>
<td>0.553±0.0065</td>
</tr>
<tr>
<td>9/3</td>
<td>baseline method</td>
<td>0.768±0.0037</td>
<td>0.401±0.0058</td>
</tr>
<tr>
<td>10/2</td>
<td>DEXUS</td>
<td>0.764±0.0038</td>
<td>0.517±0.0068</td>
</tr>
<tr>
<td>10/2</td>
<td>baseline method</td>
<td>0.603±0.0036</td>
<td>0.357±0.0055</td>
</tr>
<tr>
<td>11/1</td>
<td>DEXUS</td>
<td>0.733±0.0038</td>
<td>0.451±0.0064</td>
</tr>
<tr>
<td>12/12</td>
<td>baseline method</td>
<td>0.522±0.0031</td>
<td>0.274±0.0039</td>
</tr>
<tr>
<td>12/12</td>
<td>DEXUS</td>
<td>0.758±0.0034</td>
<td>0.598±0.0058</td>
</tr>
<tr>
<td>18/6</td>
<td>baseline method</td>
<td>0.669±0.0035</td>
<td>0.515±0.0063</td>
</tr>
<tr>
<td>18/6</td>
<td>DEXUS</td>
<td>0.782±0.0032</td>
<td>0.603±0.0054</td>
</tr>
<tr>
<td>20/4</td>
<td>baseline method</td>
<td>0.645±0.0038</td>
<td>0.447±0.0061</td>
</tr>
<tr>
<td>20/4</td>
<td>DEXUS</td>
<td>0.764±0.0034</td>
<td>0.587±0.0059</td>
</tr>
<tr>
<td>22/2</td>
<td>baseline method</td>
<td>0.627±0.0036</td>
<td>0.409±0.0057</td>
</tr>
<tr>
<td>22/2</td>
<td>DEXUS</td>
<td>0.741±0.0034</td>
<td>0.519±0.0060</td>
</tr>
<tr>
<td>22/2</td>
<td>baseline method</td>
<td>0.591±0.0032</td>
<td>0.356±0.0042</td>
</tr>
</tbody>
</table>

DEXUS consistently outperforms the baseline method.

## Table S10: Results of DEXUS and Gaussian mixtures (mclust) for unknown conditions (two conditions).

<table>
<thead>
<tr>
<th>C1/C2</th>
<th>Method</th>
<th>AUC&lt;sub&gt;ROC&lt;/sub&gt;</th>
<th>AUC&lt;sub&gt;PR&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/6</td>
<td>DEXUS</td>
<td>0.838±0.0035</td>
<td>0.745±0.0056</td>
</tr>
<tr>
<td>6/6</td>
<td>baseline method</td>
<td>0.728±0.0041</td>
<td>0.662±0.0070</td>
</tr>
<tr>
<td>9/3</td>
<td>DEXUS</td>
<td>0.843±0.0027</td>
<td>0.714±0.0060</td>
</tr>
<tr>
<td>9/3</td>
<td>baseline method</td>
<td>0.702±0.0039</td>
<td>0.568±0.0069</td>
</tr>
<tr>
<td>10/2</td>
<td>DEXUS</td>
<td>0.832±0.0028</td>
<td>0.663±0.0060</td>
</tr>
<tr>
<td>10/2</td>
<td>baseline method</td>
<td>0.673±0.0036</td>
<td>0.495±0.0056</td>
</tr>
<tr>
<td>11/1</td>
<td>DEXUS</td>
<td>0.792±0.0041</td>
<td>0.559±0.0070</td>
</tr>
<tr>
<td>11/1</td>
<td>baseline method</td>
<td>0.512±0.0029</td>
<td>0.314±0.0038</td>
</tr>
<tr>
<td>12/12</td>
<td>DEXUS</td>
<td>0.833±0.0026</td>
<td>0.755±0.0042</td>
</tr>
<tr>
<td>12/12</td>
<td>baseline method</td>
<td>0.764±0.0035</td>
<td>0.706±0.0068</td>
</tr>
<tr>
<td>18/6</td>
<td>DEXUS</td>
<td>0.851±0.0032</td>
<td>0.771±0.0054</td>
</tr>
<tr>
<td>18/6</td>
<td>baseline method</td>
<td>0.743±0.0036</td>
<td>0.632±0.0067</td>
</tr>
<tr>
<td>20/4</td>
<td>DEXUS</td>
<td>0.847±0.0034</td>
<td>0.745±0.0059</td>
</tr>
<tr>
<td>20/4</td>
<td>baseline method</td>
<td>0.719±0.0032</td>
<td>0.573±0.0062</td>
</tr>
<tr>
<td>22/2</td>
<td>DEXUS</td>
<td>0.817±0.0035</td>
<td>0.648±0.0060</td>
</tr>
<tr>
<td>22/2</td>
<td>baseline method</td>
<td>0.674±0.0035</td>
<td>0.484±0.0051</td>
</tr>
</tbody>
</table>

DEXUS consistently outperforms the baseline method.
Table S11: Results of DEXUS and Gaussian mixtures (mclust) for unknown conditions (two conditions). “C1/C2” reports the number of samples for each condition. “Method” gives the name of the method and AUC_{ROC} and AUC_{PR} the according performances. The library size was 10^8 for all experiments. DEXUS consistently outperforms the baseline method.

<table>
<thead>
<tr>
<th>C1/C2</th>
<th>Method</th>
<th>AUC_{ROC}</th>
<th>AUC_{PR}</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/6</td>
<td>DEXUS</td>
<td>0.914±0.0025</td>
<td>0.874±0.0038</td>
</tr>
<tr>
<td>6/6</td>
<td>baseline method</td>
<td>0.834±0.0035</td>
<td>0.824±0.0045</td>
</tr>
<tr>
<td>9/3</td>
<td>DEXUS</td>
<td>0.921±0.0020</td>
<td>0.852±0.0054</td>
</tr>
<tr>
<td>9/3</td>
<td>baseline method</td>
<td>0.813±0.0035</td>
<td>0.762±0.0065</td>
</tr>
<tr>
<td>10/2</td>
<td>DEXUS</td>
<td>0.908±0.0025</td>
<td>0.791±0.0053</td>
</tr>
<tr>
<td>10/2</td>
<td>baseline method</td>
<td>0.785±0.0033</td>
<td>0.681±0.0062</td>
</tr>
<tr>
<td>11/1</td>
<td>DEXUS</td>
<td>0.862±0.0027</td>
<td>0.652±0.0060</td>
</tr>
<tr>
<td>11/1</td>
<td>baseline method</td>
<td>0.513±0.0026</td>
<td>0.328±0.0038</td>
</tr>
<tr>
<td>12/12</td>
<td>DEXUS</td>
<td>0.912±0.0023</td>
<td>0.880±0.0031</td>
</tr>
<tr>
<td>12/12</td>
<td>baseline method</td>
<td>0.863±0.0030</td>
<td>0.856±0.0037</td>
</tr>
<tr>
<td>18/6</td>
<td>DEXUS</td>
<td>0.931±0.0023</td>
<td>0.899±0.0036</td>
</tr>
<tr>
<td>18/6</td>
<td>baseline method</td>
<td>0.849±0.0031</td>
<td>0.816±0.0050</td>
</tr>
<tr>
<td>20/4</td>
<td>DEXUS</td>
<td>0.926±0.0024</td>
<td>0.872±0.0047</td>
</tr>
<tr>
<td>20/4</td>
<td>baseline method</td>
<td>0.828±0.0031</td>
<td>0.762±0.0051</td>
</tr>
<tr>
<td>22/2</td>
<td>DEXUS</td>
<td>0.897±0.0028</td>
<td>0.770±0.0055</td>
</tr>
<tr>
<td>22/2</td>
<td>baseline method</td>
<td>0.786±0.0032</td>
<td>0.654±0.0070</td>
</tr>
</tbody>
</table>

Strong signals (log fold changes of 1.5) can still be reliably detected even if they appear in only a few samples (“11/1” or “22/2”).
Table S12: The performance of DEXUS in terms of sensitivity and specificity at the detection of differential expression with unknown conditions. The first row reports the different thresholds that were used for the I/NI value. The first column “C1/C2” reports the number of replicates for the first and second condition. The other columns report sensitivity and specificity of DEXUS at different I/NI thresholds. The library size was 10^6 for all experiments.

<table>
<thead>
<tr>
<th>I/NI threshold</th>
<th>0.025</th>
<th>0.05</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1/C2 specificity</td>
<td>sensitivity</td>
<td>specificity</td>
<td>sensitivity</td>
</tr>
<tr>
<td>6/6</td>
<td>0.938 ± 0.003</td>
<td>0.327 ± 0.009</td>
<td>0.955 ± 0.003</td>
</tr>
<tr>
<td>9/3</td>
<td>0.938 ± 0.003</td>
<td>0.362 ± 0.008</td>
<td>0.955 ± 0.002</td>
</tr>
<tr>
<td>10/2</td>
<td>0.938 ± 0.003</td>
<td>0.362 ± 0.010</td>
<td>0.955 ± 0.002</td>
</tr>
<tr>
<td>11/1</td>
<td>0.938 ± 0.003</td>
<td>0.319 ± 0.009</td>
<td>0.955 ± 0.002</td>
</tr>
<tr>
<td>12/12</td>
<td>0.959 ± 0.002</td>
<td>0.281 ± 0.009</td>
<td>0.978 ± 0.002</td>
</tr>
<tr>
<td>18/6</td>
<td>0.959 ± 0.002</td>
<td>0.332 ± 0.010</td>
<td>0.978 ± 0.002</td>
</tr>
<tr>
<td>20/4</td>
<td>0.959 ± 0.002</td>
<td>0.337 ± 0.009</td>
<td>0.979 ± 0.002</td>
</tr>
<tr>
<td>22/2</td>
<td>0.959 ± 0.002</td>
<td>0.295 ± 0.008</td>
<td>0.978 ± 0.002</td>
</tr>
<tr>
<td>Mean</td>
<td>0.949 ± 0.011</td>
<td>0.327 ± 0.029</td>
<td>0.967 ± 0.012</td>
</tr>
</tbody>
</table>

Table S13: The performance of DEXUS in terms of sensitivity and specificity at the detection of differential expression with unknown conditions. The first row reports the different thresholds that were used for the I/NI value. The first column “C1/C2” reports the number of replicates for the first and second condition. The other columns report sensitivity and specificity of DEXUS at different I/NI thresholds. The library size was 10^7 for all experiments.

<table>
<thead>
<tr>
<th>I/NI threshold</th>
<th>0.025</th>
<th>0.05</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1/C2 specificity</td>
<td>sensitivity</td>
<td>specificity</td>
<td>sensitivity</td>
</tr>
<tr>
<td>6/6</td>
<td>0.896 ± 0.004</td>
<td>0.550 ± 0.011</td>
<td>0.940 ± 0.003</td>
</tr>
<tr>
<td>9/3</td>
<td>0.897 ± 0.004</td>
<td>0.600 ± 0.009</td>
<td>0.941 ± 0.003</td>
</tr>
<tr>
<td>10/2</td>
<td>0.898 ± 0.004</td>
<td>0.597 ± 0.009</td>
<td>0.941 ± 0.003</td>
</tr>
<tr>
<td>11/1</td>
<td>0.898 ± 0.004</td>
<td>0.516 ± 0.010</td>
<td>0.941 ± 0.003</td>
</tr>
<tr>
<td>12/12</td>
<td>0.940 ± 0.003</td>
<td>0.516 ± 0.009</td>
<td>0.975 ± 0.002</td>
</tr>
<tr>
<td>18/6</td>
<td>0.941 ± 0.003</td>
<td>0.590 ± 0.010</td>
<td>0.976 ± 0.002</td>
</tr>
<tr>
<td>20/4</td>
<td>0.940 ± 0.003</td>
<td>0.590 ± 0.010</td>
<td>0.975 ± 0.002</td>
</tr>
<tr>
<td>22/2</td>
<td>0.940 ± 0.003</td>
<td>0.497 ± 0.010</td>
<td>0.975 ± 0.002</td>
</tr>
<tr>
<td>Mean</td>
<td>0.919 ± 0.023</td>
<td>0.557 ± 0.042</td>
<td>0.958 ± 0.018</td>
</tr>
</tbody>
</table>
Table S14: The performance of DEXUS in terms of sensitivity and specificity at the detection of differential expression with unknown conditions. The first row reports the different thresholds that were used for the I/NI value. The first column “C1/C2” reports the number of replicates for the first and second condition. The other columns report sensitivity and specificity of DEXUS at different I/NI thresholds. The library size was $10^6$ for all experiments.

<table>
<thead>
<tr>
<th>I/NI threshold</th>
<th>C1/C2</th>
<th>specificity</th>
<th>sensitivity</th>
<th>specificity</th>
<th>sensitivity</th>
<th>specificity</th>
<th>sensitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.025</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6/6</td>
<td>0.893 ± 0.003</td>
<td>0.775 ± 0.009</td>
<td>0.951 ± 0.002</td>
<td>0.720 ± 0.009</td>
<td>0.985 ± 0.002</td>
<td>0.646 ± 0.009</td>
<td></td>
</tr>
<tr>
<td>9/3</td>
<td>0.893 ± 0.004</td>
<td>0.827 ± 0.006</td>
<td>0.766 ± 0.007</td>
<td>0.985 ± 0.001</td>
<td>0.580 ± 0.008</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/2</td>
<td>0.893 ± 0.003</td>
<td>0.819 ± 0.008</td>
<td>0.950 ± 0.002</td>
<td>0.656 ± 0.009</td>
<td>0.985 ± 0.001</td>
<td>0.325 ± 0.009</td>
<td></td>
</tr>
<tr>
<td>11/1</td>
<td>0.893 ± 0.003</td>
<td>0.677 ± 0.009</td>
<td>0.951 ± 0.002</td>
<td>0.351 ± 0.008</td>
<td>0.985 ± 0.001</td>
<td>0.020 ± 0.003</td>
<td></td>
</tr>
<tr>
<td>12/12</td>
<td>0.945 ± 0.002</td>
<td>0.735 ± 0.008</td>
<td>0.982 ± 0.001</td>
<td>0.665 ± 0.008</td>
<td>0.996 ± 0.001</td>
<td>0.610 ± 0.009</td>
<td></td>
</tr>
<tr>
<td>18/6</td>
<td>0.945 ± 0.003</td>
<td>0.816 ± 0.008</td>
<td>0.982 ± 0.002</td>
<td>0.743 ± 0.009</td>
<td>0.996 ± 0.001</td>
<td>0.570 ± 0.011</td>
<td></td>
</tr>
<tr>
<td>20/4</td>
<td>0.945 ± 0.003</td>
<td>0.810 ± 0.008</td>
<td>0.982 ± 0.002</td>
<td>0.625 ± 0.009</td>
<td>0.996 ± 0.001</td>
<td>0.308 ± 0.009</td>
<td></td>
</tr>
<tr>
<td>22/2</td>
<td>0.946 ± 0.002</td>
<td>0.650 ± 0.009</td>
<td>0.982 ± 0.001</td>
<td>0.325 ± 0.008</td>
<td>0.996 ± 0.001</td>
<td>0.006 ± 0.002</td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>0.919 ± 0.028</td>
<td>0.764 ± 0.069</td>
<td>0.966 ± 0.017</td>
<td>0.606 ± 0.172</td>
<td>0.991 ± 0.006</td>
<td>0.383 ± 0.261</td>
<td></td>
</tr>
</tbody>
</table>

Table S15: The performance of DEXUS in terms of sensitivity and specificity in detecting differential expression with unknown conditions. The results are separately reported for the three different fold change categories. The first column “C1/C2” contains the numbers of replicates for the first and second condition. The other columns list sensitivity and specificity of DEXUS at different I/NI thresholds as the average across 100 data sets. The first, second, and third value in each cell corresponds to a log fold change of 0.5, 1, and 1.5, respectively. The library size was $10^6$ for all experiments.

<table>
<thead>
<tr>
<th>I/NI threshold</th>
<th>C1/C2</th>
<th>specificity</th>
<th>sensitivity</th>
<th>specificity</th>
<th>sensitivity</th>
<th>specificity</th>
<th>sensitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.025</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6/6</td>
<td>0.940(0.940)</td>
<td>0.10(0.28)/0.45</td>
<td>0.950(0.950)</td>
<td>0.08(0.27)/0.45</td>
<td>0.980(0.980)</td>
<td>0.04(0.24)/0.43</td>
<td></td>
</tr>
<tr>
<td>9/3</td>
<td>0.940(0.940)</td>
<td>0.13/0.32(0.48)</td>
<td>0.950(0.950)</td>
<td>0.09/0.30(0.47)</td>
<td>0.980(0.980)</td>
<td>0.03(0.22)/0.45</td>
<td></td>
</tr>
<tr>
<td>10/2</td>
<td>0.940(0.940)</td>
<td>0.13/0.32(0.47)</td>
<td>0.950(0.950)</td>
<td>0.08/0.29(0.47)</td>
<td>0.980(0.980)</td>
<td>0.03(0.08)/0.38</td>
<td></td>
</tr>
<tr>
<td>11/1</td>
<td>0.940(0.940)</td>
<td>0.10/0.28(0.43)</td>
<td>0.950(0.950)</td>
<td>0.06/0.12(0.38)</td>
<td>0.980(0.980)</td>
<td>0.03(0.04)/0.06</td>
<td></td>
</tr>
<tr>
<td>12/12</td>
<td>0.960(0.960)</td>
<td>0.07/0.27(0.44)</td>
<td>0.980(0.980)</td>
<td>0.04/0.24(0.42)</td>
<td>0.990(0.990)</td>
<td>0.01(0.20)/0.40</td>
<td></td>
</tr>
<tr>
<td>18/6</td>
<td>0.960(0.960)</td>
<td>0.10/0.32(0.49)</td>
<td>0.980(0.980)</td>
<td>0.06/0.29(0.47)</td>
<td>0.990(0.990)</td>
<td>0.01(0.20)/0.43</td>
<td></td>
</tr>
<tr>
<td>20/4</td>
<td>0.960(0.960)</td>
<td>0.10/0.33(0.49)</td>
<td>0.980(0.980)</td>
<td>0.03/0.27(0.47)</td>
<td>0.990(0.990)</td>
<td>0.01(0.03)/0.37</td>
<td></td>
</tr>
<tr>
<td>22/2</td>
<td>0.960(0.960)</td>
<td>0.06/0.29(0.45)</td>
<td>0.980(0.980)</td>
<td>0.03/0.07(0.37)</td>
<td>0.990(0.990)</td>
<td>0.01(0.01)/0.02</td>
<td></td>
</tr>
</tbody>
</table>
Table S16: The performance of DEXUS in terms of sensitivity and specificity in detecting differential expression with unknown conditions. The results are separately reported for the three different fold change categories. The first column “C1/C2” contains the numbers of replicates for the first and second condition. The other columns list sensitivity and specificity of DEXUS at different I/NI thresholds as the average across 100 data sets. The first, second, and third value in each cell corresponds to a log fold change of 0.5, 1, and 1.5, respectively. The library size was $10^8$ for all experiments.

<table>
<thead>
<tr>
<th>I/NI threshold</th>
<th>0.025</th>
<th>0.05</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1/C2</td>
<td>specificity</td>
<td>sensitivity</td>
<td>specificity</td>
</tr>
<tr>
<td>6/6</td>
<td>0.90/0.90</td>
<td>0.25/0.62</td>
<td>0.94/0.94</td>
</tr>
<tr>
<td>9/3</td>
<td>0.90/0.90</td>
<td>0.33/0.65</td>
<td>0.94/0.94</td>
</tr>
<tr>
<td>10/2</td>
<td>0.90/0.90</td>
<td>0.33/0.65</td>
<td>0.94/0.94</td>
</tr>
<tr>
<td>11/1</td>
<td>0.90/0.90</td>
<td>0.19/0.59</td>
<td>0.94/0.94</td>
</tr>
<tr>
<td>12/12</td>
<td>0.94/0.94</td>
<td>0.18/0.60</td>
<td>0.98/0.98</td>
</tr>
<tr>
<td>18/6</td>
<td>0.94/0.94</td>
<td>0.30/0.66</td>
<td>0.98/0.98</td>
</tr>
<tr>
<td>20/4</td>
<td>0.94/0.94</td>
<td>0.30/0.66</td>
<td>0.98/0.98</td>
</tr>
<tr>
<td>22/2</td>
<td>0.94/0.94</td>
<td>0.12/0.60</td>
<td>0.98/0.98</td>
</tr>
</tbody>
</table>

Table S17: The performance of DEXUS in terms of sensitivity and specificity in detecting differential expression with unknown conditions. The results are separately reported for the three different fold change categories. The first column “C1/C2” contains the numbers of replicates for the first and second condition. The other columns list sensitivity and specificity of DEXUS at different I/NI thresholds as the average across 100 data sets. The first, second, and third value in each cell corresponds to a log fold change of 0.5, 1, and 1.5, respectively. The library size was $10^8$ for all experiments.

<table>
<thead>
<tr>
<th>I/NI threshold</th>
<th>0.025</th>
<th>0.05</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1/C2</td>
<td>specificity</td>
<td>sensitivity</td>
<td>specificity</td>
</tr>
<tr>
<td>6/6</td>
<td>0.89/0.89</td>
<td>0.47/0.89</td>
<td>0.95/0.95</td>
</tr>
<tr>
<td>9/3</td>
<td>0.89/0.89</td>
<td>0.60/0.91</td>
<td>0.95/0.95</td>
</tr>
<tr>
<td>10/2</td>
<td>0.89/0.89</td>
<td>0.59/0.90</td>
<td>0.95/0.95</td>
</tr>
<tr>
<td>11/1</td>
<td>0.89/0.89</td>
<td>0.24/0.85</td>
<td>0.95/0.95</td>
</tr>
<tr>
<td>12/12</td>
<td>0.95/0.95</td>
<td>0.36/0.88</td>
<td>0.98/0.98</td>
</tr>
<tr>
<td>18/6</td>
<td>0.95/0.95</td>
<td>0.56/0.91</td>
<td>0.98/0.98</td>
</tr>
<tr>
<td>20/4</td>
<td>0.95/0.95</td>
<td>0.55/0.91</td>
<td>0.98/0.98</td>
</tr>
<tr>
<td>22/2</td>
<td>0.95/0.95</td>
<td>0.14/0.85</td>
<td>0.98/0.98</td>
</tr>
</tbody>
</table>
S4.3.3 The “Nigerian HapMap” data set

Pickrell et al. (2010) sequenced RNA from 69 Nigerian HapMap individuals to study expression quantitative trait loci (eQTLs). The read count data was provided by the ReCount repository (Frazee et al. 2011). As in previous experiments, DEXUS was applied to this data with its default parameters and ranked genes according to the I/NI value. The read counts of top-ranked genes and the conditions identified by DEXUS are visualized as a heatmap in Fig. S16.

Figure S16: Heatmap of the normalized read counts of the twelve genes with the largest I/NI values for the “Nigerian HapMap” data set. Colors range from white for low expression to blue for high expression. The columns displays different HapMap individuals. The rows show the gene symbols of the top-ranked genes. Red crosses indicate that these samples belong to the minor condition. At the right hand side of the heatmap, each gene is annotated by the minimum (“>”), the median of two conditions (“m1” and “m2”), and the maximum (“<”) read count.

Five out of the twelve top-ranked genes are located on the Y chromosome (RPS4Y1, CY-orf15A, EIF1AY, TMSB4Y, RPS4Y2). For these genes the conditions that DEXUS identified are related to the sex. For four of the twelve top-ranked genes at least one eQTL is known. For ZFP57 the eQTL is the single nucleotide polymorphism (SNP) rs1736924 with a minor allele frequency (MAF) of 0.14 (Pickrell et al. 2010). CDH1 has 6 eQTLs, one of which is SNP rs7196495 with a MAF of 0.22 (Zeller et al. 2010). CLLU1OS possesses the eQTL SNP rs12580153 with a MAF of 0.19 (Dimas et al. 2009). L1TD1 has 2 eQTLs, one of which is SNP rs12137088 with a MAF 0.30 (Veyrieras et al. 2008). Since the MAFs are large, it is plausible that the minor alleles are observed in the HapMap data set and that they lead to differential expressions of the associated genes. The conditions that were found by DEXUS correspond to the alleles of corresponding SNPs.

The HapMap samples are lymphoblastoid cells, therefore we confirmed that the genes detected by DEXUS are indeed expressed in lymphoblastoid cell lines. The gene NLRP2, ranked 11th by DEXUS, is expressed in lymphoblastoid cells but with large variability (Halbritter et al. 2011) as shown in Figure S17. NLRP2 is expressed in the HapMap individuals but in some very low. Schlattl et al. (2011) identified a copy number variable region that covers NLRP2 partially and may be the cause of differential expression. Therefore, the conditions that DEXUS identified for NLRP2 seem to be related to copy number states of the samples. Copy number states might also
cause differential expression of MKRN3 that was ranked 12th by DEXUS. Pinto et al. (2007) and Redon et al. (2006) identified a copy number variable region covering MKRN3. However, the interpretation of MKRN3’s conditions is difficult since only the paternal copy of MKRN3 is expressed.

We analyzed DEXUS’ I/NI value ranking of transcripts. Genes on the X chromosome were ranked significantly higher than other genes ($p = 3.0e-12$) which can be explained by sex related transcripts. An analog test for the Y chromosome was not significant, because too few genes were expressed. However, as already mentioned, out of the twelve top-ranked genes, five are located on the Y chromosome. At an I/NI threshold of 0.1, DEXUS called 366 differentially expressed genes. Gene enrichment analysis showed that the called genes are associated with the extracellular region. Significant GO terms were “extracellular space”, “extracellular region part”, and “extracellular region” with $p = 2.2e-5$, $p = 8.8e-5$, and $p = 0.01$, respectively ($p$-values were corrected for multiple testing by the Benjamini-Hochberg procedure). These GO terms are in agreement with characteristics of lymphoblastoid cells. Tab. S18 shows all significant GO terms of this data set.

<table>
<thead>
<tr>
<th>Term</th>
<th>Count</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO:0005615 extracellular space</td>
<td>35</td>
<td>2e-5</td>
</tr>
<tr>
<td>GO:0044421 extracellular region part</td>
<td>41</td>
<td>9e-5</td>
</tr>
<tr>
<td>GO:0005529 sugar binding</td>
<td>16</td>
<td>0.001</td>
</tr>
<tr>
<td>GO:0042379 chemokine receptor binding</td>
<td>9</td>
<td>0.001</td>
</tr>
<tr>
<td>GO:0008009 chemokine activity</td>
<td>9</td>
<td>0.001</td>
</tr>
<tr>
<td>GO:0005125 cytokine activity</td>
<td>15</td>
<td>0.003</td>
</tr>
<tr>
<td>GO:0007267 cell-cell signaling</td>
<td>30</td>
<td>0.004</td>
</tr>
<tr>
<td>GO:0005886 plasma membrane</td>
<td>95</td>
<td>0.011</td>
</tr>
<tr>
<td>GO:0031982 vesicle</td>
<td>27</td>
<td>0.011</td>
</tr>
<tr>
<td>GO:0031988 membrane-bounded vesicle</td>
<td>24</td>
<td>0.012</td>
</tr>
<tr>
<td>GO:0044459 plasma membrane part</td>
<td>63</td>
<td>0.014</td>
</tr>
<tr>
<td>GO:0030246 carbohydrate binding</td>
<td>19</td>
<td>0.014</td>
</tr>
<tr>
<td>GO:0030054 cell junction</td>
<td>22</td>
<td>0.016</td>
</tr>
<tr>
<td>GO:0005576 extracellular region</td>
<td>59</td>
<td>0.016</td>
</tr>
<tr>
<td>GO:0031410 cytoplasmic vesicle</td>
<td>25</td>
<td>0.018</td>
</tr>
<tr>
<td>GO:0016023 cytoplasmic membrane-bounded vesicle</td>
<td>22</td>
<td>0.027</td>
</tr>
<tr>
<td>GO:0003002 regionalization</td>
<td>14</td>
<td>0.044</td>
</tr>
<tr>
<td>GO:0005865 striated muscle thin filament</td>
<td>4</td>
<td>0.046</td>
</tr>
<tr>
<td>GO:0008021 synaptic vesicle</td>
<td>7</td>
<td>0.049</td>
</tr>
</tbody>
</table>

Table S18: Significant GO terms of the differentially expressed genes of the “Nigerian HapMap” data set. The first column presents the GO identifier and the short name of the GO term. The second column the number of genes that belong to that GO term, and the last column shows the $p$-values after Benjamini-Hochberg correction.

S4.3.4 The “European HapMap” data set

We analyzed the RNA-Seq data of 60 individuals from the HapMap cohort from Montgomery et al. (2010) which were provided by the ReCount repository (Frazee et al. 2011). Again, DEXUS
Figure S17: Expression values of gene NLRP2 in log10 RPM (reads per million mapped reads) format as provided by the GeneProf data base (Halbritter et al. 2011). The data is taken from 113 public data sets. The gene NLRP2 is expressed and highly variable in lymphoblastoid cell lines.

was applied to these data with its default parameters and ranked genes according to the I/NI value. The read counts of top-ranked genes and the identified conditions are visualized as a heatmap in Fig. S18.

RPS4Y1 is the gene with the largest I/NI value, differentially expressed between males and females, and located on the Y chromosome. The genes CYorf15A and TMSB4Y, ranked fourth and fifth according to the I/NI value, are located on the Y chromosome, too. As in “Nigerian HapMap” data set, ZFP57 was detected as being differentially expressed. Two of the twelve top-ranked genes have eQTLs. CLLU1OS has as eQTL the SNP rs12580153 with a minor allele frequency of 0.19 (Dimas et al. 2009). POU2F3 has as eQTL the SNP rs2847497 with a MAF of 0.14 (Schadt et al. 2008). As in the “Nigerian HapMap” data set some top ranked genes, like NLRP2 (rank 11, again), were differentially expressed due to variable copy numbers (Schlattl et al. 2011). Again the conditions are associated with copy numbers. For the genes T, PRSS21, and RASSF10 DEXUS identified two conditions the interpretation of which is yet to be found. We could neither interpret the conditions by sex, nor allele, nor copy number state. DEXUS hints at a new source of variability in gene expression. The second ranked gene T, the third ranked gene PRSS21, and the twelfth ranked gene RASSF10 are expressed in B-lymphoblastoid cells (Wu et al. 2009; The ENCODE Project Consortium 2012), the cell type of the HapMap samples. The high expression variability of T and PRSS21 in the B-lymphoblastoid cell line was already reported by the ENCODE Project (The ENCODE Project Consortium 2012). The ENCODE Project expression values for the genes T, PRSS21, and RASSF10 are visualized in Fig. S19, S20, and S21.

When analyzing the I/NI value ranking, we found that genes on the X chromosome are ranked
Figure S18: Heatmap of the read counts of the twelve genes with the largest I/NI values for the “European HapMap” data set. Colors range from white for low expression to blue for high expression. The columns displays different HapMap individuals. The rows show the genes symbols of the top-ranked genes. Red crosses indicate that these samples belong to the minor condition. At the right hand side of the heatmap, each gene is annotated by the minimum (“>”), the median of two conditions (“m1” and “m2”), and the maximum (“<”) read count.

significantly higher ($p = 8.0e-6$, Wilcoxon test). The analogous test for the Y chromosome was not significant as too few genes were expressed. However, three out of the twelve top-ranked genes with the largest I/NI value are located on the Y chromosome.

At an I/NI threshold of 0.1, DEXUS called 680 differentially expressed genes. Gene enrichment analysis showed that the called genes are associated with ion transport. Significant GO terms were “ion transport”, “potassium ion transport”, and “plasma membrane part” with $p = 0.04$, $p = 4.3e-03$, and $p = 0.027$, respectively ($p$-values were corrected for multiple testing by the Benjamini-Hochberg procedure). These GO terms are in agreement with characteristics of lymphoblastoid cells. Tab. S19 shows all significant GO terms of this data set.

S4.3.5 The “Primate Liver” data set

Blekhman et al. (2010) investigated the differences in alternative splicing in liver tissue between humans, chimpanzees and rhesus macaques. For this purpose, they sequenced the RNA of three male and three female liver samples from each species. They focused on the expression values of exons that had reliably determined orthologs in all species. Read counts for exons were provided by the original publication which used gene models from Ensemble (Release 50). After pooling technical replicates, DEXUS ranked genes according to the I/NI value using its default parameters. The ten top-ranked genes are visualized in Fig. S22 which shows strong differential expression between the species. For all these genes DEXUS determined one of the three species as minor condition without having been provided with this information. Interestingly, out of the ten top-ranked genes, six are human pseudogenes: AC010591.10, AC105383.3, AC093874.3-1, AC105383.3, AL132855.4, and UOX. These genes are inactive in humans because of recent structural rearrangements (Balasubramanian et al. 2009). Since the rearrangements are recent, their
Figure S19: Expression values of gene T in RPM (reads per million mapped reads) format as provided by the GeneProf data base (Halbritter et al. 2011). The data is taken from 113 public data sets. The gene T is expressed and highly variable in lymphoblastoid cell lines.

Figure S20: Expression values of PRSS21 in log10 RPM (reads per million mapped reads) format as provided by the GeneProf data base (Halbritter et al. 2011). The data is taken from 113 public data sets. The gene PRSS21 is expressed and highly variable in lymphoblastoid cell lines.
<table>
<thead>
<tr>
<th>Term</th>
<th>Count</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO:0005261 cation channel activity</td>
<td>28</td>
<td>0.001</td>
</tr>
<tr>
<td>GO:0006813 potassium ion transport</td>
<td>20</td>
<td>0.004</td>
</tr>
<tr>
<td>GO:0022838 substrate specific channel activity</td>
<td>31</td>
<td>0.005</td>
</tr>
<tr>
<td>GO:0022843 voltage-gated cation channel activity</td>
<td>17</td>
<td>0.005</td>
</tr>
<tr>
<td>GO:0005267 potassium channel activity</td>
<td>16</td>
<td>0.005</td>
</tr>
<tr>
<td>GO:0046873 metal ion transmembrane transporter activity</td>
<td>29</td>
<td>0.005</td>
</tr>
<tr>
<td>GO:0022803 passive transmembrane transporter activity</td>
<td>20</td>
<td>0.005</td>
</tr>
<tr>
<td>GO:0005244 voltage-gated ion channel activity</td>
<td>20</td>
<td>0.006</td>
</tr>
<tr>
<td>GO:0022832 voltage-gated channel activity</td>
<td>20</td>
<td>0.006</td>
</tr>
<tr>
<td>GO:0005216 ion channel activity</td>
<td>30</td>
<td>0.006</td>
</tr>
<tr>
<td>GO:0015267 channel activity</td>
<td>32</td>
<td>0.007</td>
</tr>
<tr>
<td>GO:0030955 potassium ion binding</td>
<td>16</td>
<td>0.008</td>
</tr>
<tr>
<td>GO:0031420 alkali metal ion binding</td>
<td>22</td>
<td>0.008</td>
</tr>
<tr>
<td>GO:0022836 gated channel activity</td>
<td>27</td>
<td>0.009</td>
</tr>
<tr>
<td>GO:0005249 voltage-gated potassium channel activity</td>
<td>13</td>
<td>0.011</td>
</tr>
<tr>
<td>GO:0044459 plasma membrane part</td>
<td>103</td>
<td>0.027</td>
</tr>
<tr>
<td>GO:0051254 positive regulation of RNA metabolic process</td>
<td>34</td>
<td>0.030</td>
</tr>
<tr>
<td>GO:0034702 ion channel complex</td>
<td>19</td>
<td>0.031</td>
</tr>
<tr>
<td>GO:0030001 metal ion transport</td>
<td>33</td>
<td>0.031</td>
</tr>
<tr>
<td>GO:0045944 positive regulation of transcription from RNA polymerase II promoter</td>
<td>28</td>
<td>0.032</td>
</tr>
<tr>
<td>GO:0051173 positive regulation of nitrogen compound metabolic process</td>
<td>41</td>
<td>0.033</td>
</tr>
<tr>
<td>GO:0045941 positive regulation of transcription</td>
<td>38</td>
<td>0.033</td>
</tr>
<tr>
<td>GO:0031328 positive regulation of cellular biosynthetic process</td>
<td>43</td>
<td>0.034</td>
</tr>
<tr>
<td>GO:0015672 monovalent inorganic cation transport</td>
<td>27</td>
<td>0.034</td>
</tr>
<tr>
<td>GO:0009891 positive regulation of biosynthetic process</td>
<td>43</td>
<td>0.035</td>
</tr>
<tr>
<td>GO:0045893 positive regulation of transcription, DNA-dependent</td>
<td>34</td>
<td>0.036</td>
</tr>
<tr>
<td>GO:0006811 ion transport</td>
<td>48</td>
<td>0.042</td>
</tr>
<tr>
<td>GO:0010628 positive regulation of gene expression</td>
<td>39</td>
<td>0.043</td>
</tr>
<tr>
<td>GO:0034703 cation channel complex</td>
<td>15</td>
<td>0.046</td>
</tr>
</tbody>
</table>

Table S19: Significant GO terms of the differentially expressed genes of the “European HapMap” data set. The first column presents the GO identifier and the short name of the GO term. The second column the number of genes that belong to that GO term, and the last column shows the p-values after Benjamini-Hochberg’s correction.

orthologs can reliably be identified in other primates. Differential expression is detected because these orthologs are still transcribed in chimpanzees or in rhesus macaques.

Many of the ten top-ranked genes are associated with liver pathways. Differential expression of these genes between species might have arisen from different diets. Examples of such genes are the human pseudogene UOX that is required to catalyze the oxidation of uric acid to allantoin in *Macaca mulatta*, APB1 and GSTM5 which participate in degradation and detoxification pathways, VNN3 which helps to recycle vitamin B5, and CHFR2 which is associated with lipoproteins.

Thresholding the I/NI call at 0.1, DEXUS called 3384 genes (16% of all genes) as differentially expressed. A gene set enrichment analysis found GO-Terms “intrinsic to plasma membrane” \((p = 7.9e-7)\) and “integral to plasma membrane” \((p = 4.0e-6)\) to be significant. Thus, genes that encode membrane proteins seem to be more often differentially expressed between species than other genes. Interestingly also “response to extracellular stimulus”, “response to nutrient”, and
Figure S21: Expression values of gene RASSF10 in RPM (reads per million mapped reads) format as provided by the GeneProf data base (Halbritter et al. 2011). The data is taken from 113 public data sets. The gene T is expressed and highly variable in lymphoblastoid cell lines.

“response to nutrient levels” were significant (all $p$-values below 7.6e-5), which supports the hypothesis that some genes are differentially expressed due to the different diets of the species. All $p$-values were corrected by the Benjamini-Hochberg procedure.

S4.3.6 The “Maize Leafs” data set

Li et al. (2010) studied the developmental dynamics of the maize transcriptome using RNA-Seq data from different locations of maize plant leaves. For each location two biological replicates were sequenced with Illumina’s Genome Analyzer II. The reads were mapped to the TE-masked Zea maize ZmB73 reference genome version 2 (AGPv2), release 5a using the GSNAP splicing short read mapper (Wu and Nacu 2010). We counted the overlaps between mapped reads and the Zea maize gene definitions from the Ensemble Plants database (Release 14). Reads that have multiple possible alignments or that overlap with more than one gene are discarded. DEXUS was applied to this data with its default parameters.

Fig. S23 shows the genes with the largest I/NI value and the conditions that were identified by DEXUS. DEXUS found differentially expressed genes between different tissues, therefore distinguished them without having been provided with any information on the tissue type. DEXUS almost always assigned the two replicates to the same condition without knowing replicates or tissue types. Thus, DEXUS assigns conditions very reliable.

Eight of the ten top ranked genes were also measured by microarrays across different tissues of Zea mays (Sekhon et al. 2011). In this microarray experiment all eight genes show an absolute...
Figure S22: Heatmap of the normalized read counts of the ten genes with the largest I/NI values for the “Primate Liver” data set. Colors range from white for low expression to blue for high expression. The columns give female and male individuals from the three species human *Homo sapiens* (HS), chimpanzee *Pan troglodytes* (PT), and rhesus macaques *Macaca mulatta* (MM). The rows display the gene symbols of the top-ranked genes. Red crosses mark samples that were assigned to the minor condition. At the right hand side of the heatmap, each gene is annotated by the minimum (“>”), the median of two conditions (“m1” and “m2”), and the maximum (“<”) read count.

Log fold change of at least 1 between base and tip. Six of these eight genes show an absolute log fold change greater than 4.

The two remaining genes, GRMZM2G331518 and AC213612.3_FG001, were not annotated on the microarray. The function of the top ranked gene GRMZM2G331518 is not known. However, the associated peptide is similar to the defensin-like protein 91 of *Arabidopsis thaliana* that plays a role in immune response. The gene AC213612.3_FG001 was ranked ninth. It is a glycine-rich cell wall structural protein which hints at the fact that cell walls at different locations have different structure.

At a threshold of 0.1 for the I/NI call, DEXUS called 15,756 differentially expressed genes. Gene set enrichment analysis using the R package *goseq* (Young *et al.* 2010) led to the significant GO terms “chloroplast” (*p* = 1.8e-92), and “plasma membrane” (*p* = 1.3e-34). Further the GO terms “cytosolic ribosome” (*p* = 9.8e-32), “chloroplast thylakoid membrane” (*p* = 5.4e-31), and “chloroplast stroma” (*p* = 1.8e-30) were significant. All *p*-values were corrected by the Benjamini-Hochberg procedure. It is plausible that different locations of the maize plant leaf are different with respect to chloroplasts. Moreover the GO term “cell wall” was highly significant (*p* = 3.9e-18) which supports the above mentioned hypothesis that the cell walls differ at the different locations of the plant leaf.
Figure S23: Heatmap of the normalized read counts of the ten genes with the largest DEXUS I/NI values for the “Maize Leafs” data set. Colors range from white for low expression to blue for high expression. The columns show samples from different locations of the maize plant leaf. The rows display the gene symbols of the top-ranked genes. Red crosses indicate that the according samples belong to the minor condition. At the right hand side of the heatmap, each gene is annotated by the minimum (“>”), the median of two conditions (“m1” and “m2”), and the maximum (“<”) read count.

S4.4 RNA-Seq Data with Subconditions

We demonstrate that DEXUS is capable of detecting subconditions in data sets with known major conditions, which are typically the study conditions. Either (a) the higher level conditions are given or (b) the hierarchy of the conditions is unknown. In both cases we can explain the hierarchy by following model:

\[
p(x) = \sum_{j=1}^{l} \beta_j \sum_{i=1}^{k_j} \alpha_{ij} p(x \mid C = j, D_j = i) \quad \text{with} \quad \beta_j = p(C = j) \quad (S75)
\]

\[
p(x \mid C = j, D_j = i) = \text{NB}(x; \mu_{ij}, r_{ij} ) ,
\]

where \( \beta_j = p(C = j) \) with \( \sum_{j=1}^{l} \beta_j = 1 \) are the probabilities for the higher level condition \( j \). \( C \) is the random variable for all the higher level conditions, \( D_j \) is the random variable for the lower level conditions for higher level condition \( j \), and \( \alpha_{ij} = p(D_j = i \mid C = j) \) with \( \sum_{i=1}^{k_j} \alpha_{ij} = 1 \) is the probability to observe a lower level condition \( i \) given the higher level condition \( j \). We obtain the mixture of negative binomials for condition \( j \):

\[
\sum_{i=1}^{k_j} \alpha_{ij} p(x \mid C = j, D_j = i) = \sum_{i=1}^{k_j} p(D_j = i \mid C = j) p(x \mid C = j, D_j = i) = \sum_{i=1}^{k_j} p(x, D_j = i \mid C = j) = p(x \mid C = j) . \quad (S76)
\]

We define the probabilities for the lower level conditions

\[
\pi_{ij} = \alpha_{ij} \beta_j = p(D_j = i \mid C = j) p(C = j) = p(D_j = i, C = j) \quad \text{and} \quad \alpha_{ij} = \frac{\pi_{ij}}{\beta_j} . \quad (S77)
\]
Thus, \( \beta_j = \beta_j \sum_{i=1}^{k_j} \alpha_{ij} = \sum_{i=1}^{k_j} \alpha_{ij} \beta_j = \sum_{i=1}^{k_j} \pi_{ij} \) (S78)

and \( \sum_{j=1}^{l} \sum_{i=1}^{k_j} \pi_{ij} = 1 \). We have as full model using the \( \pi_{ij} \):

\[
p(x) = \sum_{j=1}^{l} \sum_{i=1}^{k_j} \pi_{ij} p(x \mid C = j, D_j = i),
\]

which is just a mixture of negative binomials with index set \( \{(i, j)\} \).

The posterior of a condition \( j \) after observing read count \( x \) is given via the Bayes formula by

\[
p(C = j \mid x) = \frac{p(x \mid C = j) p(C = j)}{p(x)}
\]  

(S80)

Using this model we first consider case (a), in which the higher level conditions are given. For each read count \( x_k \), its higher level condition \( \iota \) is known:

\[
p(C = j \mid x_k) = \begin{cases} 1 & \text{if } j = \iota \\ 0 & \text{otherwise} \end{cases},
\]

(S81)

and, therefore, the \( \beta_j \) can be approximated:

\[
\beta_j = p(C = j) = \sum_x p(x) p(C = j \mid x) = E_x (p(C = j \mid x))
\]

(S82)

\[
\approx \frac{1}{N} \sum_{k=1}^{N} p(C = j \mid x_k)
\]

To estimate the \( \alpha_{ij} \), we perform model selection on the mixture of negative binomials of Eq. (S76) using only the \( x_k \) that belong to condition \( j \). If we do model selection for each higher level condition \( j \) then all parameters of the hierarchical model are known. We just apply our standard mixture of negative binomials model to each of the higher level conditions \( j \).

Next we consider case (b), in which the higher level conditions are not known. DEXUS is applied to the full data set using the mixture of negative binomials model in Eq. (S79). Model selection supplies the \( \theta_s = \pi_{ij} \), such that we do not know the index \( j \) of the \( \pi_{ij} \). To identify \( j \) and \( \theta_s \), which belong to the same higher level condition \( j \), the lower level conditions \( s \) can be joined by agglomerative clustering. In such a way we obtain a hierarchy of the conditions. The variable \( \beta_j \) is obtained by summing up the \( \theta_s \) which belong to higher level condition \( j \), that are the \( \theta_s = \pi_{ij} \): \( \beta_j = \sum_{j=1}^{k_j} \pi_{ij} \). Then the \( \alpha_{ij} \) are obtained by \( \alpha_{ij} = \frac{\pi_{ij}}{\pi_j} \). This approach is just our standard mixture of negative binomials model applied to all data followed by a agglomerative clustering to obtain a hierarchy of conditions.

In our experiments, in which the higher level conditions were known, both approaches led to similar results, as we show in Figure S24. This figure shows using four genes of the “Primate Liver” data set as exemplars for a hierarchy of conditions (or groups with subgroups of samples).
Figure S24: Heatmap of the normalized read counts of four exemplar genes of the “Primate Liver” data set that contain subconditions. Colors range from white for low expression to blue for high expression. Different individuals are denoted along the $x$-axis, while genes are denoted by their gene symbols along the $y$-axis. Red crosses indicate that the according samples belong to the minor condition. Red diamonds indicate that the according samples belong to the minor subcondition. At the right hand side of the heatmap, each gene is annotated by the minimum (”$>$”), the median of three conditions (“m1”, “m2”, and “m3”), and the maximum (“$<$”) read count. The three conditions found in both cases if the high level conditions are unknown and if the high level conditions are known. In the former case, a DEXUS model with multiple unknown conditions finds the three conditions and then two of them are merged by agglomerative clustering to a high level condition. In the latter case DEXUS is applied to data from one high level condition and then finds the subconditions.
S5  Additional Information

S5.1  Data set overview

Tab. S20 gives an overview over the data sets used in this supplement. Except for the “Primate Liver” data set, all count matrices were downloaded from the ReCount (Frazee et al. 2011) repository. The count matrix for the “Primate Liver” data was taken from GEO (Accession number GSE17274). The raw counts were normalized using UpperQuartile normalization.

<table>
<thead>
<tr>
<th>Name</th>
<th>Reference</th>
<th>Organism</th>
<th>S</th>
<th>R</th>
<th>Counts</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primate Liver</td>
<td>Blekhman et al. (2010)</td>
<td>H.s./M.m./P.T.</td>
<td>18</td>
<td>2</td>
<td>Pub.</td>
<td>U</td>
</tr>
<tr>
<td>European HapMap</td>
<td>Montgomery et al. (2010)</td>
<td>H. sapiens</td>
<td>60</td>
<td>1</td>
<td>ReCount</td>
<td>U</td>
</tr>
<tr>
<td>Nigerian HapMap</td>
<td>Pickrell et al. (2010)</td>
<td>H. sapiens</td>
<td>69</td>
<td>1</td>
<td>ReCount</td>
<td>U</td>
</tr>
<tr>
<td>Worm</td>
<td>Hillier et al. (2009)</td>
<td>C. elegans</td>
<td>46</td>
<td>1</td>
<td>ReCount</td>
<td>U</td>
</tr>
<tr>
<td>Yeast</td>
<td>Nagalakshmi et al. (2008)</td>
<td>S. cerevisiae</td>
<td>4</td>
<td>1</td>
<td>ReCount</td>
<td>U</td>
</tr>
<tr>
<td>Mice Strains</td>
<td>Bottomly et al. (2011)</td>
<td>M. musculus</td>
<td>21</td>
<td>1</td>
<td>ReCount</td>
<td>K</td>
</tr>
<tr>
<td>Maize Leaves</td>
<td>Li and Tibshirani (2011)</td>
<td>Z. mays</td>
<td>12</td>
<td>2</td>
<td>Mapped</td>
<td>U</td>
</tr>
</tbody>
</table>

Table S20: Overview of the data sets used in the manuscript. “Name” gives the name used for the data set in the manuscript, “Reference” lists the according publications, “Organism” gives the organism from which the RNA-Seq data was obtained (“H.s./M.m./P.T.” means Homo sapiens/Pan troglodytes/Macaca mulatta which is human, chimpanzee, and rhesus macaques), “S” reports the number of samples, “R” gives the number of replicates for each condition, “Counts” reports the way the read counts are obtained (“Pub.” means from the publication, “ReCount” means the mapped reads are counted per transcript, “Mapped” we preprocessed the data ourselves (read mapping and counting), the column “C” lists whether the conditions were known (K) or unknown (U).

S5.2  Alternative Way to Derive the Update Rule for Mixture Weights

The update rule Eq. (S57) can be obtained in an alternative way. The Dirichlet distribution is conjugate to the multinomial distribution, that is the posterior $p(\alpha | \{\alpha_1, \ldots, \alpha_k, \ldots, \alpha_N\})$ is a Dirichlet distribution as is the prior $p(\alpha)$ with $\alpha_k = p(\alpha | x_k)$. The Dirichlet prior $p(\alpha) = D(\alpha; \gamma)$ with parameters $\gamma$ leads to the conjugate posterior $p(\alpha | \{\alpha_1, \ldots, \alpha_k, \ldots, \alpha_N\})$ with parameters

$$\hat{\gamma} = \gamma + \sum_{k=1}^{N} \alpha_k = \gamma + N \alpha,$$

(S83)

where we used Eq. (S38). We obtain update rule Eq. (S57) from Eq. (S83) component-wise by first replacing the unknown values $\alpha_{ik}$ by their estimates $\hat{\alpha}_{ik}$ and then computing the posterior’s mode because we search for the maximum posterior.
S5.3 Posteriors in Our Framework

In our Bayesian framework, we introduced two different posterior distributions: (i) in Eq. (S37) the posterior \( \alpha_{ik} = p(i \mid x_k, \alpha, \mu, r) \) of the data \( x_k \) arising from the \( i \)-th condition with prior \( \alpha_i = p(i) \) — this posterior is defined for fixed model parameters \( (\alpha, \mu, r) \); (ii) in Eq. (S20) the parameter posterior \( p(\alpha, \mu, r \mid x) \) with priors \( p(\alpha), p(\mu) \), and \( p(r) \) — this posterior is the objective that we maximize during model selection. In previous subsection we introduced another posterior, the posterior \( p(\alpha \mid \{\alpha_1, \ldots, \alpha_k, \ldots, \alpha_N\}) \) used in Eq. (S83) with prior \( p(\alpha) \). In contrast to (ii) this posterior is not the posterior for the full mixture of negative binomials model but only for the multinomial distribution given by \( \alpha \), where the posteriors \( \alpha_{ik} = p(i \mid x_k, \alpha, \mu, r) \) from (i) serve as data.

S5.4 Maximum A Posterior for the Size Parameter of a Negative Binomial

The maximum likelihood solution \( r_{ML} \) Eq. (S17) for the negative binomial tends to overestimate the true size parameter \( r \) (Piegorsch 1990). Therefore we introduce a prior \( p(r) \) on \( r \), which prefers small \( r \)-values. An appropriate prior distribution is the exponential distribution \( p(r) = \text{EXP}(r) = \eta e^{-\eta r} \).

Using a Bayesian approach, we obtain the posterior \( p(r \mid x) \) for a data point \( x \) as the normalized product between the likelihood \( p(x \mid r) \) and the prior \( p(r) \). We want to maximize the posterior

\[
\alpha = \int p(x \mid r) p(r) \, dr .
\]

The logarithm of the posterior is

\[
\log p(r \mid x) = \log p(x \mid r) + \log p(r) - \log(c(x)) ,
\]

where \( c(x) \) is a function of \( x \). Using the negative binomial distribution

\[
p(x \mid r) = \prod_{k=1}^{N} \frac{\Gamma(x_k + r)}{\Gamma(x_k + 1) \Gamma(r)} \left( \frac{\mu}{\mu + r} \right)^{x_k} \left( \frac{r}{\mu + r} \right)^r
\]

and the exponential prior \( \text{EXP}(r) = \eta e^{-\eta r} \) on \( r \), we obtain

\[
\log p(r \mid x) = \sum_{k=1}^{N} \left[ \log (\Gamma(x_k + r)) - \log (\Gamma(x_k + 1)) - \log (\Gamma(r)) + r \log \left( \frac{r}{\mu + r} \right) + x_k \log \left( \frac{\mu}{\mu + r} \right) \right] + \log(\eta) - \eta r - \log(c(x)) .
\]

In order to maximize the posterior, we set the derivative with respect to \( r \) to zero:

\[
\frac{\partial}{\partial r} \log(p(r \mid x)) = \sum_{k=1}^{N} \psi(x_k + r) - N \psi(r) + N \log \left( \frac{r}{\mu + r} \right) - \eta = 0 ,
\]

where \( \psi \) is the digamma function.
where $\psi$ is the digamma function. We call the solution of the above equation “maximum a posterior estimator” $r_{\text{MAP}}$ for the size parameter of the negative binomial distribution. Note that this is identical to the maximum-likelihood solution Eq. (S17) without prior except for the additional term $-\eta$. $\eta$ is the parameter of the exponential prior.

Note the similarity of Eq. (S88) for a single negative binomial distribution to Eq. (S49) for the whole mixture model. The difference is that, for the whole mixture model, each data point is weighted by its contribution to component $i$, that is, $\alpha_{i,k}$.

\section{S5.5 Summary of the parameters and input values of DEXUS}

\subsection{S5.5.1 Unknown Conditions}

\textbf{Input values and parameters:}

- $X$ The input matrix of read counts. Rows are assumed to be genes and columns samples. An entry is the read count of sample $k$ in gene $g$.
- $n$ Number of conditions. For further information see Subsection S3.1. \textit{Default setting:} $n = 2$.
- $\alpha^{\text{INIT}}$ The initial values for $\alpha_i$. For further information see Subsection S3.2.4. \textit{Default setting:} $\alpha_{i}^{\text{INIT}} = 1/n$.
- \textbf{normalization} We implemented “RLE” (relative log expression) that is used by DESeq (Anders and Huber 2010) and “UpperQuartile” normalization (Bullard et al. 2010). \textit{Default setting:} normalization = RLE.
- $\text{kmeansIter}$ The number of iterations of the kmeans algorithm for initializing. For further information see Subsection S3.2.4. \textit{Default setting:} kmeansIter = 10.
- $\text{cyc}$ The number of cycles of the EM algorithm. Convergence is usually reached after 5 to 10 cycles. For further information see Subsection S3.2.1. \textit{Default setting:} cyc = 20.

\textbf{Hyperparameters:}

- $G$ The weight of the prior of $\alpha$. The parameter of the Dirichlet distribution is set to $\gamma = (1 + G, 1, \ldots, 1)$. For further information see Subsection S3.2.1. \textit{Default setting:} $G = 1$.
- $\theta$ The hyperparameter that governs the setting of the regularization parameter $\eta$ on the size parameter $r$. For further information see Subsection S3.1.2 and S3.2.5. \textit{Default setting:} $\theta = 2.5$.
- $r_{\text{max}}$ The upper bound for the size parameter $r$ of the negative binomial distribution. Corresponds to a lower bound of $1/r_{\text{max}}$ for the overdispersion. \textit{Default setting:} $r_{\text{max}} = 13.0$.
- $\mu_{\text{min}}$ The minimal value for $\mu_i$ that is the mean parameter of the negative binomial distribution. For further information see Subsection S3.3. \textit{Default setting:} $\mu_{\text{min}} = 0.5$. 
S5.5.2 Known Conditions

Input values and parameters:

- **X**: The input matrix of read counts. Rows are assumed to be genes and columns samples. An entry is the read count of sample \( k \) in gene \( g \).

- **labels**: A vector containing the condition for each sample. Must be the same length as the number of rows of \( X \). For further information see Subsection S3.3.1.

- **normalization**: We implemented “RLE” (relative log expression) that is used by DESeq (Anders and Huber 2010) and “UpperQuartile” normalization (Bullard et al. 2010). Default setting: normalization = RLE.

Hyperparameters:

- \( \theta \) The hyperparameter that governs the setting of the regularization parameter \( \eta \) on the size parameter \( r \). For further information see Subsection S3.1.2 and S3.2.5. Default setting: \( \theta = 2.5 \).

- \( r_{\text{max}} \) The upper bound for the size parameter \( r \) of the negative binomial distribution. Corresponds to a lower bound of \( 1/r_{\text{max}} \) for the overdispersion. Default setting: \( r_{\text{max}} = 13.0 \).

- \( \mu_{\text{min}} \) The minimal value for \( \mu \), that is the mean parameter of the negative binomial distribution. For further information see Subsection S3.3. Default setting: \( \minMu = 0.5 \).

S5.6 Software Details of DEXUS and Experiments

- In case of two known conditions we use the function `nbinomTestForMatrices` of the R package DESeq.

- To detect differential expression for multiple known conditions, DEXUS fits a generalized linear model with the R package statmod.

- The Gaussian clustering method `mclust` is available as package for the R. We used the most recent stable version `mclust 4.0` of the implementations as provided by the original authors.

- For initialization of the EM algorithms the k-means clustering algorithm as implemented in `kmeans` of the R base package is used.

- We calculated the AUC\textsubscript{ROC} with the function of the R package ROCR (Sing et al. 2005) and the AUC\textsubscript{PR} with the algorithm suggested by Davis and Goadrich (2006).

- A function to calculate the maximum a posterior estimator \( r_{\text{MAP}} \) for the size parameter of a negative binomial is efficiently implemented in the function `getRNBbisection` of the DEXUS software package.
<table>
<thead>
<tr>
<th>C1/C2</th>
<th>precision</th>
<th>FDR</th>
<th>recall</th>
<th>F-score</th>
<th>accuracy</th>
<th>specificity</th>
<th>sensitivity</th>
<th>b. accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/6</td>
<td>0.64</td>
<td>0.36</td>
<td>0.33</td>
<td>0.43</td>
<td>0.78</td>
<td>0.94</td>
<td>0.33</td>
<td>0.63</td>
</tr>
<tr>
<td>9/3</td>
<td>0.67</td>
<td>0.33</td>
<td>0.36</td>
<td>0.47</td>
<td>0.79</td>
<td>0.94</td>
<td>0.36</td>
<td>0.65</td>
</tr>
<tr>
<td>10/2</td>
<td>0.66</td>
<td>0.34</td>
<td>0.36</td>
<td>0.47</td>
<td>0.79</td>
<td>0.94</td>
<td>0.36</td>
<td>0.65</td>
</tr>
<tr>
<td>11/1</td>
<td>0.63</td>
<td>0.37</td>
<td>0.32</td>
<td>0.42</td>
<td>0.78</td>
<td>0.94</td>
<td>0.32</td>
<td>0.63</td>
</tr>
<tr>
<td>12/12</td>
<td>0.72</td>
<td>0.28</td>
<td>0.28</td>
<td>0.40</td>
<td>0.77</td>
<td>0.96</td>
<td>0.28</td>
<td>0.62</td>
</tr>
<tr>
<td>18/6</td>
<td>0.75</td>
<td>0.25</td>
<td>0.33</td>
<td>0.46</td>
<td>0.79</td>
<td>0.96</td>
<td>0.33</td>
<td>0.65</td>
</tr>
<tr>
<td>20/4</td>
<td>0.76</td>
<td>0.24</td>
<td>0.34</td>
<td>0.47</td>
<td>0.79</td>
<td>0.96</td>
<td>0.34</td>
<td>0.65</td>
</tr>
<tr>
<td>22/2</td>
<td>0.73</td>
<td>0.27</td>
<td>0.30</td>
<td>0.42</td>
<td>0.78</td>
<td>0.96</td>
<td>0.30</td>
<td>0.63</td>
</tr>
<tr>
<td>Mean</td>
<td><strong>0.70</strong></td>
<td><strong>0.30</strong></td>
<td><strong>0.33</strong></td>
<td><strong>0.44</strong></td>
<td><strong>0.79</strong></td>
<td><strong>0.95</strong></td>
<td><strong>0.33</strong></td>
<td><strong>0.64</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C1/C2</th>
<th>precision</th>
<th>FDR</th>
<th>recall</th>
<th>F-score</th>
<th>accuracy</th>
<th>specificity</th>
<th>sensitivity</th>
<th>b. accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/6</td>
<td>0.70</td>
<td>0.30</td>
<td>0.31</td>
<td>0.43</td>
<td>0.79</td>
<td>0.96</td>
<td>0.31</td>
<td>0.63</td>
</tr>
<tr>
<td>9/3</td>
<td>0.72</td>
<td>0.28</td>
<td>0.34</td>
<td>0.46</td>
<td>0.80</td>
<td>0.96</td>
<td>0.34</td>
<td>0.65</td>
</tr>
<tr>
<td>10/2</td>
<td>0.71</td>
<td>0.29</td>
<td>0.33</td>
<td>0.45</td>
<td>0.80</td>
<td>0.96</td>
<td>0.33</td>
<td>0.64</td>
</tr>
<tr>
<td>11/1</td>
<td>0.62</td>
<td>0.38</td>
<td>0.22</td>
<td>0.32</td>
<td>0.77</td>
<td>0.96</td>
<td>0.22</td>
<td>0.59</td>
</tr>
<tr>
<td>12/12</td>
<td>0.82</td>
<td>0.18</td>
<td>0.25</td>
<td>0.39</td>
<td>0.78</td>
<td>0.98</td>
<td>0.25</td>
<td>0.62</td>
</tr>
<tr>
<td>18/6</td>
<td>0.84</td>
<td>0.16</td>
<td>0.30</td>
<td>0.44</td>
<td>0.79</td>
<td>0.98</td>
<td>0.30</td>
<td>0.64</td>
</tr>
<tr>
<td>20/4</td>
<td>0.83</td>
<td>0.17</td>
<td>0.29</td>
<td>0.42</td>
<td>0.79</td>
<td>0.98</td>
<td>0.29</td>
<td>0.63</td>
</tr>
<tr>
<td>22/2</td>
<td>0.75</td>
<td>0.25</td>
<td>0.17</td>
<td>0.28</td>
<td>0.76</td>
<td>0.98</td>
<td>0.17</td>
<td>0.57</td>
</tr>
<tr>
<td>Mean</td>
<td><strong>0.75</strong></td>
<td><strong>0.25</strong></td>
<td><strong>0.28</strong></td>
<td><strong>0.40</strong></td>
<td><strong>0.78</strong></td>
<td><strong>0.97</strong></td>
<td><strong>0.28</strong></td>
<td><strong>0.62</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C1/C2</th>
<th>precision</th>
<th>FDR</th>
<th>recall</th>
<th>F-score</th>
<th>accuracy</th>
<th>specificity</th>
<th>sensitivity</th>
<th>b. accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/6</td>
<td>0.80</td>
<td>0.20</td>
<td>0.28</td>
<td>0.41</td>
<td>0.80</td>
<td>0.98</td>
<td>0.28</td>
<td>0.63</td>
</tr>
<tr>
<td>9/3</td>
<td>0.80</td>
<td>0.20</td>
<td>0.28</td>
<td>0.41</td>
<td>0.80</td>
<td>0.98</td>
<td>0.28</td>
<td>0.63</td>
</tr>
<tr>
<td>10/2</td>
<td>0.73</td>
<td>0.27</td>
<td>0.19</td>
<td>0.31</td>
<td>0.78</td>
<td>0.98</td>
<td>0.19</td>
<td>0.58</td>
</tr>
<tr>
<td>11/1</td>
<td>0.39</td>
<td>0.61</td>
<td>0.05</td>
<td>0.08</td>
<td>0.74</td>
<td>0.98</td>
<td>0.05</td>
<td>0.51</td>
</tr>
<tr>
<td>12/12</td>
<td>0.92</td>
<td>0.08</td>
<td>0.22</td>
<td>0.36</td>
<td>0.78</td>
<td>0.99</td>
<td>0.22</td>
<td>0.61</td>
</tr>
<tr>
<td>18/6</td>
<td>0.92</td>
<td>0.08</td>
<td>0.23</td>
<td>0.37</td>
<td>0.79</td>
<td>0.99</td>
<td>0.23</td>
<td>0.61</td>
</tr>
<tr>
<td>20/4</td>
<td>0.89</td>
<td>0.11</td>
<td>0.15</td>
<td>0.26</td>
<td>0.76</td>
<td>0.99</td>
<td>0.15</td>
<td>0.57</td>
</tr>
<tr>
<td>22/2</td>
<td>0.43</td>
<td>0.57</td>
<td>0.01</td>
<td>0.03</td>
<td>0.73</td>
<td>0.99</td>
<td>0.01</td>
<td>0.50</td>
</tr>
<tr>
<td>Mean</td>
<td><strong>0.73</strong></td>
<td><strong>0.27</strong></td>
<td><strong>0.18</strong></td>
<td><strong>0.28</strong></td>
<td><strong>0.77</strong></td>
<td><strong>0.98</strong></td>
<td><strong>0.18</strong></td>
<td><strong>0.58</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C1/C2</th>
<th>precision</th>
<th>FDR</th>
<th>recall</th>
<th>F-score</th>
<th>accuracy</th>
<th>specificity</th>
<th>sensitivity</th>
<th>b. accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/6</td>
<td>0.87</td>
<td>0.13</td>
<td>0.25</td>
<td>0.39</td>
<td>0.80</td>
<td>0.99</td>
<td>0.25</td>
<td>0.62</td>
</tr>
<tr>
<td>9/3</td>
<td>0.82</td>
<td>0.18</td>
<td>0.18</td>
<td>0.29</td>
<td>0.78</td>
<td>0.99</td>
<td>0.18</td>
<td>0.58</td>
</tr>
<tr>
<td>10/2</td>
<td>0.55</td>
<td>0.45</td>
<td>0.05</td>
<td>0.08</td>
<td>0.75</td>
<td>0.99</td>
<td>0.05</td>
<td>0.52</td>
</tr>
<tr>
<td>11/1</td>
<td>0.36</td>
<td>0.64</td>
<td>0.02</td>
<td>0.04</td>
<td>0.75</td>
<td>0.99</td>
<td>0.02</td>
<td>0.50</td>
</tr>
<tr>
<td>12/12</td>
<td>0.96</td>
<td>0.04</td>
<td>0.20</td>
<td>0.34</td>
<td>0.78</td>
<td>1.00</td>
<td>0.20</td>
<td>0.55</td>
</tr>
<tr>
<td>18/6</td>
<td>0.94</td>
<td>0.06</td>
<td>0.14</td>
<td>0.24</td>
<td>0.76</td>
<td>1.00</td>
<td>0.14</td>
<td>0.57</td>
</tr>
<tr>
<td>20/4</td>
<td>0.62</td>
<td>0.38</td>
<td>0.02</td>
<td>0.03</td>
<td>0.73</td>
<td>1.00</td>
<td>0.02</td>
<td>0.51</td>
</tr>
<tr>
<td>22/2</td>
<td>0.37</td>
<td>0.63</td>
<td>0.01</td>
<td>0.01</td>
<td>0.73</td>
<td>1.00</td>
<td>0.01</td>
<td>0.50</td>
</tr>
<tr>
<td>Mean</td>
<td><strong>0.69</strong></td>
<td><strong>0.31</strong></td>
<td><strong>0.11</strong></td>
<td><strong>0.18</strong></td>
<td><strong>0.76</strong></td>
<td><strong>0.99</strong></td>
<td><strong>0.11</strong></td>
<td><strong>0.55</strong></td>
</tr>
</tbody>
</table>

Table S21: Results of DEXUS for unknown conditions (two conditions). “C1/C2” reports the number of samples for each condition. Each line represents one experiment that consists of 100 data sets. The column names give the different performance measures. The I/NI thresholds are given in table headings. The library size was $10^6$ for all experiments.
<table>
<thead>
<tr>
<th>C1/C2</th>
<th>precision</th>
<th>FDR</th>
<th>recall</th>
<th>F-score</th>
<th>accuracy</th>
<th>specificity</th>
<th>sensitivity</th>
<th>b. accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/6</td>
<td>0.69</td>
<td>0.31</td>
<td>0.55</td>
<td>0.61</td>
<td>0.79</td>
<td>0.90</td>
<td>0.55</td>
<td>0.72</td>
</tr>
<tr>
<td>9/3</td>
<td>0.71</td>
<td>0.29</td>
<td>0.60</td>
<td>0.65</td>
<td>0.81</td>
<td>0.90</td>
<td>0.60</td>
<td>0.75</td>
</tr>
<tr>
<td>10/2</td>
<td>0.71</td>
<td>0.29</td>
<td>0.60</td>
<td>0.65</td>
<td>0.81</td>
<td>0.90</td>
<td>0.60</td>
<td>0.75</td>
</tr>
<tr>
<td>11/1</td>
<td>0.68</td>
<td>0.32</td>
<td>0.52</td>
<td>0.59</td>
<td>0.78</td>
<td>0.90</td>
<td>0.52</td>
<td>0.71</td>
</tr>
<tr>
<td>12/12</td>
<td>0.79</td>
<td>0.21</td>
<td>0.52</td>
<td>0.62</td>
<td>0.81</td>
<td>0.94</td>
<td>0.52</td>
<td>0.73</td>
</tr>
<tr>
<td>18/6</td>
<td>0.81</td>
<td>0.19</td>
<td>0.59</td>
<td>0.68</td>
<td>0.84</td>
<td>0.94</td>
<td>0.59</td>
<td>0.77</td>
</tr>
<tr>
<td>20/4</td>
<td>0.81</td>
<td>0.19</td>
<td>0.59</td>
<td>0.68</td>
<td>0.84</td>
<td>0.94</td>
<td>0.59</td>
<td>0.77</td>
</tr>
<tr>
<td>22/2</td>
<td>0.78</td>
<td>0.22</td>
<td>0.50</td>
<td>0.61</td>
<td>0.81</td>
<td>0.94</td>
<td>0.50</td>
<td>0.72</td>
</tr>
<tr>
<td>Mean</td>
<td>0.75</td>
<td>0.25</td>
<td>0.56</td>
<td>0.64</td>
<td>0.81</td>
<td>0.92</td>
<td>0.56</td>
<td>0.74</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C1/C2</th>
<th>precision</th>
<th>FDR</th>
<th>recall</th>
<th>F-score</th>
<th>accuracy</th>
<th>specificity</th>
<th>sensitivity</th>
<th>b. accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/6</td>
<td>0.79</td>
<td>0.21</td>
<td>0.52</td>
<td>0.62</td>
<td>0.81</td>
<td>0.94</td>
<td>0.52</td>
<td>0.73</td>
</tr>
<tr>
<td>9/3</td>
<td>0.80</td>
<td>0.20</td>
<td>0.56</td>
<td>0.66</td>
<td>0.83</td>
<td>0.94</td>
<td>0.56</td>
<td>0.75</td>
</tr>
<tr>
<td>10/2</td>
<td>0.78</td>
<td>0.22</td>
<td>0.51</td>
<td>0.62</td>
<td>0.81</td>
<td>0.94</td>
<td>0.51</td>
<td>0.72</td>
</tr>
<tr>
<td>11/1</td>
<td>0.68</td>
<td>0.32</td>
<td>0.30</td>
<td>0.42</td>
<td>0.75</td>
<td>0.94</td>
<td>0.30</td>
<td>0.62</td>
</tr>
<tr>
<td>12/12</td>
<td>0.89</td>
<td>0.11</td>
<td>0.47</td>
<td>0.62</td>
<td>0.82</td>
<td>0.98</td>
<td>0.47</td>
<td>0.72</td>
</tr>
<tr>
<td>18/6</td>
<td>0.90</td>
<td>0.19</td>
<td>0.53</td>
<td>0.67</td>
<td>0.84</td>
<td>0.98</td>
<td>0.53</td>
<td>0.75</td>
</tr>
<tr>
<td>20/4</td>
<td>0.89</td>
<td>0.11</td>
<td>0.48</td>
<td>0.62</td>
<td>0.83</td>
<td>0.98</td>
<td>0.48</td>
<td>0.73</td>
</tr>
<tr>
<td>22/2</td>
<td>0.82</td>
<td>0.18</td>
<td>0.26</td>
<td>0.40</td>
<td>0.76</td>
<td>0.98</td>
<td>0.26</td>
<td>0.62</td>
</tr>
<tr>
<td>Mean</td>
<td>0.82</td>
<td>0.18</td>
<td>0.45</td>
<td>0.58</td>
<td>0.81</td>
<td>0.96</td>
<td>0.45</td>
<td>0.71</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C1/C2</th>
<th>precision</th>
<th>FDR</th>
<th>recall</th>
<th>F-score</th>
<th>accuracy</th>
<th>specificity</th>
<th>sensitivity</th>
<th>b. accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/6</td>
<td>0.89</td>
<td>0.11</td>
<td>0.46</td>
<td>0.61</td>
<td>0.82</td>
<td>0.98</td>
<td>0.46</td>
<td>0.72</td>
</tr>
<tr>
<td>9/3</td>
<td>0.89</td>
<td>0.11</td>
<td>0.44</td>
<td>0.59</td>
<td>0.82</td>
<td>0.98</td>
<td>0.44</td>
<td>0.71</td>
</tr>
<tr>
<td>10/2</td>
<td>0.83</td>
<td>0.17</td>
<td>0.27</td>
<td>0.41</td>
<td>0.77</td>
<td>0.98</td>
<td>0.27</td>
<td>0.62</td>
</tr>
<tr>
<td>11/1</td>
<td>0.39</td>
<td>0.61</td>
<td>0.04</td>
<td>0.07</td>
<td>0.70</td>
<td>0.98</td>
<td>0.04</td>
<td>0.51</td>
</tr>
<tr>
<td>12/12</td>
<td>0.96</td>
<td>0.04</td>
<td>0.42</td>
<td>0.59</td>
<td>0.82</td>
<td>0.99</td>
<td>0.42</td>
<td>0.71</td>
</tr>
<tr>
<td>18/6</td>
<td>0.96</td>
<td>0.04</td>
<td>0.42</td>
<td>0.58</td>
<td>0.82</td>
<td>0.99</td>
<td>0.42</td>
<td>0.71</td>
</tr>
<tr>
<td>20/4</td>
<td>0.94</td>
<td>0.06</td>
<td>0.24</td>
<td>0.39</td>
<td>0.77</td>
<td>0.99</td>
<td>0.24</td>
<td>0.62</td>
</tr>
<tr>
<td>22/2</td>
<td>0.42</td>
<td>0.58</td>
<td>0.01</td>
<td>0.02</td>
<td>0.70</td>
<td>0.99</td>
<td>0.01</td>
<td>0.50</td>
</tr>
<tr>
<td>Mean</td>
<td>0.78</td>
<td>0.22</td>
<td>0.29</td>
<td>0.41</td>
<td>0.78</td>
<td>0.98</td>
<td>0.29</td>
<td>0.64</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C1/C2</th>
<th>precision</th>
<th>FDR</th>
<th>recall</th>
<th>F-score</th>
<th>accuracy</th>
<th>specificity</th>
<th>sensitivity</th>
<th>b. accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/6</td>
<td>0.94</td>
<td>0.06</td>
<td>0.42</td>
<td>0.58</td>
<td>0.82</td>
<td>0.99</td>
<td>0.42</td>
<td>0.71</td>
</tr>
<tr>
<td>9/3</td>
<td>0.90</td>
<td>0.10</td>
<td>0.25</td>
<td>0.39</td>
<td>0.77</td>
<td>0.99</td>
<td>0.25</td>
<td>0.62</td>
</tr>
<tr>
<td>10/2</td>
<td>0.57</td>
<td>0.43</td>
<td>0.03</td>
<td>0.07</td>
<td>0.70</td>
<td>0.99</td>
<td>0.03</td>
<td>0.51</td>
</tr>
<tr>
<td>11/1</td>
<td>0.37</td>
<td>0.63</td>
<td>0.02</td>
<td>0.03</td>
<td>0.70</td>
<td>0.99</td>
<td>0.02</td>
<td>0.50</td>
</tr>
<tr>
<td>12/12</td>
<td>0.98</td>
<td>0.02</td>
<td>0.39</td>
<td>0.56</td>
<td>0.82</td>
<td>1.00</td>
<td>0.39</td>
<td>0.70</td>
</tr>
<tr>
<td>18/6</td>
<td>0.97</td>
<td>0.03</td>
<td>0.23</td>
<td>0.38</td>
<td>0.77</td>
<td>1.00</td>
<td>0.23</td>
<td>0.62</td>
</tr>
<tr>
<td>20/4</td>
<td>0.65</td>
<td>0.35</td>
<td>0.01</td>
<td>0.02</td>
<td>0.70</td>
<td>1.00</td>
<td>0.01</td>
<td>0.50</td>
</tr>
<tr>
<td>22/2</td>
<td>0.35</td>
<td>0.65</td>
<td>0.00</td>
<td>0.01</td>
<td>0.70</td>
<td>1.00</td>
<td>0.00</td>
<td>0.50</td>
</tr>
<tr>
<td>Mean</td>
<td>0.72</td>
<td>0.28</td>
<td>0.17</td>
<td>0.26</td>
<td>0.75</td>
<td>0.99</td>
<td>0.17</td>
<td>0.58</td>
</tr>
</tbody>
</table>

Table S22: Results of DEXUS for unknown conditions (two conditions). “C1/C2” reports the number of samples for each condition. Each line represents one experiment that consists of 100 data sets. The column names give the different performance measures. The I/NI thresholds are given in table headings. The library size was $10^7$ for all experiments.
<table>
<thead>
<tr>
<th>C1/C2</th>
<th>precision</th>
<th>FDR</th>
<th>recall</th>
<th>F-score</th>
<th>accuracy</th>
<th>specificity</th>
<th>sensitivity</th>
<th>b. accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/6</td>
<td>0.76</td>
<td>0.24</td>
<td>0.78</td>
<td>0.77</td>
<td>0.86</td>
<td>0.89</td>
<td>0.78</td>
<td>0.83</td>
</tr>
<tr>
<td>9/3</td>
<td>0.77</td>
<td>0.23</td>
<td>0.83</td>
<td>0.80</td>
<td>0.87</td>
<td>0.89</td>
<td>0.83</td>
<td>0.86</td>
</tr>
<tr>
<td>10/2</td>
<td>0.77</td>
<td>0.23</td>
<td>0.82</td>
<td>0.79</td>
<td>0.87</td>
<td>0.89</td>
<td>0.82</td>
<td>0.86</td>
</tr>
<tr>
<td>11/1</td>
<td>0.73</td>
<td>0.27</td>
<td>0.68</td>
<td>0.70</td>
<td>0.83</td>
<td>0.89</td>
<td>0.68</td>
<td>0.79</td>
</tr>
<tr>
<td>12/12</td>
<td>0.85</td>
<td>0.15</td>
<td>0.74</td>
<td>0.79</td>
<td>0.88</td>
<td>0.94</td>
<td>0.74</td>
<td>0.84</td>
</tr>
<tr>
<td>18/6</td>
<td>0.86</td>
<td>0.14</td>
<td>0.82</td>
<td>0.84</td>
<td>0.91</td>
<td>0.95</td>
<td>0.82</td>
<td>0.88</td>
</tr>
<tr>
<td>20/4</td>
<td>0.86</td>
<td>0.14</td>
<td>0.81</td>
<td>0.84</td>
<td>0.90</td>
<td>0.95</td>
<td>0.81</td>
<td>0.88</td>
</tr>
<tr>
<td>22/2</td>
<td>0.84</td>
<td>0.16</td>
<td>0.65</td>
<td>0.73</td>
<td>0.86</td>
<td>0.95</td>
<td>0.65</td>
<td>0.80</td>
</tr>
<tr>
<td>Mean</td>
<td><strong>0.80</strong></td>
<td><strong>0.20</strong></td>
<td><strong>0.76</strong></td>
<td><strong>0.78</strong></td>
<td><strong>0.87</strong></td>
<td><strong>0.92</strong></td>
<td><strong>0.76</strong></td>
<td><strong>0.84</strong></td>
</tr>
</tbody>
</table>

**Table S23:** Results of DEXUS for unknown conditions (two conditions). "C1/C2" reports the number of samples for each condition. Each line represents one experiment that consists of 100 data sets. The column names give the different performance measures. The I/NI thresholds are given in table headings. The library size was $10^8$ for all experiments.
References


References


Chapter 4

Discussion

Although cn.MOPS was suggested for whole genome sequencing data of humans, it has a much wider application area. With a small change to the prior assumptions it can be transferred to haploid genomes and exome sequencing data. cn.MOPS was successfully applied to bacterial strains, see Section 4.1, and to exome sequencing data, see Section 4.2. Other possible research and application areas of cn.MOPS are presented in Section 4.3.

DEXUS made the analysis of data from many different study designs possible, which has already been explained before, see Chapter 3. In this chapter I will explain a typical setting of an RNA-Seq drug design study in a pharma company. In this setting the results of DEXUS are highly important and will improve the efficiency of drug development, see Section 4.5.

4.1 The german enterohemorrhagic Escherichia coli O104:H4 outbreak

In May 2011 a series of infections with a novel strain of Escherichia coli, later named O104:H4, was reported in northern Germany (Mellmann et al. 2011; Frank et al. 2011). About 4,000 people were infected, and more than 50 died of this disease. Only hours after the first bacteria were isolated, their genome was analyzed using a high throughput sequencing technique, called IonTorrent sequencing (Loman et al. 2012). The experiment was carried out in the Beijing Genomics Institute and the data were made publicly available (Rohde et al. 2011). Before the genome of O104:H4 was fully assembled, the cn.MOPS algorithm discovered that this new strain had two copies of the gene $stx_2$, see Figure 4.1. The gene $stx_2$ codes for the Shiga toxin that inhibits protein synthesis, which was the reason for the severe illness caused
Figure 4.1: Genome plot showing copy numbers variable regions of O104:H4 mapped to the reference genome of the related strain O157:H7. Red arrows: shiga toxin stx2 is amplified in O104:H4. Days later, whole genome analysis confirmed the two copies of stx2, and it is assumed that these were acquired by horizontal gene transfer.

4.2 Copy number detection in exome sequencing data

Recently exome sequencing came in the focus of researchers’ interest, see Figure 4.2, and is currently developing into a key technology for revealing causes of human diseases (Ng et al. 2010; Varela et al. 2011; Sanders et al. 2012; Yu et al. 2013). For exome sequencing an enrichment strategy is used
4.2. COPY NUMBER DETECTION IN EXOME SEQUENCING DATA

Figure 4.2: Barplot displaying the number of publications per year that present research based on exome sequencing data. The x-axis denotes the years from 2009 to 2013, and the y-axis the number of publications. The numbers were calculated by Google Scholar counting only publications that have the exact phrase “exome sequencing” in the publication title. The use of this technology in biological and medical research has been increasing since 2009.

In order to sequence only the transcribed part of the human genome, that is about 1% of the whole genome. The data is then usually analyzed for variants that may disrupt the functionality of a gene. These variants can either be single nucleotide variants or InDels, but also copy number variations.

In general, copy number variation detection methods suggested for whole genome sequencing data cannot be used for exome sequencing data, because the distribution assumptions of the algorithms are not fulfilled. The distribution of read counts or depth of coverage along the DNA segments substantially differs between genome and exome sequencing, see Figure 4.3. This difference arises from the enrichment strategy, which introduces biases for each of the enriched DNA segments. cn.MOPS was designed to estimate the biases of each DNA segment by local modeling across samples, and is therefore perfectly suited for exome sequencing data. It was shown by Guo et al. (2013) that cn.MOPS is the best-performing copy number detection method for exome sequencing data. Guo et al. (2013) show that, although having the lowest false discovery rate (FDR), the number of CNV detections of cn.MOPS is four times higher than the number of detections of the second best method. Increased discoveries and at the same time lowest FDR means that it outperforms all other methods with respect to precision and recall.
4.3 Research suggested by cn.MOPS

As I have shown in my work, copy number variations can reliably be detected from high throughput sequencing data. This offers a number of new opportunities for research in the area of genetics and related fields.

4.3.1 Installation of CNV data bases

Since the cn.MOPS algorithm is computationally efficient and can be parallelized,\(^1\) it is possible to analyze even very large data sets, such as the data of the 1000 Genomes Project (The 1000 Genomes Project Consortium 2010). A catalog of CNVs of a large number of healthy individuals is highly desirable for population genetics analysis and for increasing the discovery power of genome wide association studies (GWAS), because CNVs appearing in healthy individuals can be removed from the set of hypotheses (Barnes et al. 2008; Clevert et al. 2013).

4.3.2 Improving the discovery power of genome-wide association studies

Another improvement for genome-wide association studies (GWAS) is that both single nucleotide variants and copy number variations can be detected with a single technology. Some diseases, e.g. autism spectrum disorder, are supposed to be caused by two malfunctioning genes ("two hit model")

---

\(^1\) A parallel version of cn.MOPS is already implemented in the Bioconductor package `cn.mops`
4.3. RESEARCH SUGGESTED BY CN.MOPS

(Leblond et al. 2012; Girirajan et al. 2010). This malfunctioning of genes can be caused by both a single nucleotide variant or a copy number variant. Therefore studies that consider only SNVs or CNVs cannot detect all associations between genetics and the disease. Standard NGS analysis pipelines can be extended with cn.MOPS to provide both SNV and CNV calls. This enables researchers to identify causes of disease in situations as described above.

4.3.3 Reliable genetic testing for CNVs

Genetic testing is a process in the clinic that should identify disease-related mutations in the DNA. CNVs are also considered by genetic testing as potential causes for diseases (Rodriguez-Revenga et al. 2007) and various technologies, such as microscopy, array comparative genomic hybridization, and SNP microarrays, are used for this purpose. Typical for genetic testing is that single samples have to be analyzed on-demand, as opposed to a study in which samples are analyzed in batches. A genetic test must be highly accurate, since both a false positive or a false negative result can lead to wrong medication or counseling of a person. At this point of time and in the future, high-throughput DNA sequencing is used and will be used for genetic testing for clinical purposes (Desai and Jere 2012). Therefore, a reliable computational method as part of a genetic test for CNVs based on high-throughput sequencing data is needed.

cn.MOPS is perfectly suited for this purpose, although it needs multiple samples for CNV detection. CNV detection using only data from one individual is possible, but the performance of CNV detection methods improves with multiple samples, see Section 2. Although in the clinic a genetic test is done for one person at a time, there are simple ways to include more samples in a genetic test. Databases of samples that have already been sequenced with a certain protocol and technology can be built up. A new sample can then be analyzed together with the samples from the database to provide reliable CNV calls. This is a very straight-forward extension to the current cn.MOPS algorithm.

In summary, cn.MOPS suggests CNV analyses even at population scale, which is highly relevant for various applications including population genetics analyses, and genome wide association studies. Additionally cn.MOPS can be used for genetic testing for clinical purposes.
4.4 The unknown conditions of DEXUS

DEXUS is designed to analyze data for which the groups or sample conditions are “unknown”. This terminology, “unknown conditions”, needs further explanation, because it can lead to misunderstandings. There are different layers of conditions that can be present or absent in a single cell at the same time. One of these layers is the cell’s genetic condition that is determined by the genome and affects the transcriptome in many ways. The expression of $RPS26$, e.g., is determined by several distinct SNVs (e.g. rs1131017), so-called expression quantitative trait loci (eQTLs) (Plagnol et al. 2009; Montgomery et al. 2010). Secondly, there is a layer of metabolic conditions of a cell. It is also determined by many factors, such as the metabolites that are present inside and outside the cell. For instance, amino acid deprivation of human cells triggers the expression of the asparagine synthetase which catalyzes asparagine and glutamate synthesis (Chen et al. 2004). Another layer of conditions, that partly includes the metabolic one, is the layer of environmental conditions of sample. Samples are often individuals or organisms that live in a certain environment that can also interfere with gene expression. As a prominent example I can mention that human skin cells react to exposure to sunlight with a very distinct pattern of gene expression, i.e., by rapidly up-regulating the expression of $IL-8$ (Strickland et al. 1997). Furthermore, there are experimental conditions that are present, when the actual experiment is carried out. This includes the temperature at which the cells were colonized, the lab technician carrying out the experiment, the biotechnology instruments utilized. All these parameters can affect gene expression and can thereby confound gene expression data analysis (Johnson et al. 2007).

The layers of conditions mentioned above are all present at the same time in a cell and affect the transcriptome. When we take a snapshot of the transcriptome by RNA sequencing, we have the chance to get insight in the most prominent transcriptional effects caused by entirety of these conditions.

From the statistical point of view an “unknown condition” is a latent variable, a variable that can not directly be measured, but has to be estimated via the observed variables. In case of the DEXUS algorithm the observed variable is the read count and the hidden variable is the condition.

4.5 Research suggested by DEXUS

At a seminar on “Computational Methods Aiding Early-Stage Drug Design” (Bender et al. 2013) I presented a typical drug-design study as a perfect example for the case of “unknown conditions” for which DEXUS was designed.
4.5. RESEARCH SUGGESTED BY DEXUS

4.5.1 Identification of compound-induced transcriptional effects

When a drug-like compound is administered to a cell line, it is usually unknown which transcriptional changes the compound will induce. However, transcriptional changes are to be expected since drugs usually interfere with a target protein, which can ultimately lead – via changes on the protein level – to alterations on the transcriptional level (Lamb et al. 2006). Even side effects of a compound can manifest themselves as changes in transcription (Baum et al. 2010).

The transcriptional effects of drug-like compounds can be measured either by gene expression microarrays or by RNA-Seq. Although microarrays are better suited for the quantification of transcripts (Labaj et al. 2011; Klambauer et al. 2012) RNA-Seq is becoming the standard technique, since it also provides information about the transcript sequence, alternative splicing, post-transcriptional modifications, and RNA editing (Marguerat and Bähler 2010). Therefore, RNA-Seq data analysis is highly desirable, especially in the drug design context. As stated before in Section 3, previous RNA-Seq methods are restricted to a priori defined conditions. However, the conditions for a drug-design study involving newly designed compounds, are a priori unknown. In this context the conditions of a compound can be interpreted as “not changing the transcription of a certain gene”, “increasing the transcription of a certain gene”, and “decreasing the transcription of a certain gene”. Exactly for this situation DEXUS has been designed. The analysis of an RNA-Seq drug-design study by DEXUS will lead to the detection of a compound’s on- and off-target effects, both desired and undesired.

If an adverse side effect of a compound is detected in an early stage of drug development, such as the hit-to-lead or the lead optimization phase, both time and capital can be saved. To detect new drug targets can help to develop a new or alternative way of treatment of diseases. This means that identification of on- and off-target effects makes the drug development process more efficient, i.e., more and better drugs can be brought to the market.

In summary, DEXUS suggests RNA-Seq drug-design studies investigating transcriptional effects of new drug-like compounds, which is highly important to the field of drug development.
Bibliography


Bibliography


Bibliography


