T.R.
HACETTEPE UNIVERSITY
INSTITUTE OF HEALTH SCIENCES

DEVELOPMENT AND APPLICATION OF NOVEL MACHINE LEARNING APPROACHES FOR RNA-SEQ DATA CLASSIFICATION

Gökmen ZARARSIZ

BIOSTATISTICS PROGRAM
PHILOSOPHY OF DOCTORATE THESIS

Ankara
2015

T.R.
HACETTEPE UNIVERSITY
INSTITUTE OF HEALTH SCIENCES

DEVELOPMENT AND APPLICATION OF NOVEL MACHINE LEARNING APPROACHES FOR RNA-SEQ DATA CLASSIFICATION

Gökmen ZARARSIZ

BIOSTATISTICS PROGRAM
PHILOSOPHY OF DOCTORATE THESIS

ADVISOR
Ass. Prof. Dr. Erdem KARABULUT
CO-ADVISOR
Ass. Prof. Dr. Ahmet ÖZTÜRK

Ankara
2015

Anabilim Dalı :**BİYOİSTATİSTİK**

Program :**BİYOİSTATİSTİK**

Tez Başlığı :**DEVELOPMENT AND APPLICATION OF NOVEL MACHINE LEARNING APPROACHES FOR RNA-SEQ DATA CLASSIFICATION**

Öğrenci Adı-Soyadı :**GÖKMEN ZARARSIZ**

Savunma Sınavı Tarihi :**23.06.2015**

Bu çalışma jürimiz tarafından yüksek lisans/doktora tezi olarak kabul edilmiştir.

Jüri Başkanı: **PROF.DR.ERGUN KARAAĞAOĞLU**

(Hacettepe Üniversitesi)

Tez danışmanı: **DOÇ.DR.ERDEM KARABULUT**

(Hacettepe Üniversitesi)

Üye: **PROF.DR.OSMAN SARAÇBAŞI**

(Hacettepe Üniversitesi)

Üye: **DOÇ.DR.DERYA ÖZTUNA**

(Ankara Üniversitesi)

Üye: **DOÇ.DR.TURGAY ÜNVER**

(Çankırı Karatekin Üniversitesi)

ONAY

Bu tez Hacettepe Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca yukarıdaki jüri tarafından uygun görülmüş ve Sağlık Bilimleri Enstitüsü Yönetim Kurulu kararıyla kabul edilmiştir.

Prof.Dr. Ersin FADILLIOĞLU

Müdür

TEŞEKKÜR

Akademik kariyerimin başından itibaren beni yönlendiren, destek olan ve bu tezin gerçekleşmesinde katkıları bulunan danışmanlarım Doç.Dr. Erdem Karabulut ve Doç.Dr. Ahmet Öztürk'e,

Bu tezin planlamasından, tasarımına ve gerçekleştirilmesine kadar tüm süreç boyunca büyük emekleri olan, maddi ve manevi destekleri her zaman yanımda olan çalışma arkadaşlarım ve dostlarım Ar.Gör. Dinçer Göksülük, Ar.Gör. Selçuk Korkmaz ve Ar.Gör. Vahap Eldem'e,

Desteklerini esirgemeyen değerli hocalarım Prof.Dr. Reha Alpar, Prof.Dr. Ergun Karaağaoğlu, Prof.Dr. Osman Saraçbaşı'na, bölüm sekreterimiz Menekşe Tarla'ya, Yrd.Doç.Dr. Anıl Barak Dolgun'a ve diğer Hacettepe Biyoistatistik Ana Bilim Dalı personeline,

Doktora eğitimim sürecinde Pekin Genom Merkezi'ne gitmeme ve bu alanda bilgi birikimimin artmasına vesile olan, tez boyunca da desteklerini esirgemeyen manevi danışmanım Doç.Dr. Turgay Ünver'e ve değerli hocam Prof.Dr. Mustafa Çetin'e,

Bu tezin ortaya çıkışında fikirleriyle destekte bulunan, yöntem ve yazılım kısımlarında da sürekli yardımcı olan Bernd Klaus ve Wolfgang Huber'e,

Tezde kullandığım Alzheimer verisini tereddüt etmeden benimle paylaşan Andreas Keller'e,

Tezin analiz boyutunda sunucu ve iş istasyonlarına erişimimi sağlayan Hacettepe Üniversitesi, Biyoistatistik Anabilim Dalı'na; Erciyes Üniversitesi, Biyoistatistik Anabilim Dalı'na; Erciyes Üniversitesi, Genom ve Kök Hücre Merkezi'ne; Marmara Üniversitesi, Fizik Bölümü'ne ve University of California, San Diego Supercomputer Center birimine,

Tezin İngilizce dil düzeltmelerinde bana yardımcı olan sevgili kuzenim Seren Öncel'e,

Doğum anımdan itibaren her anımda benimle olan, maddi ve manevi destekleriyle beni bu günlere getiren rahmetli annem Reyhan Zararsız'a ve babam Kemal Zararsız'a,

Biricik kardeşim, arkadaşım, yaşam kaynağım Yasin Güven Zararsız'a,

Teşekkürü bir borç bilirim.

ABSTRACT

Zararsız, G. Development and Application of Novel Machine Learning Approaches for RNA-Seq Data Classification. Hacettepe University Institute of Health Sciences, Ph.D. Thesis in Biostatistics, Ankara, 2015. RNA-Seq is a recent and efficient technique that uses the capabilities of next-generation sequencing technology in characterizing and quantifying transcriptomes. This technique has revolutionized the gene-expression profiling with major advantages over microarrays: (i) providing less noisy data, (ii) detecting novel transcripts and isoforms, and (iii) unnecessity of prearranged transcripts of interest. One important task using gene-expression data is to identify a small subset of genes and classify the data for diagnostic purposes, particularly for cancer diseases. Microarray based classifiers are not directly applicable due to the discrete nature of RNA-Seq data. Overdispersion is another problem that requires careful modeling of mean and variance relationship of the RNA-Seq data. Voom is a recent method that estimates the mean and variance relationship of the log-counts and provides precision weights for each observation to be used for further analysis. In this study, we developed VoomNSC method, which brings together voom and a powerful microarray classifier nearest shrunken centroids approaches for the purpose of "gene-expression based classification". VoomNSC is a sparse classifier that models the mean and variance relationship using voom method, incorporates the outputs of voom method (i.e. log-cpm values and precision weights) into NSC using weighted statistics. We also provided two non-sparse classifiers voomDLDA and voomDQDA, the extensions of diagonal linear and quadratic discriminant classifiers for RNA-Seq classification. A comprehensive simulation study is designed and four real datasets are used to assess the performance of developed approaches. Results revealed that voomNSC method performs as the sparsest classifier, also provides the most accurate results with power transformed Poisson linear discriminant analysis, and rlog transformed support vector machines and random forests algorithms. In conclusion, voomNSC is a fast, accurate and sparse classifier that can successfully be applied for diagnostic biomarker discovery and classification problems in medicine. This algorithm can also be used in other transcriptomics studies, such as separating developmental differences, cellular responses against stressors, or diverse phenotypes. An interactive web application is freely available at http://www.biosoft.hacettepe.edu.tr/voomDDA/ .

Keywords: Diagonal covariance matrix, discriminant analysis, gene expression, nearest shrunken centroids, next generation sequencing, RNA sequencing, voom, weighted statistic.

Supported by Research Fund of the Erciyes University, Ph. D. Thesis Grant (TDK-2015-5468).

ÖZET

Zararsız, G. RNA Dizileme Verilerinin Sınıflandırılmasında Yeni Makine Öğrenimi Yaklaşımlarının Geliştirilmesi ve Uygulanması. Hacettepe Üniversitesi, Sağlık Bilimleri Enstitüsü, Biyoistatistik Programı Doktora Tezi, Ankara, 2015. RNA dizileme, transkriptom karakterizasyonu ve nicelleştirmesinde yeni nesil dizileme teknolojisinin imkânlarını kullanan güncel ve etkin bir tekniktir. Bu teknik mikrodizin teknolojisine olan önemli avantajları ile gen ifadesi profillemesinde önemli gelişmeler kaydetmiştir: (i) daha az tutarsız veri üretme, (ii) yeni transkript ve izoformalarını tespit edebilme ve (iii) ilgilenilen transkriptler için ön hazırlık gerektirmeme. Gen ifadesi verisi kullanılarak yapılan önemli işlemlerden biri genlerin küçük bir alt setinin belirlenmesi ve özellikle kanser hastalıklarında tanı amaçlı verinin sınıflandırılmasıdır. RNA dizileme verilerinin kesikli veri yapısından dolayı, mikrodizin temelli sınıflandırıcılar doğrudan kullanılamamaktadır. Aşırı yaygınlık diğer bir problem olup, RNA dizileme verisinin ortalama ve varyans ilişkisinin dikkatli modellemesini gerektirmektedir. Voom, log-sayma değerlerinin ortalama ve varyans ilişkisini tahmin eden ve izleyen analizlerde kullanılmak üzere her gözlem için ağırlık katsayıları üreten güncel bir yöntemdir. Bu çalışmada biz güçlü bir mikrodizin sınıflandırıcısı olan en yakın küçültülmüş merkezler ve voom yaklaşımlarını bir araya getiren voomNSC yöntemini geliştirdik. VoomNSC ortalama ve varyans ilişkisini voom yöntemi ile modelleyen, voom yöntemi çıktılarını (log-cpm değerleri ve ağırlık katsayıları) ağırlıklandırılmış istatistikler kullanarak en yakın küçültülmüş merkezler yöntemine dâhil eden spars bir sınıflandırıcıdır. Ayrıca biz köşegenel doğrusal ve karesel ayırma analizlerinin RNA dizileme sınıflandırmasındaki uyarlamaları olan voomDLDA ve voomDQDA spars olmayan sınıflandırıcılarını da sağladık. Geliştirilen yaklaşımların performanslarının değerlendirilmesi için kapsamlı bir benzetim çalışması tasarladık ve dört adet gerçek veri seti kullandık. Bulgular, voomNSC yönteminin en spars sınıflandırıcı olduğunu, ayrıca üs dönüşümü uygulanmış Poisson doğrusal ayırma analizi, ve rlog dönüşümü uygulanmış destek vektör makineleri ve random forests yöntemleri ile birlikte en doğru sonuçları ürettiğini göstermiştir. Sonuç olarak, voomNSC, tıp alanında tanı biyobelirteçlerinin tespiti ve sınıflandırılma probleminde başarıyla uygulanabilir hızlı, tutarlı ve spars bir sınıflandırıcıdır. Ayrıca, bu algoritma gelişim farklılıklarının ayırt edilmesi, stres ajanlarına karşı hücresel yanıtın tespiti gibi çeşitli fenotiplerin ayrımında da kullanılabilir. İnteraktif web uygulamasına http://www.biosoft.hacettepe.edu.tr/voomDDA/ adresinden ücretsiz olarak ulaşılabilir.

Anahtar Kelimeler: Ağırlıklandırılmış istatistik, ayırma analizi, en yakın küçültülmüş merkezler, gen ifadesi, köşegenel kovaryans matrisi, RNA dizileme, yeni nesil dizileme, voom.

Erciyes Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon Birimince Desteklenmiştir. Doktora Tezi Projesi (TDK-2015-5468)

INDEX

	Page
Approval page	iii
Teşekkür	iv
Abstract	v
Özet	vi
Index	vii
Symbols and Abbreviations Index	ix
Figures Index	xii
Tables Index	xiv
1.INTRODUCTION	1
1.1.Problem Overview	1
1.2.Contribution	6
1.3.Organization of This Thesis	6
2.GENERAL INFORMATION	8
2.1.Machine Learning and Gene-Expression Based Classification	8
2.2.Next-Generation Sequencing	10
2.3.RNA-Sequencing	14
2.4.RNA-Sequencing Data Analysis Workflow	15
2.5.RNA-Sequencing Data	18
2.5.1.Notations	18
2.5.2.Discrete Models	19
2.5.3.Normalization	22
2.5.4.Transformation	24
2.6.Linear and Quadratic Discriminant Analysis	28
2.7.Diagonal Linear and Quadratic Discriminant Analysis	29
2.8.Nearest Shrunken Centroids	30
2.9.Poisson Linear Discriminant Analysis	34
2.10.Negative-Binomial Linear Discriminant Analysis	35
2.11.MLSeq Software for RNA-Seq Classification	36
3.MATERIAL and METHODS	38
3.1.VoomDDA Classifiers	38

3.1.1.Calculation of Log-Cpm Values and Estimation of Precision Weights 38
3.1.2.Classification Models Based on Diagonal Weighted Sample Covariance Matrices 40
3.1.3.Prediction of Test Observations for VoomDLDA and VoomDQDA Classifiers 40
3.1.4.Sparse VoomNSC Classifier for RNA-Seq Classification 42
3.1.5.Selection of the Optimal Threshold Parameter (λ) 44
3.1.6.Prediction of Test Observations for VoomNSC Classifier 45
3.2.Implementation of Classifiers 46
3.3.Evaluation of VoomDDA Classifiers 48
3.3.1.Simulation Study 48
3.3.2.Application to Real RNA-Sequencing Datasets 52
3.3.3.Evaluation Criteria 54
3.3.4.Computational Infrastructure and Parallel Programming 55
3.4.Development of a Web-Based Platform 56
4.RESULTS 58
4.1.Simulation Results 58
4.2.Real Dataset Results 96
4.3.Computational Cost of Classifiers 100
4.4.VoomNSC Classifiers in Diagnostic Biomarker Discovery Problems 100
5.DISCUSSION 105
6.CONCLUSION 109
REFERENCES 111
SUPPLEMENTARY MATERIAL
Supplementary File 1: User Guide Of VoomDDA Web Application
Supplementary File 2: All results for simulation and real datasets (CD)
Supplementary File 3: Analysis codes (CD)
Supplementary File 4: Web application files (CD)
Supplementary File 5: Figures in high-quality formats (CD)

SYMBOLS and ABBREVIATIONS

ADAS-Cog	Alzheimer Disease Assessment Scale-Cognitive Subscale
ALL	Acute Lymphoblastic Leukemia
AML	Acute Myeloid Leukemia
ASCII	American Standard Code for Information Interchange
BWA	Burrows-Wheeler Aligner
CART	Classification and Regression Trees
CDR	Clinical Dementia Rating
ChIP-Seq	Chromatin Immunoprecipitation Followed by Sequencing
DLDA	Diagonal Linear Discriminant Analysis
DNA	Deoxyribonucleic Acid
DNA-Seq	DNA Sequencing
DNase-seq	DNAse I Hypersensitive Site Sequencing
DQDA	Diagonal Quadratic Discriminant Analysis
FAIRE-Seq	Formaldehyde-Assisted Isolation of Regulatory Elements Followed by Sequencing
FN	False Negative
FP	False Positive
Indel	Insersion or the Deletion of the Bases
GC Content	Guanine-Cytosine Content
KICH	Kidney Chromophobe Carcinomas
KIRC	Kidney Renal Clear Cell
KIRP	Kidney Renal Papillary Cell
KNN	K-Nearest Neighbors
Lasso	Least Absolute Shrinkage and Selection Operator
LDA	Linear Discriminant Analysis
Limma	Linear Models for Microarray and RNA-Seq Data
lncRNA	Long Non-Coding RNA
Log-Cpm	Log Counts per Million
LOWESS	Locally Weighted Scatter Plot Smoothing
LUAD	Lung Adenocarcinoma

LUSC	Lung Squamous Cell with Carcinoma
miRNA	Micro RNA
MLE	Maximum Likelihood Estimation
MMSE	Mini-Mental State Exam
MNase-Seq	Micrococcal Nuclease Digestion Followed by Sequencing
mRNA	Messenger RNA
NGS	Next Generation Sequencing
NB	Negative Binomial
NBLDA	Negative Binomial Linear Discriminant Analysis
NSC	Nearest Shrunken Centroids
PCR	Polymerase Chain Reaction
PLDA	Poisson Linear Discriminant Analysis
QDA	Quadratic Discriminant Analysis
RCC	Renal Cell Cancer
Rlog	Regularized Logarithmic Transformation
RNA	Ribonucleic Acid
RNA-Seq	Transcriptome Sequencing or RNA Sequencing
RF	Random Forests
RPKM	Reads per Kilobase per Million Mapped Reads
SAM	Significance Analysis of Microarrays
SBS	Sequence-by-Synthesis
SNP	Single Nucleotide Polymorphism
SVM	Support Vector Machines
TC	Total Count
TCGA	The Cancer Genome Atlas
TMM	Trimmed Mean of M-Values
TN	True Negative
TP	True Positive
UQ	Upper Quartile
Voom	Variance Modeling at the Observational Level
VoomDDA	Voom Based Diagonal Discriminant Analysis
VoomDLDA	Voom Based Diagonal Linear Discriminant Analysis

VoomDQDA	Voom Based Diagonal Quadratic Discriminant Analysis
VoomNSC	Voom Based Nearest Shrunken Centroids
Vst	Variance Stabilizing Transformation
WMS	Wechsler Memory Scale

FIGURES

		Page
2.1.	Illumina (Solexa) sequencing workflow. (A) DNA or cDNA sample preparation for sequencing, (B) Bridge amplification and cluster generation of adapter ligated DNA fragments, (C) Sequencing by synthesis and imaging	13
2.2.	Voom mean-variance modeling for cervical data	27
2.3.	Optimization of shrinkage parameter in colon cancer microarray data	32
2.4.	Shrunken centroids for the colon cancer microarray data	33
2.5.	Gene expression level distributions of selected 15 genes in colon cancer microarray data	33
2.6.	A screenshot of MLSeq package in R/BIOCONDUCTOR network	37
3.1.	A flowchart of the steps of voomNSC algorithm	43
3.2.	Selection of voomNSC threshold parameter for cervical data	45
3.3.	Simulation design and the evaluation process	50
4.1.	Accuracy results for the simulation scenario K=2, e_{gk}=1%, σ=0.1	59
4.2.	Sparsity results for the simulation scenario K=2, e_{gk}=1%, σ=0.1	60
4.3.	Accuracy results for the simulation scenario K=2, e_{gk}=5%, σ=0.1	61
4.4.	Sparsity results for the simulation scenario K=2, e_{gk}=5%, σ=0.1	62
4.5.	Accuracy results for the simulation scenario K=2, e_{gk}=10%, σ=0.1	63
4.6.	Sparsity results for the simulation scenario K=2, e_{gk}=10%, σ=0.1	64
4.7.	Accuracy results for the simulation scenario K=3, e_{gk}=1%, σ=0.1	65
4.8.	Sparsity results for the simulation scenario K=3, e_{gk}=1%, σ=0.1	66
4.9.	Accuracy results for the simulation scenario K=3, e_{gk}=5%, σ=0.1	67
4.10.	Sparsity results for the simulation scenario K=3, e_{gk}=5%, σ=0.1	68
4.11.	Accuracy results for the simulation scenario K=3, e_{gk}=10%, σ=0.1	69
4.12.	Sparsity results for the simulation scenario K=3, e_{gk}=10%, σ=0.1	70
4.13.	Accuracy results for the simulation scenario K=4, e_{gk}=1%, σ=0.1	71
4.14.	Sparsity results for the simulation scenario K=4, e_{gk}=1%, σ=0.1	72
4.15.	Accuracy results for the simulation scenario K=4, e_{gk}=5%, σ=0.1	73
4.16.	Sparsity results for the simulation scenario K=4, e_{gk}=5%, σ=0.1	74
4.17.	Accuracy results for the simulation scenario K=4, e_{gk}=10%, σ=0.1	75

4.18. Sparsity results for the simulation scenario K=4, e_{gk}=10%, σ=0.1 76
4.19. Accuracy results for the simulation scenario K=2, e_{gk}=1%, σ=0.2 77
4.20. Sparsity results for the simulation scenario K=2, e_{gk}=1%, σ=0.2 78
4.21. Accuracy results for the simulation scenario K=2, e_{gk}=5%, σ=0.2 79
4.22. Sparsity results for the simulation scenario K=2, e_{gk}=5%, σ=0.2 80
4.23. Accuracy results for the simulation scenario K=2, e_{gk}=10%, σ=0.2 81
4.24. Sparsity results for the simulation scenario K=2, e_{gk}=10%, σ=0.2 82
4.25. Accuracy results for the simulation scenario K=3, e_{gk}=1%, σ=0.2 83
4.26. Sparsity results for the simulation scenario K=3, e_{gk}=1%, σ=0.2 84
4.27. Accuracy results for the simulation scenario K=3, e_{gk}=5%, σ=0.2 85
4.28. Sparsity results for the simulation scenario K=3, e_{gk}=5%, σ=0.2 86
4.29. Accuracy results for the simulation scenario K=3, e_{gk}=10%, σ=0.2 87
4.30. Sparsity results for the simulation scenario K=3, e_{gk}=10%, σ=0.2 88
4.31. Accuracy results for the simulation scenario K=4, e_{gk}=1%, σ=0.2 89
4.32. Sparsity results for the simulation scenario K=4, e_{gk}=1%, σ=0.2 90
4.33. Accuracy results for the simulation scenario K=4, e_{gk}=5%, σ=0.2 91
4.34. Sparsity results for the simulation scenario K=4, e_{gk}=5%, σ=0.2 92
4.35. Accuracy results for the simulation scenario K=4, e_{gk}=10%, σ=0.2 93
4.36. Sparsity results for the simulation scenario K=4, e_{gk}=10%, σ=0.2 94
4.37. Principal component analysis plots for each dataset 97
4.38. Distribution of dispersion statistics for each dataset 97
4.39. Heatmap plot for the selected miRNAs in cervical dataset 103
4.40. Heatmap plot for the selected miRNAs in alzheimer dataset 103
4.41. Heatmap plot for the selected genes in lung cancer dataset 103
4.42. Heatmap plot for the selected genes in renal cell cancer dataset 104
5.1. A Venn-diagram displaying the number of selected miRNAs from voomNSC algorithm and Witten et al. 108

TABLES

		Page
2.1.	Properties of various next-generation sequencing platforms	12
2.2.	An example count data matrix format for *p* genes, *n* samples and *K* classes.	20
3.1.	Confusion matrix for a classification model	54
3.2.	Computational properties of the used workstations in analysis	56
4.1.	Accuracy results of classifiers for real datasets	99
4.2.	Sparsity results of classifiers for real datasets	99
4.3.	Computational costs of classifiers for real datasets	101
4.4.	Summary of voomNSC models and selected genes in real datasets	102

1. INTRODUCTION

1.1. Problem Overview

In molecular biological studies, gene-expression profiling is among the most widely applied genomic technique to understand the role and the molecular mechanism of particular genes in interested conditions (1). Recent high-throughput technologies allow researchers to quantify the expression levels of thousands of genes simultaneously.

During the last seventeen years or more, microarray technology was very popular in gene expression profiling. Although microarray technology has high-throughput abilities, it has two major drawbacks (2):

1. Cross-hybridization[1] may occur and increase the noise of data,
2. Only prearranged transcripts can be measured on the array, thus novel transcripts cannot be detected.

With the recent developments in molecular biology, next-generation sequencing (NGS) has become the premier technology and preferred approach for –omics studies, including genomics, transcriptomics, epigenomics, metagenomics, etc. Transcriptome sequencing (RNA-Seq) benefits from the capabilities of NGS and is a powerful technique for examining comprehensive catalog of protein-coding and non-coding RNAs, and investigating the transcriptional activity of genomes (Wang et al., 2009). With diverse range of applications, RNA-Seq has already proved itself as a promising tool; (i) discovering novel transcripts, (ii) detecting and quantifying spliced isoforms, (iii) detecting fusions[2], (iv) detecting sequence variations (e.g, SNPs, indels). Beyond these applications, RNA-Seq is widely and effectively being used in gene expression studies due to some advantages over microarrays: (i) providing less noisy data, (ii) detecting novel transcripts and isoforms, and (iii) unnecessity of prearranged transcripts of interest. These advantages lead RNA-Seq technology to replace microarrays as the technology of choice and become the de facto standard in gene-expression studies (3).

Identifying the relevant genes across the conditions (e.g. tumor and non-tumor tissue samples) is a common research interest in gene-expression studies. In this gene selection, researchers are often interested in one of the following objectives (4):

---

[1]Mismatch of a DNA probe to a DNA molecule.
[2]A hybrid form of two distinct genes.

i. to detect a large set of differentially expressed genes that are related with the condition and apply further analysis to investigate their molecular roles to understand their association with the condition,

ii. to detect a small set of genes for diagnostic purpose in medicine that involves the identification of the minimal subset of genes that achieves maximal predictive performance.

This thesis focuses here on the second objective, which refers to the 'classification analysis' in statistical terminology. Classification analysis has great importance in gene-expression studies and used especially in medicine field to develop decision support systems for molecular diagnosis of diseases. The task is to classify and predict whether an individual has a disease (or a specific type of disease, e.g. subtype) or not based on the gene expression profile of biopsy or serum sample (5).

A particular interest of gene-expression based classification is the cancer classification problem. Traditional diagnosis is based on the morphological appearances of tissues under the light microscope and is subjective. The successful diagnosis of cancer is highly associated with the expertise of pathologists. High-throughput gene expression technologies provide objective and accurate solutions to assist and support clinicians in their decision (6). Apart from the disease diagnosis, this classification task can be used in other transcriptomics studies to separate developmental differences, cellular responses against stressors, or diverse phenotypes (3).

In order to make a successful classification based on the gene-expression profiles, powerful statistical algorithms are highly required. These algorithms should be able to cope with the high-throughput abilities of the recent technologies, identify the best minimal subset of genes and predict the condition categories accurately. Since, the number of genes (mostly in thousands) is much higher than the number of samples (mostly in tens to hundreds), curse of dimensionality[3] problem arises and the classical statistical algorithms do not work in this condition.

In microarrays, a great deal of machine learning algorithms is proposed and applied for the purpose of gene-expression based classification. Brown et al. (7) adapted support vector machines (SVM) algorithm for microarray classification and

[3]A problem that exists when an algorithm is not capable to scale with the high-dimension property of data.

showed its performance on yeast data. Dudoit et al. (8) derived discriminant analysis work with high-dimensional data with two of its extensions: (i) diagonal linear discriminant analysis (DLDA), (ii) diagonal quadratic discriminant analysis (DQDA). The authors compared the performance of these and several other classification algorithms in three real datasets. Tibshirani et al. (6) proposed a powerful approach, nearest shrunken centroids classifier, which determines the best subset of genes and classifies the data. The authors showed the effectiveness of the algorithm in two real datasets, developed software as well for the applicability of the proposed algorithm. Díaz-Uriarte et al. (4) evaluated the random forest (RF) algorithm for gene selection and classification purpose. The authors simulated data, also used 9 real datasets and found that their approach yields less number of genes than k-nearest neighbors (KNN), DLDA and SVM approaches while preserving the classification accuracy. Rapaport et al. (9) integrated a priori knowledge of a gene network and obtained both accurate and more interpretable classification results. Many other studies can be found from a simple 'PubMed' or 'ScienceDirect' searches with the following keyword: 'microarray classification'.

These algorithms cannot be directly applied to RNA-Seq data, since the type of the data is totally different. Albeit the continuous data format of microarrays, RNA-Seq data is summarized with nonnegative and integer-valued counts which exist from the number of mapped sequencing reads to genomic regions of interested specie. These mapped read counts are found to be correlated with the abundance of the target transcript (10). Since the data types are different, algorithms developed for microarrays are not directly applicable for RNA-Seq based gene-expression classification.

In the past few years, much effort was invested in modeling RNA-Seq data for differential expression analysis. Earlier studies applied microarray based methods after normalizing and taking the logarithm of counts (11-14). Later publications analyzed RNA-Seq data with specific methods designed for counts. Several Poisson distribution based statistical methods have been used for differential expression analysis (15-17). However, this distribution has a single parameter (λ) stands for both mean and variance; hence there is no need to estimate the variance. When we have only technical replicates, which means we have just one individual with technical

steps replicated, Poisson distribution based methods may be applicable. However, Nagalakshmi et al. (18) reported that the variance exceeds the mean if biological replicates (multiple individuals) are available in RNA-Seq data. This problem refers to overdispersion problem. To make an inference to the population and obtain more convincing results, biological replicates should be used. Thus, Poisson based methods are inapplicable and more care must be taken to model RNA-Seq data by considering the overdispersion problem (10). Much interest has been given to negative binomial (NB) distribution to overcome this problem. This distribution has two parameters uniquely determined by mean and variance. DESeq and edgeR are the two widely used powerful NB based approaches to model RNA-Seq data. Since the number of replicates in RNA-Seq data is usually small, these methods model the counts by estimating the mean and variance relationship. edgeR method uses a single proportionality constant, while DESeq applied local regression in this estimation (10,19).

Recently, variance modeling at the observational level (voom) method is proposed to open access microarray based methods for RNA-Seq analysis. Voom estimates the mean and variance relationship from the log counts and provides precision weights for downstream analysis. This method is integrated with limma (linear models for microarray and RNA-Seq data) method (20) and showed the best performance as compared to count based methods in controlling the type-I error rate, having the best power and lowest false discovery rate. Voom has various advantages than other methods: (i) observed mean-variance relationship matches more perfectly to theoretical mean-variance relationship after voom transformation, (ii) mean-variance trend is more precise, mostly for different RNA samples with different sequence depths, (iii) gives access to make use of empirical Bayes estimation theory, (iv) voom transformed normal distributed data with variance modeling partly supported by generalized linear model theory, (v) faster (1). The advantages of voom method and its well coordination with limma method in differential expression analysis point out to high performance results also for the other analysis types such as classification and clustering.

For the classification purpose, there is still less advancements for RNA-Seq data until recently. Witten proposed Poisson linear discriminant analysis classifier

(PLDA), which is an extension of Fisher's linear discriminant analysis to high-dimensional count data. PLDA shrinks the class differences to identify a subset of genes, and apply Poisson log linear model for classification (2). Dong et al. (5) extended this algorithm to build a new classification method based on NB distribution. The authors used a shrinkage method to predict the additional overdispersion parameter.

Law et al. (1) mentions that the count-based statistical methodology, such as differential expression is limited as compared to microarrays. This is also same for classification analysis. Even a wide range of algorithms are presented for microarray technology, this progress is very slow due to the type of the RNA-Seq data. This technological exchange may outdate most of the microarray classification methods that are based on normal distribution. Moreover, the mathematical theories of Poisson and NB distributions are not practicable as the normal distribution (1). Overdispersion problem is the main issue, which may have significant effect on classification accuracies. Poisson linear discriminant analysis (PLDA) algorithm is not capable to deal with this problem on its own. To overcome this, Witten et al. (2) applied a power transformation to stabilize the variance of genes and make the mean and variance relationship linear. However, the power transformation does not guarantee to account for the overdispersion. Since, there may be a trade-off between linearity and homoscedasticity[4]. This trade-off may limit the use of simple transformations in providing optimal results for both. Negative-binomial linear discriminant analysis (NBLDA) is a recent method, which uses a shrinkage estimator to predict the overdispersion parameter (21). This algorithm treats the estimated dispersion as a known parameter of asymptotic NB distribution and does not allow for the uncertainty of estimation. An imprecise dispersion estimate may be biased, which may lead NBLDA to give overly liberal results and directly affect the classification results.

Another solution may be to transform the data into continuous format in order to make the RNA-Seq data hierarchically closer to microarray data and make use of the flexibility of normal distribution. Zararsız et al. (3) transformed the data using variance stabilizing transformation (vst) and applied several machine learning algorithms including single SVM, bagging SVM, random forest, classification and

[4]A statistical assumption referring to the equality of variance across the predictor variables.

regression trees (CART) and PLDA. The authors developed MLSeq R/BIOCONDUCTOR package (22) to make computation less complicated for researchers and allow one to fit a model using the mentioned algorithms with one single function. These simple transformations (e.g. logarithmic and vst) have the limitation of obtaining less extreme values but still have unequal variances (23). If transformation is the strategy of choice in classification, one should use an appropriate transformation method that correctly models the mean and variance relationship of data, which is mentioned to be more important than specifying the exact probability distribution of counts (1). After transformation, powerful statistical algorithms are strongly required due to the small sample size settings.

1.2. Contribution

In this thesis, we present a sparse classifier voomNSC that brings together two powerful methods, voom method and nearest shrunken centroids algorithm, for the classification of RNA-Seq data. Basicly, voomNSC accepts either a normalized or non-normalized count data as input, applies voom method to data and provides precision weights for each observation, fits a NSC classifier by taking account of these weights. Thus, the main objective of proposing this approach is twofold:

1. to extend voom method for RNA-Seq classification studies,
2. to make NSC algorithm available for RNA-Seq technology,

Using voom method, we also made available the diagonal discriminant classifiers able to work with RNA-Seq data. Two diagonal RNA-Seq discriminant classifiers, voomDLDA and voomDQDA, will also be presented within the scope of this thesis.

1.3. Organization of This Thesis

We organized the rest of this thesis as follows. We detail the RNA-Sequencing in both the technological and the methodological view in 'General Information' section. We describe the bioinformatics analysis workflow, with important aspects and the commonly used methods and tools in each step. Here, we give a particular interest to normalization and transformation methods, which are crucial for statistical analysis of RNA-Seq data (i.e. differential expression, classification, clustering, etc.). We

describe how PLDA and NBLDA algorithms work in RNA-Sequencing based classification. We will also mention about low and high dimensional discriminant analysis approaches that underpin the basis of developed algorithms. In 'Material and Methods' section, we present the underlying theory of voomDDA classifiers (i.e. voomNSC, voomDLDA and voomDQDA). A comprehensive simulation study is designed for performance assessment. Four real studies are also used to illustrate the use of the proposed approaches. We discuss the details of these datasets and the evaluation process of voomDDA classifiers and other compared algorithms. In 'Results' section, we give the results of simulation and real dataset results. We discuss and conclude our study in 'Discussion' and 'Conclusion' sections. Illustration of voomDDA web application is given in the Supplementary section.

2. GENERAL INFORMATION

2.1. Machine Learning and Gene-Expression Based Classification

Machine learning (or statistical learning) is a subfield of statistics and computer science. It is concerned with the construction of computer algorithms that enables computers to assist humans make data-driven predictions and enhance with experience. The huge amount of data provided by the recent genetics technologies, such as microarrays and NGS, increased the use of these algorithms. These advances led machine-learning to be applied in various fields in genomics. These fields include DNA sequencing (DNA-Seq), RNA-Seq, small RNA-Seq, DNAse I hypersensitive site sequencing (DNase-seq), chromatin immunoprecipitation followed by sequencing (ChIP-Seq), formaldehyde-assisted isolation of regulatory elements followed by sequencing (FAIRE-Seq), micrococcal nuclease digestion followed by sequencing (MNase-Seq) and metagenome sequencing. Several examples of machine-learning applications in these genomics fields are as follows: (i) identification the location of transcription factor binding sites, alternative splicing sites, promoters, etc., (ii) classification of biological samples and prediction of clinical or other outcomes, (iii) detection of the functional annotations of genes, (iv) understanding the molecular mechanism of gene expression, (v) annotation of the genome and identification of novel functional classes, etc. (24).

Machine learning algorithms are mostly categorized into two parts: supervised and unsupervised learning. Supervised learning refers to training statistical models based on the labeled examples and make predictions based on the trained model to the unlabeled examples. Unsupervised learning algorithms do not require labels and aim to find patterns or structures in a dataset. In this thesis, we focus on the supervised learning category of machine learning in the application of gene-expression studies.

In gene-expression studies, researchers mostly collect samples from different conditions to identify the relevant genomic features and making predictions to in order to separate these conditions based on these features. These conditions may be the tumor classes (e.g. AML and ALL), tumor growth (e.g. growing and stable), treatment response (e.g. yes and no), survival status (e.g. surviving and exitus), pathogenic bacteria type (e.g. brucella and helicobacter), chemical compound type

(e.g. known and unknown), etc. Identification of the relevant genomic features, which may be genes, transcripts, micro RNAs (miRNAs[5]), etc., corresponds to biomarker discovery or feature selection problem, where the prediction of these conditions refers to the classification problem. The aim is mostly to get highly accurate predictions with the minimal subset of genomic features for separation of these conditions.

Using gene-expression data, both biomarker discovery and classification are crucial in medicine to assist physicians and other health professionals with decision making tasks, such as determining diagnosis of patient data. With the use of the capabilities of next-generation sequencing technology, detecting the most relevant genes (or exons, transcripts and isoforms) with an interested condition and developing a decision support system for clinical diagnosis will lead physicians to make more accurate diagnosis, develop and implement personalized, patient centered therapeutic interventions and improve the life quality of patients with better treatments.

A particular interest is the cancer classification problem. Current diagnostic approaches rely on the morphological appearance of tissue specimens, clinical and molecular variables. These methods have uncertainties in the diagnosis and subjective. For the same tissue specimen, there is not an obvious agreement among pathologists. Thus, the success of the diagnosis is highly dependent to the expertise of the pathologists. Gene-expression technologies and machine learning approaches make this classification objective, finer and more reliable. Since, expression levels of RNA are very dynamic, integrate both genetic and epigenetic information, and reflect the functional state of the cell. Because of its numerous advantages (detailed in Introduction section), RNA-Seq has been progressively becoming the standard technique in quantifying gene-expression. Therefore, RNA-Seq based discovery of biomarkers, and RNA-Seq based prediction of cancer classes (e.g. tumor type, tumor grade, chemotherapy type, recurrence, survival, etc.) will lead to the development of novel diagnostic methods, assays, even drugs (6,8,25,26). This will improve the management of cancer chemotherapy. With an accurate RNA-based diagnostic test (e.g. multigene expression assays[6]), it may be possible to detect the patients who are not responding to chemotherapy. Getting this information will guide oncologists on

---

[5]A short form of RNA molecules in approximately 21-23 bp. These molecules are non-protein coding RNAs and have role in post transcriptional gene-expression regulation and RNA silencing.
[6]Prognostic tests developed based on the expression profiles of multiple genes.

chemotherapy regimens, make them switch to alternative therapies, avoid the toxic side effects and improve the survival rate of patients. Besides increasing the survival probability, this kind of tests may improve the life-quality of patients by reducing the use of invasive tests such as the painful, sometimes life-threatening biopsy tests.

As well as messenger RNAs (mRNAs), non-coding RNAs play significant role on tumor progression and tumorigenesis. Again, these types of RNA molecules may be used for both disease diagnosis and monitoring of treatments. miRNAs have been used in detection of various types of cancers, such as colorectal cancer, pancreatic cancer, osteosarcoma and clear renal cell carcinoma. Discovering the miRNA biomarkers and using them for predictive purposes (e.g. via machine learning) have great potential in developing therapeutics. Since, miRNAs are short molecules and they have conserved known sequences among species. These properties make them challenging biomarkers for drug development (27). Apart from miRNAs, long non-coding RNAs[7] (lncRNAs) are the new players in cancer detection. For instance, it has been shown that MVIH lncRNA has over-expression, while H19 has under-expression in hepatocellular carcinoma. CRNDE has been found as over-expressed in various cancer types including hepatocellular, pancreatic, colorectal, prostate, leukemia, ovarian and gliomas (28). Similarly to miRNAs, identifying the potential lncRNAs will lead to an early diagnosis, prognosis and more accurate and personalized treatment of cancer diseases.

Such classification systems can also be used in other applications as to identify the types of species, separate developmental differences, cellular responses against stressors, or diverse phenotypes in transcriptomics.

2.2. Next-Generation Sequencing

Basically, DNA sequencing is the process of determining the precise order of nucleotide bases in a DNA molecule. Until now, various approaches using different chemical techniques have been attempted to determine DNA sequence. With the recent rapid advancement in sequencing technologies, determining the DNA sequence of an organism's genome shifted from low-throughput Sanger sequencing to higher throughput next-generation sequencing (NGS) platforms. Currently, a

[7]Non-coding RNAs with a length of longer than 200bp.

number of NGS platforms have been developed and commercialized for the accurate detection of DNA.

These sequencing approaches can be grouped into three main categories; (i) Second-generation sequencing platforms (Illumina, Roche 454, ABI/SOLID, Helicos BioScience), (ii) Third-generation sequencing platforms (Ion Torrent, Pasific Bio, Complete Genomics) and (iii) fourth-generation sequencing (Oxford Nanopore). All platforms have advantages and disadvantages in terms of sequencing and detection chemistries, accuracy, run time, throughput and so on (Table 2.1).

However, nowadays, Illumina is the mostly used platform and preferred by end-users due to its high-throughput, accuracy, fast and easy sample preparation procedure. Illumina (Solexa-based sequencing) technology is based on the sequence-by-synthesis (SBS) incorporation of fluorescent nucleotides. The stage of Illumina sequencing can be divided into three steps: sample preparation, cluster generation and sequencing. Sample preparation is a pre-sequencing process including; (i) fragmenting DNA or fragmenting RNA, first and second strand cDNA synthesis, (ii) repairing 3’ and 5’ ends of DNA or cDNA fragments, (iii) adding an Adenine base to the 3’ ends, (iv) ligating pair-end adaptors to the end of fragments, (v) PCR reactions and validation of libraries. All these steps render sequencing libraries compatible for cluster generation and sequencing-by-synthesis. After libraries are constructed, DNA or cDNA fragments with specific adaptors are passed through a flow cell which will hybridize the individual molecules on flow cell based on complementary with adaptor sequences. It is important to note that before attachment to the flow cell, the library fragments are denatured, and thus a single-stranded copy of the library fragment is sequenced. Following this step, hybridized sequences held at both ends of the adaptor on a solid phase will be amplified as a bridge. After all, one million copies of each template in successive cycles of denaturation, amplification and hybridization between oligonucleotides on flow cell and DNA fragments. This entire process is known as cluster generation. After the clusters are generated and one strand removed from DNA fragments, sequencing reagents (mainly, DNA polymerase and fluorescent nucleotide) are passed through the flow cell to do

Table 2.1. Properties of various next-generation sequencing platforms

Platform	**Illumina**	**Roche 454**	**ABI/SOLID**	**Helicos**	**Pacific Biosciences**	**Ion Torrent**	**Complete Genomics**	**Oxford Nanopore**
Next-generation sequencing	Second generation	Second generation	Second generation	Second generation	Third generation	Third generation	Third generation	Fourth generation
Sequencing chemistry	Sequencing by synthesis	Pyrosequencing	Sequencing by ligation	Single molecular sequencing	Real-time single polymerase	Sequencing by synthesis	Probe-anchor capture and ligation	Single molecule sequencing
Detection chemistry	Florescence	Luminescence	Florescence	Florescence	Florescence	Proton/pH detection	Florescence	Scanning tunneling electron microscope
Read length (bp)	150	~400	25-35	25-30	10,000	100-200	10	5,400
Output (Gb)	1000-1200	0.70	120	25	0.50-1.0	100	3000	Unknown
Throughput	Very high	High	Very high	High	High	Very high	Very high	Unknown
Accuracy	99.90%	99.90%	99.99%	99.00%	80.00%	98.00%	99.50%	95.00%
Run time	1-11 days	10 hours	8 days	8 days	0.5-2 hours	2 hours	11 days	0.15-6 hours
Advantages	• Sample material less than 100 pg • High throughput • High accuracy	• Long read length	• High accuracy	• Nonbiased DNA sequence	• Real-time monitoring of nucleotide incorporation • No amplification of template DNA required	• High accuracy • Fast turn-around time	• High accuracy	• Fastest sequencing platform (15 min, human genome 6 hours)
Limitations	• Short read length (<300 bp)	• High error rate in homopolymer regions • Long sample preparation procedure • Expensive	• Long run time • Require more starting material • Long sample preparation procedure	• High error rate	• High error rate	• High frequencies of homopolymers		• High error rate (4%)

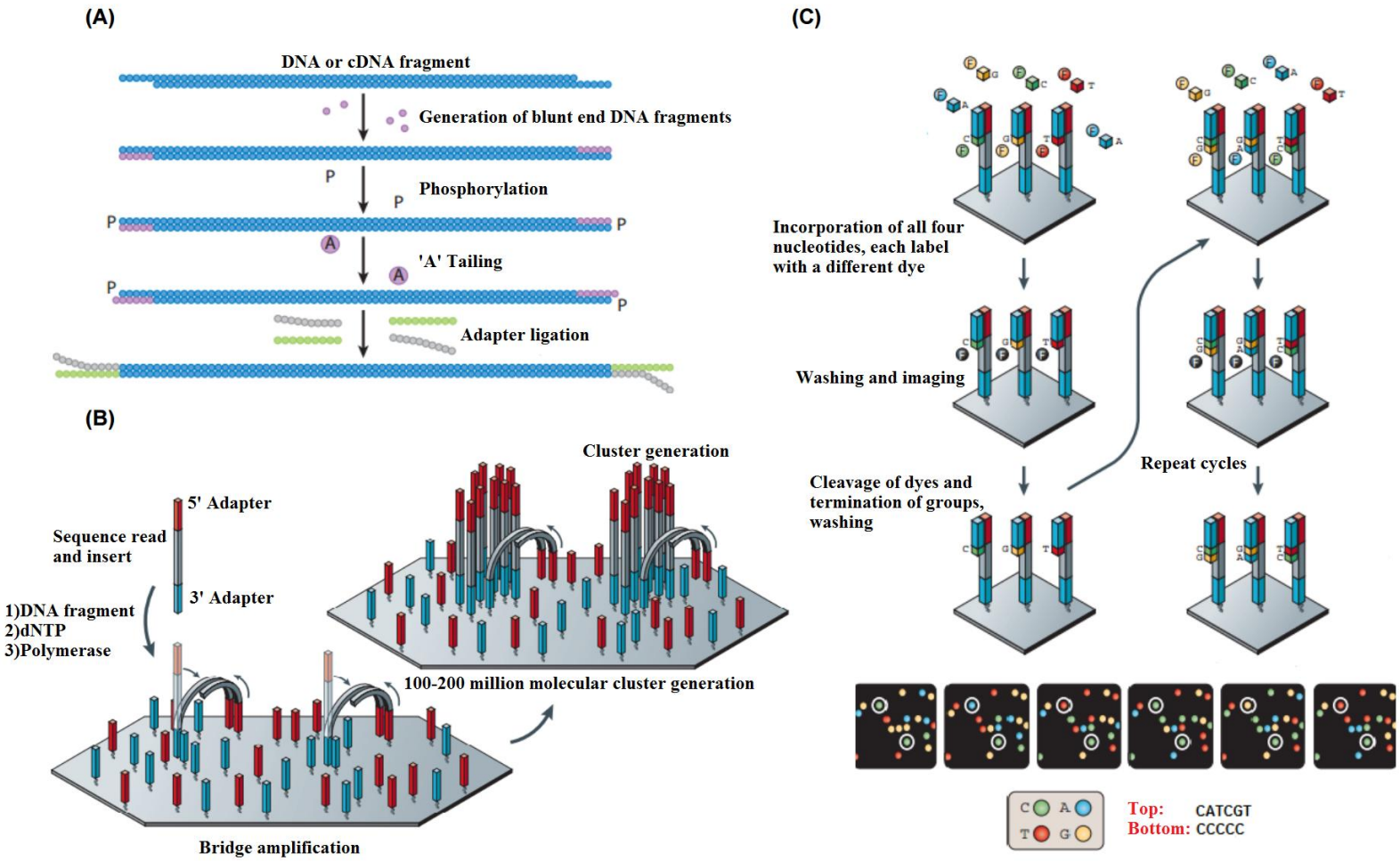

Figure 2.1. Illumina (Solexa) sequencing workflow. (A) DNA or cDNA sample preparation for sequencing, (B) Bridge amplification and cluster generation of adapter ligated DNA fragments, (C) Sequencing by synthesis and imaging

sequencing by synthesis. Sequencing by synthesis defines a reaction where in each synthesis cycle, the addition of a single nucleotide, which can be A, C, G, or T, as determined by a fluorescent signal, then is imaged, so that the location and added nucleotide can be determined, stored, and analyzed. Reconstruction of the sequence of additions in a specific location on the flow cell, which corresponds to a generated DNA cluster, provides the precise nucleotide sequence for an original piece of DNA fragments (Figure 2.1) (29-31).

2.3. RNA-Sequencing

In the biological perspective, the term of transcriptome is defined as the complete set of all expressed RNA transcripts including; protein-coding (mRNA) and non-coding (rRNA, tRNA, snRNA, snoRNA, miRNA, lncRNA, piwi-interacting RNA and so on) RNA species. Uncovering all functioning transcripts at genome-wide level by various methods provide a considerable amount of information regarding the molecular mechanism of specific cellular functions.

Currently, two experimental techniques are in use for identifying transcripts on a genome-wide scale; microarray and next generation RNA sequencing (RNA-Seq) technologies. In most cases, when comparing two methods, it is reported that RNA sequencing has a superior performance over microarray technologies in terms of the dynamic range of transcripts (RNA-Seq is at least 8000-fold, compared with ~60-fold for microarrays), less noisy data, detecting novel transcripts and isoforms. Therefore, RNA-Seq replaced microarrays as the technology of choice in finding novel transcripts and gene expression profiling.

As a more comprehensive way, RNA-Seq based transcriptome approaches are being routinely used for various purposes; (i) characterizing transcriptome profile, (ii) measuring the expression level of transcripts, (iii) detecting splicing isoforms and fusion transcripts, (iv) finding novel transcripts in the genome, (v) profiling small regulator RNAs and (vi) identifying coding variants. Since the expression dynamics of protein-coding genes must be regulated precisely during development, disease and other physiological conditions in specific cell types, the measuring gene expression levels is particularly appealing among these applications. In addition, the differences in spatial and temporal expression pattern can be associated with cancer

development. Therefore, robust statistical methods are urgently needed for classifying the gene expression information obtained from RNA-Seq based expression profile.

2.4. RNA-Sequencing Data Analysis Workflow

In this section, we provide a pipeline and describe the commonly used computer-assisted statistical methods to give a quick snapshot view to how to handle the high-throughput RNA-Seq data.

Experimental Design: In statistical hypothesis testing, it is vital to plan a good experimental design to obtain the maximal information with minimum cost. This is also same in RNA-Seq analysis, since a common interest in RNA-Seq analysis is the identification of the relevant genes with the interested condition. Thus, we aim to model the relationship between each gene and class condition, and need a good experimental design. A good experimental design should include the basic principles of Fisher: (i) replication, (ii) randomization and (iii) blocking (32). Replication refers to the number of samples in each run. Technical and biological replicates are the two types of replication. Technical replicates refer to one individual with technical steps replicated, while biological replicates refer to multiple individuals. In a good experimental design, biological replicates should be preferred and increased to increase the power of the used statistical test, and generalize the results. Pooling the biological samples may be alternative design, but will provide less powerful results. Randomization is the process of randomly allocating the samples across the conditions and should be applied during sample preparation to minimize the technical variation. It is also recommended to index and multiplex[8] the samples to reduce the effect of lanes and flow cells. If this is not possible, a blocking design should be preferred including some samples in each condition arranged in each sequencing lane (33,34).

Power Analysis: Another issue in RNA-Seq analysis is the power estimation based on the used statistical method. In this way, one can determine a satisfactory sample size for differential expression analysis. Most of the current differential expression analyses are based on complex negative binomial models. Several power estimation

[8]Sequencing multiple samples in a lane simultaneously.

methods can be found in (35-37). Scotty is another tool that can be used for this purpose with an extra option to estimate the sequencing depth (38).

Raw Data: In a single run, NGS platforms provide millions of short sequence reads with corresponding quality scores for each base call. The data format may differ based on the sequencing platform. Illumina produces these reads in FASTQ format. These FASTQ files consists four lines for each read. An example format is as below:

```
@genkokbioinf_3615
GATAGTAGGTTCTAAGCAGTATCGATCAAATAGTAAATCCTCTTGTT
+
!''*(+)(%**+))%%%++)(%%%%).1***-+*''))**55CCF>>
```

First line begins with the @ character and followed with an identifier of the read. Second line consists the raw sequence reads. Third line begins with the + character and optionally followed by the same identifier. Last line includes the quality scores for each base-call in ASCII code.

Quality Assessment: Before starting to analysis, quality assessment of the data is a crucial step that may affect the results of further analysis (e.g. alignment accuracy). NGS platforms provide quality assessment results by themselves. However, these results focus on detecting the problems in the sequencing process. Quality assessment process should consider detecting the problems in both sequencing and the other experimental processes such as starting library material. FASTQC is a widely preferred software in assessing the quality of NGS reads (39). This tool contains several modules including basic statistics, per base sequence quality, per sequence quality scores, per base sequence content, per base GC content[9], per sequence GC content, per base N content, sequence length distribution, duplicate sequences, overrepresented sequences and overrepresented Kmers. Other quality assessment tools include HTSeq (40), R ShortRead (41) and PRINSEQ (42).

Filtering: After the quality assessment, next step is to preprocess the data to obtain high-quality and clean reads. These preprocessing include quality filtering, quality trimming, removing sequencing adapters, masking nucleotides with 'N', etc. FASTX

[9]Percentage of guanine or cytosine nucleotides in a DNA sequence ((G+C)/(A+T+G+C)).

toolkit is a collection of these and some other modules (43). QTrim is another popular tool used for this purpose (44).

Alignment: Following the quality assessment and filtering procedures, next step is the read alignment. In alignment step, high-quality and clean reads are mapped to a reference genome or transcriptome. Each model organism has reference FASTA files similar to FASTQ format, but have no quality scores. An example format is as below:

```
>1 dna:chromosome chromosome:GRCm38:1:1:195471971:1 REF
TTCTGTTTCTATTTTGTGGTTACTTTGAGGAGAGTTGGAATTAGGTCT
TCTTTGAAGGTCTGGTAGAACTCTGCATTAAACCCATCTGGTCCTGG
GCTTTTTTTTTTTTTTTTTTTTTTTTTTGGGTGGGAGACTATTGATGAC
TGCCTCTATTTCTTTAGGGGAAATGGGACTTTTAGTCCATGAATCTG
ATCCTGATTTAGCTTTGGTACCTGGTATCTGTCTAGGAAGTTGTCCAT
TTCATCCAGGTTTTCCTGGTTTTTTTTTAGTATAGCCTTTCATAGTAA
AATCTGATGATGTTTTTGATATCCTCATGTTCTGTTGGTATGTCTCCT
TTTTCATTTCTGATTTTGTTAATTATAGTACAGTCCCTATGCCCTCTA
GTTAGTCTGGCTAAGGGTTTATCTATCTTGTTGACTTTCTCAAAGAAC
CAGCTACTATTTTGGTTGATTCTTTGAATATTTCTTTTTGTTTCCACTT
GGTTGATTTCAGCTCTGAGTTTGATTATTTCCTGCTGTCTACTCATCT
TGGGTGAATTTGCTTCCTTTTGTTCTAGAGCTTCTAGATTTGCTGTCA
GGCTGCTAGTGTATACTCTAGTTTCCTTTTGGAGGCACACAGGCCTG
TGAGTTTTACTCTTAGGACTGCCTCATTGTGCCCCAT...
```

First line begins with the > character and followed with a single line description. Followed lines contain the sequences in each genomic position. Burrows-Wheeler Aligner (BWA) (45) and Bowtie2 (46) are the standard aligners for DNA based NGS analysis. These aligners are also applicable for RNA-Seq analysis, if all coding regions or splice junctions are known. If one cares about detecting novel transcripts or alternative-splicing, it is critical to use splice-aware aligners. Tophat2 is a splice-aware extension of Bowtie2 algorithm and among the

most widely applied tool in RNA-Seq analysis (47). MapSplice (48) and Star (49) are the other commonly used splice-aware aligners.

De-novo Transcriptome Assembly: One can use de-novo transcriptome assembly[10] methods while working with non-model organisms. In this way, a reference file can be generated from the data itself and the transcript abundances can be provided for further analysis. Popular de-novo transcriptome assembly tools include Trinity (50), Oases (51), SOAPdenovo-Trans (52) and Trans-ABySS (53). After generating assemblies, RSEM (54) and CORSET (55) algorithms can be used to estimate the transcript abundances.

Feature Counting: It has been reported that the number of mapped reads is correlated with the abundance of transcripts. Thus, we need to count the number of mapped reads to each transcript to obtain the expression levels. This step can be accomplished using featureCounts (56), HTSeq (40) and bedtools (57) tools.

After this step, the raw count data is obtained which is the input for differential expression analysis, also the developed algorithms in this thesis.

2.5. RNA-Sequencing Data

2.5.1. Notations

The main difference between RNA-Seq and microarray data matrices is that the matrix elements are nonnegative and integer-valued counts in RNA-Seq, where they are the log-intensities in microarrays. In this section, we will describe RNA-Seq data and introduce some notations that will be used throughout the thesis.

After feature counting process, we obtain a pxn dimensional $\mathbf{X}$ count matrix, where p is the number of genomic features $g = \{1, \dots, p\}$, and n is the number of observations (e.g. tissue samples) $i = \{1, \dots, n\}$. These genomic features may refer to genes, transcripts, exons, spliced mRNA isoforms, non-coding RNAs or any predefined transcriptome subsets. For simplicity of language, we will use the term gene to refer genomic features and the term sample to refer observations throughout the thesis. Each matrix element, let's say x_{gi}, is the number of mapped read counts to g^{th} gene for i^{th} sample. We define $x_g = (X_{g1}, \dots, X_{gn})$ denote the row g and $x_i = (X_{1i}, \dots, X_{pi})$ denote the column i of $\mathbf{X}$ matrix. We define $X_{.i} = \sum_{g=1}^{p} x_{ij}$ the library size or the total number of counts for sample i, $X_{g.} = \sum_{i=1}^{n} x_{ij}$ total number of

[10]Construction of a transcriptome without the information of a reference genome.

counts that mapped to g^{th} gene and $X_{..} = \sum_{g=1}^{p} \sum_{i=1}^{n} x_{ij}$ the total library size or the sum of the number of counts within the **X** matrix. Let **y,** a vector with a length *n*, containing the class labels of each sample and let *K* the number of biological conditions or classes $y_i \in k = \{1, \dots, K\}$. Let $C_k = \{i: y_i = k\}$ denotes the set of sample indices belonging to k^{th} class. Finally, $\boldsymbol{x}_* = (X_{*1}, \dots, X_{*g})^T$ is a vector of new test samples whose class labels y_* will be predicted.

An example of RNA-Seq data matrix is given in Table 2.2. This data contains the read counts of Witten et al. (58) cervical data. This data matrix **X** contains the counts of 714 miRNAs ($p = 714$), belonging to 58 samples ($n = 58$). These samples belong to 2 classes ($K = 2$) where 29 of them are non-tumor ($n_1 = 29$) and the remaining 29 are tumor ($n_2 = 29$). The first element of the matrix has the count value of 865 ($X_{11} = 865$), corresponding to 865 mapped sequence reads to the let-7a miRNA for the first non-tumor sample. The library size for this sample is 22,449 ($X_{.1} = 22{,}429$), total mapped read count for let-7a miRNA is 284,257 ($X_{1.} = 284{,}257$) and the total library size is 13,701,148 ($X_{..} = 13{,}701{,}148$).

2.5.2. Discrete Models

The discrete type of RNA-Seq data lead researchers to present models based on discrete probability distributions. Poisson based models are considered in several studies (15-17),

$$x_{gi} | y_i = k \sim Poisson(g_g s_i) \quad (2.1)$$

This model allows for the variability in both g_g term, the total number of counts per gene, and the s_i term, the total number of counts per sample, where $\sum_{i=1}^{n} s_i = 1$. Poisson models are valid when technical replicates are used in experiments and $g_g s_i$ represents both mean and the variance (18). Due to the reproducibility of RNA-Seq protocols and to make an inference to

Table 2.2. An example count data matrix format for *p* genes, *n* samples and *K* classes.

miRNA	**Non-tumor samples**					**Tumor samples**					**Total**
	Sample-1	**Sample-2**	**Sample-3**	**...**	**Sample-29**	**Sample-30**	**Sample-31**	**Sample-32**	**...**	**Sample-58**	
let-7a	865	810	5,505	...	38	3,343	4,990	5,193	...	1,422	284,257
let-7g	447	173	1,922	...	126	737	4,141	3,760	...	2,081	128,348
miR-125b	1,038	5,007	2,595	...	106	381	1,463	201	...	1	337,394
miR-18a	5	4	10	...	0	10	9	56	...	0	634
miR-29a	320	447	904	...	4	413	4,619	1,398	...	1	134,382
miR-490-5p	0	1	0	...	0	0	0	0	...	0	19
miR-874	2	4	9	...	0	4	0	3	...	0	509
...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...
miR-93	10	18	126	...	0	36	133	211	...	0	7,028
miR-99b	24	92	97	...	177	36	62	19	...	20	11,718
Total	22,449	39,798	71,717	...	8,034	58,362	431,247	84,850	...	16,338	13,701,148

In this matrix, the rows indicate genes, where the columns indicate samples. The numbers in each cell are the mapped sequence read counts (to a reference genome or transcriptome) of gth gene for i^{th} sample. First column of this table describe the names of genes, first row describe the names of classes and second row describe the name of samples. The numbers in last column are the total number of counts for g^{th} gene. The numbers in last row are the library sizes (i.e. the total number of counts for i^{th} sample).

population, biological replicates are used (59). In case of the presence of biological replicates, counts have variance excluding the mean and the overdispersion problem arises. Negative binomial (NB) distribution can model the overdispersed counts with an extra dispersion parameter of $\phi > 0$, which reduces to Poisson distribution when $\phi \to 0$,

$$x_{gi}|y_i = k \sim NB\left(g_g s_i, \phi_g\right) \quad (2.2)$$

In this condition, NB distribution is parameterized with mean $\mu_{gi} = g_g s_i$, and variance $Var(x_{gi}) = g_g s_i + \left(g_g s_i\right)^2 \phi_g$. These two models are extended in some studies as follows (10,15,58,60,61):

$$x_{gi}|y_i = k \sim Poisson\left(g_g s_i e_{gk}\right) \quad (2.3)$$
$$x_{gi}|y_i = k \sim NB\left(g_g s_i e_{gk}, \phi_g\right) \quad (2.4)$$

Here, e_{gk} term allows the gth gene to be differentially expressed in the kth class. Witten et al. (2) considered the Poisson model and maximum-likelihood parameter estimation (MLE) method in fitting counts. For an expected size of counts $\mu_{gi} = g_g s_i$, the authors used the MLE as $\bar{x}_{gi} = X_{g.} X_{.i} / X_{..}$ (62), $\hat{s}_i = X_{.i} / X_{..}$ and $\hat{g}_g = X_{g.}$. They fit the model as follows:

$$x_{gi}|y_i = k \sim Poisson\left(\bar{x}_{gi} e_{gk}\right) \quad (2.5)$$

Assuming the prior distribution of e_{gk} to be Gamma(θ, θ), MLE estimate is provided with $\hat{e}_{gk} = \frac{X_{gC_k} + \theta}{\sum_{i \in C_k} \bar{x}_{gi} + \theta}$. $\hat{e}_{gk}$ can be interpreted as follows:

- if $\hat{e}_{gk} > 1$, then the gth gene is up-regulated[11] in class *k*,
- else if $\hat{e}_{gk} < 1$, then the gth gene is down-regulated[12] in class *k*.

[11]Increase in the expression level of a gene or other cellular component.

[12]Decrease in the expression level of a gene or other cellular component.

2.5.3. Normalization

Total counts $X_{g.}$ and $X_{.i}$ are dependent on the experimental design. This leads to existence of technical biases. These biases can have significant effect on the statistical results, should be detected and corrected. The source of variation may come out either from non-systematic and systematic biases (63).

Firstly, different samples may have very different library sizes within a single RNA-Seq experiment. A sample may have higher counts not only from the abundance of RNAs for the gth gene, but also from the sequencing depth. The library size $X_{.i}$ would be higher, if i^{th} sample is deeply sequenced. For instance, we can say that 31st sample in cervical dataset has sequence depth at least 50 times higher than sample 29 (Table 2.1). This may lead to substantial problems in downstream analysis, unless it is corrected.

Another source of variation may arise from the gene length. At the same expression level, we expect that more reads will be aligned to a longer gene than a shorter one. In this way, statistical tests will have higher power when analyzing longer genes than for shorter genes (63). Thus, the total number of counts for g^{th} gene $X_{g.}$ will be higher for longer genes.

Other technical biases may arise from the sequence composition (e.g. GC content), the presence of majority fragments in experiment, sampling bias in library construction and PCR amplification (33). Due to all these technical biases, each count should be normalized before conducting statistical analysis. Here we describe a number of methods used in this correction.

Total count normalization: MLE estimate $\hat{s}_i = X_{.i}/X_{..}$ can be considered as a size factor estimate (15). This is known as total count (TC) normalization method. The limitation of this normalization method is that the library size for sample i is strongly associated with a few highly expressed genes, which may skew the analysis results.

Upper quartile normalization: Bullard et al. (60) overcame this limitation with using the upper quartile (UQ) of counts (q_i), instead of total counts. The size factor estimate is $\hat{s}_i = q_i / \frac{1}{n}\sum_{i=1}^{n} q_i$. Note that only genes with non-zero counts are included to this calculation (33).

Median normalization: Median normalization is similar to upper quartile normalization. Only difference is that median value is used in place of third quartile ($q_i = median(X_{gi})$).

Quantile normalization: This normalization is primarily used in microarrays. All quantiles are matched between lanes, in normalizing counts (64).

Reads per kilobase per million mapped reads normalization: Together with the sequence depth, reads per kilobase per million mapped reads (RPKM) method also adjust the counts for gene length. In RPKM method, size factors are estimated using $\hat{s}_i = \frac{10^3 x 10^6}{X_{.i} L_g}$. L_g refers to the length of gth gene. RPKM allows comparing expression levels of genes within a sample; however this may lead to unbiased variance estimates of counts (65).

Trimmed mean of M-values normalization: Trimmed mean of M-values (TMM) method firstly trims the data based on the log-fold-changes[13] (M_{gi}^r, as default 30% in edgeR software) and the absolute intensity (A_g, as default 5% in edgeR software). Next, TMM calculates the weighted mean of gene-wise log-fold-changes (M_g). The weights are calculated using the delta method (66) which can be considered as the inverse of the asymptotic variances. In this normalization, TMM uses a reference sample (r) in calculating normalization factors (61):

$$log_2(TMM_i^r) = \frac{\sum_{g=1}^{p'} \varpi_{gi}^r M_{gi}^r}{\sum_{g=1}^{p'} \varpi_{gi}^r} \qquad (2.6)$$

where $M_{gi}^r = \frac{log_2(x_{gi}/X_{.i})}{log_2(x_{gr}/X_{.r})}$ and $\varpi_{gi}^r = \frac{X_{.i} - x_{gi}}{X_{.i}\, x_{gi}} + \frac{X_{.r} - x_{gr}}{X_{.r}\, x_{gr}}$; $x_{gi}, x_{gr} > 0$.

In the formula, p' corresponds to genes which are not trimmed and used in the calculation. $X_{.r}$ is the library size for the reference sample. If the user does not provide any reference sample, edgeR software (19) normalizes the data using UQ method and selects the reference sample that has count values closer to mean value than the other samples.

[13] A value describing the variation from an initial to final value.

Deseq median ratio normalization: This method generates a pseudo-reference sample by calculating the geometric means across samples. Size factors are estimated from the median of the count ratio of ith and pseudo-reference sample, over all g genes:

$$m_i = median_g \left\{ \frac{x_{gi}}{\left(\prod_{i=1}^{n} x_{gi}\right)^{1/n}} \right\} \quad (2.7)$$

Size factors are estimated from $\hat{s}_i = m_i / \sum_{i=1}^{n} m_i$ (10).

Using the methods described above, normalized counts can be obtained from $x'_{gi} = x_{gi}/\hat{s}_i$. TMM and deseq median ratio normalizations are found to be more effective approaches in correcting for sequence depths. Both approaches minimize the effect of majority sequences (67).

2.5.4. Transformation

Even the count data is normalized, microarray based statistical methods are not applicable due to the highly skewed nature of the data. RNA-Seq has a large dynamic range (0 to 10^5) than microarrays (68). The sample-to-sample variation is higher in genes with larger expression than the lower ones. This brings the heteroscedasticity problem. Instead of using discrete models, one may consider transform the data matrix **X** to **Z** and assume $z_{gi} \sim N(\mu_{z_{gi}}, \sigma^2_{z_{gi}})$. Rendering the RNA-Seq count data approximately homoscedastic will allow the use of various methods such as classification, clustering, principal component analysis, etc. Here, we introduce several specialized transformation methods in this section:

Shifted logarithmic transformation: The simplest way to scale RNA-Seq data is the log2 transformation. However, the existence of zero counts will lead to infinite values. A basic solution is to add a small constant of 1 to avoid this problem:

$$z_{gi} = log_2(x_{gi} + 1) \quad (2.8)$$

Even the transformed data has less skew distribution with less extreme values, the variances are still unequal for all genes.

Variance-stabilizing transformation: Anders et al. (10) presented variance-stabilizing transformation (vst) to transform RNA-Seq data. Vst is based on the variance stabilization concept (69) and error modeling. After transformation, vst provides variances that are independent from the mean. We obtain the vst transformed data as follows:

$$z_{gi} = \int_{0}^{x_{gi}} \frac{1}{var(x_g)} dx_g \quad (2.9)$$

where $var(x_g) = \mu_g + \phi_g \mu_g^2$, and $\phi_g = \phi_0 + \frac{\phi_1}{\mu_g}$.

In the formula, μ_g is the mean and $var(x_g)$ is the variance for g^{th} gene. ϕ_0 and ϕ_1 are estimated using generalized linear models (70). Vst is an effective transformation method in variance stabilization. However, it does not work so well for data with unequal library sizes (23).

Regularized logarithmic transformation: Love et al. (23) proposed regularized logarithmic (rlog) transformation by taking into account the unequal library sizes. For this purpose, rlog transformation applies a shrinkage approach as used in DESeq2 method (23). Transformed values are similar to vst or log2 transformed values for genes with high counts, while shrunken together for genes with low counts. Rlog transformation ignores the class label of data and considers all samples as replicates. The rlog transformation is calculated as follows:

$$z_{gi} = log_2(g_{gi}) = \beta_{g0} + \beta_{gi} \quad (2.10)$$

In this formula, β_{g0} corresponds to the baseline gene expression level for each sample, while β_{gi} corresponds to the shrunken log-fold-changes of the normalized counts in base 2 scale.

Power transformation: Witten et al. (2) considered applying a simple power transformation to the counts. For $\alpha \in (0,1]$, transformed count values can be obtained from the following formula:

$$z_{gi} = x_{gi}^{\alpha} \quad (2.11)$$

α is selected using a grid search based on the goodness of fit of the model (62):

$$\sum_{g=1}^{p}\sum_{i=1}^{n}\frac{\left(z_{gi} - \frac{Z_{.i}Z_{g.}}{Z_{..}}\right)}{\left(\frac{Z_{.i}Z_{g.}}{Z_{..}}\right)} \approx (p-1)(n-1) \quad (2.12)$$

In this formula, $z_{gi} \in \boldsymbol{Z}$, $Z_{.i}$ denotes to the total number of transformed counts for sample i, $Z_{g.}$ denotes to total number of transformed counts for gene g, and $Z_{..}$ is the sum of the number of transformed counts within the **Z** matrix. Witten et al. (2) assumed that z_{gi} follow a Poisson distribution ($z_{gi}|y_i = k \sim Poisson(\mu_{z_{gi}})$ since **Z** contains non-integer values.

Voom transformation: Unlike other transformation methods, voom takes **X** as input and extracts **Z** and **W**, also pxn dimensional, matrices. **Z** contains z_{gi}, the log-counts per million (log-cpm) values for each sample and each gene, and **W** is the precision weight matrix containing the variances of each log-cpm value. Counts per million (cpm) is a basic measure which is calculated from the ratio of each count (+ 0.5, a small constant) to its library size in millions. Law et al. (1) considered log-cpm values analogous to the log-intensities in microarrays with the difference that log-cpm values do not have constant variances. Unlike logarithmic transformation, log-cpm accounts for the unequal library sizes and makes libraries scaled and comparable with each other. However, depending on the heteroscedasticity of the data, genes with higher counts have larger variances. The authors demonstrated that the standard deviation of log-cpm values is nearly equal to the coefficient of variation of raw counts. Further, it is mentioned that the mean-variance trend for log-cpm values are smoothly decreasing with count size and asymptotes to a value for genes with higher counts depending on the dispersion or biological variability (Figure 2.2) (1).

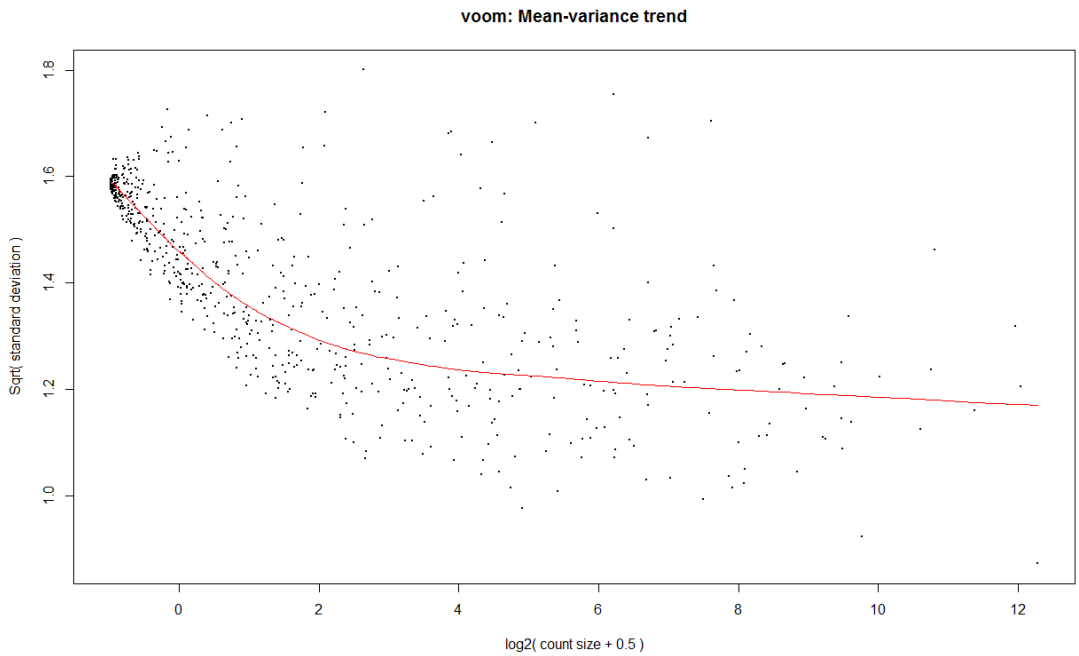

Figure 2.2. Voom mean-variance modeling for cervical data (2)

To eliminate the mean-variance effect of log-cpm values, the authors provided the precision weights as well. These weights are estimated non-parametrically from the mean-variance relationship at the observational level instead of gene level as other methods. This yields robust estimates even for data with unequal library sizes. Locally weighted scatter plot smoothing (Lowess) curves[14] are used for this purpose, the variances of log-cpm values are estimated using this relationship. Inverse of these variances are considered as the weights $w_{gi} \in \boldsymbol{W}$ for sample i and gene g. This enables users to downweight unreliable samples or measurements and increases the power in gene expression analysis. These weights can be considered as analogous to empirical quality weights of microarrays, where the poor-quality samples are downweighted similarly. Voom method has several advantages as compared to other methods:

1. observed mean-variance relationship matches more perfectly to theoretical mean-variance relationship after voom transformation,
2. mean-variance trend is more precise, mostly for different RNA samples with different sequence depths,
3. gives access to make use of empirical Bayes estimation theory,

[14]A nonparametric regression method which smoothes a curve between two variables on a scatter plot.

4. voom transformed normal distributed data with variance modeling partly supported by generalized linear model theory.

The strong capabilities of voom method allow to use similar workflows for microarrays and RNA-Seq techniques, also allow obtaining results comparable with each other. Law et al. (1) entered these weights into the limma analysis pipeline with the log-cpm values for differential expression analysis. Incorporation of these weights into a linear modeling pipeline led limma to perform better than the other count-based differential expression methods in controlling the type-I error rate, having the best power and lowest false discovery rate. The advantages of voom method given above and its well coordination with limma method in differential expression analysis points out to high performance results also for the other analysis types such as classification and clustering.

Zwiener et al. described several other transformation approaches, such as Box-Cox, Blom, rank transformation, etc. to scale RNA-Seq data, where the details can be found in (70).

2.6. Linear and Quadratic Discriminant Analysis

Let **X** and **Y** random variables for a matrix of expression data and a vector of class labels, respectively. Let $k = 1,2,\ldots,K$ the classes, π_k the prior probabilities for class *k*, where $\sum_{k=1}^{K} \pi_k = 1$. Let $f_k(x) = P(X = x|Y = k)$ class-conditional density function of **X**, for class *k*.

In statistical decision theory, posterior class probabilities $P(Y = k|X = x)$ are needed to be estimated for optimal classification. Bayes rule can simply be used to motivate this estimation:

$$P(Y = k|X = x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^{K} f_l(x)\pi_l} \quad (2.13)$$

Modeling the class-conditional density function has the most importance in estimating posterior probabilities and many techniques are present to model $f_k(x)$, such as:

1. Gaussian densities,

2. Mixtures of Gaussian densities,
3. Nonparametric density estimates,
4. Naïve Bayes models.

Linear discriminant analysis (LDA) uses the first model, multivariate normal distribution in modeling $P(X = x|Y = k)$:

$$P(X = x|Y = k) = \frac{1}{(2\pi)^{\frac{p}{2}}\Sigma^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)} \quad (2.14)$$

Here, μ_k and Σ are parameters referring to class-specific mean vector and the common within variance-covariance matrix, respectively. After we replace Formula 2.14 into Formula 2.13 and do some algebra, we obtain the linear discriminating function, which is linear in x_*:

$$\delta_k^{LDA}(x_*) = x_*^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + log\ \hat{\pi}_k \quad (2.15)$$

x_* is a new test observation here, which will be assigned to the class maximizes the $\delta_k^{LDA}(x_*)$. μ_k, Σ and π_k are unknown parameters and are estimated from the training data:

- $\bar{x}_k = \sum_{i=1}^{n} x_i/n_k$: sample mean vector for class k,
- $\hat{\Sigma} = \sum\sum(x_i - \bar{x}_k)\,(x_i - \bar{x}_k)^T$: sample pooled variance-covariance matrix,
- $\hat{\pi}_k = n_k/n$.

Quadratic discriminant analysis (QDA) is a basic extension of LDA. QDA assumes that the variance-covariance matrices are not common within classes ($\Sigma_k \neq \Sigma\forall_k$). Thus, the sample variance-covariance matrices should be calculated separately in each class to estimate Σ_k.

2.7. Diagonal Linear and Quadratic Discriminant Analysis

LDA is inapplicable in high-dimensional settings ($n < p$). The problem arises due to the singularity of the estimated variance-covariance matrix ($\hat{\Sigma}$). The inverse of the

estimated variance-covariance matrix ($\hat{\Sigma}^{-1}$) cannot be calculated, which is required in the computation of $\delta_k^{LDA}(x_*)$.

In microarray classification, diagonal extensions of LDA and QDA are presented (8). Here, the class-conditional densities are estimated using Naïve Bayes models. These models assume that the genes are independent in each class, thus the class densities are obtained from the products of marginal densities. Since the genes are independent with each other in this setting, the covariances are assumed to be zero. This is also known as 'independence rule[15]' and leads to calculation of diagonal covariance matrices $\hat{\Sigma}_{C=k} = diag(s_{1k}^2, \dots, s_{pk}^2)$, where the off-diagonal elements are all set to be zero (8,71). It has been showed that diagonal extensions of discriminant classifiers outperform the traditionally used LDA and QDA in high-dimensional classification analysis. If we use diagonal variance-covariance matrices, we obtain the following discriminant rule for class *k*:

$$\delta_k^{DLDA}(x_*) = -\sum_{j=1}^{p} \frac{(x_{g*} - \bar{x}_{gk})^2}{s_g^2} + 2\log(\hat{\pi}_k) \quad (2.16)$$

This discrimination rule is called as diagonal linear discriminant analysis (DLDA). $\bar{x}_{gk}$ is the sample mean of gth gene in class *k* and s_g^2 is the sample pooled variance of gth gene. Again, a new test observation (x_*) will be assigned to the class which maximizes the $\delta_k^{DLDA}(x_*)$ discriminating function. The second rule, diagonal quadratic discriminant analysis (DQDA) assumes that the gene specific variances are not equal across groups. So, each class variance should be calculated separately (s_{gk}^2 instead of s_g^2) for g^{th} gene to estimate σ_{gk}^2.

2.8. Nearest Shrunken Centroids

Despite the high-dimensional capabilities of DLDA and DQDA classifiers, decision boundaries are generated from all genes. This leads obtaining very complex models in classifying high-dimensional data. When the number of genes is large, it is essential to work with a significant gene subset that contributes most to class prediction. In this way, we can get more simple, interpretable and reduced variance

[15]Ignoring the covariance structure of the data in model building.

models (63). To overcome the complexity of diagonal discriminant classifiers, Tibshirani et al. (6) developed nearest shrunken centroids (NSC) algorithm, an extension of diagonal discriminant classifiers. NSC is a sparse classifier which uses a shrinkage procedure to identify the most relevant gene subsets in class prediction. NSC shrinks the standardized class means (or centroids) of genes to the standardized overall means, then eliminates the genes which have shrunken means and finally builds a classification model with the remaining genes.

The steps of NSC algorithm is as follows: (i) calculation of difference scores between class means and overall mean, (ii) shrinkage of difference scores to zero using soft-thresholding, (iii) elimination the shrunken genes and keeping the remaining of them, (iv) building a DLDA classifier with the updated means.

A new test observation (x_*) will be assigned to the class that will maximize the following $\delta_k^{NSC}(x_*)$ discriminating function:

$$\delta_k^{NSC}(x_*) = -\sum_{j=1}^{p} \frac{(x_{g*} - \bar{x}_{gk})^2}{(s_g + s_0)^2} + 2\log(\hat{\pi}_k) \quad (2.17)$$

s_0 is set to be a positive constant to make the gene expression levels independent across genes. Mostly s_0 is calculated from the median value of s_g over the set of genes. Posterior class probabilities $P(Y = k|X = x)$ for both diagonal discriminant analysis and NSC classifier can be calculated as follows:

$$\hat{p}_k(x_*) = \frac{e^{-\delta_k(x_*)/2}}{\sum_{l=1}^{K} e^{-\delta_l(x_*)/2}} \quad (2.18)$$

In identifying the subset of genes, k-fold cross-validation[16] is applied to find the optimal shrinkage parameter. The parameter which gives the most accurate and sparse classification model is considered as the optimal parameter and used for prediction. Here, we illustrate it with an example of microarray data. We use the colon cancer data of Alon et al. (72) which contains the expression levels of 2,000 genes belonging to 62 samples. Of these 62 samples, 40 of them are tumor, while the remaining 22 are normal samples. We consider detecting the optimal gene subset that

[16]A repeated model validation procedure that the data is split into k folds in which the training procedure is applied in the k-1 fold and tested on the remaining 1 fold.

will accurately predict whether the class of a new test observation is tumor or normal. After performing cross-validation technique and a grid search of shrinkage parameter between 0 and 5.5, we find 3.218 as optimal value. This value gives the most accurate and sparse model in predicting colon cancer samples. Using this optimal value, only 15 genes will be considered in the classification process with an accuracy of 85.5% (Figure 2.3).

Shrunken centroids and the distribution of expression levels of selected genes are displayed in Figure 2.4 and Figure 2.5.

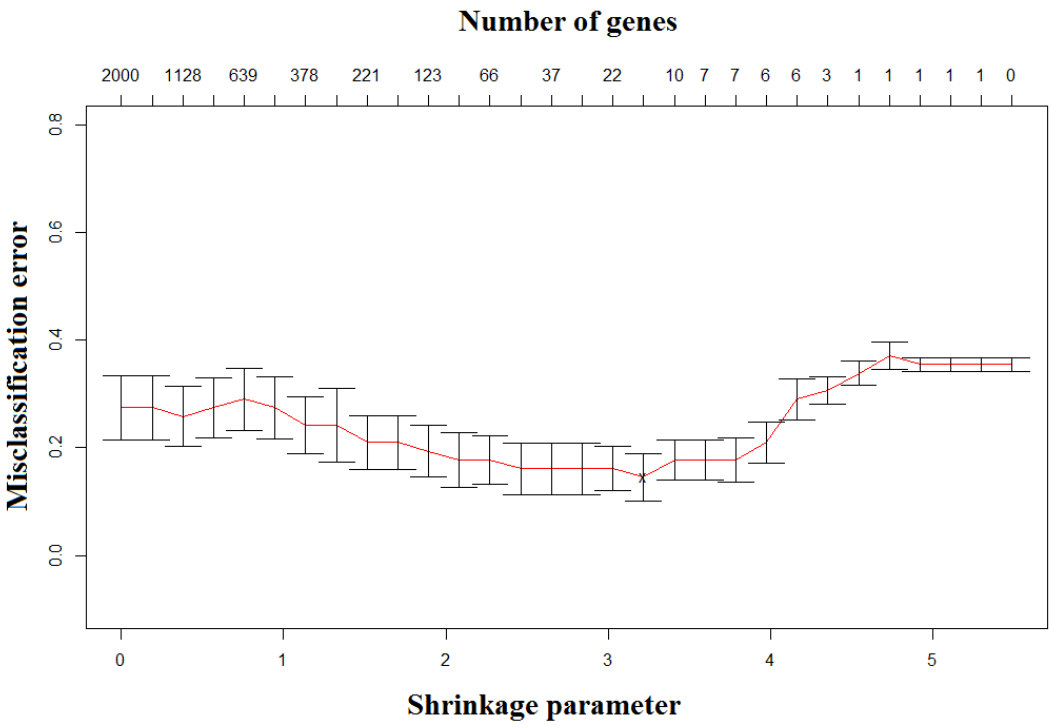

Figure 2.3. Optimization of shrinkage parameter in colon cancer microarray data (72)

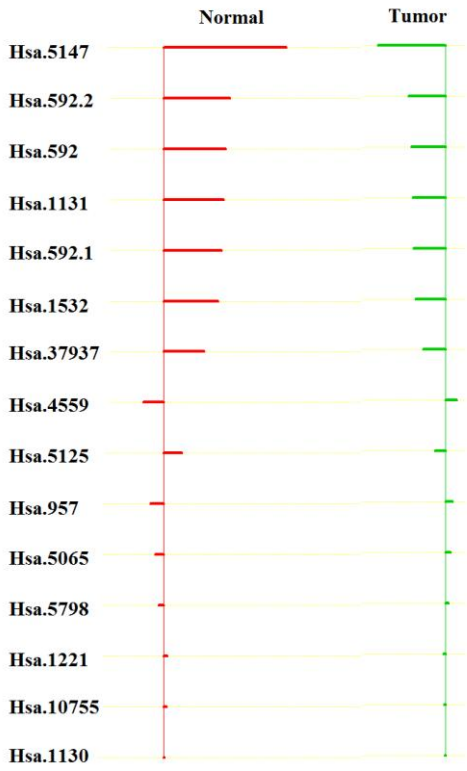

Figure 2.4. Shrunken centroids for the colon cancer microarray data (72)

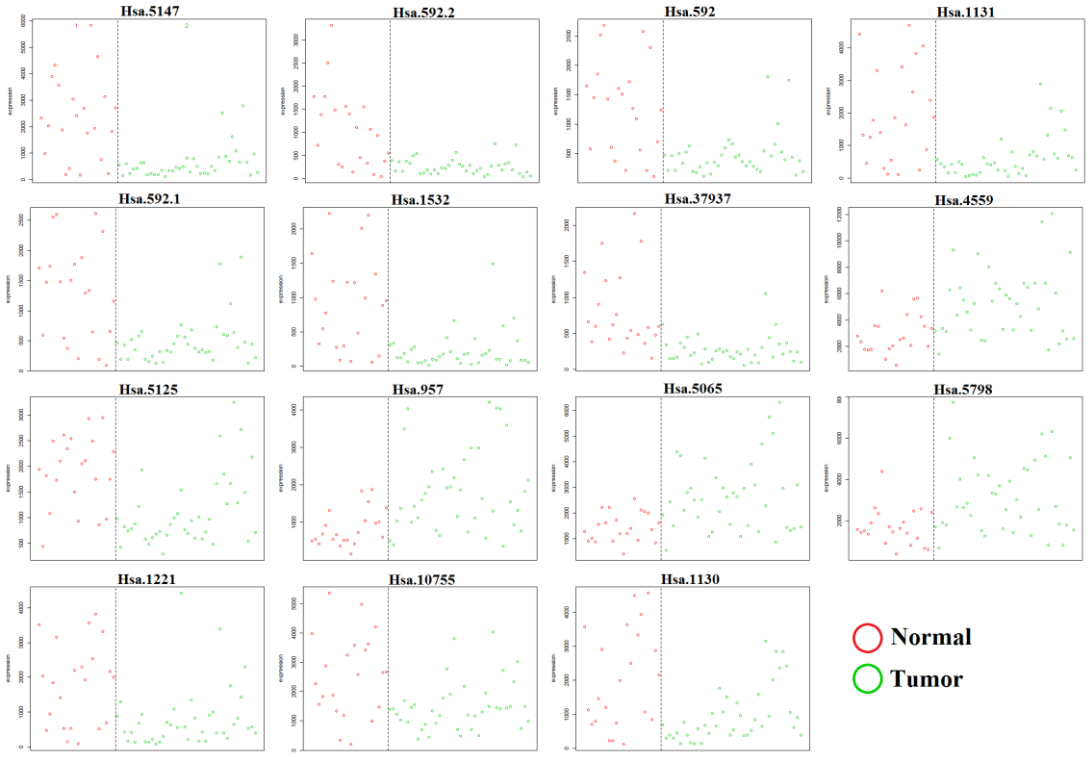

Figure 2.5. Gene expression level distributions of selected 15 genes in colon cancer microarray data (72)

2.9. Poisson Linear Discriminant Analysis

Poisson linear discriminant analysis (PLDA) is developed by Witten et al. (2) for the classification of RNA-Seq data. PLDA is an extension of Tibshirani et al. (6) NSC algorithm for high-dimensional discrete data. Witten et al. (2) considered modeling class-conditional densities from the product of marginal Poisson densities. In other words, samples in the kth class are assumed to have a Poisson distribution of Formula 2.3 and the genes are independent with each other. Writing Formula 2.3 into Formula 2.13 and performing some algebra, we obtain the following discriminating function:

$$\delta_k^{PLDA}(x_*) = \sum_{g=1}^{p} x_{g*} log\hat{e}_{gk} - \hat{s}_* \sum_{g=1}^{p} \hat{e}_{gk}\hat{g}_g + log\hat{\pi}_k \quad (2.19)$$

A new test observation x_* will be assigned to the class that maximizes $\delta_k^{PLDA}(x_*)$. Formula 2.19 is linear in x_* and contains all genes in the model unless $\hat{e}_{gk} = 1$ for all classes k. $\hat{g}_g$ is estimated as $\hat{g}_g = X_{g.}$. Estimation of $\hat{s}_*$ will be discussed in Section 3.1.1.

Similar with the shrinkage of difference scores to zero in NSC classifier, PLDA shrinks $\hat{e}_{gk}$ towards 1 using soft-thresholding method. Assuming the prior distribution of e_{gk} to be Gamma(θ, θ), $\hat{e}_{gk}$ is estimated as follows:

$$\hat{e}_{gk} = \begin{cases} \frac{a}{b} - \frac{\lambda'}{b}, & if\ \sqrt{b}\left(\frac{a}{b} - 1\right) > \lambda' \\ \frac{a}{b} + \frac{\lambda'}{b}, & if\ \sqrt{b}\left(1 - \frac{a}{b}\right) > \lambda' \\ 1 & if\ \sqrt{b}\left|1 - \frac{a}{b}\right| < \lambda' \end{cases} \quad (2.20)$$

where $a = X_{gC_k} + \theta$, $b = \sum_{i \in C_k} \bar{x}_{gi} + \theta$. λ' is the threshold parameter, which is optimized using cross-validation method. Shrunken differential expression rates can be obtained from:

$$e'_{gk} = sign(e_{gk})(|e_{gk}| - \lambda') \quad (2.21)$$

2.10. Negative-Binomial Linear Discriminant Analysis

Dong et al. (5) extended PLDA considering negative-binomial distribution as class-conditional densities as in Formula 2.4. Due to the variability which arises from biological replicates, RNA-Seq data is overdispersed and Poisson models are inappropriate in this setting. Witten et al. (2) suggested a power transformation to overcome this problem. Alternatively, Dong et al. (5) proposed negative-binomial linear discriminant analysis (NBLDA), which takes advantage of the extra dispersion parameter of negative-binomial distribution in solution of overdispersion problem. NBLDA estimates the dispersion parameter using a shrinkage approach of (21).

Class-conditional densities used in NBLDA method is given as follows:

$$P(X = x|Y = k) = \frac{\Gamma\left(x_{gi} + \frac{1}{\phi_g}\right)}{x_{gi}!\,\Gamma\left(\frac{1}{\phi_g}\right)} \left(\frac{g_g s_i e_{gk} \phi_g}{1 + g_g s_i e_{gk} \phi_g}\right)^{x_{gi}} \left(\frac{1}{1 + g_g s_i e_{gk} \phi_g}\right)^{\frac{1}{\phi_g}} \quad (2.22)$$

After plugging Formula 2.22 into Formula 2.13 and performing some algebra, we obtain the discriminating function of NBLDA:

$$\delta_k^{NBLDA}(x_*) = \sum_{g=1}^{p} x_{g*}(log\hat{e}_{gk} - \log(1 + \hat{g}_g \hat{s}_* \hat{e}_{gk} \hat{\phi}_g)) - \sum_{g=1}^{p} \frac{1}{\hat{\phi}_g} \log(1 + \hat{g}_g \hat{s}_* \hat{e}_{gk} \hat{\phi}_g) + log\hat{\pi}_k \quad (2.23)$$

A new test observation x_* will be assigned to the class that maximizes $\delta_k^{NBLDA}(x_*)$. The relationship between NBLDA and PLDA can be addressed as follows:

$$If \quad \hat{\phi}_g \to 0 \qquad then \qquad \log\left(1 + \hat{g}_g \hat{s}_* \hat{e}_{gk} \hat{\phi}_g\right) \to \hat{g}_g \hat{s}_* \hat{e}_{gk} \quad (2.24)$$

It means that decreasing the dispersion parameter will approximate the data distribution towards Poisson, thus will approximate NBLDA towards PLDA. For this

reason, Dong et al. (5) describes NBLDA as a generalized version of PLDA classifier.

Estimation of $\hat{g}_g$ and $\hat{e}_{gk}$ is similar with PLDA algorithm. Estimation of $\hat{s}_*$ will be discussed in Section 3.1.1. To estimate $\hat{\phi}_g$ NBLDA uses the shrinkage method of (21) which shrinks the gene-specific estimates towards a target value using method of moments:

$$\hat{\phi}_g = \varphi\xi + (1-\varphi)\tilde{\phi}_g \quad (2.25)$$

where φ is a weight defined as:

$$\varphi = \frac{\sum_{g=1}^{p}\left\{\tilde{\phi}_g - \frac{1}{p}\sum_{g=1}^{p}\tilde{\phi}_g\right\}^2 / (p-1)}{\sum_{g=1}^{p}\left(\tilde{\phi}_g - \xi\right)^2 / (p-2)} \quad (2.26)$$

In the formula, $\tilde{\phi}_g$ refers to the initial dispersion estimates obtained from the method of moments, ξ is the target value calculated from:

$$\xi = min\left\{\sum_{g=1}^{p}\left(\hat{\phi}_g - \tilde{\phi}_g\right)^2\right\} \quad (2.27)$$

2.11. MLSeq Software for RNA-Seq Classification

Zararsız et al. (22) presented an R package in BIOCONDUCTOR network to make RNA-Seq classification less complicated for researchers and allow users to fit classifiers using single functions. MLSeq package requires from users to upload their raw count data in which can be obtained from feature counting tools (e.g. HTSeq (40), featureCounts (56), etc.) and allow them to normalize, transform and build classifiers including SVM, bagging SVM, RF and CART (Figure 2.6).

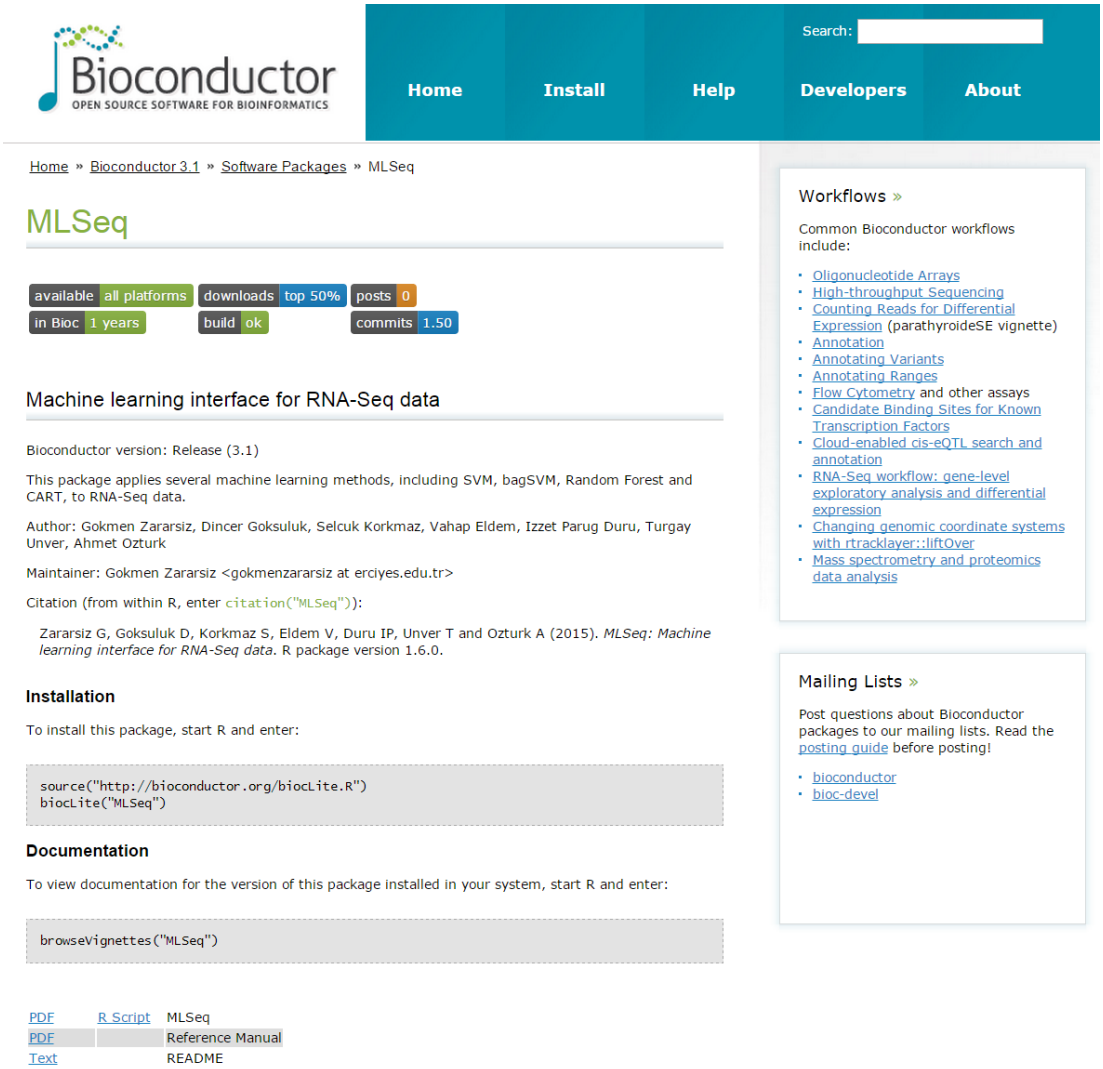

Figure 2.6. A screenshot of MLSeq package in R/BIOCONDUCTOR network

3. MATERIAL and METHODS

3.1. VoomDDA Classifiers

In this section, we detail the methodology of voomDDA classifiers. We assume that the input data is a pxn dimensional count data matrix, which consists of either x_{gi} raw or x'_{gi} normalized count values. Moreover, genes with zero or very low counts should be filtered before starting to analysis. For simplicity, we will assume throughout this section that the input data is a pxn dimensional, filtered and non-normalized count data matrix **X**.

3.1.1. Calculation of Log-Cpm Values and Estimation of Precision Weights

Firstly, we calculate the log-cpm values using the following formula:

$$z_{gi} = log_2\left(\frac{x_{gi} + 0.5}{X_{.i} + 1} x10^6\right) \quad (3.1)$$

Small constant values 0.5 and 1 in the formula are used to avoid taking logarithm of zero and guaranteeing that $0 < (x_{gi} + 0.5)/(X_{.i} + 1) < 1$.

To estimate the weight matrix **W**, containing the variances of log-cpm values w_{gi}, we benefit from the delta rule, linear models and lowess smoothing curves. We assume a linear model between the expected size of the log-cpm values and the class conditions as follows:

$$E(z_{gi}) = \mu_{z_{gi}} = y_i^T \beta_g \quad (3.2)$$

In the formula, β_g corresponds to a vector of regression coefficients to be estimated. These coefficients are the log-fold-changes between class conditions (1). Matrix notation of this equation is as follows:

$$E(z_g) = D\beta_g \quad (3.3)$$

D in the formula represents the design matrix[17] with the rows y_i. z_g is a vector containing the log-cpm values for g^{th} gene. For each gene, we fit these models using ordinary least squares method and obtain the fitted coefficient $\hat{\beta}_g$, fitted log-cpm values $\hat{\mu}_{z_{gi}} = y_i^T \hat{\beta}_g$ and the standard deviations of residuals s_g.

Let $\bar{z}_g = \sum_{i=1}^{n} \hat{\mu}_{z_{gi}}/n$ the mean log-cpm value for g^{th} gene, and $\tilde{X}_{.n} = (\prod_{i=1}^{n}(X_{.i} + 1))^{1/n}$, the geometric mean of the library sizes plus one. Using delta rule, we obtain the mean log-counts $\tilde{x}_g$ as follows:

$$\tilde{x}_g \approx \bar{z}_g + log_2(\tilde{X}_{.n}) - 6log_2(10) \quad (3.4)$$

Fitted counts are calculated from the fitted log-cpm values accordingly:

$$\hat{\mu}_{gi} \approx \hat{\mu}_{z_{gi}} + log_2(X_{.i} + 1) - 6log_2(10) \quad (3.5)$$

Now, we estimate the mean-variance relationship for each gene, using the mean log counts $\tilde{x}_g$ and the square root of residual standard deviations $s_g^{1/2}$. A lowess curve (73) is fitted with the $g(.)$ smoothing function as follows:

$$s_g^{1/2} = g(\tilde{x}_g) \quad (3.6)$$

A piecewise linear function $lo(.)$ is obtained from the fitted lowess curve by interpolating the curve for the $\tilde{x}_g$ values in order. Finally, we obtain the w_{gi} precision weights (i.e. variances of log-cpm values) as follows:

$$w_{gi} = lo(\hat{\mu}_{gi})^{-4} \quad (3.7)$$

Log-cpm values z_{gi}, and the associated precision weights w_{gi} will be used in model building process of voomDDA classifiers.

[17]A matrix containing the values of exploratory variables and an optional indicator variable (0 or 1).

3.1.2. Classification Models Based on Diagonal Weighted Sample Covariance Matrices

We assume the genes independent from each other in building classification rules. Let $\bar{z}_{w^{gk}} = \left(\sum_{i=1}^{n} w_{gik}\, z_{gik}\right) / \sum_{i=1}^{n} w_{gik}$ class-specific weighted mean for k^{th} class, $\bar{z}_{w^{g}} = \left(\sum_{i=1}^{n} w_{gi}\, z_{gi}\right) / \sum_{i=1}^{n} w_{gi}$ overall weighted mean, $\hat{\Sigma}_{\mathrm{w}^{\mathrm{C=k}}} = diag\left(s^2_{w^{1k}},, \ldots, s^2_{w^{pk}}\right)$ diagonal weighted sample covariance matrices for k^{th} class and $\hat{\Sigma}_{\mathrm{w}} = diag\left(s^2_{w^{1}},, \ldots, s^2_{w^{p}}\right)$ weighted pooled covariance matrix. The diagonal elements of these matrices are obtained from the class specific and pooled weighted variances respectively. The off-diagonal elements of these matrices are all set to be zero. The weighted pooled variance of gth gene can be calculated as follows:

$$s^2_{w^{g}} = \frac{\sum_{i=1}^{n} w_{gi}}{\left(\sum_{i=1}^{n} w_{gi}\right)^2 - \sum_{i=1}^{n} w^2_{gi}} \sum_{i=1}^{n} w_{gi}\left(z_{gi} - \bar{z}_{g}\right) \quad (3.8)$$

The weighted variance of gth gene in class k can be calculated as follows:

$$s^2_{w^{gk}} = \frac{\sum_{i=1}^{n} w_{gik}}{\left(\sum_{i=1}^{n} w_{gik}\right)^2 - \sum_{i=1}^{n} w^2_{gik}} \sum_{i=1}^{n} w_{gik}\left(z_{gik} - \bar{z}_{gk}\right) \quad (3.9)$$

Here, we define voomDLDA and voomDQDA classifiers, which are extensions of DLDA and DQDA classifiers for RNA-Seq data with the weighted parameter estimates. voomDLDA assumes that the gene specific weighted variances are equal across groups and use the weighted pooled covariance matrix in modeling class-conditional densities $f_k(x)$. The second rule voomDQDA uses separate covariance matrices $\hat{\Sigma}_{\mathrm{w}^{\mathrm{C=k}}}$ obtained by class-specific weighted variance statistics.

3.1.3. Prediction of Test Observations for VoomDLDA and VoomDQDA Classifiers

Discriminant rules for voomDLDA and voomDQDA classifiers are given as below:

$$\delta_k^{voomDLDA}(x_*) = -\sum_{g=1}^{p} \frac{(z_{g*} - \bar{z}_{wgk})^2}{s_{wg}^2} + 2\log(\hat{\pi}_k) \quad (3.10)$$

$$\delta_k^{voomDQDA}(x_*) = -\sum_{g=1}^{p} \frac{(z_{g*} - \bar{z}_{wgk})^2}{s_{wgk}^2} + 2\log(\hat{\pi}_k) \quad (3.11)$$

A new test observation (x_*) will be assigned to class which maximizes the $\delta_k^{voomDLDA}(x_*)$ or $\delta_k^{voomDQDA}(x_*)$. An important point here is that same parameters should be used for both training and test sets to guarantee that both sets are on the same scale and homoscedastic with each other. Thus, z_{g*} should be obtained after normalizing and transforming x_* based on the properties of training dataset.

Suppose that the training dataset is normalized using the deseq median ratio normalization method. Then the size factor of a test observation $\hat{s}_*$ will be calculated as follows:

$$m_* = median_g \left\{ \frac{x_{g*}}{\left(\prod_{i=1}^{n} x_{gi}\right)^{1/n}} \right\} \quad (3.12)$$

$$\hat{s}_* = \frac{m_*}{\sum_{i=1}^{n} m_i} \quad (3.13)$$

If we use TMM normalization method, then the reference sampled that is selected in training set, will be used for the calculations of test dataset. Let $X_{.*}$ the library size for the test observation. Then, we calculate TMM normalization factors as below:

$$log_2(TMM_*^r) = \frac{\sum_{g=1}^{p'} \varpi_{g*}^r M_{g*}^r}{\sum_{g=1}^{p'} \varpi_{g*}^r} \quad (3.14)$$

where $M_{g*}^r = \frac{log_2(x_{g*}/X_{.*})}{log_2(x_{gr}/X_{.r})}$ and $\varpi_{g*}^r = \frac{X_{.*} - x_{g*}}{X_{.*}\, x_{g*}} + \frac{X_{.r} - x_{gr}}{X_{.r}\, x_{gr}}$; $x_{g*}, x_{gr} > 0$.

In voom transformation, log-cpm values for x_* can be calculated as:

$$z_{g*} = log_2\left(\frac{x_{g*} + 0.5}{X_{.*} + 1} x10^6\right) \quad (3.15)$$

If a normalization (e.g. deseq median ratio, TMM, etc.) applied before, then $x'_{g*} = x_{g*}/\hat{s}_*$ is used instead of x_{g*} in the formula.

3.1.4. Sparse VoomNSC Classifier for RNA-Seq Classification

Similar to microarrays, RNA-Seq data is high-dimensional as well. Therefore, it is common to obtain very complex models with voomDLDA and voomDQDA classifiers. Here, we present voomNSC algorithm to overcome this complexity and obtain more simple, interpretable and reduced variance models. voomNSC is an extension of Tibshirani et al. (6) NSC algorithm with incorporating both log-cpm values and the associated weights together into the estimation of model parameters by using the weighted statistics. A flowchart displaying the steps of voomNSC algorithm is given in Figure 3.1.

Similar with NSC algorithm, voomNSC aims to identify the most significant gene subset for class prediction. Briefly, the standardized class specific weighted gene expression means are shrunken to the standardized overall gene expression weighted means, then the shrunken genes are eliminated and a voomDLDA classification model is built with the remaining genes. Mean expressions can be called as centroids as well. Let $d_{w^{gk}}$ the weighted difference scores, between weighted centroids of kth class and overall weighted centroids:

$$d_{w^{gk}} = \frac{\bar{z}_{w^{gk}} - \bar{z}_{w^g}}{m_k(s_{w^g} + s_{w^0})} \quad (3.16)$$

In the formula, m_k is a standard error adjustment term set as $\sqrt{1/n_k + 1/n}$. s_{w^0} is a small positive constant added to the denominator of Formula 3.16 to ensure

that the variance of the difference scores are independent from the gene expression level. s_{w^0} is calculated from the median value of s_{w^g} across genes.

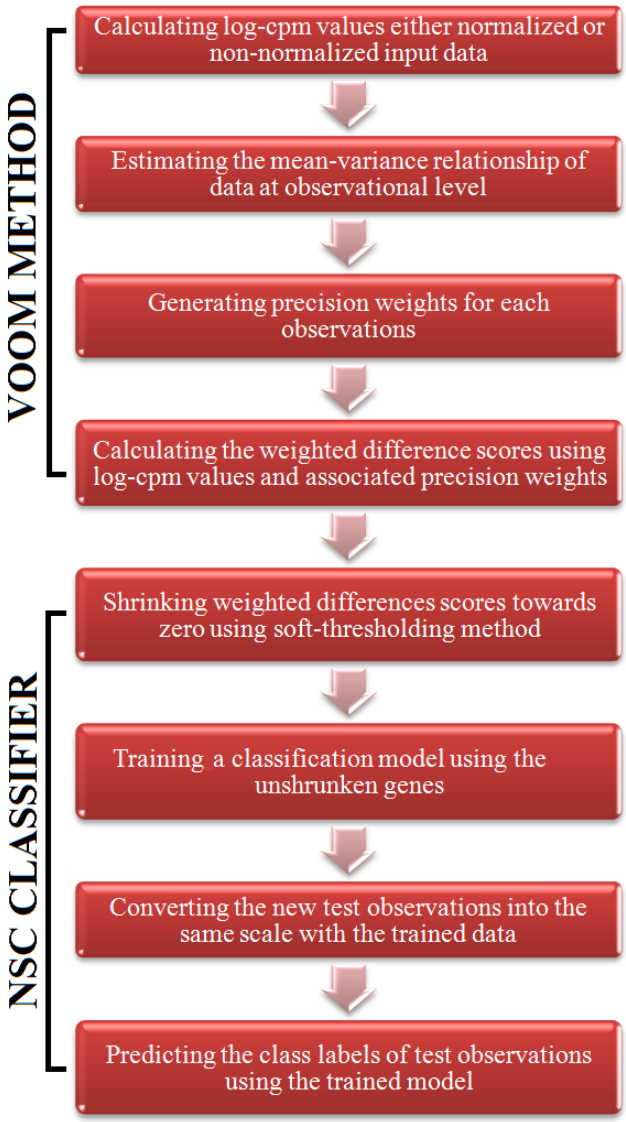

Figure 3.1. A flowchart of the steps of voomNSC algorithm

These weighted difference scores can be considered as the voom extension of the "relative differences" mentioned in (74). One can use these scores for the purpose of differential expression analysis with the significance analysis of microarrays (SAM) method. Formula 3.16 can be rewritten as:

$$\bar{z}_{w^{gk}} = \bar{z}_{w^g} + m_k(s_{w^g} + s_{w^0})d_{w^{gk}} \quad (3.17)$$

Next, each $d_{w^{gk}}$ is shrunken to zero using the soft-thresholding shrinkage method. Soft-thresholding is equivalent to lasso derivation and commonly used approach due to its reliable mean estimates. Using soft-thresholding with an amount of λ, threshold parameter, weighted shrunken differences can be obtained as follows:

$$d'_{w^{gk}} = sign(d_{w^{gk}})max(|d_{w^{gk}}| - \lambda, 0) \quad (3.18)$$

After shrinking $d_{w^{gk}} \rightarrow d'_{w^{gk}}$, we update the weighted centroids as follows:

$$\bar{z}'_{w^{gk}} = \bar{z}_{w^{g}} + m_k(s_{w^{g}} + s_{w^{0}})d'_{w^{gk}} \quad (3.19)$$

Increasing λ will lead to obtain more sparse models by eliminating most of the genes from the class prediction. Suppose that $d'_{w^{gk}}$ is zero for a gene g, for all classes, then the weighted centroids will be same across the classes. In this way, this gene will not contribute to the class prediction.

3.1.5. Selection of the Optimal Threshold Parameter (λ)

Selection of λ is very important on the model sparsity. Increasing λ will lead to obtain sparser models, but may give inaccurate results. Small values of λ may give more accurate results, but may provide very complex models. Thus, it is necessary to select λ that yields to both accurate and sparse results. Figure 3.2 displays the test set errors for a set of λ parameters for cervical dataset of (58). It is seen that we obtain the minimum misclassification errors for the values of $\lambda = \{0.561, 0.654, 0.748, 1.028, 1.121, 1.215, 1.308, 1.402, 1.495, 1.588, 1.682, 1.775\}$.

Among these values, selecting the maximum one will give us the sparsest solution. For this reason, we select 1.775 and obtain 96.5% accuracy with using only 16 features. One can also use cross-validation technique and select the sparsest model that minimizes cross-validation error.

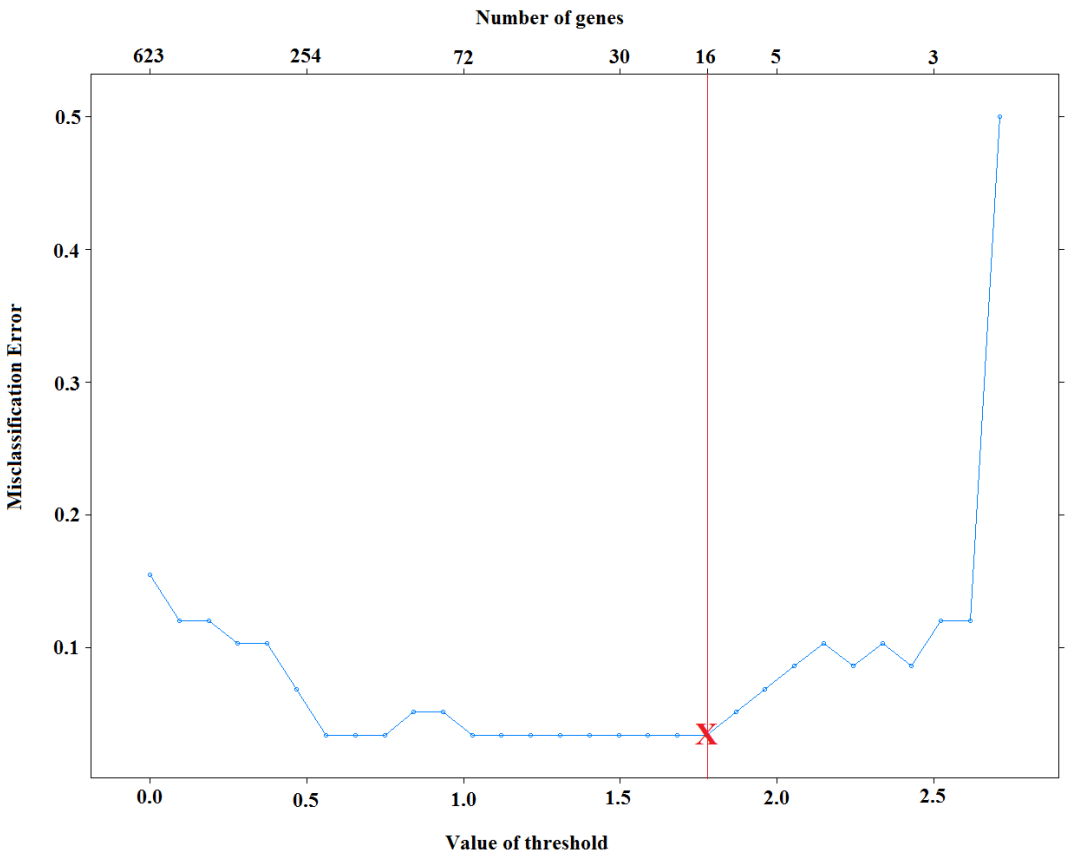

Figure 3.2. Selection of voomNSC threshold parameter for cervical data

3.1.6. Prediction of Test Observations for VoomNSC Classifier

Test observations are normalized and transformed based on the training set parameters which are detailed in Section 3.1.3. Again, a standardization is applied to the z_{g*}, log-cpm values of test observations, by $s_{w^g} + s_{w^0}$. We classify a test observation to the class that maximizes the following discriminating function:

$$\delta_k^{voomNSC}(x_*) = -\frac{1}{2}\sum_{g=1}^{p}\frac{(z_{g*} - \bar{z}_{w^{gk}})^2}{(s_{w^g} + s_{w^0})^2} + \log(\hat{\pi}_k) \quad (3.20)$$

Posterior probabilities can be obtained from (2.18).

3.2. Implementation of Classifiers

To assess the performance of developed algorithms, we compared our results with several classifiers. In this section, we give implementation details of the used classifiers.

Firstly, we selected discrete RNA-Seq classifiers (i.e. PLDA and NBLDA), since they are the only algorithms proposed for RNA-Seq classification. We also applied the diagonal discriminant classifiers, after transforming the data hierarchically closer to microarrays. SVM and RF algorithms are also considered due to their accurate performances in microarray based classification studies. Implementation of each algorithm including voomDDA classifiers are given below:

$PLDA_1$: The data is normalized using deseq median ratio method. Normalized count values are taken as input to PLDA algorithm. Five-fold cross validation is performed to identify the optimal ρ tuning parameter. A grid search (number of search: 30) is applied and the sparsest model with the minimum misclassification error is selected to optimize ρ. PLDA is applied with the optimum ρ in PoiClaClu R package (75).

$PLDA_2$: After normalization, a power transformation ($X_{ij}' = \sqrt{X_{ij} + 3/8}$) is applied to reduce the effect of overdispersion and make genes have nearly constant variance (76). Normalized and transformed expression values are the input to PLDA algorithm. Other procedures are same as $PLDA_1$.

NBLDA: Deseq median ratio method is used for normalization. Yu et al. (21) shrinkage method is applied for estimation of dispersion parameter. Normalized counts are used as input to NBLDA algorithm. NBLDA is applied in R software with the necessary codes available in (5).

NSC: Deseq median ratio method is used for data normalization and rlog transformation is applied to normalized count data. Normalized and transformed expression values are used as input data for NSC algorithm. Proportions of class sample sizes are used as class prior probabilities. Five-fold cross validation is used to determine the optimal threshold value. Optimum threshold value is obtained from the sparsest model with the minimum misclassification error after a grid search (number of search: 30). NSC is applied in pamr package of R (77).

DLDA: Deseq median ratio method is applied for data normalization and rlog transformation is applied to normalized count data. Normalized and transformed

expression values are used as input data for DLDA algorithm. Proportions of class sample sizes are used as class prior probabilities. Then DLDA is applied in sfsmisc package of R (78).

DQDA: Same procedure is applied with DLDA algorithm (78).

SVM: Deseq median ratio method is used for data normalization and rlog transformation is applied to normalized count data. Normalized and transformed expression values are used as input data for SVM algorithm. Five-fold cross validation is performed, repeated for three times and a grid search (with tune length of 10) is made to identify the optimal sigma and cost parameters. Radial basis function was used to allow SVM for nonlinear classification. SVM is applied in caret package of R (79).

RF: The applied procedure is similar with SVM. Here, the optimized parameter is the number of variables randomly sampled as candidates at each split. Number of trees are set as 500. RF is applied in caret package of R (79).

voomNSC$_1$: Deseq median ratio normalization is applied to data and the normalized data is used as input to voomNSC classifier. Proportions of class sample sizes are used as class prior probabilities. To optimize the threshold value, the sparsest model with the minimum misclassification error is selected. A grid search (number of search: 30) is applied to determine the optimal threshold value.

voomNSC$_2$: Raw read counts are directly used as input for voomNSC algorithm. All other procedures are remained to be same with voomNSC$_1$.

voomNSC$_3$: TMM method is applied to normalize the data. Normalized data is used as input to voomNSC classifier. Other procedures are same with voomNSC$_1$ and voomNSC$_2$.

voomDLDA$_1$: Deseq median ratio normalization is applied to data and the normalized data is used as input to voomDLDA classifier. Proportions of class sample sizes are used as class prior probabilities.

voomDLDA$_2$: Raw count data is not normalized and directly used as input to voomDLDA classifier. Other procedures are same with voomDLDA$_1$.

voomDLDA$_3$: TMM method is used for normalization. Other procedures are same with voomDLDA$_1$ and voomDLDA$_2$.

voomDQDA$_1$: Deseq median ratio normalization is applied to data and the normalized data is used as input to voomDQDA classifier. Proportions of class sample sizes are used as class prior probabilities.

voomDQDA$_2$: Raw count data is not normalized and directly used as input to voomDQDA classifier. Other procedures are same with voomDQDA$_1$.

voomDQDA$_3$: TMM method is performed for normalization. Other procedures are same with voomDQDA$_1$ and voomDQDA$_2$.

All necessary codes for voomDDA classifiers are available as supplementary material to this thesis.

3.3. Evaluation of voomDDA Classifiers

To evaluate the performance of the developed algorithms, we performed a comprehensive simulation study. Four real datasets were also used to illustrate the applicability of voomDDA classifiers and assess their performance in real experiments.

3.3.1. Simulation Study

Simulation Setup

We simulated data (pxn dimensional matrix) under 648 scenarios using negative binomial distribution as follows:

$$x_{gi}|y_i = k \sim NB(\mu_{gi} e_{gk}, \phi_g) \quad g = 1, \dots, p; \;\; i = 1, \dots, n; \;\; k = 1, \dots, K \qquad (3.21)$$

where NB corresponds to negative binomial distribution, μ_{gi} corresponds to $g_g s_i$, e_{gk} is the differential expression probability for each of the $p = 10{,}000$ genes among classes, and ϕ_g is the dispersion parameter. For a given $y_i = k$, x_{gi} has mean $\mu_{gi} e_{gk}$ and variance $(\mu_{gi} e_{gk})^2 \phi_g$. s_i is the size factor for each sample and simulated identically and independently from $s_i \sim Unif(0.2, 2.2)$. g_g refers to the total number of counts per gene and also simulated identically and independently from $g_g \sim Exp(1/25)$. If a gene is not differentially expressed among classes k, then e_{gk} is set to 1. Otherwise, $\log(e_{gk}) = \tilde{z}_{gk}$, where $\tilde{z}_{gk}$'s are identically and independently

distributed from $\tilde{z}_{gk} \sim N(0, \sigma^2)$. σ is set to 0.10 or 0.20 in simulations. Of the total $p = 10{,}000$ genes, 500, 1,000 and 2,000 genes with maximum variances are selected. We added a small constant ($\varepsilon = 1$) to count values of each simulated data to avoid taking the logs of zero in following analysis.

The simulated datasets contain all possible combinations of:

- number of genes; $p' = (500, 1000, 2000)$,
- number of biological samples; $n = (40, 60, 80, 100)$,
- number of classes; $K = (2, 3, 4)$,
- probability of differential expression: $e_{gk} = (1\%, 5\%, 10\%)$,
- standard deviation parameter: $\sigma = (0.1, 0.2)$
- dispersion parameter; (ϕ_g=0.01: very slight, ϕ_g=0.1: substantial; ϕ_g=1, very high overdispersion).

Simulation codes are obtained from the *CountDataSet* function of the PoiClaClu R package (75) and manipulated based on the simulation details given above. Seed number is set to a constant of '10072013' in all analysis steps.

Evaluation Process

After simulating the datasets, the following steps are applied in order. A flow chart is provided for the reader to better understand the evaluation process (Figure 3.3).

Data splitting: The data are randomly split into training and test sets with 70% and 30%, respectively. The feature data can be denoted as $\mathbf{X}_{tr}$ and $\mathbf{X}_{ts}$, where the class labels can be denoted as $\mathbf{y}_{tr}$ and $\mathbf{y}_{ts}$.

Near-zero filtering: Since the genes with low counts can affect the further analysis (e.g. linear modeling inside voom transformation), genes having near zero variances are detected in the training dataset and filtered in this step. For this purpose, two criteria are used for filtering: (i) the frequency ratio of the most frequent value to most frequent second value is higher than 19 (95/5), (ii) the number of unique values divided by the sample size is less than 10%. Selected genes with near zero variances are both filtered from training and test datasets.

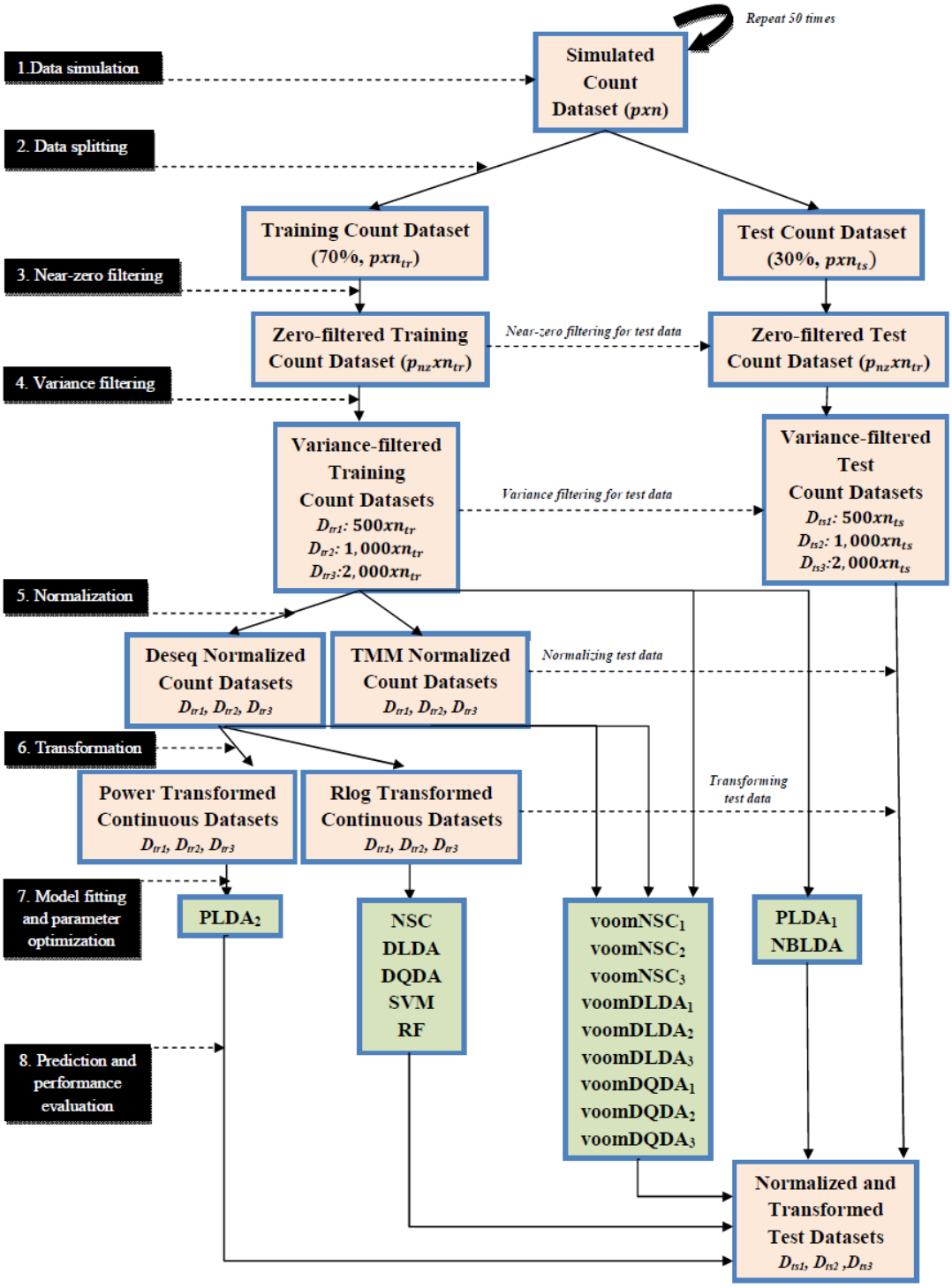

Figure 3.3. Simulation design and the evaluation process

Variance filtering: Next, a second filtering is applied to keep only the informative genes in the model. In training dataset, 500, 1,000 and 2,000 genes with maximum

variances are selected and other genes are filtered from both training and test datasets. In this selection, data are normalized using deseq median ratio method, transformed using vst transformation and genes are sorted in decreasing order based on their variances. The count values of the selected genes were fetched again for further analysis.

Normalization: After filtering steps, the datasets are normalized to adjust the sample specific differences using deseq median ratio method or TMM method depending on the used classification methods. The datasets are not normalized for voomNSC$_3$, voomDLDA$_3$ and voomDQDA$_3$ classifiers. Note that, the normalization of test datasets are made based on the information obtained from the training datasets. Since, training and test sets should be in the same scale and be homoscedastic relative to each other. Therefore, the test samples should each be independently normalized using the same parameters calculated from the training set. This procedure is explained in detail in Section 3.1.3.

Transformation: After normalization, several transformations are applied to data to estimate the mean and variance relationship of the data and convert it from discrete to continuous format. The transformations are not used for PLDA$_1$, NBLDA and classifiers. VoomDDA classifiers use voom method inside the algorithm for transformation. A power transformation is applied for PLDA$_2$ classifier. Rlog transformation is performed for other classifiers, due to its capability of accounting for variations in sequencing depth across samples (23). Similar with the normalization, test sets are transformed based on the mean and variance relationship (of genes or samples) properties of the training sets. Thus, we do not re-estimate the mean-variance relationship of the data, but use the same β_g coefficients same with the training set.

Model fitting and parameter optimization: In order to avoid overfitting[18] and underfitting[19], we optimized the tuning parameters of classifiers before model fitting. Five-fold cross validation approach is used on the training set and the parameter that gives the minimum misclassification error is identified as optimal parameter. Same folds are used in all classifiers to make the results comparable. In case of equal misclassification errors, best parameter is chosen based on its sparsity. Next, classification models are fit on $\mathbf{X}_{tr}$ and $\mathbf{y}_{tr}$ with the optimal tuning parameters.

---

[18]The problem that arises when a statistical model captures the noise of the data.

[19]The problem that arises when a statistical model could not capture the actual trend of data.

Prediction and performance evaluation: Using the built classification models, we used $\mathbf{X}_{tr}$ to predict $\mathbf{y}_{tr}$, calculated the misclassification error of each model. Number of genes used in each model is also saved in order to assess sparsity.

Since we mimic the real datasets, sample sizes are set to be very small relative to the number of genes. Thus, the misclassification errors may be highly variable depending on the split of samples into training and test sets. To overcome this problem, all the entire simulation procedure was repeated 50 times and the summaries are given in the Results section.

3.3.2. Application to Real RNA-Sequencing Datasets

Experimental datasets

Cervical dataset: Cervical dataset is a miRNA sequencing dataset obtained from (58). miRNAs are non-coding small RNA molecules with average 21-23 bp length and take role in the regulation of gene expression. The objective of this study was to both identify the novel miRNAs and to detect the differentially expressed ones between normal and tumor cervical tissue samples. For this purpose, the authors constructed 58 small RNA libraries, prepared from 29 cervical cancer and 29 matched control tissues. After deep sequencing with Solexa/Illumina sequencing platform, they obtained a total of 25 Mb and 17 Mb RNA sequences from the normal and cancer libraries respectively. Of these 29 tumor samples, 21 of them had a diagnosis of squamous cell carcinomas, 6 of them had adenocarcinomas and 2 were unclassified. In our analysis, we used the data that contains the sequence read counts of 714 miRNAs belonging to 58 human cervical tissue samples, where 29 tumor and 29 non-tumor samples are treated as two distinct classes for prediction.

Alzheimer dataset: This dataset is another miRNA dataset provided from Leidinger et al. (80). The authors aimed to discover potential miRNAs from blood in diagnosing Alzheimer and related neurological diseases. In this purpose, the authors obtained blood samples from 48 Alzheimer patients that were evaluated after undergoing some tests including Alzheimer Disease Assessment Scale-cognitive subscale (ADAS-Cog), Wechsler Memory Scale (WMS), and Mini-Mental State Exam (MMSE) and Clinical Dementia Rating (CDR). A total of 22 age-matched control samples were obtained and all sample libraries were sequenced using

Illumina HiSeq2000 platform. After obtaining the raw read counts, the authors filtered the miRNAs with less than 50 counts in each group. We used the data including 416 read counts of 70 samples, where 48 Alzheimer and 22 control samples are considered as two separate classes for classification.

Renal cell cancer dataset: Renal cell cancer (RCC) dataset is an RNA-Seq dataset that is obtained from The Cancer Genome Atlas (TCGA) (81). TCGA is a comprehensive community resource platform for researchers to explore, download, and analyze datasets. We downloaded this dataset (with options level 3, RNASeqV2 data) from this database and obtained the raw 20,531 known human RNA transcript counts belonging to 1,020 RCC samples. This RNA-Seq data has 606, 323 and 91 specimens from kidney renal papillary cell (KIRP), kidney renal clear cell (KIRC) and kidney chromophobe carcinomas (KICH), respectively. These three classes are referred as the most common subtypes of RCC (account for nearly 90%-95% of the total malignant kidney tumors in adults) and treated as three separate classes in our analysis (82).

Lung cancer dataset: Lung cancer is another RNA-Seq dataset provided from TCGA platform. Same options were used in the download process. The resulting count file contains the read counts of 20,531 transcripts of 1,128 samples. The dataset has two distinct classes including lung adenocarcinoma (LUAD) and lung squamous cell with carcinoma (LUSC) with 576 and 552 class sizes, respectively. These two classes are used as class labels in our analysis.

Evaluation Process

A similar procedure is followed with the simulation study. The data are randomly split into two parts as training (70%) and test (30%) sets. Near zero filtering is applied to all datasets except Alzheimer, since low counts were already filtered by the authors of the study (80). Next, 2,000 transcripts with the highest variances are selected in each renal cell cancer and lung datasets. Appropriate normalization, transformation and model fitting processes are applied same with the simulation study. In prediction step misclassification errors for Alzheimer and renal cell cancer datasets are balanced due to the unbalanced class sizes.

We repeated the whole process 50 times, since cervical and Alzheimer datasets have relatively small sample size. Test set errors may differ for different train/test splits. Seed number is set between 1 to 50 the analysis steps. In results section, summary statistics are given across these 50 repeats.

3.3.3. Evaluation Criteria

To assess the performance of classifiers, we used three criteria: (i) sparsity, (ii) accuracy and (iii) computational cost. We simply assessed the sparsity of each model by calculating the sparsity, number of selected genes in each model, or relative sparsity, which is the ratio of the number of genes selected in each classification model over total number of genes. A model which uses lower number of genes in the decision rule, thus have lower relative sparsity is considered as sparser model. To measure accuracy, we calculated misclassification errors of each model in the test set. Due to the high-dimension of the RNA-Seq data, it is possible to meet with the overdispersion problem, that a classification method may perfectly classify the training data, but may not perform well in test data. Since we are mostly interested in predicting the class labels of new observations accurately in real life problems, we randomly split each dataset into training and test sets. All model building processes are applied in training set and the performance assessment is made in test sets. Misclassification errors are calculated from the confusion matrices of each prediction. A model with less misclassification error is considered as more accurate model. A confusion matrix is given in Table 3.1 and the formula of misclassification error is given below:

Table 3.1. Confusion matrix for a classification model

Prediction of a classifier ($\hat{\boldsymbol{y}}_{ts}$)	Actual labels ($\boldsymbol{y}_{ts}$)		Total
	Positive	Negative	
Positive	TP	FP	TP+FP
Negative	FN	TN	FN+TN
Total	TP+FN	TP+FN	n_{ts}

TP: True positive, FP: False positive, TN: True negative, FN: False negative, n_{ts}: sample size of the test set

$$Misclassification\ error\ = \frac{(FP\ +\ FN)}{n_{ts}} \qquad (3.22)$$

In case of unbalanced class sizes, misclassification error may be leading to measure the actual accuracy. Suppose that we are working with Alzheimer dataset and a classifier predicted all the 21 test class labels as Alzheimer. If we use misclassification error as criteria, the error will be calculated as 0.33 disregarding the variability in class sizes. In both Alzheimer and renal cell cancer datasets, class sizes are unbalanced. Here, we used balanced misclassification error as evaluation criteria:

$$\text{Balanced misclassification error} = 1 - \frac{\frac{TP}{TP + FN} + \frac{TN}{FP + TN}}{2} \qquad (3.23)$$

For multiclass problems, both measures are calculated by one-versus-all[20] method by comparing each class label to the remaining labels. Finally, computational costs are computed from the process time intervals of each classifier during the model building process.

3.3.4. Computational Infrastructure and Parallel Programming

We used R programming language and RStudio software (Version 0.98.1103) in order to develop the algorithms and apply the experiments. R is a free and open source software for statistical computing available at (83). RStudio is an integrated development environment developed to execute R codes easier as well as providing tools for plotting, debugging history and workspace management (84).

To reduce the computational cost of the entire experimental processes, we arranged the simulation codes into a special form to be able to work in multiple platforms. We split the scenarios into several workstations, run the analysis in these computers, saved the results and finally assembled together. The computational properties of each workstation are given in Table 3.2:

[20]Assuming one class as positive and the remaining classes as negative to calculate binary performance measures in multiclass classification problems.

Tablo 3.2. Computational properties of the used workstations in analysis

Workstation	**Operating system**	**CPU**	**Memory**	**Number of cores**
Hacettepe University, Department of Biostatistics	Windows 7	Core i7 3960X, 3.30 GHz	64 GB	12
Personal Computer	Ubuntu 14.04 LTS	Core i7 4770, 3.40 GHz	16 GB	8
Erciyes University, Department of Biostatistics	OS X Yosemite 10.10.2	Core i7 Quad Core, 4 GHz	32 GB	8
Erciyes University, Genome and Stem Cell Center, Division of Bioinformatics	Windows 8	Xeon E5-1650, 3.20 GHz	64 GB	12
Erciyes University, Genome and Stem Cell Center, Division of Bioinformatics	Ubuntu 14.10	Xeon E5-1650, 3.20 GHz	16 GB	12
Marmara University, Department of Physics	Windows 7	Core i7 3930K, 3.20 GHz	16 GB	8
University of California, San Diego Supercomputer Center	OS X Yosemite 10.10.2	Xeon Quad Core, 2x2.66 GHz	16 GB	8

We also used parallel programming to carry out the computations simultaneously. For this reason, we used the R packages doSNOW (85), doParallel (86), doMC (87), foreach (88) and digest (89). In this way, we obtained all experimental results in less than two weeks.

3.4. Development of a Web-Based Platform

To provide the developed algorithms applicable for researchers, we benefited from the R shiny package (90). Shiny allows users to build interactive web applications with R software. In order to build the voomNSC tool, two scripts are constructed including the user-interface script and the server script. User-

interface script is used to build the design of the interface and to control the layout, while server script contains all necessary instruction codes, which makes the tool available. After developing voomNSC, both scripts are embedded into a web-server to provide the applicability of voomNSC on web. This server is located at Hacettepe University, Faculty of Medicine, Department of Biostatistics and used for the BIOSOFT Project. This project is dedicated to develop free, up-to-date and user-friendly web tools in various scientific areas using the R language environment. Users can freely access to our application from the project website: http://www.biosoft.hacettepe.edu.tr/voomDDA/ .

These two scripts are provided in Supplementary Material 3. Users can execute the software locally in their personal machines from these files using the following R codes:

```
> library(shiny)
> runApp("voomDDA")
```

4. RESULTS

4.1. Simulation Results

Simulation results for each scenario are given in Figure 4.1 - 4.36. These figures differ with different combinations of number of classes (K), probability of differential expression (e_{gk}) and standard deviation (σ). Odd numbered figures give the accuracy results, while the even numbered figures give the sparsity results. Note that the sparsity results are only given for sparse classifiers (i.e. NSC, $PLDA_1$, $PLDA_2$, $voomNSC_1$, $voomNSC_2$ and $voomNSC_3$). All figures are given in same format in same matrix layout. Each figure displays the effect of sample size (n), number of genes (p'), dispersion parameter (ϕ_g) on the accuracy and sparsity of classification models. Axis panels give the results for sample size, ordinate panels give the results for dispersion parameter. Each panel demonstrates the error bars for each classifier on classification performance. Classifiers are displayed in axes, evaluation measures are displayed in ordinates in each panel. Each measure is in the range [0,1], with lower values corresponding to more accurate or sparser models. Error bars are generated from the arithmetic mean and 95% confidence levels of each performance measure in 50 repeats. Black, red and green bars correspond to the results for 500, 1,000 and 2,000 genes, respectively.

As can be seen from the figures, an increase in the sample size leads to an increase in the overall accuracies, unless the data is overdispersed. This relation is more distinct for very slightly overdispersed scenarios. However, this increase does not affect the amount of sparsity. Number of genes has considerable effect on both accuracy and sparsity. Including more genes into classification models mostly leads to more accurate results for PLDA ($PLDA_1$, $PLDA_2$) and voomNSC ($voomNSC_1$, $voomNSC_2$, $voomNSC_3$) classifiers, unless the data is overdispersed. Increasing the number of genes mostly provides less accurate results for other classifiers. However, this relation may change in some scenarios depending on the increase of sample size and standard deviation. VoomNSC and PLDA classifiers mostly produce sparser results depending on the increase in the number of genes. This situation is quite opposite for the NSC algorithm in most scenarios.

The change in dispersion parameter has direct effect on both model accuracies and sparsities. When the data become more spread, all methods have

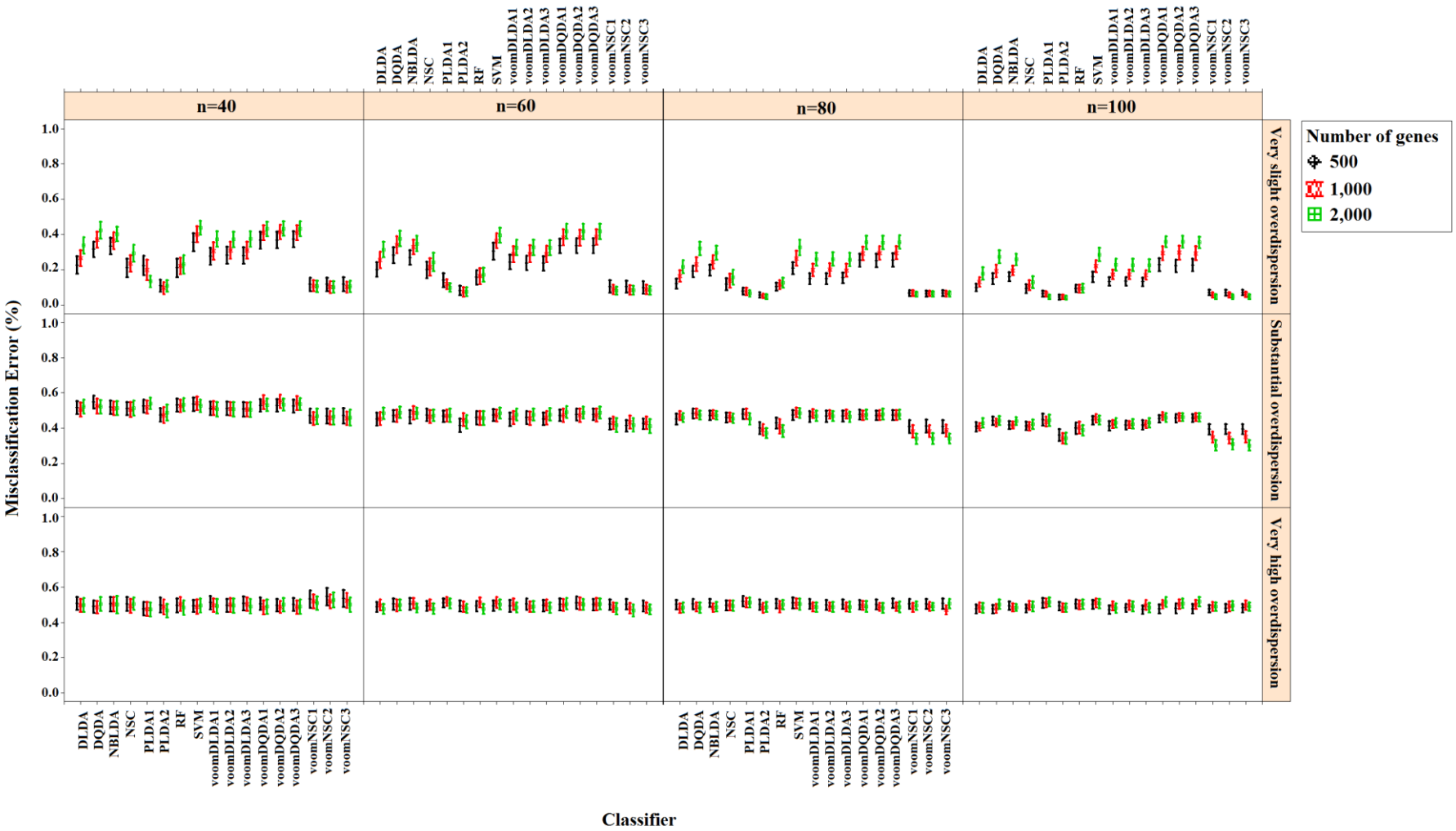

Figure 4.1. Accuracy results for the simulation scenario K=2, e_{gk}=1%, σ=0.1

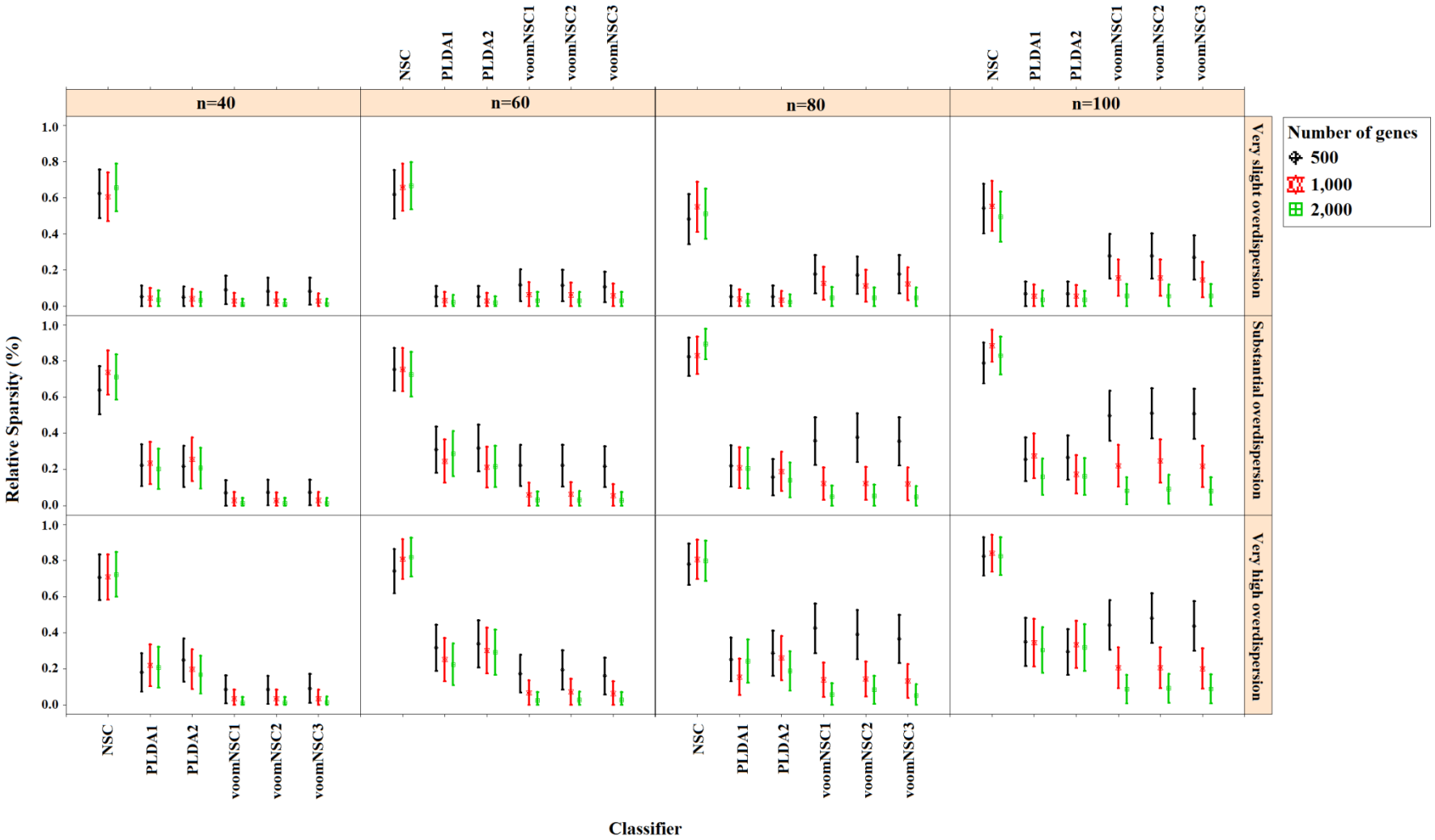

Figure 4.2. Sparsity results for the simulation scenario K=2, e_{gk}=1%, σ=0.1

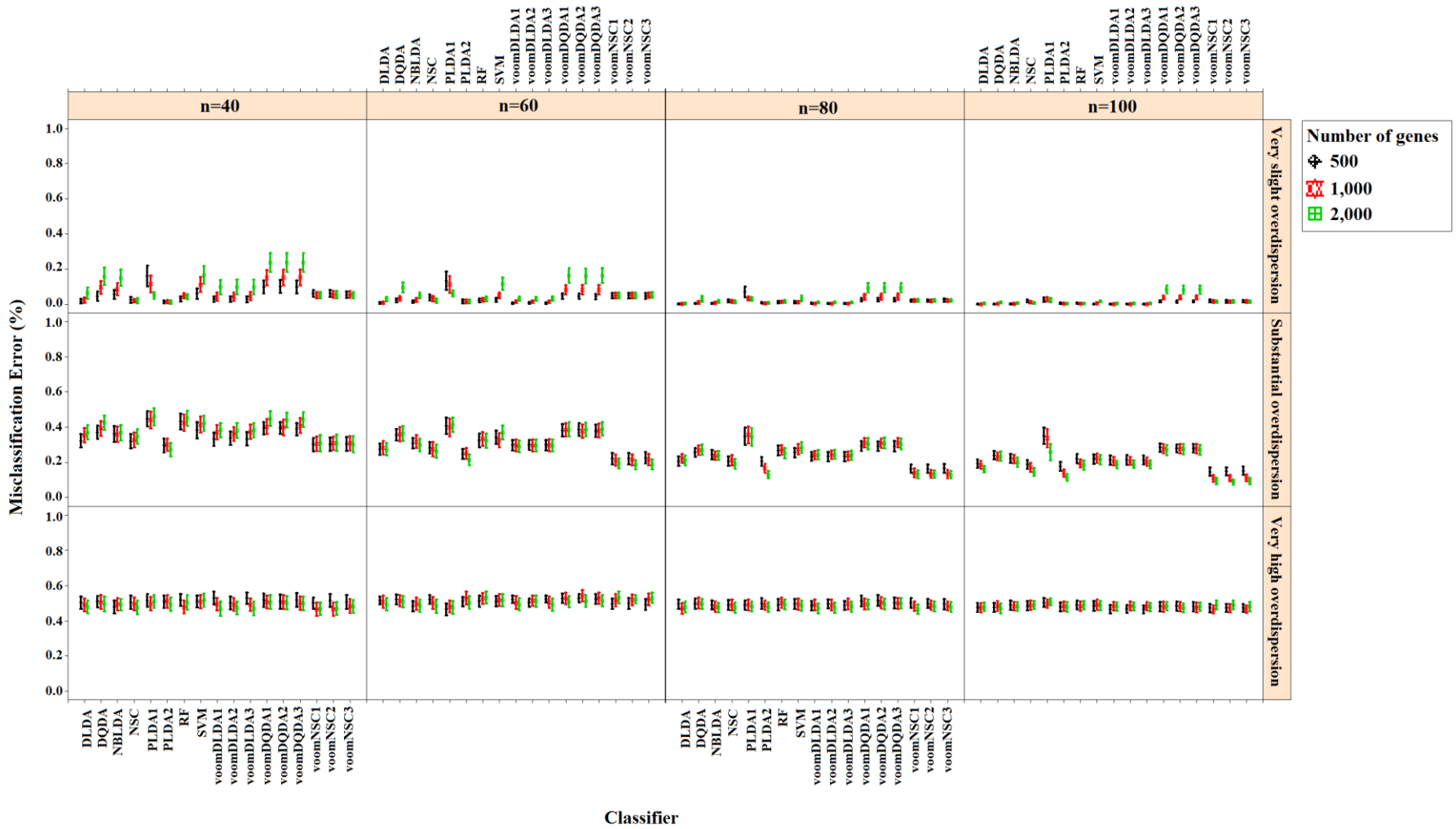

Figure 4.3. Accuracy results for the simulation scenario K=2, e_{gk}=5%, σ=0.1

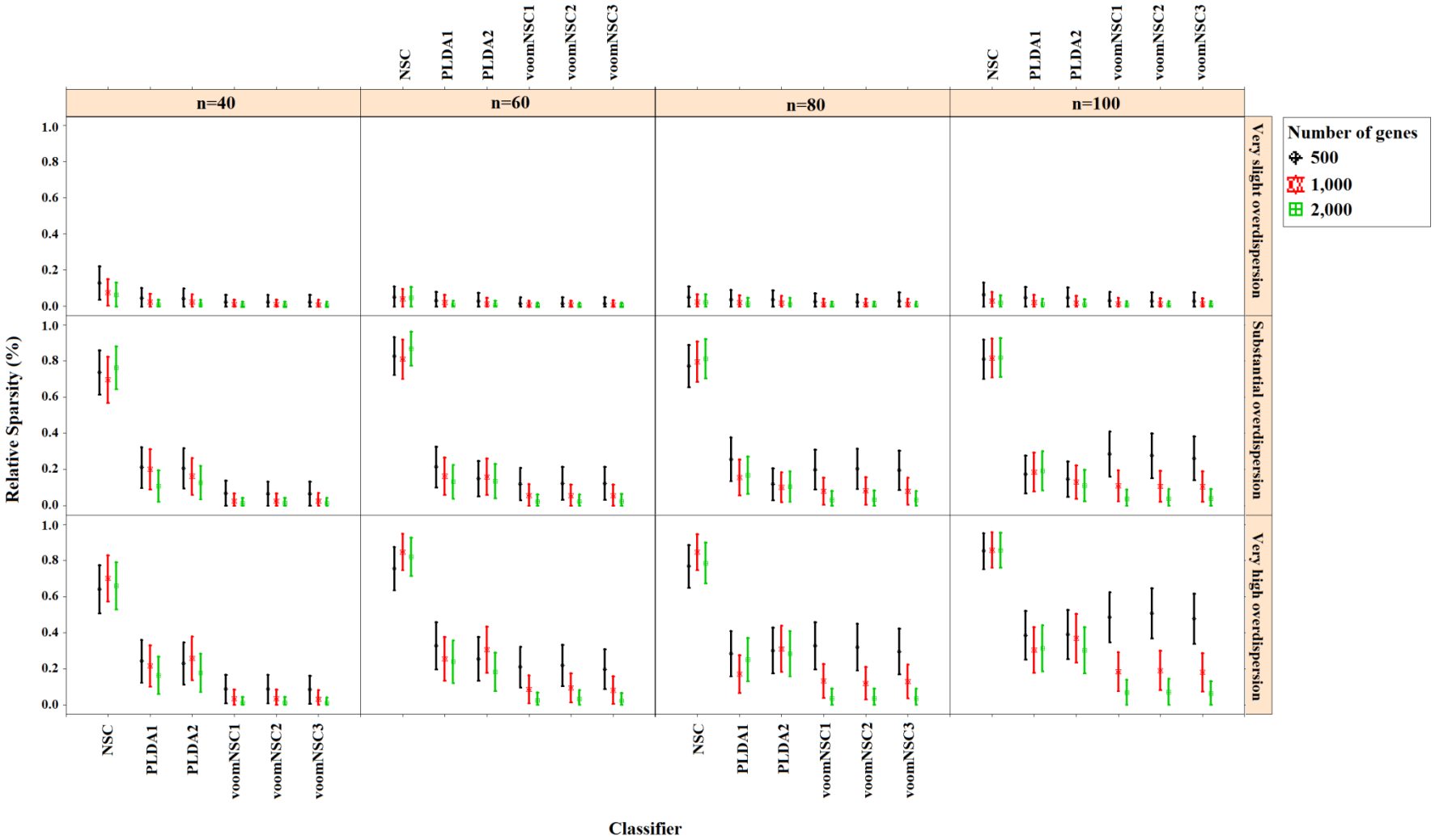

Figure 4.4. Sparsity results for the simulation scenario K=2, e_{gk}=5%, σ=0.1

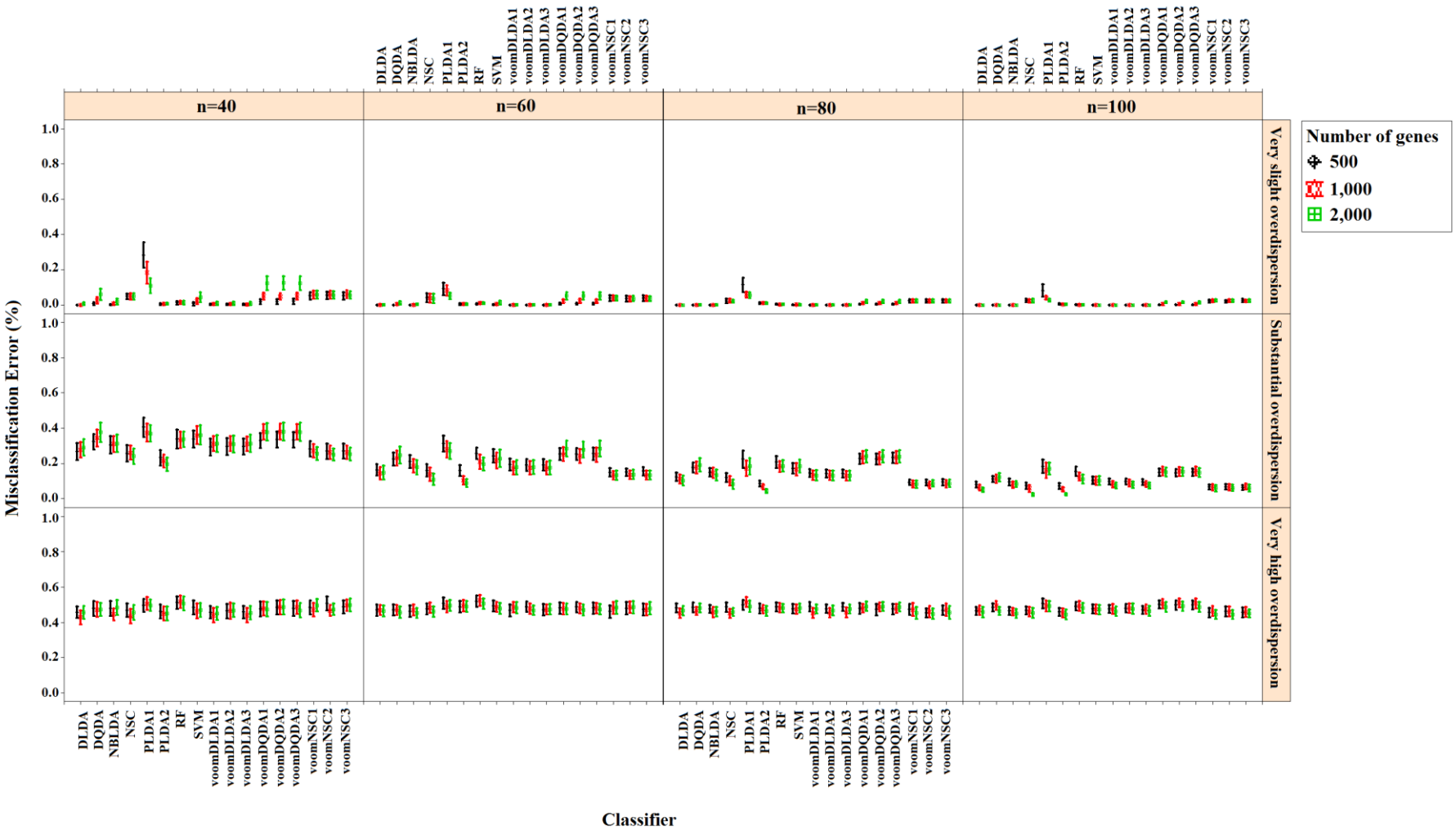

Figure 4.5. Accuracy results for the simulation scenario K=2, e_{gk}=10%, σ=0.1

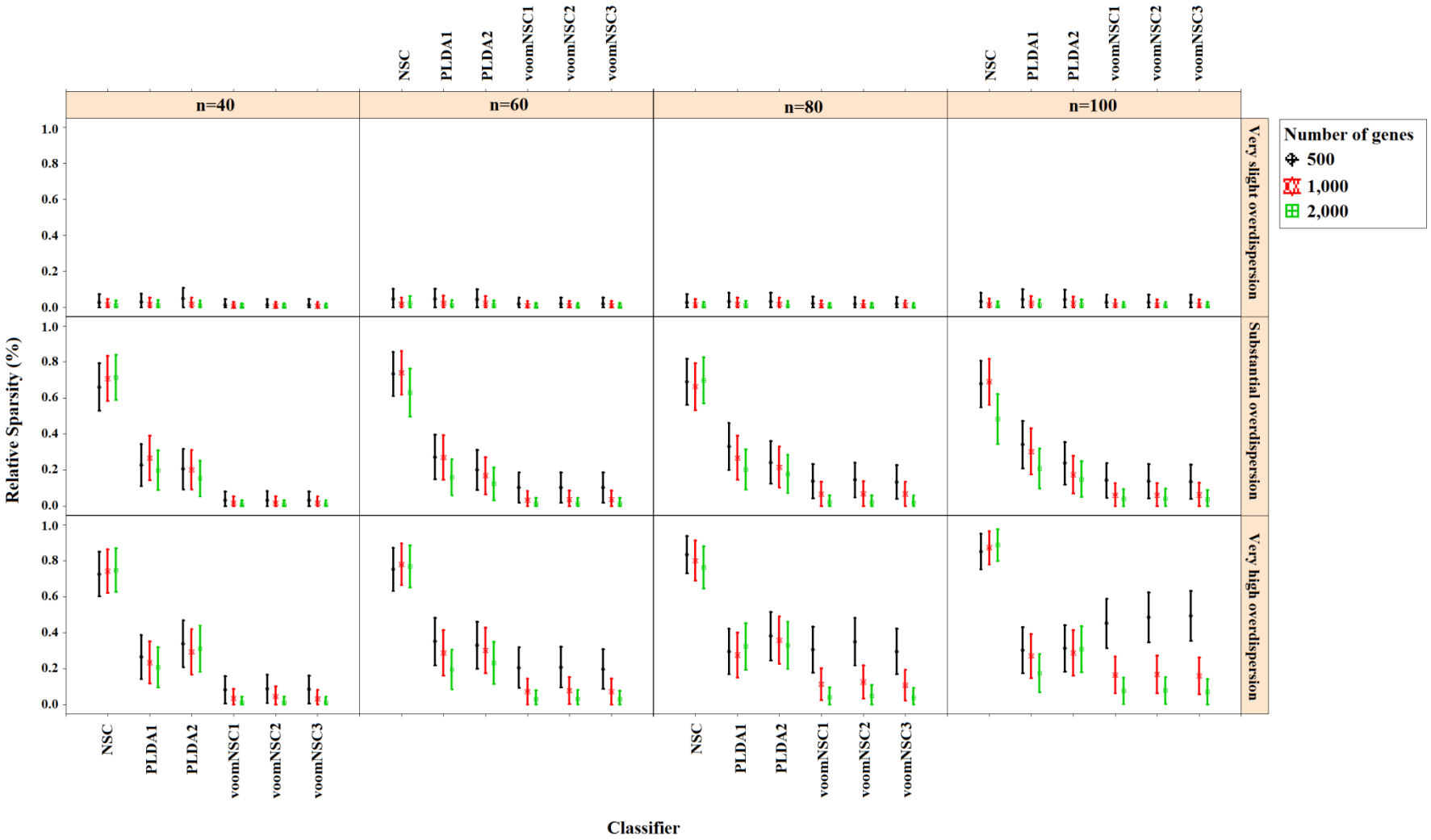

Figure 4.6. Sparsity results for the simulation scenario K=2, e_{gk}=10%, σ=0.1

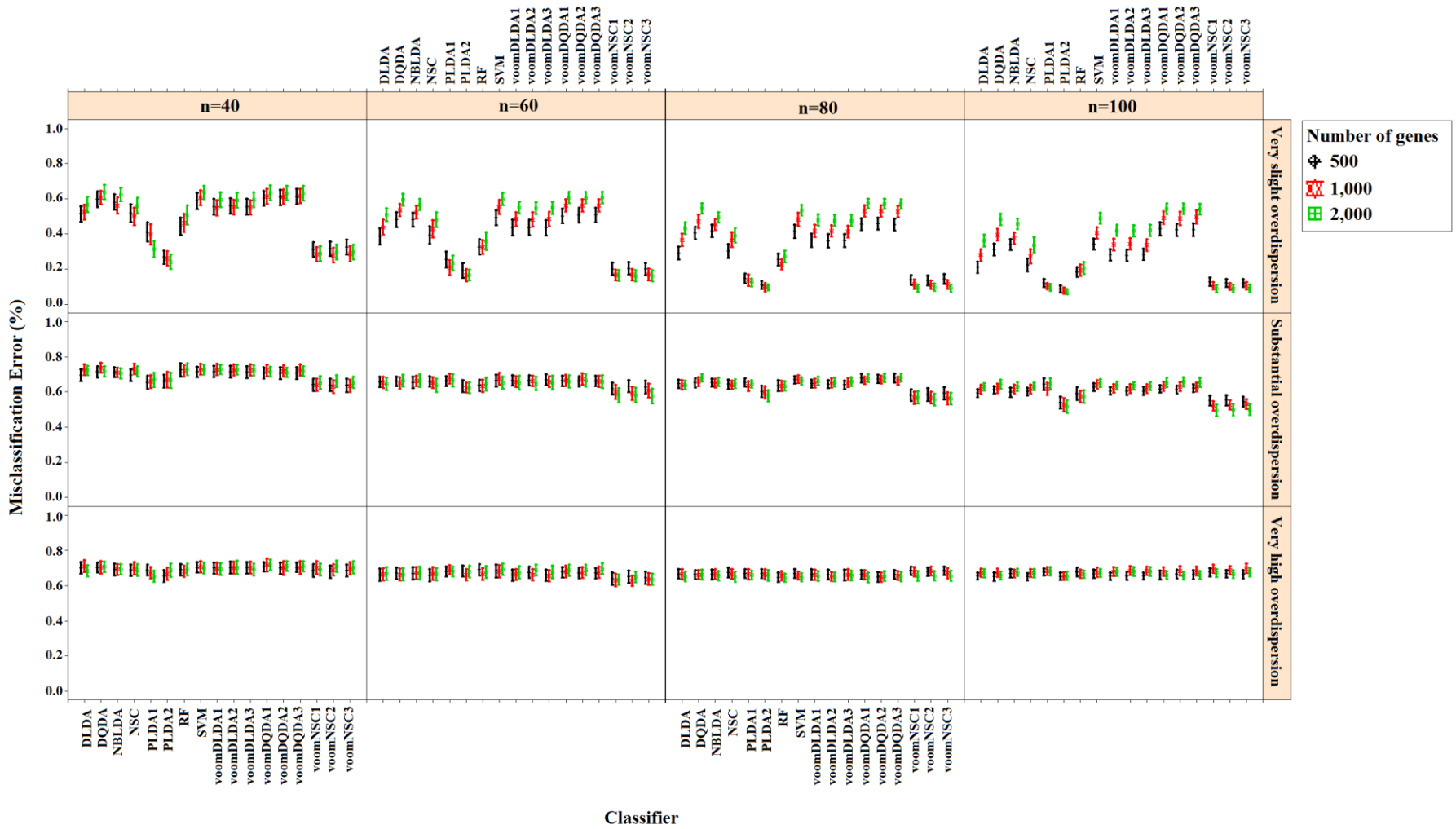

Figure 4.7. Accuracy results for the simulation scenario K=3, e_{gk}=1%, σ=0.1

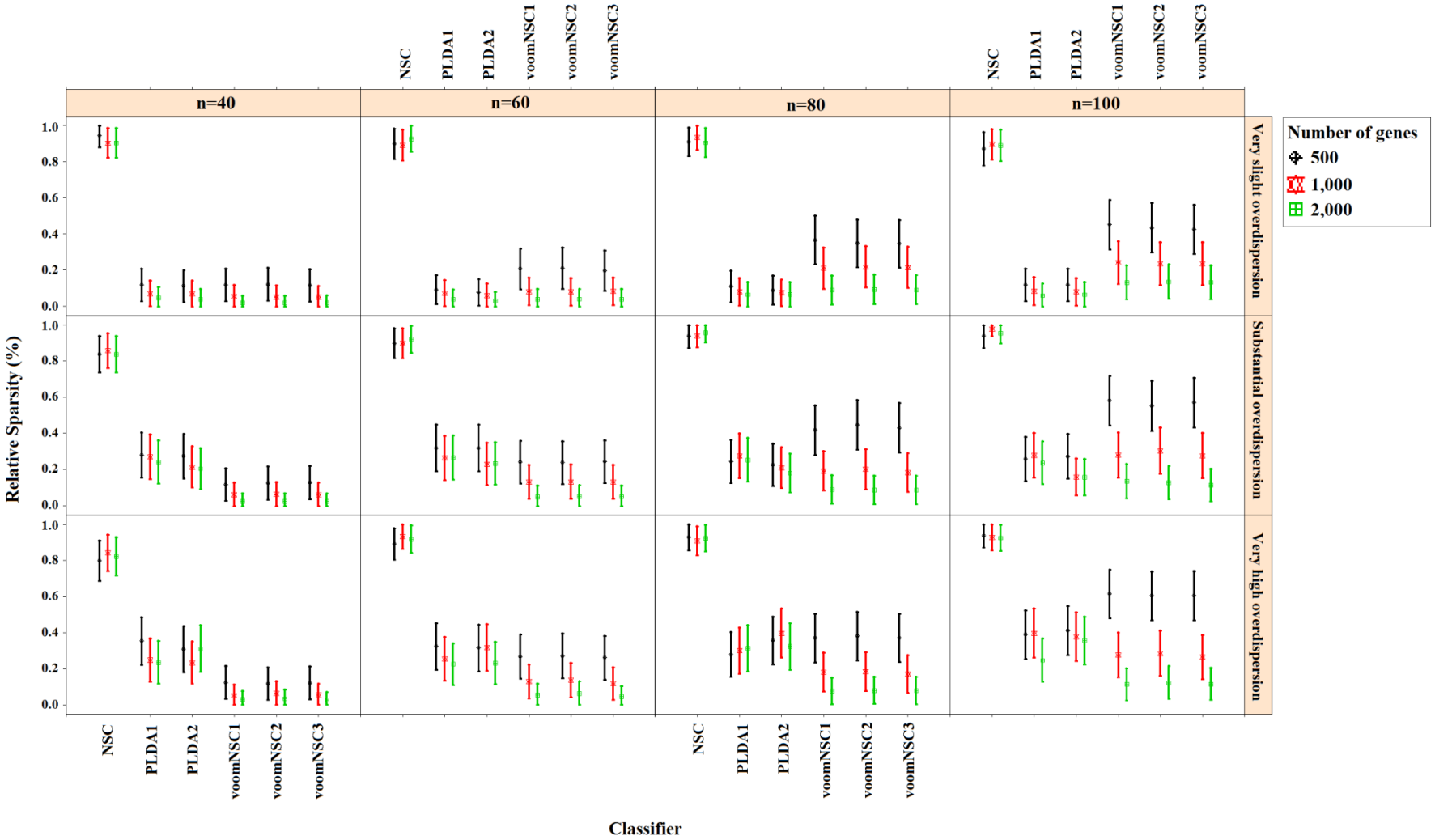

Figure 4.8. Sparsity results for the simulation scenario K=3, e_{gk}=1%, σ=0.1

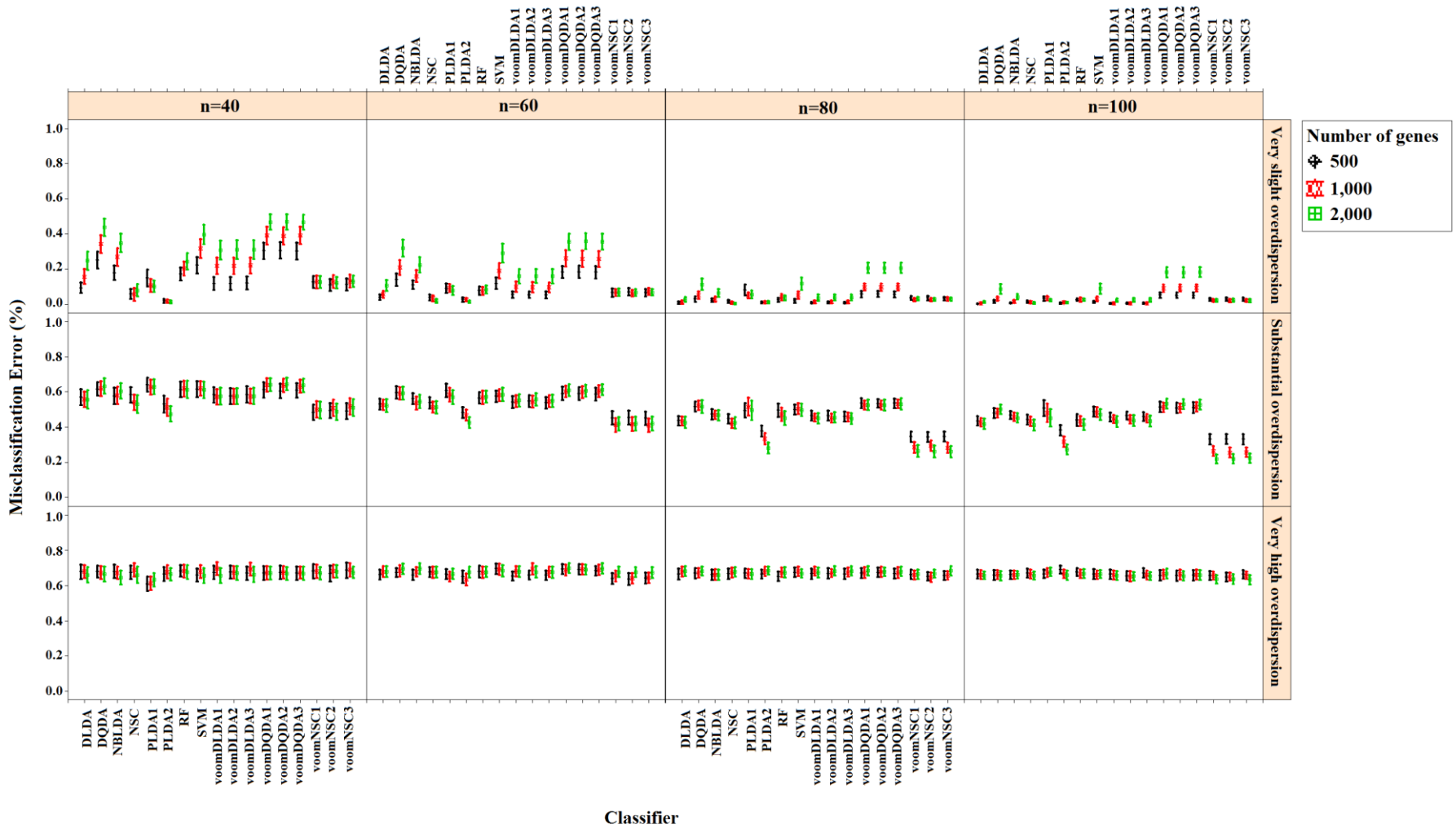

Figure 4.9. Accuracy results for the simulation scenario K=3, e_{gk}=5%, σ=0.1

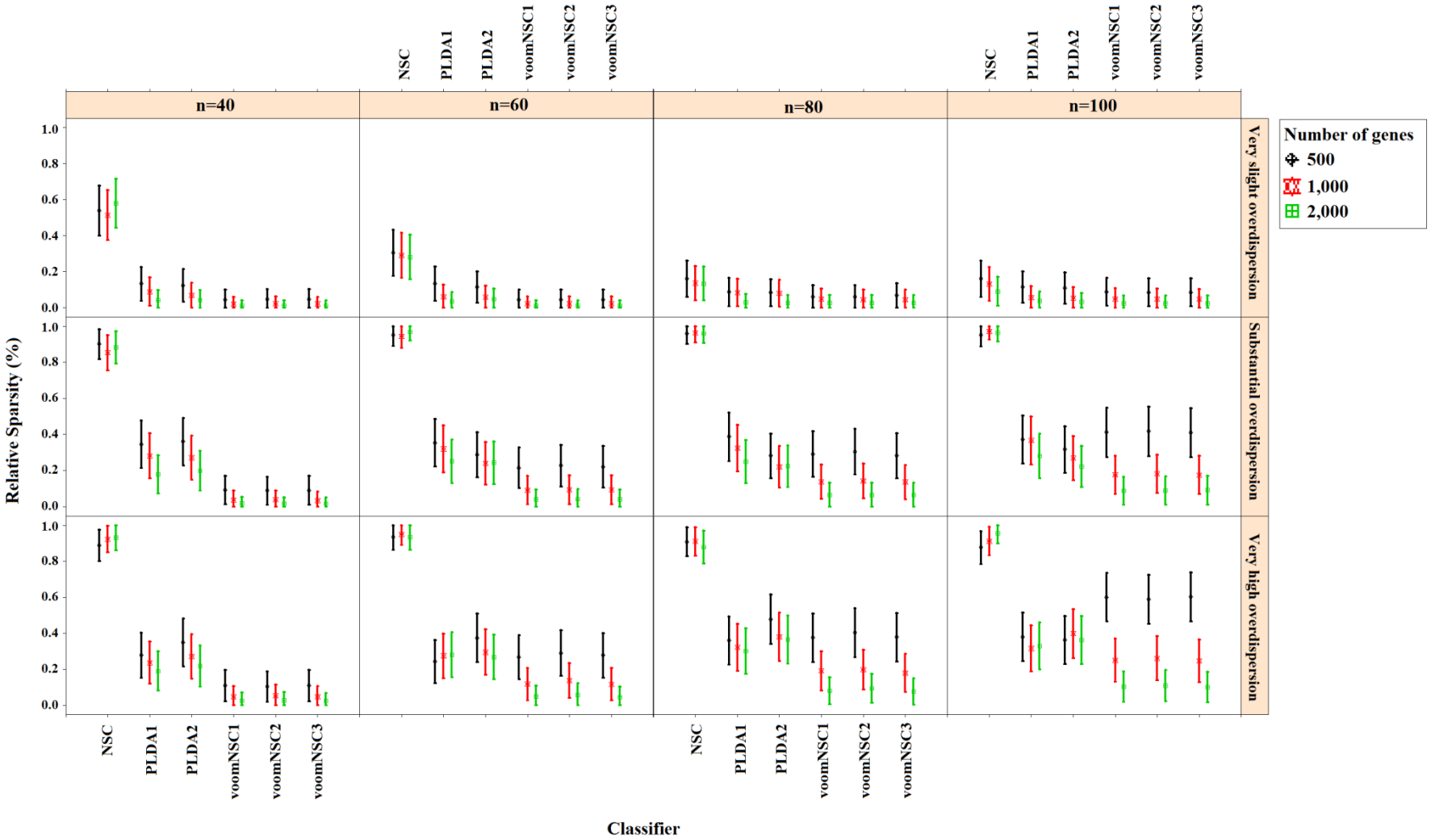

Figure 4.10. Sparsity results for the simulation scenario K=3, e_{gk}=5%, σ=0.1

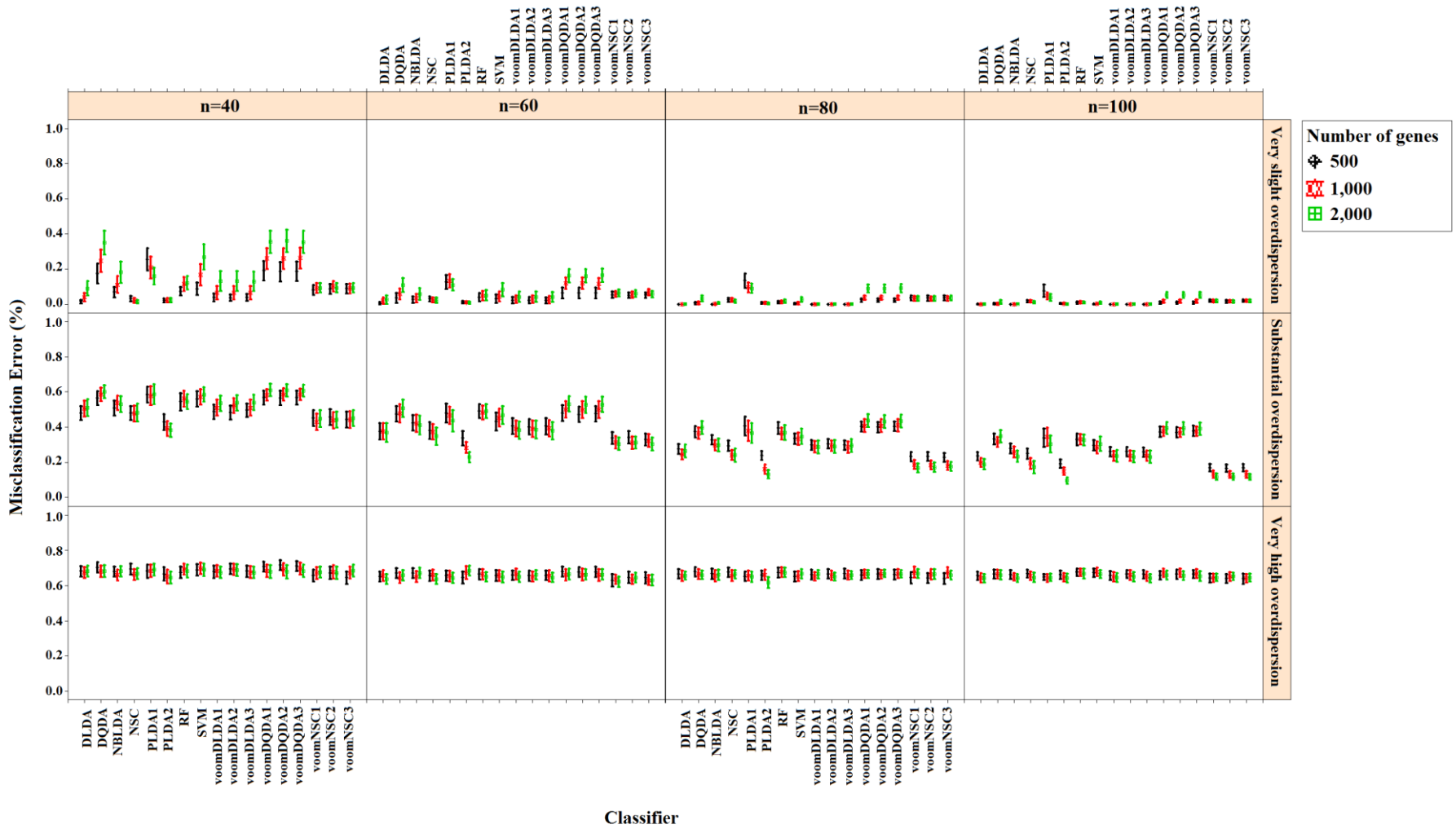

Figure 4.11. Accuracy results for the simulation scenario K=3, e_{gk}=10%, σ=0.1

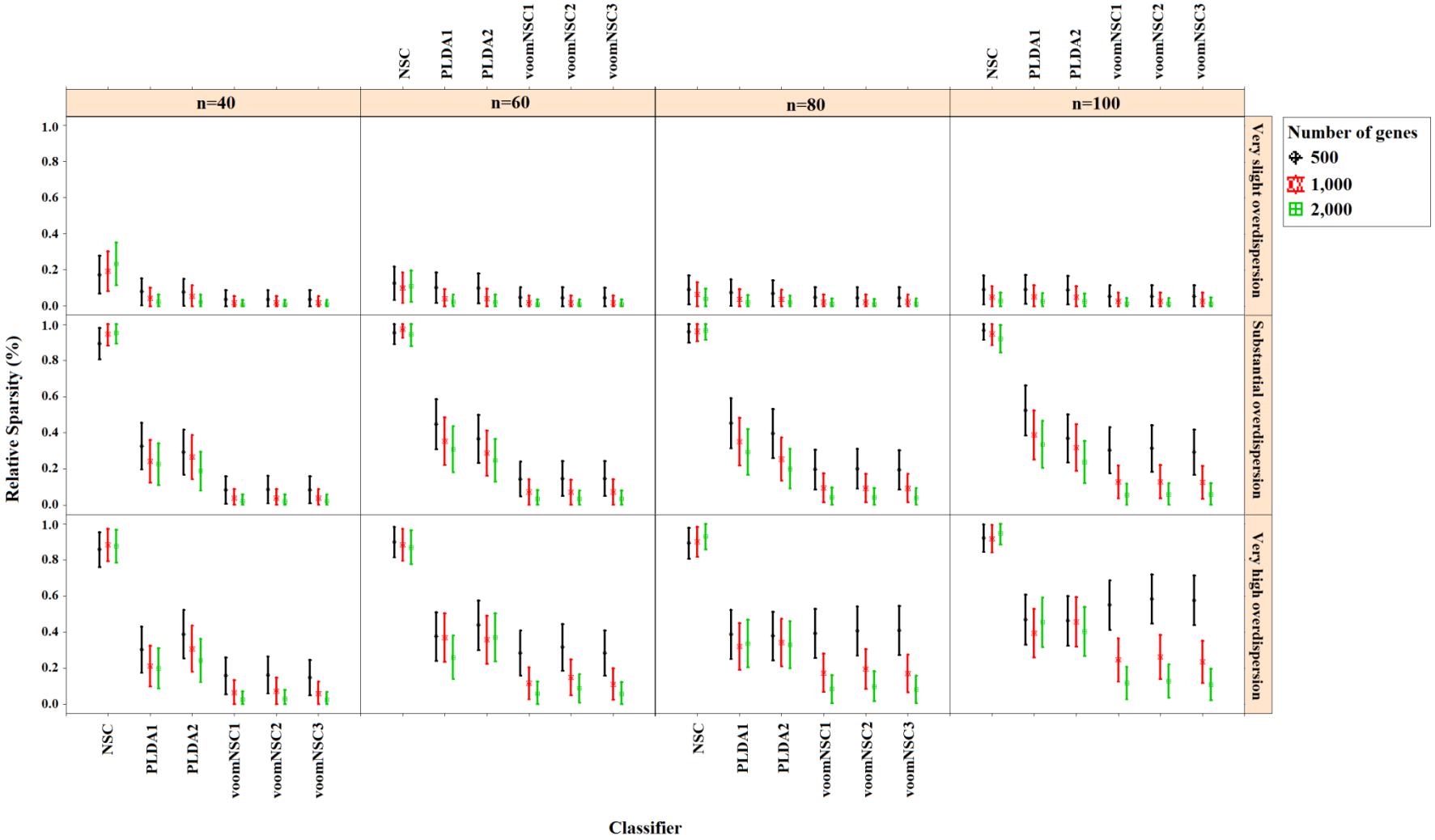

Figure 4.12. Sparsity results for the simulation scenario K=3, e_{gk}=10%, σ=0.1

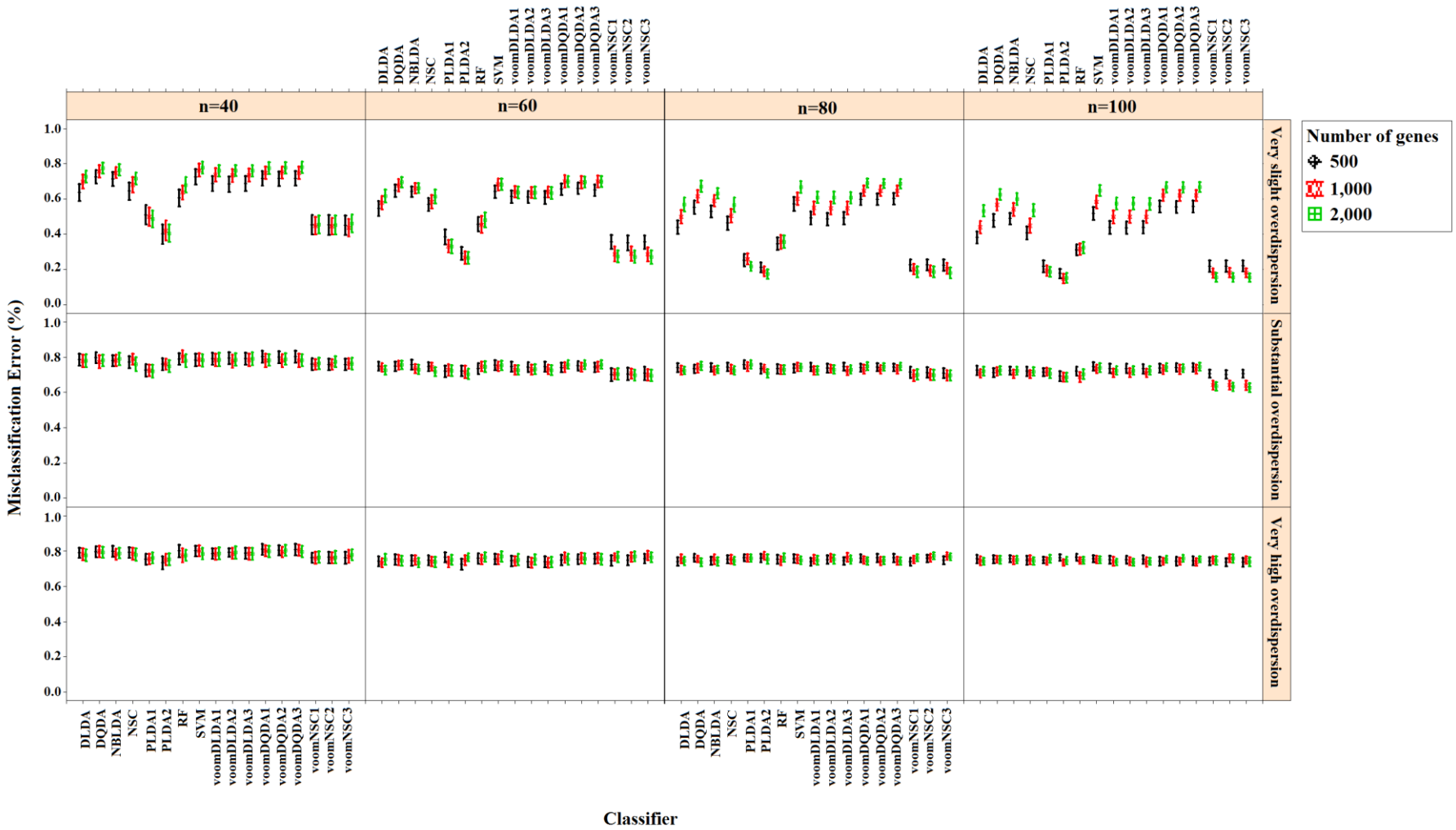

Figure 4.13. Accuracy results for the simulation scenario K=4, e_{gk}=1%, σ=0.1

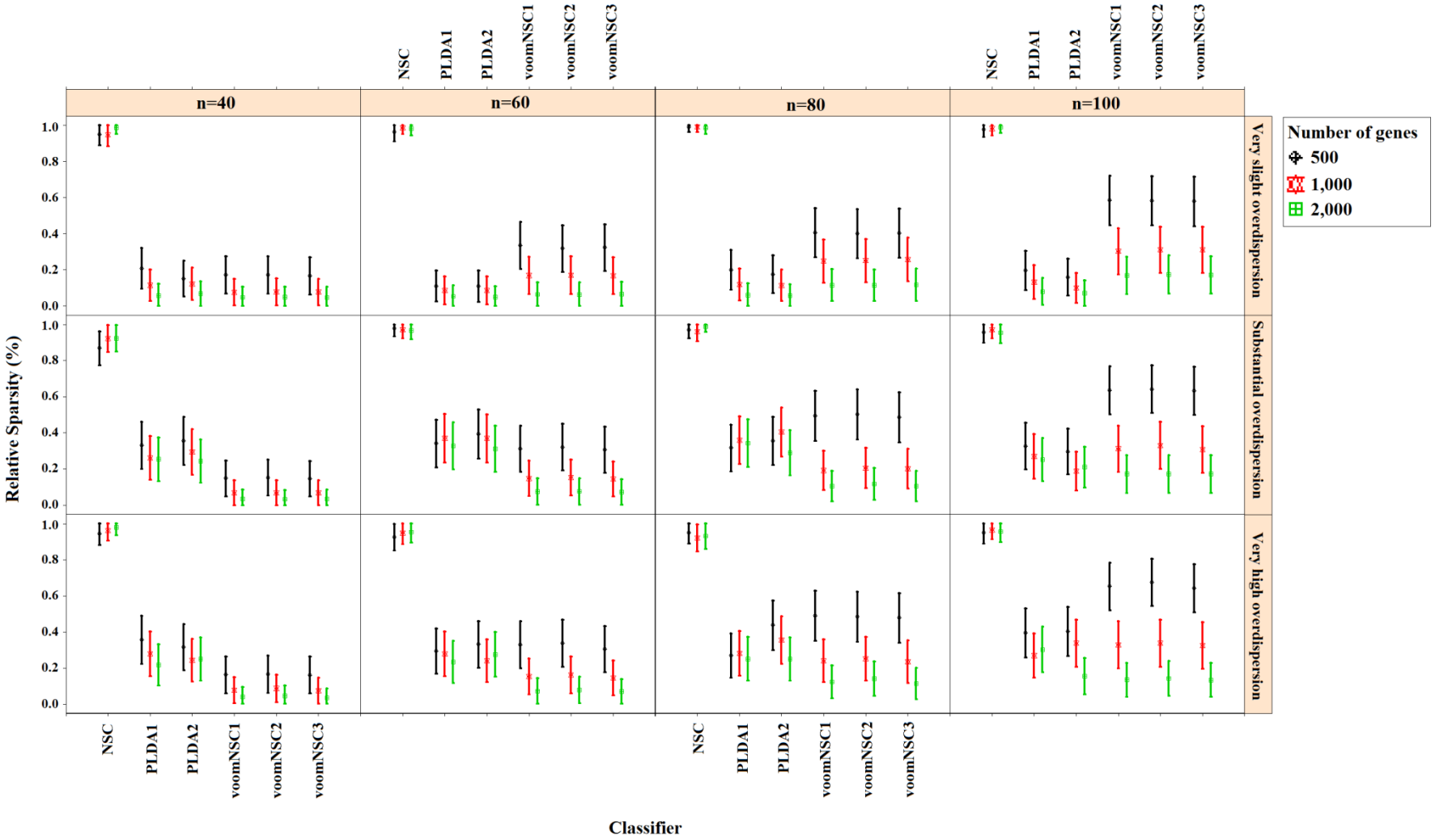

Figure 4.14. Sparsity results for the simulation scenario K=4, e_{gk}=1%, σ=0.1

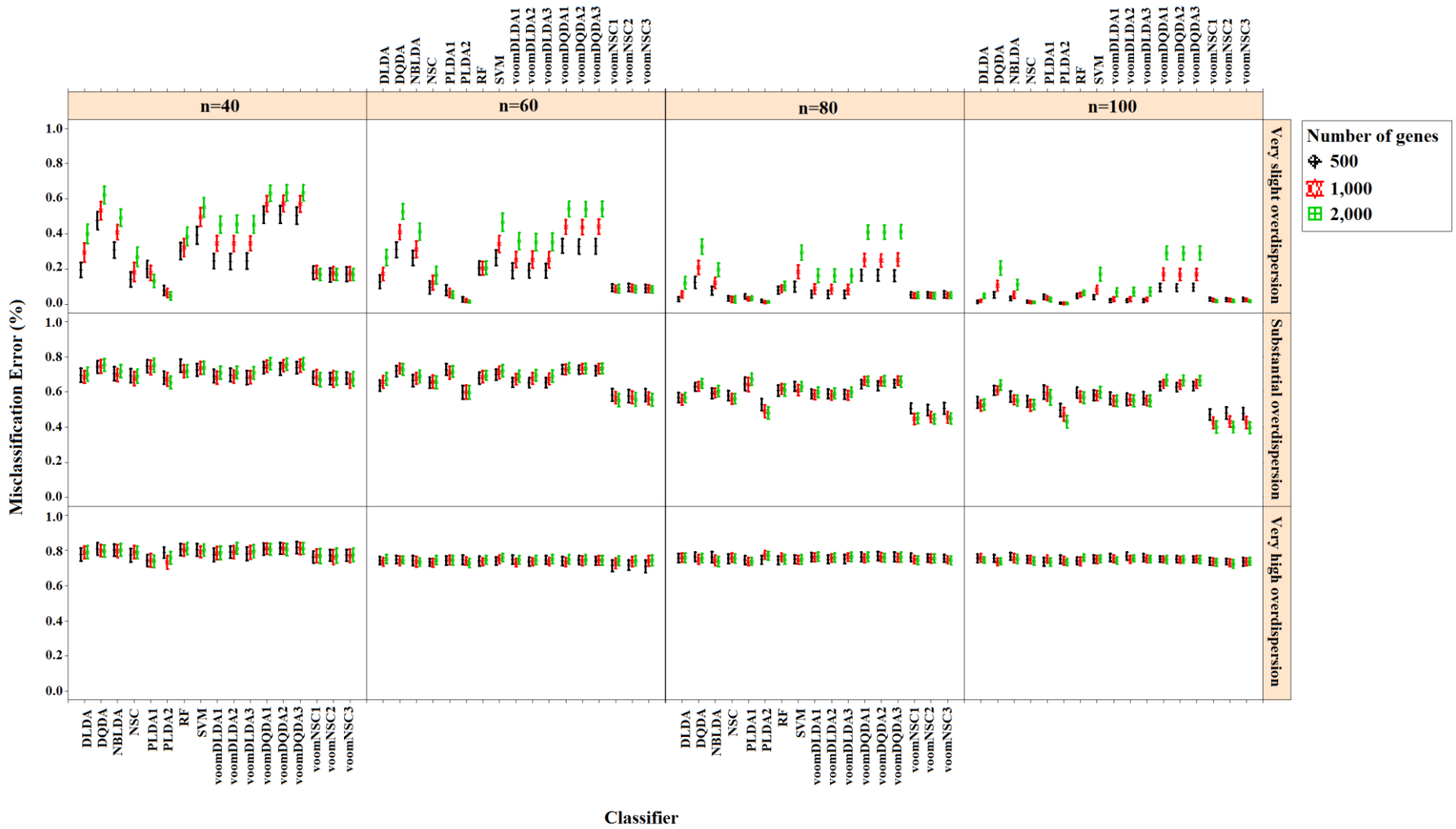

Figure 4.15. Accuracy results for the simulation scenario K=4, e_{gk}=5%, σ=0.1

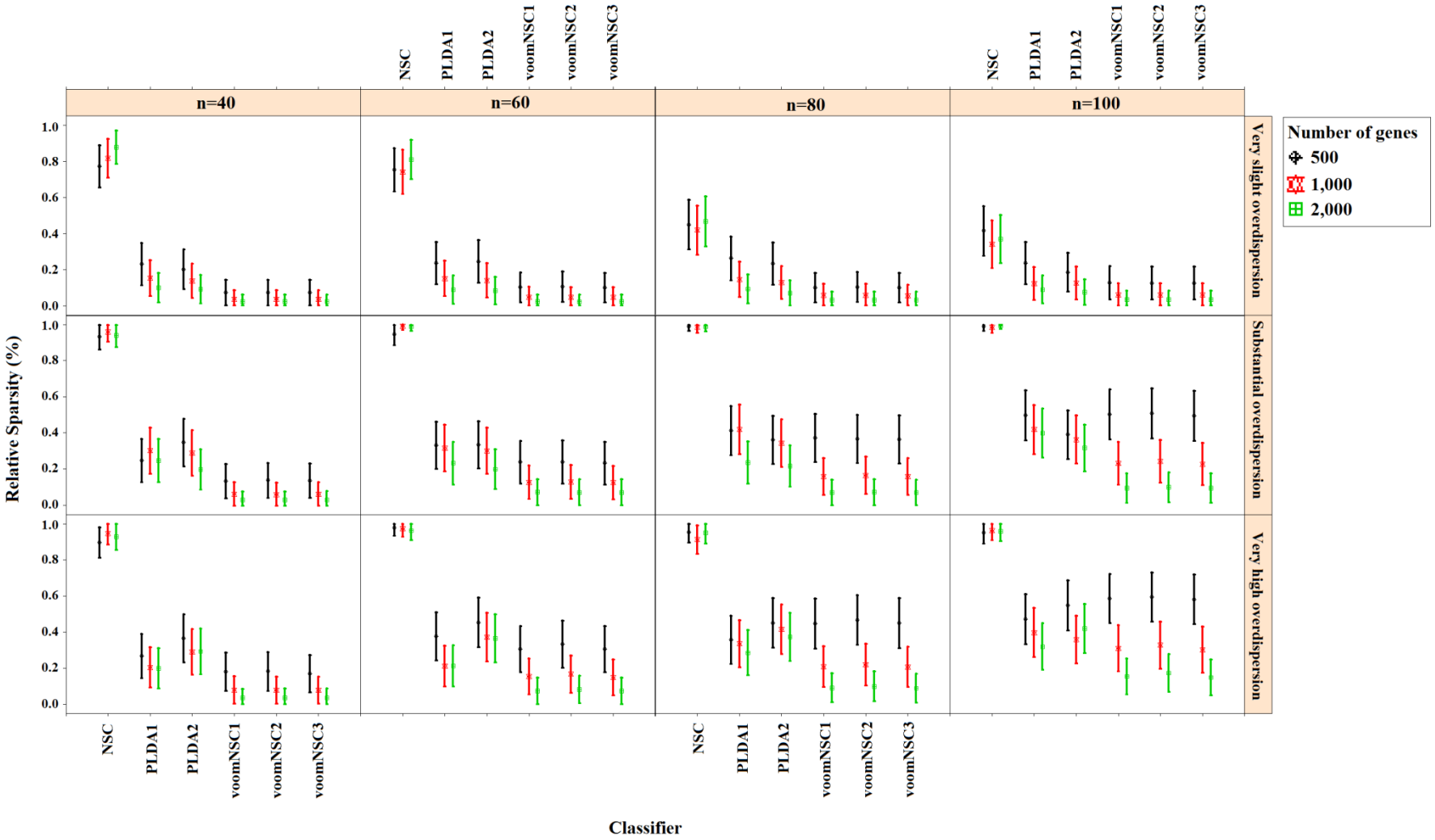

Figure 4.16. Sparsity results for the simulation scenario K=4, e_{gk}=5%, σ=0.1

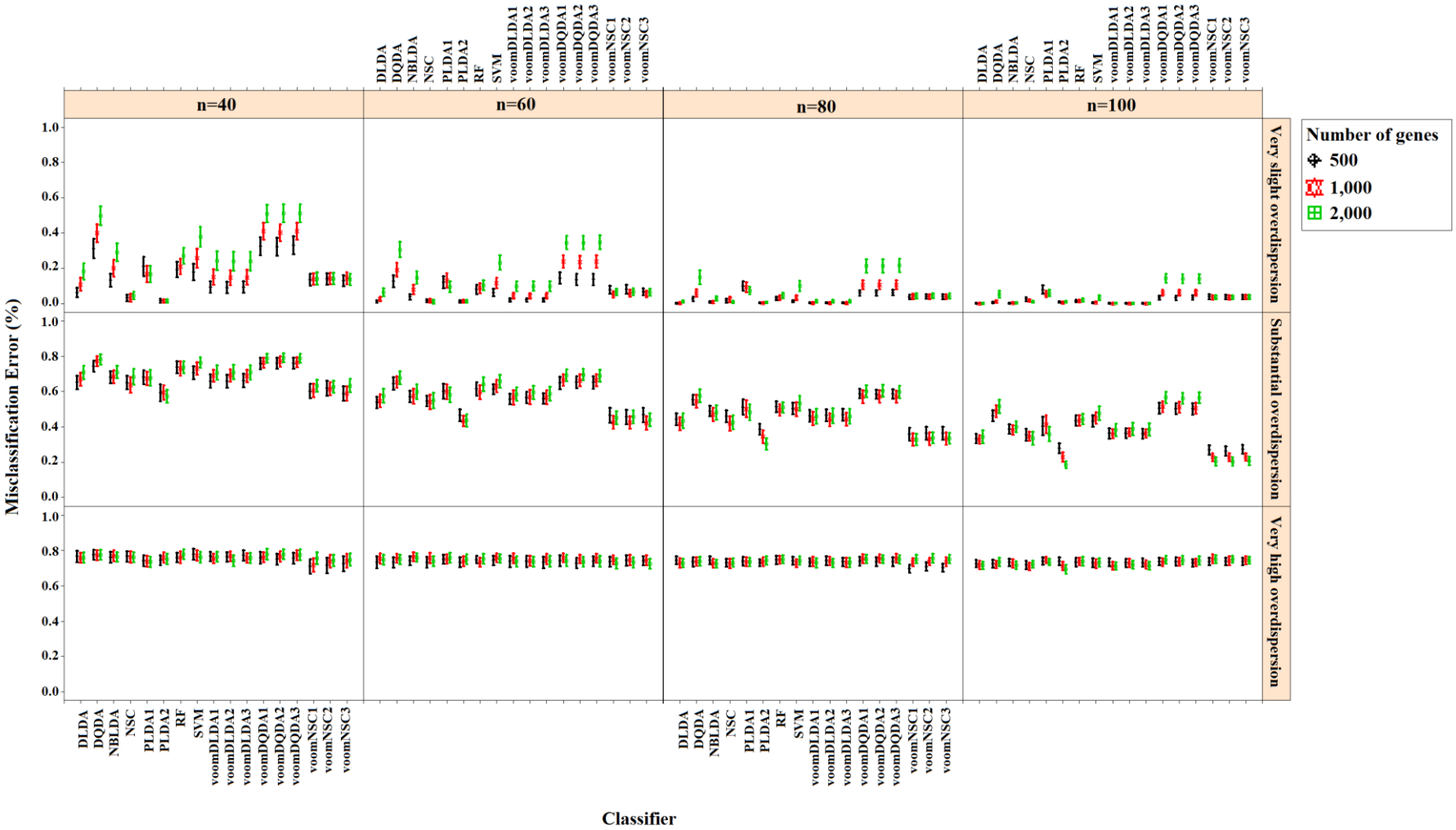

Figure 4.17. Accuracy results for the simulation scenario K=4, e_{gk}=10%, σ=0.1

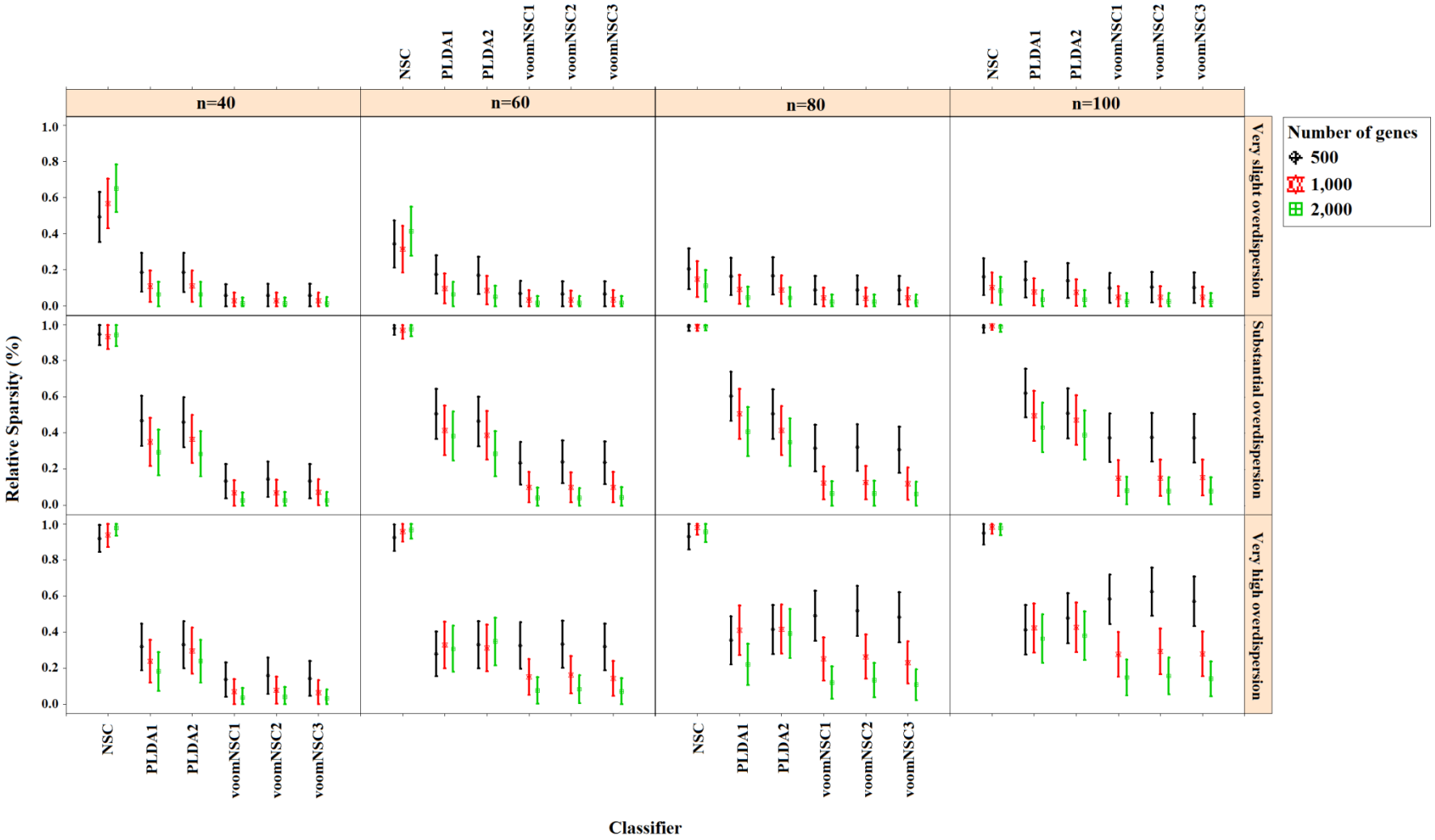

Figure 4.18. Sparsity results for the simulation scenario K=4, e_{gk}=10%, σ=0.1

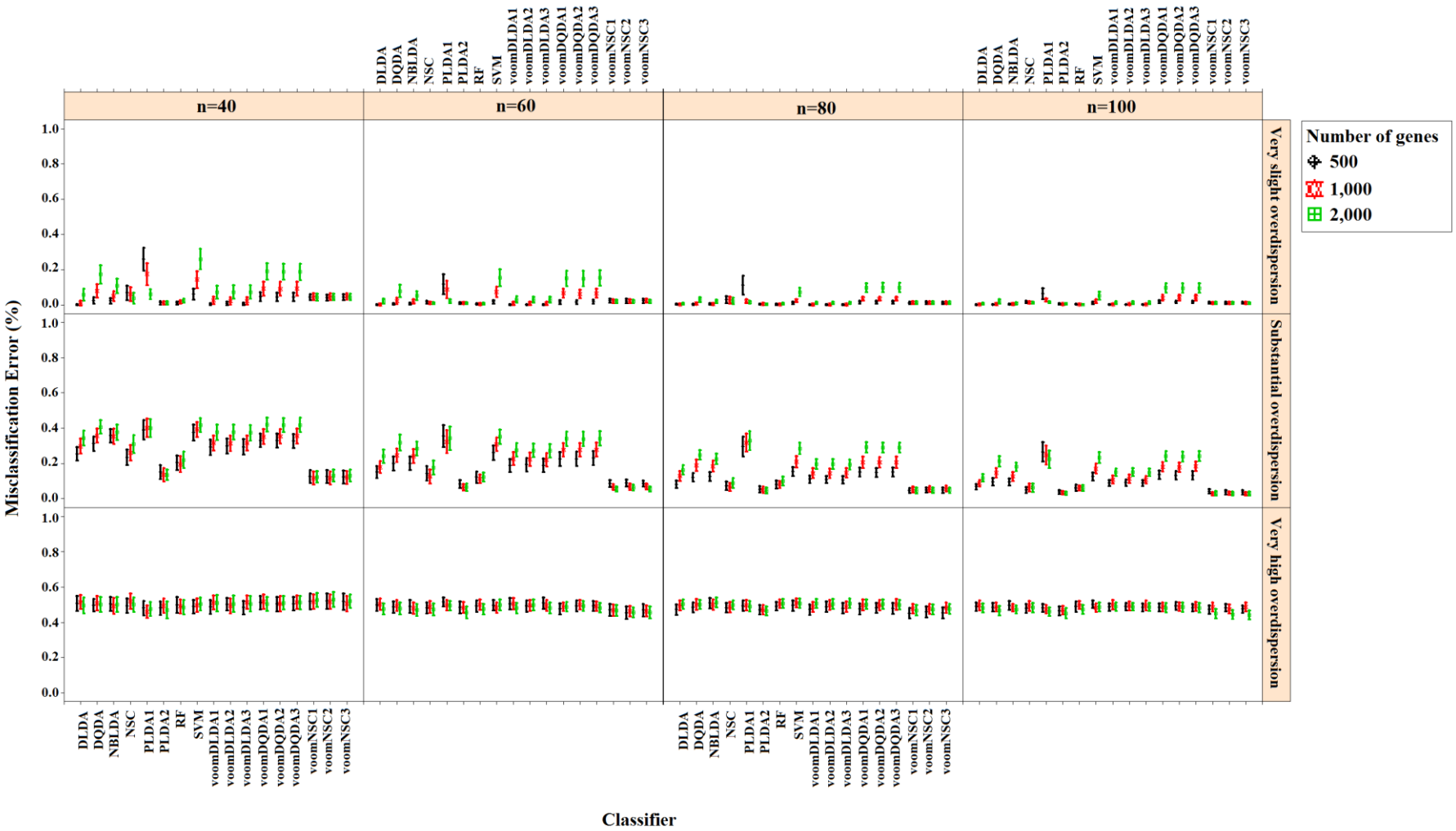

Figure 4.19. Accuracy results for the simulation scenario K=2, e_{gk}=1%, σ=0.2

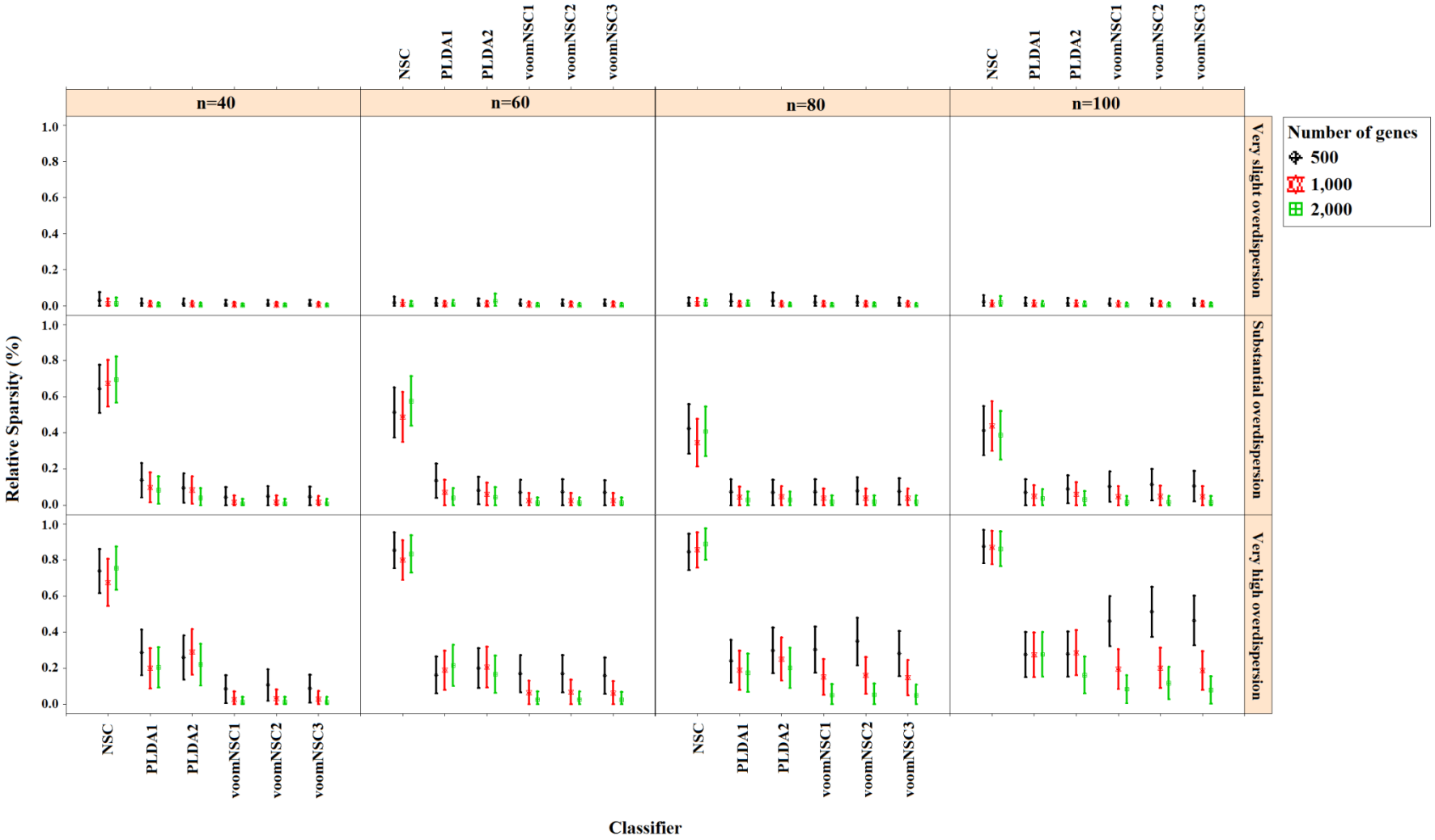

Figure 4.20. Sparsity results for the simulation scenario K=2, e_{gk}=1%, σ=0.2

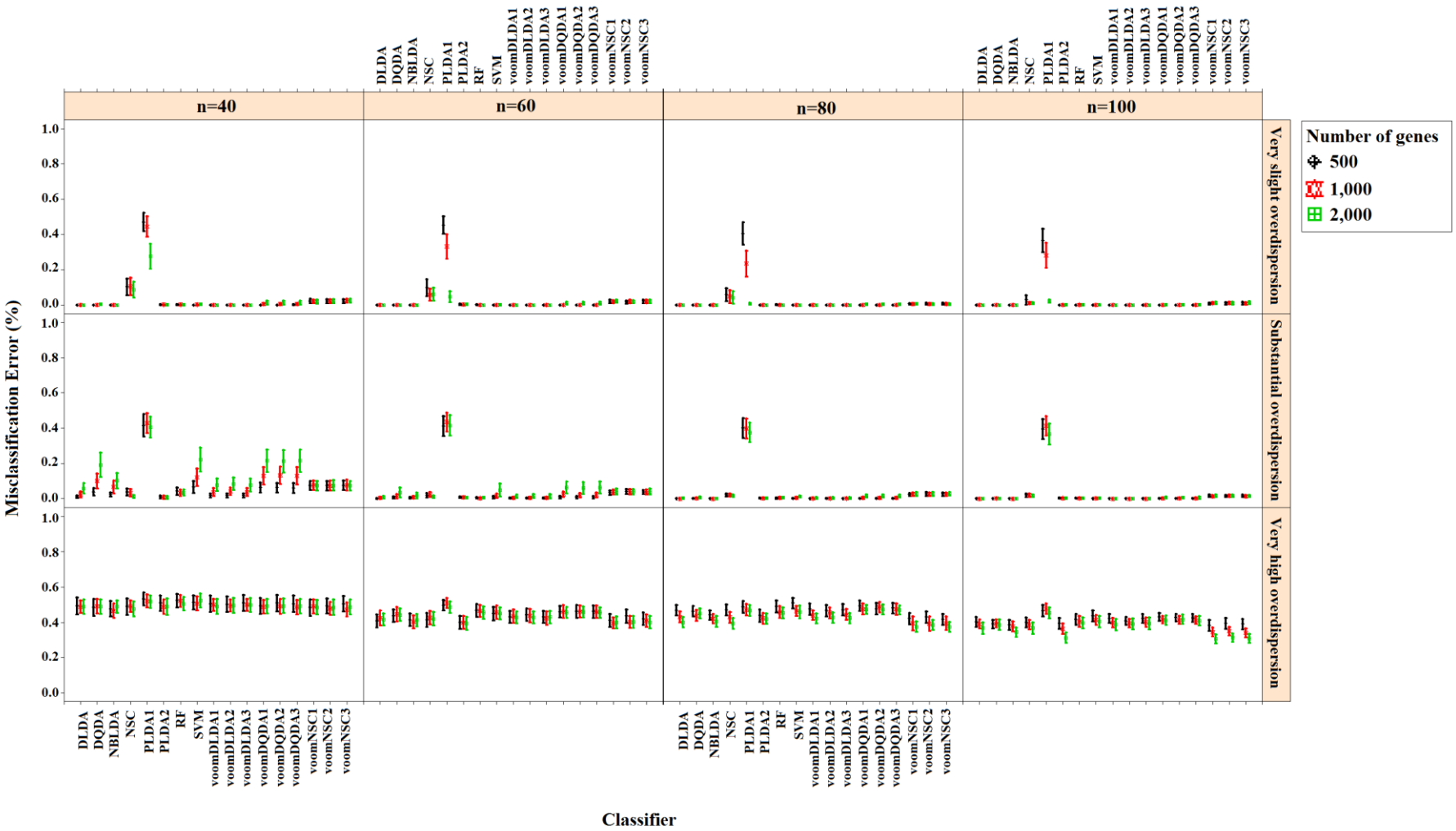

Figure 4.21. Accuracy results for the simulation scenario K=2, e_{gk}=5%, σ=0.2

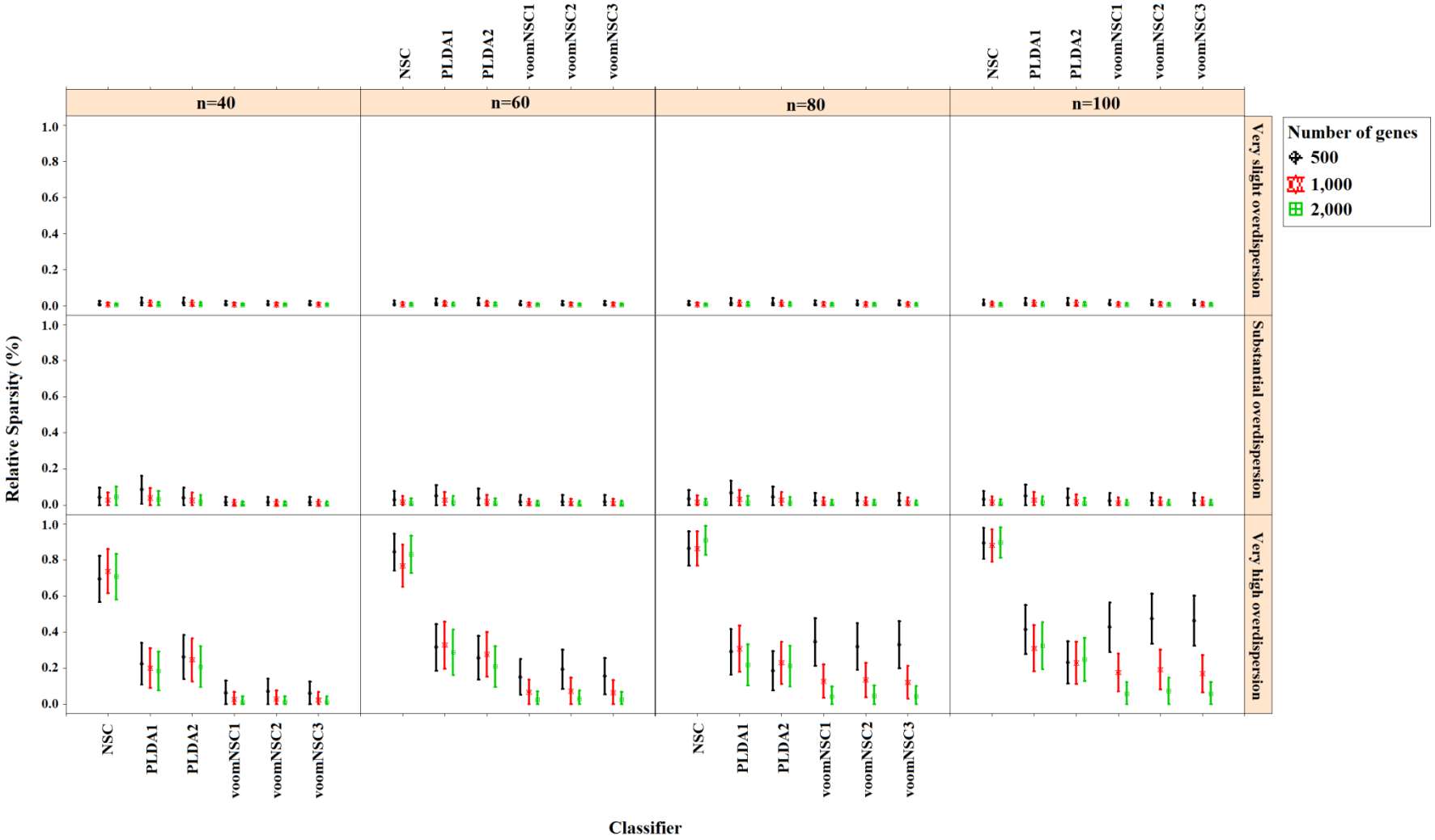

Figure 4.22. Sparsity results for the simulation scenario K=2, e_{gk}=5%, σ=0.2

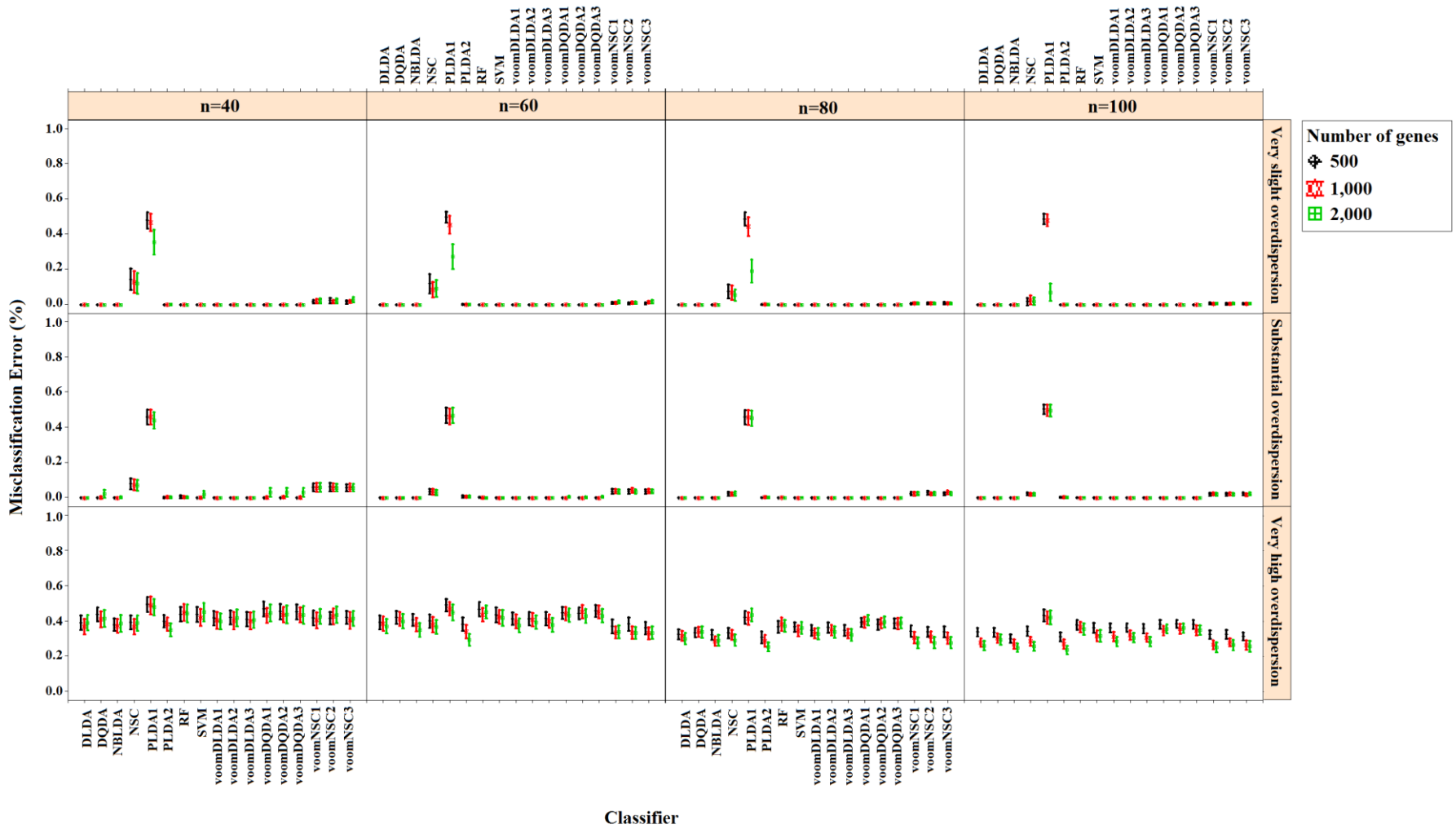

Figure 4.23. Accuracy results for the simulation scenario K=2, e_{gk}=10%, σ=0.2

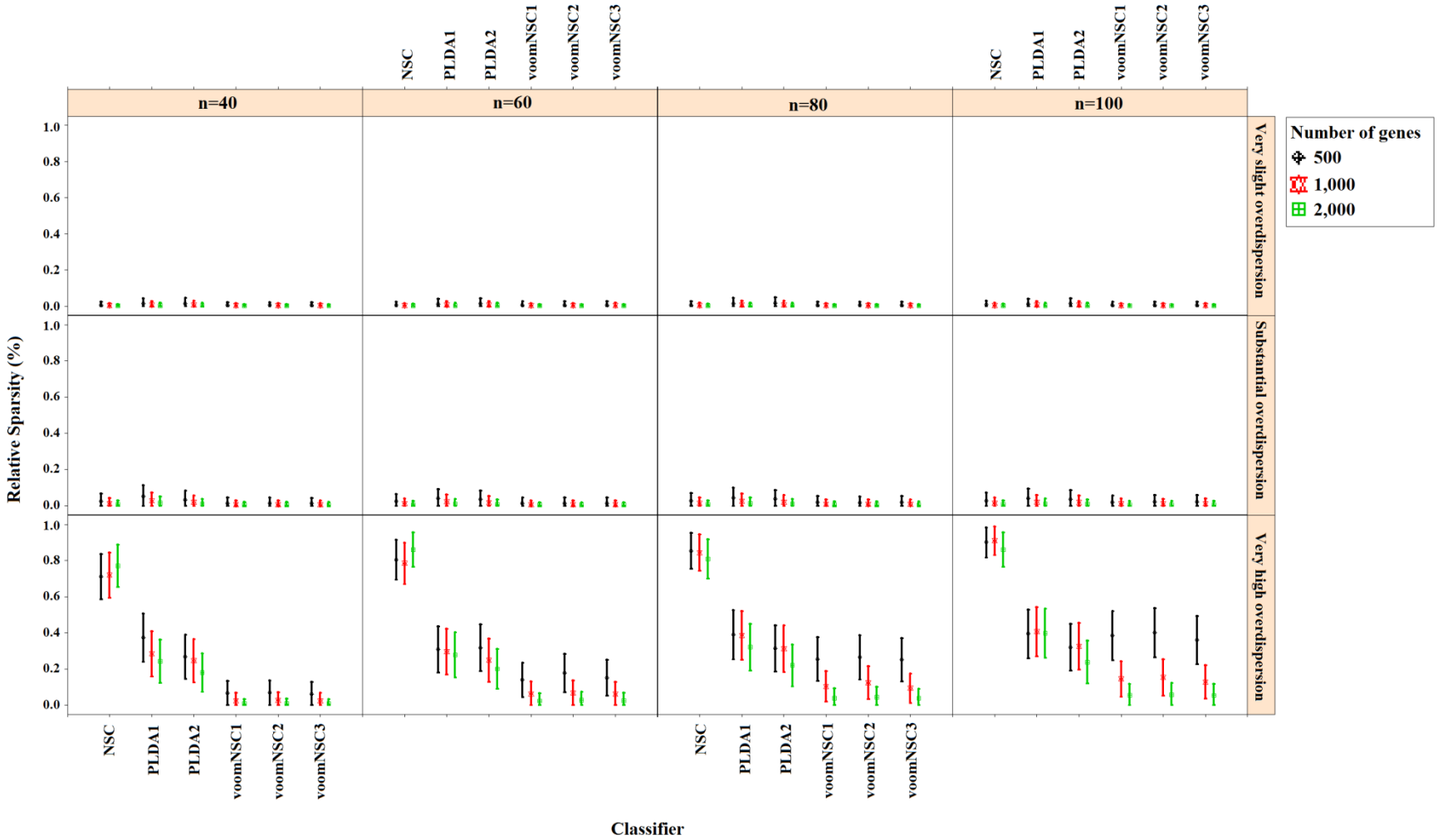

Figure 4.24. Sparsity results for the simulation scenario K=2, e_{gk}=10%, σ=0.2

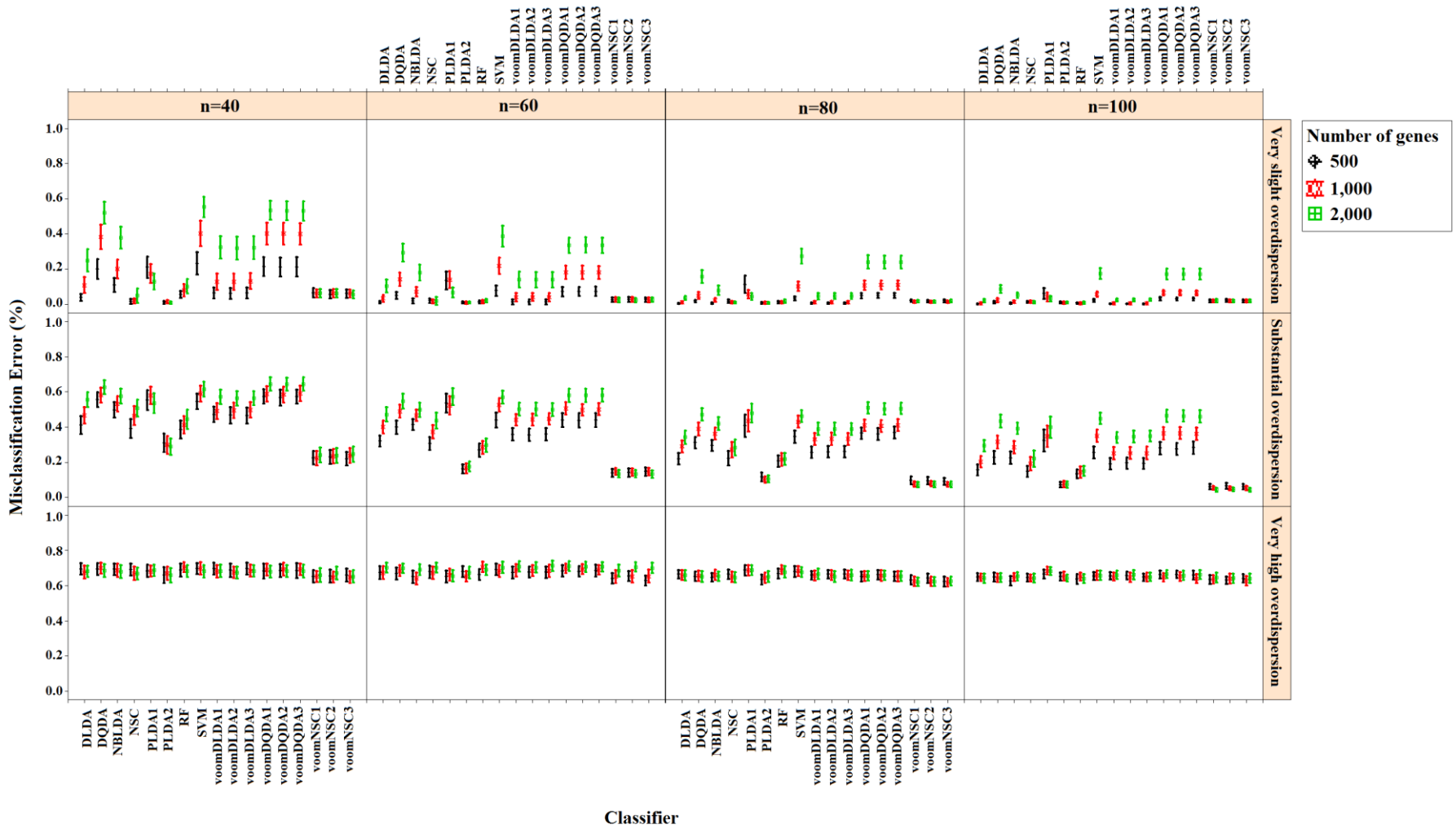

Figure 4.25. Accuracy results for the simulation scenario K=3, e_{gk}=1%, σ=0.2

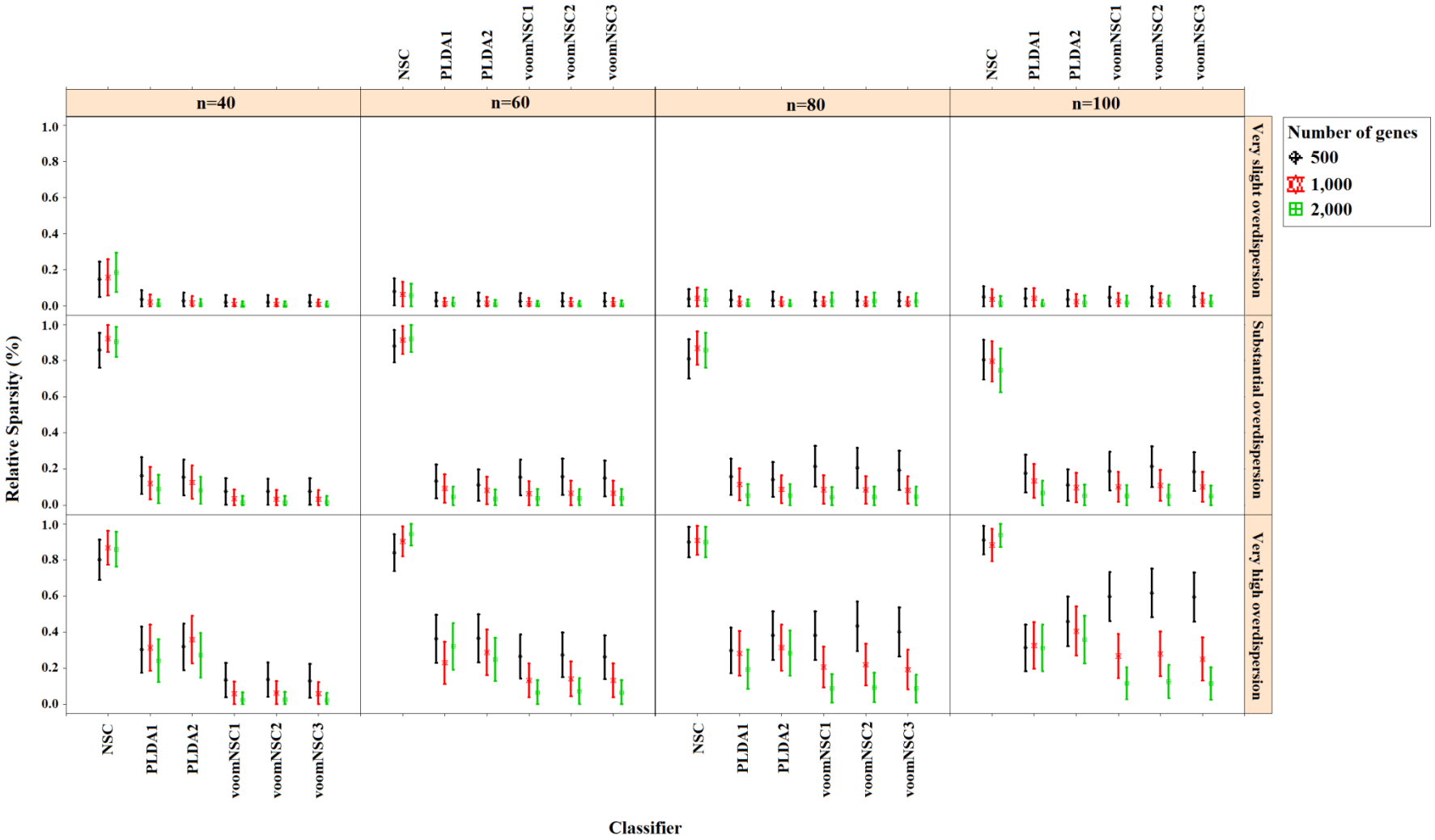

Figure 4.26. Sparsity results for the simulation scenario K=3, e_{gk}=1%, σ=0.2

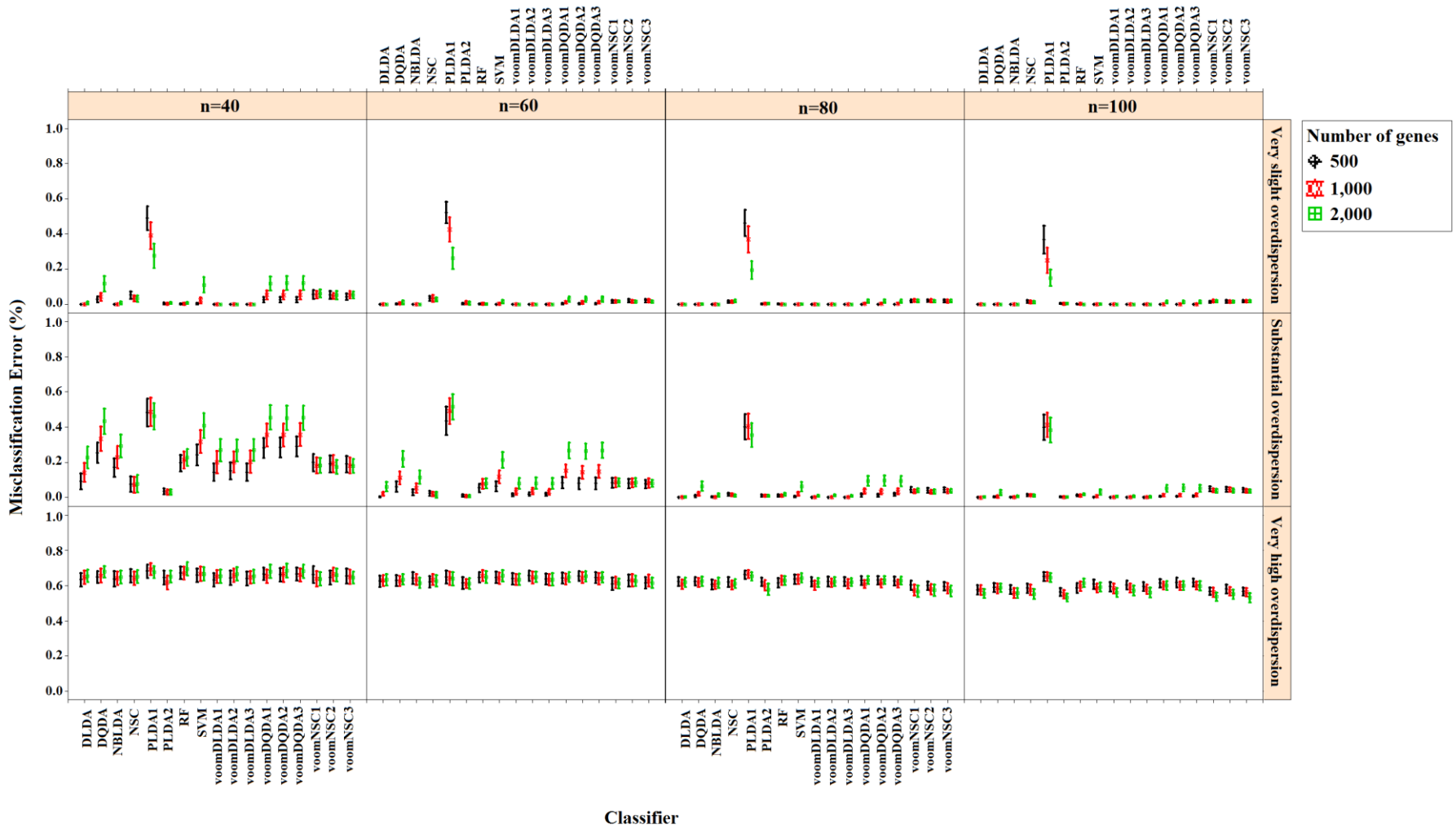

Figure 4.27. Accuracy results for the simulation scenario K=3, e_{gk}=5%, σ=0.2

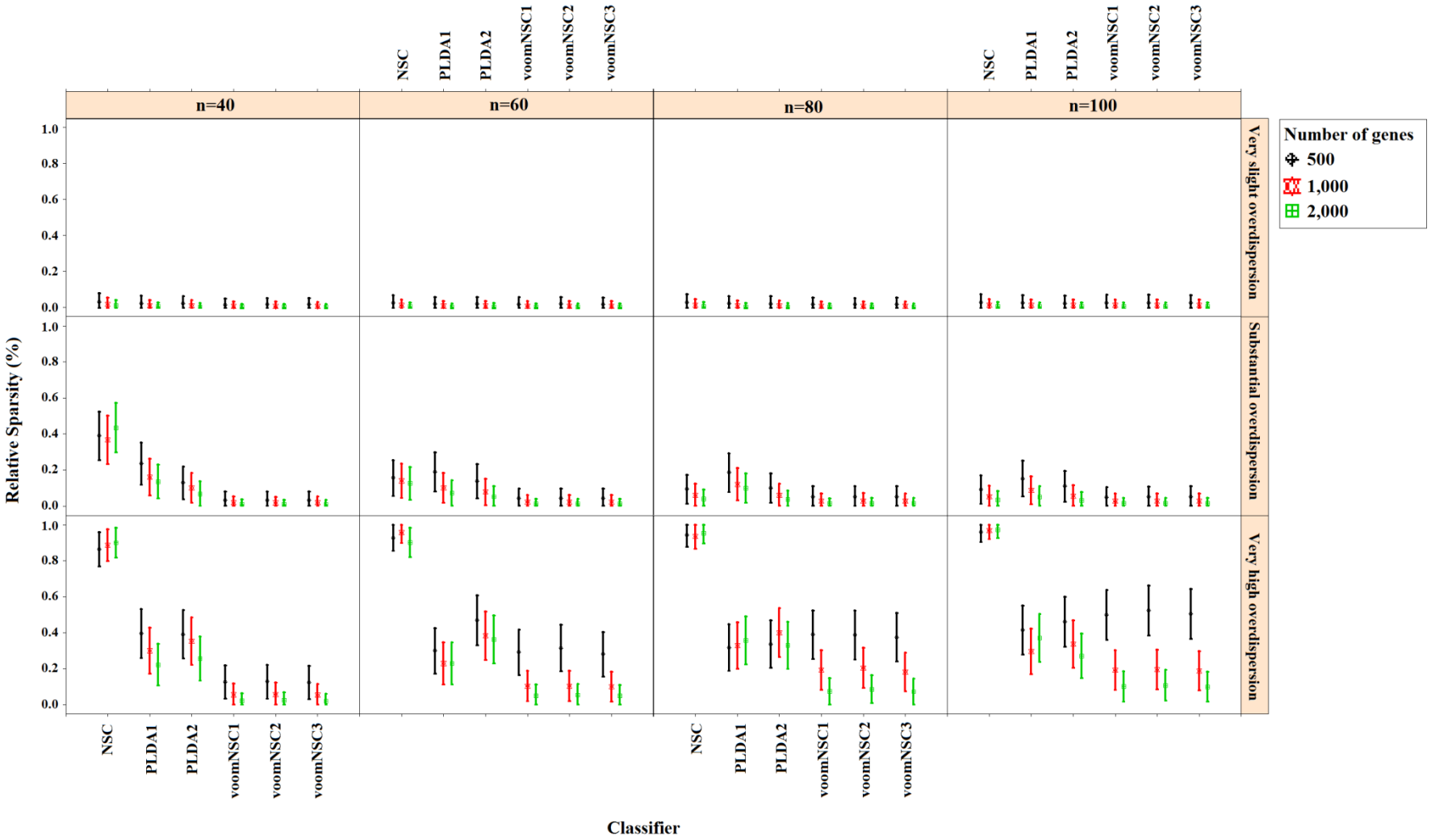

Figure 4.28. Sparsity results for the simulation scenario K=3, e_{gk}=5%, σ=0.2

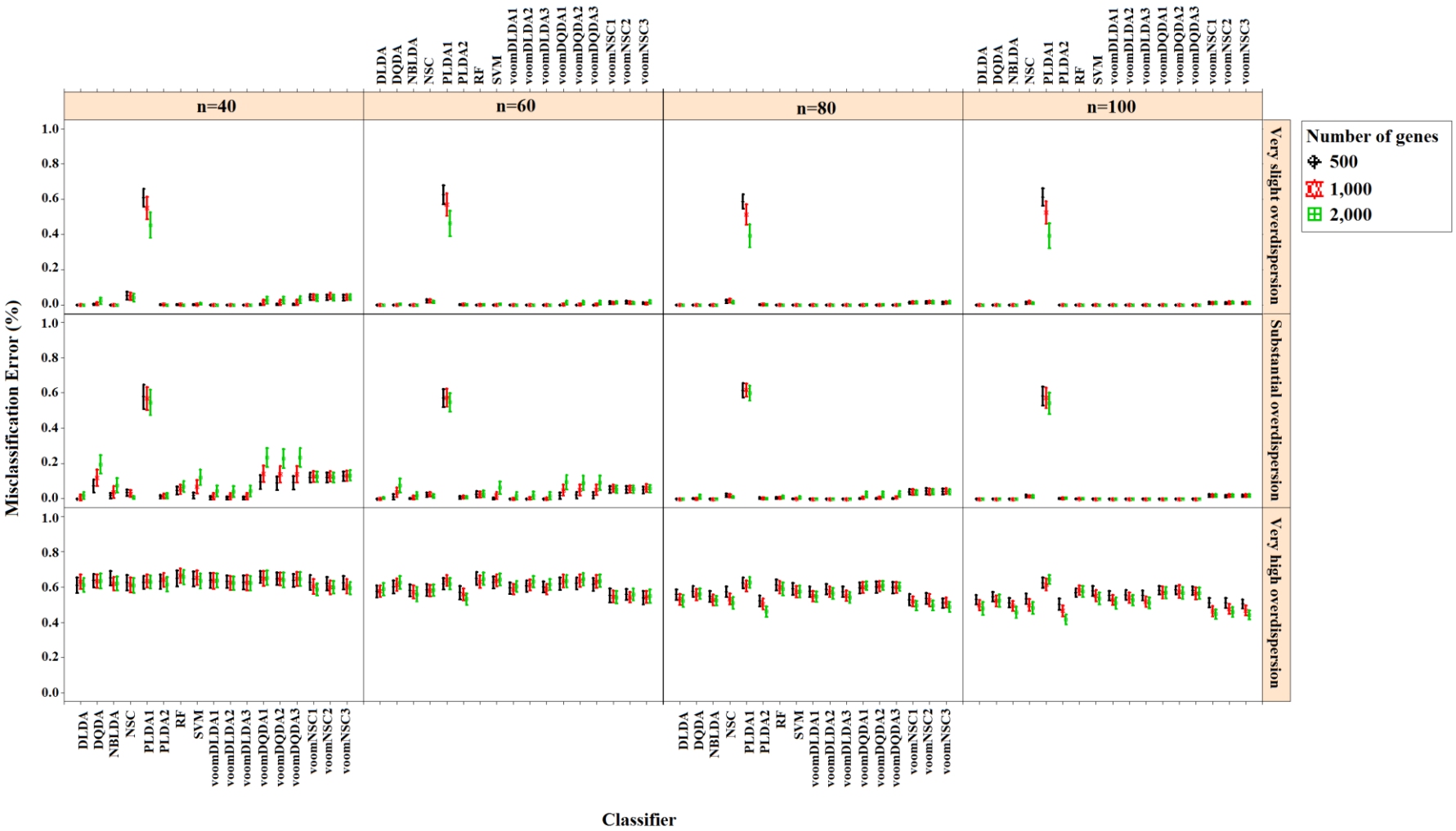

Figure 4.29. Accuracy results for the simulation scenario K=3, e_{gk}=10%, σ=0.2

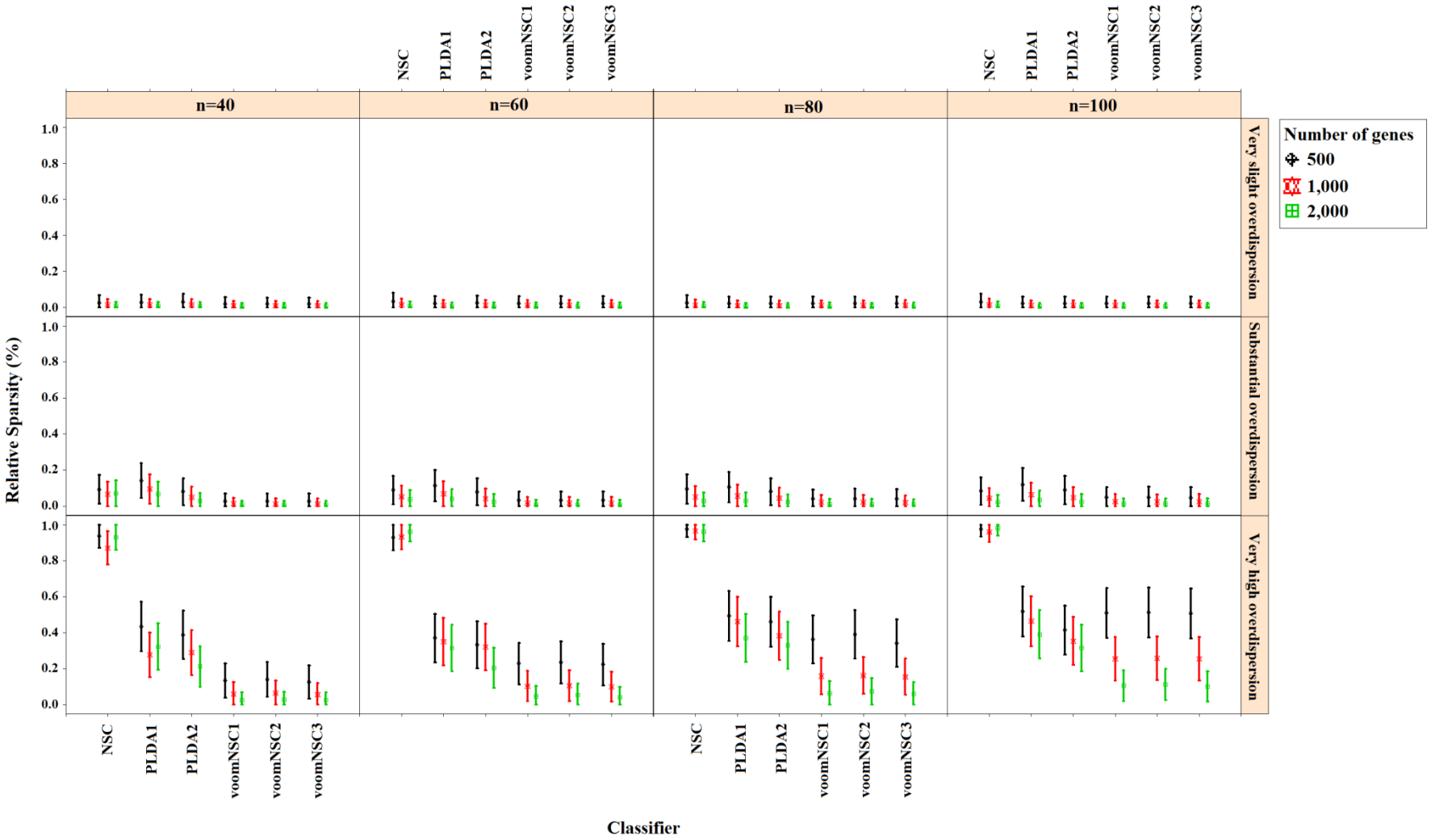

Figure 4.30. Sparsity results for the simulation scenario K=3, e_{gk}=10%, σ=0.2

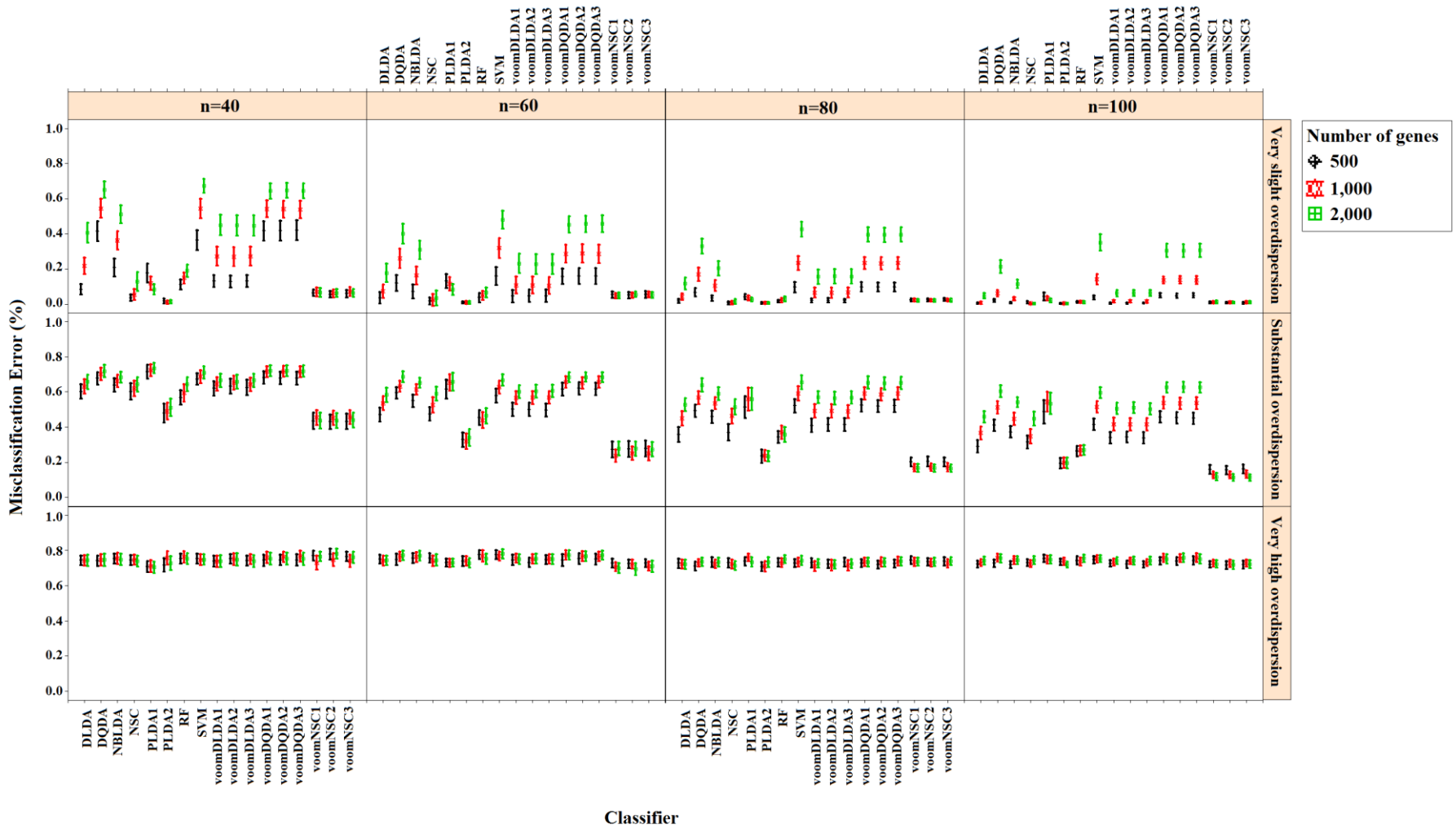

Figure 4.31. Accuracy results for the simulation scenario K=4, e_{gk}=1%, σ=0.2

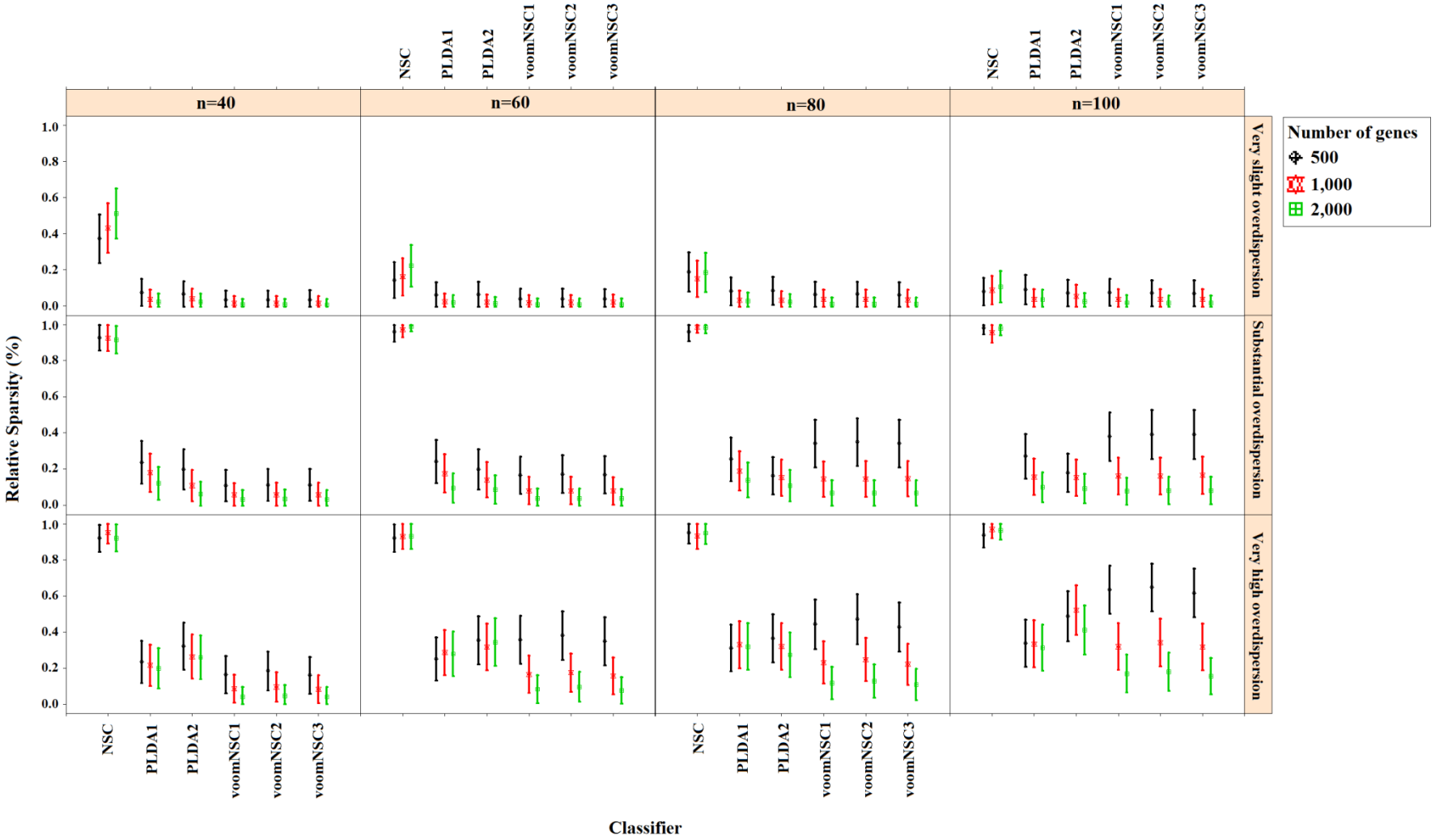

Figure 4.32. Sparsity results for the simulation scenario K=4, e_{gk}=1%, σ=0.2

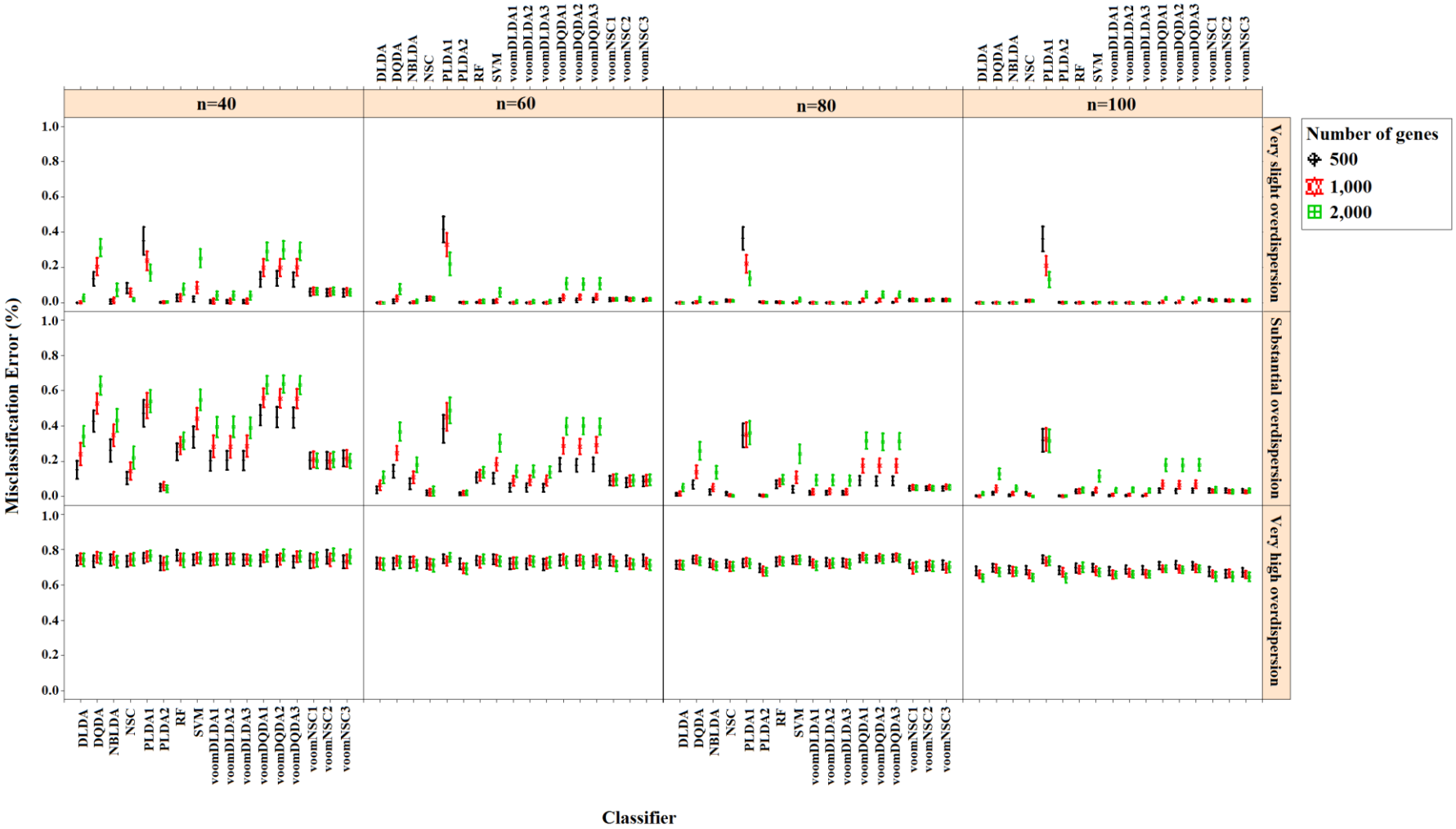

Figure 4.33. Accuracy results for the simulation scenario K=4, e_{gk}=5%, σ=0.2

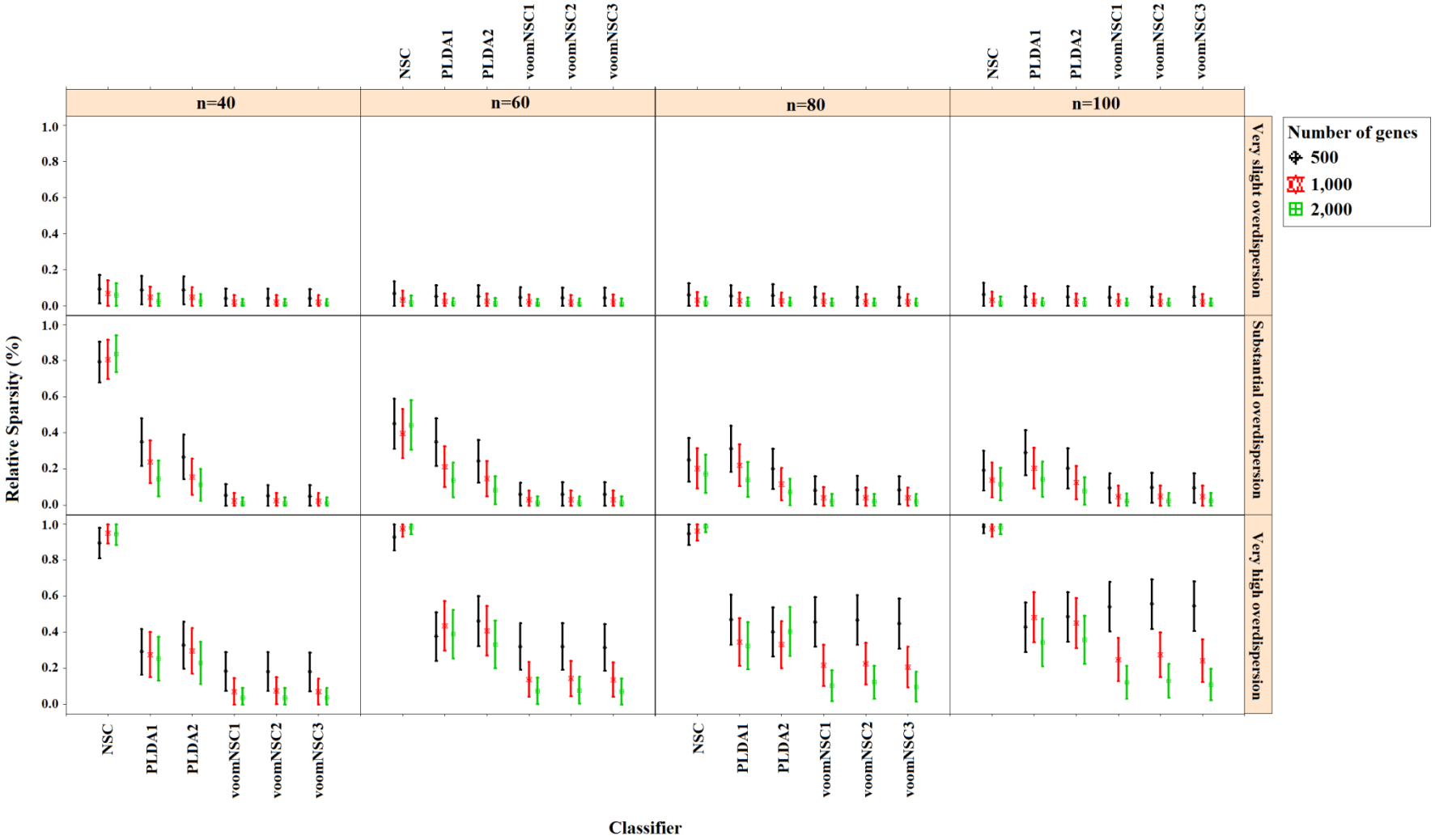

Figure 4.34. Sparsity results for the simulation scenario K=4, e_{gk}=5%, σ=0.2

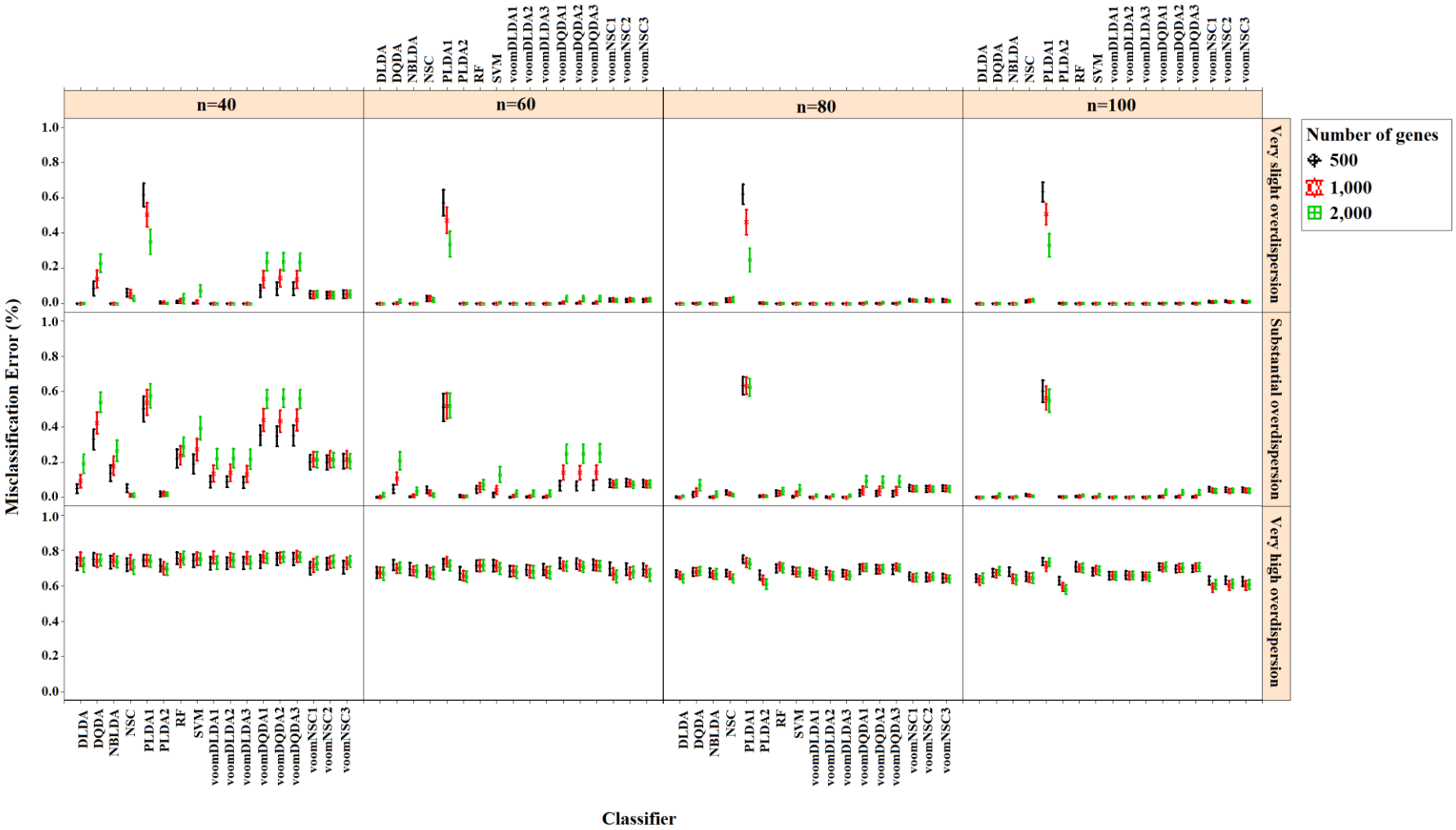

Figure 4.35. Accuracy results for the simulation scenario K=4, e_{gk}=10%, σ=0.2

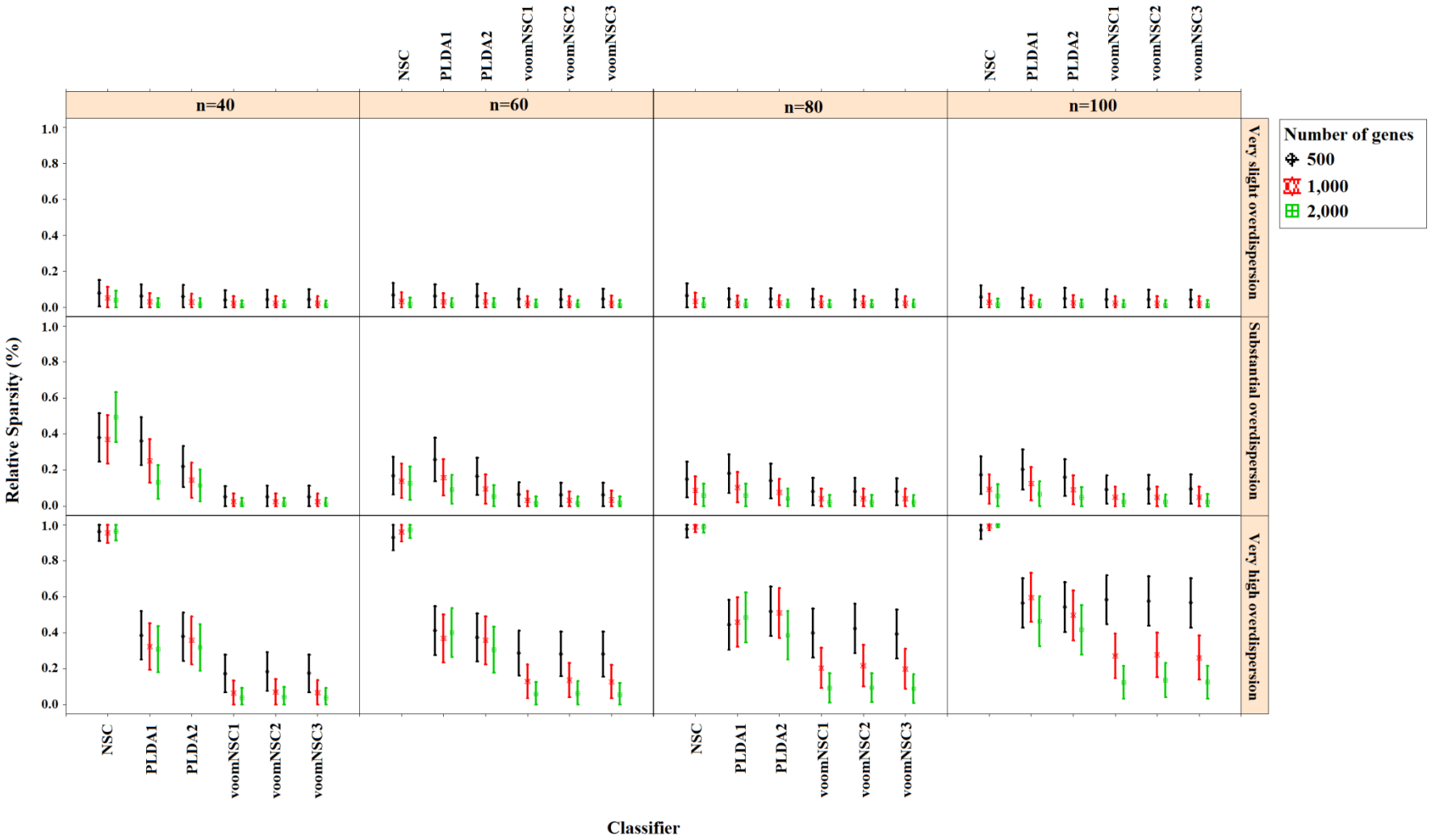

Figure 4.36. Sparsity results for the simulation scenario K=4, e_{gk}=10%, σ=0.2

decreases in their classification accuracies. In slightly overdispersed data, all classifiers, except NSC, produce sparse results. NSC gives sparser solutions based on the increase in the differential expression probability. This probability increase causes to less sparse solutions in PLDA classifier in many scenarios.

Increasing the number of classes lead to a decrease in classification accuracies. This relation particularly arises when the number of genes and the differential expression probability increases as well. The decrease in the performance of $PLDA_2$ and voomNSC classifiers is less than the other algorithms. An increase in the class numbers affect the sparsity of NSC classifier in negative way, while do not affect the other classifiers.

Nearly all classifiers demonstrate more accurate performances depending on the increase in the differential expression probability. Only $PLDA_1$ and NSC classifiers show less accurate performances in this situation depending on the increase in the standard deviation. The increase in these differential expression probabilities brings sparser model performances, mostly for NSC algorithm in slightly overdispersed datasets. The increase in the standard deviation leads to more accurate and sparser classification performances. This may be on contrary for NSC classifier in slightly overdispersed datasets. In this case, sparser solutions come out mostly in slightly overdispersed datasets.

When we assess the performances of classifiers within each other, $PLDA_2$ and voomNSC classifiers performed to be the most accurate algorithms for slightly overdispersed datasets. $PLDA_1$ may be considered as the second performer, RF as the third performer and NSC as the fourth performer classifiers. In substantially overdispersed datasets, voomNSC classifiers become to be the most accurate classifiers, mostly for the scenarios with high number genes. $PLDA_2$ gives compatible results with these classifiers. RF provides substantial results behind voomNSC and $PLDA_2$ classifiers. In highly overdispersed data, all methods generally give very poor results. Considerable performances may be seen when the number of class decreases, and the number of samples, differential expression probability and the standard deviation increases. In such cases, again $PLDA_2$ and voomNSC classifiers outperform other classifiers, mostly for the high number of genes.

In slightly overdispersed datasets, all methods, except NSC algorithm, provide very sparse results. Sparser results for NSC algorithm are seen with the increase in probability in differential expression and standard deviation. In datasets with substantial overdispersion, voomNSC classifiers seem to show their ability, and produce sparser models than the other classifiers, especially in scenarios with high number of genes. In highly overdispersed datasets, voomNSC classifiers clearly build the sparsest models. In this case, PLDA classifiers give less sparse solutions, while NSC algorithm gives the poorest results.

Nonsparse voomDDA classifiers gave compatible results with the rlog generalizations of DLDA and DQDA classifiers. Dispersion has a significant effect on PLDA classifier and $PLDA_2$ classifier outperforms $PLDA_1$ in both accuracy and sparsity in most scenarios.

To make an overall evaluation of the classifiers, we can say that $PLDA_2$ and voomNSC classifiers outperform other classifiers based on the accuracies. When we consider the sparsity measure, voomNSC classifiers are the overall winner and provide the sparsest solutions than the other methods. Finally, we note that the normalization does not have significant effect on the performance of voomNSC algorithm, since all three forms of this method performed very similar results.

4.2. Real Dataset Results

For each dataset, the principal component analysis plots for the first two dimensions are given in Figure 4.37. Again for each dataset, the distribution of dispersion statistics is given in Figure 4.38. Method of moments approach is used to estimate the parameters:

$$\hat{\phi}_g = \frac{Var(x_{gi})}{g_g s_i + \left(g_g s_i\right)^2} \quad (4.1)$$

Real dataset results are given in Table 4.1 and Table 4.2. Table 4.1 demonstrates the misclassification errors and Table 4.2 demonstrates the sparsities for each classifier across 50 repetitions.

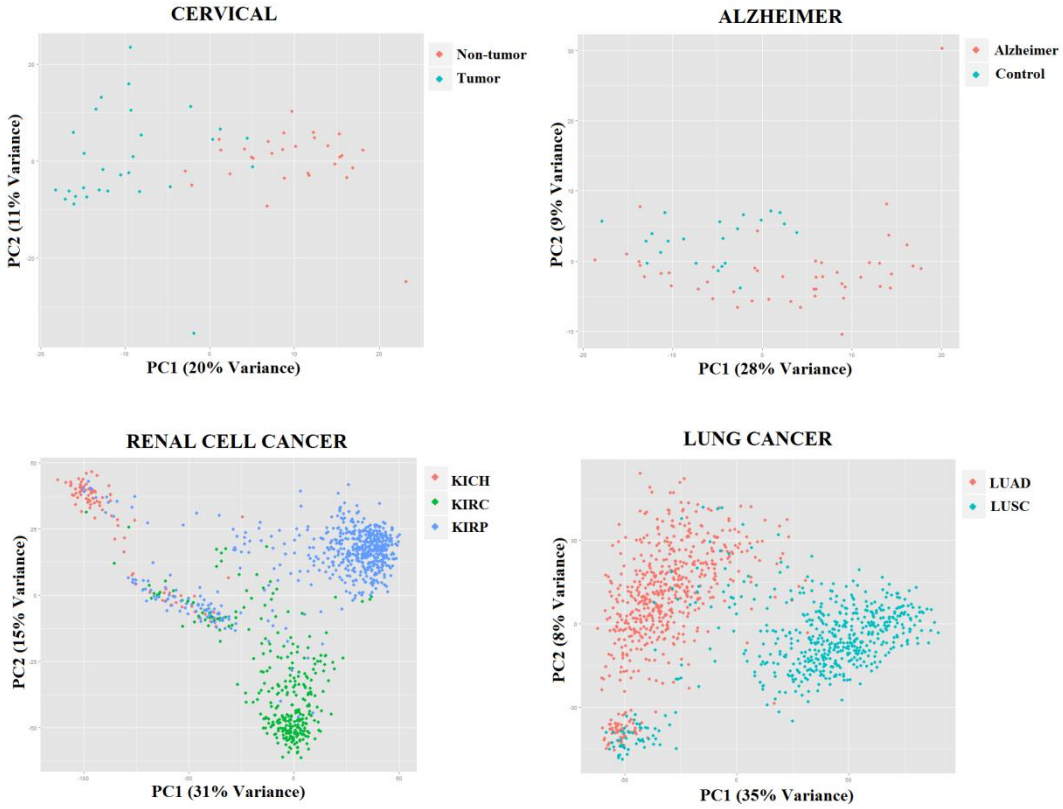

Figure 4.37. Principal component analysis plots for each dataset

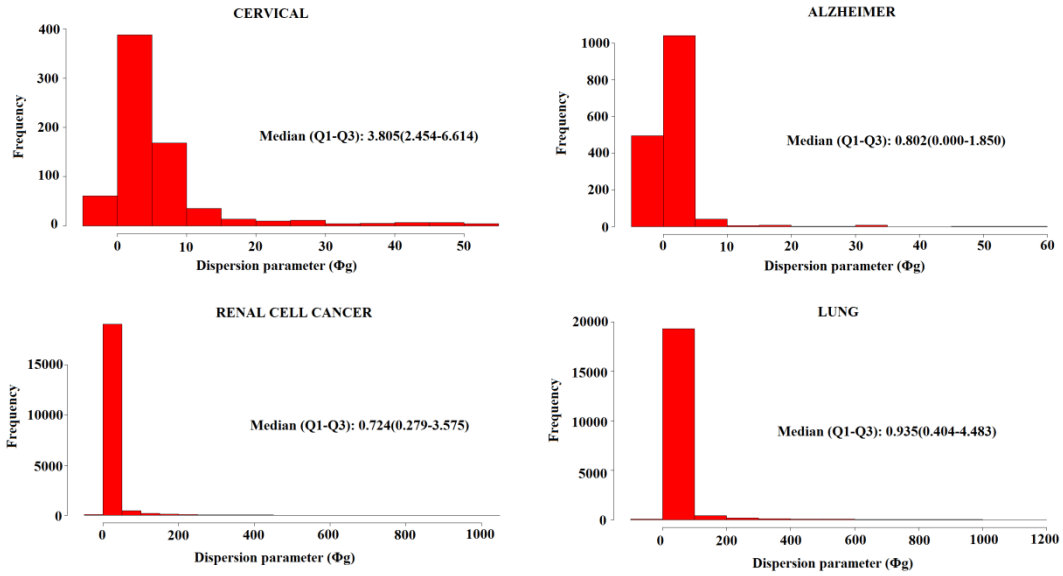

Figure 4.38. Distribution of dispersion statistics for each dataset

In cervical dataset, NBLDA, SVM and NSC algorithms gave the most accurate results with 8.9%, 10.1% and 10.8% misclassification errors, respectively. NBLDA and SVM algorithms use all miRNA's for prediction while NSC selected an average of 194 of them. VoomNSC and $PLDA_2$ classifiers errors were between 11-12%. An average of 290 miRNAs was selected for $PLDA_2$ classifier, while this number was between 56.28 and 63.34 for voomNSC classifiers. Thus, voomNSC classifiers can be considered as the best performers, since the average test errors were compatible with NBLDA, SVM and NSC algorithms; however they use substantially fewer miRNAs than the other classifiers.

In Alzheimer dataset, SVM and $voomDQDA_2$ algorithms performed more accurately than the other algorithms with 8.7% and 13.9% misclassification errors, respectively. $PLDA_1$ was the sparsest classifier with an average of 11 miRNAs, however its test error was 31.7%, which is relatively higher than the other algorithms. Among the other sparse classifiers, $voomNSC_3$ and $voomNSC_1$ fit the other sparsest models with an average of 30 and 48 miRNAs, respectively. Thus, SVM and $voomNSC_3$ classifiers can be considered as the best performers. For this dataset, SVM may build more accurate but complex models. On the other hand, $voomNSC_3$ classifier may give sparser results with relatively less accurate results than SVM algorithm.

In renal cell cancer dataset, SVM and RF are the most accurate classifiers with 6.5% and 7.7% misclassification errors, respectively. $PLDA_1$ classifier performed as the poorest algorithm with 75.6% test set error. The performance of voomNSC classifiers were around 18-19%, which is relatively less accurate than other algorithms. Misclassification errors of other classifiers were between 13-15%. When we look at the sparsity results, NSC and $PLDA_2$ classifiers provided less sparse solutions, with an average of 1989 and 1649 genes, respectively. $PLDA_1$ and voomNSC2 performed moderate results with an average of 607 and 701 genes, respectively. $VoomNSC_1$ and $voomNSC_3$ gave the sparsest results for this dataset. $VoomNSC_1$ selected an average of 178 genes, while $voomNSC_3$ selected 202 genes. In this case, we recommend using SVM and RF classifiers to obtain more accurate results and recommend $voomNSC_1$ and $voomNSC_3$ for sparsest results.

Table 4.1. Misclassification errors of classifiers for real datasets

Classifier	Cervical	Alzheimer	Renal Cell Cancer	Lung Cancer
DLDA	0.149 (0.015)	0.197 (0.012)	0.140 (0.003)	0.098 (0.002)
DQDA	0.140 (0.012)	0.188 (0.012)	0.135 (0.003)	0.098 (0.002)
NBLDA	0.089 (0.010)	0.198 (0.014)	0.139 (0.003)	0.098 (0.002)
NSC	0.108 (0.011)	0.201 (0.012)	0.140 (0.003)	0.097 (0.002)
$PLDA_1$	0.287 (0.029)	0.317 (0.014)	0.756 (0.044)	0.262 (0.028)
$PLDA_2$	0.111 (0.011)	0.223 (0.013)	0.143 (0.003)	0.100 (0.002)
RF	0.135 (0.012)	0.204 (0.013)	0.077 (0.002)	0.062 (0.002)
SVM	0.101 (0.010)	0.087 (0.010)	0.065 (0.002)	0.052 (0.002)
$voomDLDA_1$	0.148 (0.015)	0.210 (0.012)	0.141 (0.003)	0.097 (0.002)
$voomDLDA_2$	0.211 (0.019)	0.228 (0.015)	0.139 (0.003)	0.097 (0.002)
$voomDLDA_3$	0.146 (0.015)	0.203 (0.012)	0.142 (0.003)	0.097 (0.002)
$voomDQDA_1$	0.164 (0.014)	0.181 (0.012)	0.134 (0.002)	0.097 (0.002)
$voomDQDA_2$	0.165 (0.013)	0.139 (0.010)	0.138 (0.003)	0.098 (0.002)
$voomDQDA_3$	0.153 (0.014)	0.170 (0.011)	0.137 (0.003)	0.095 (0.002)
$voomNSC_1$	0.119 (0.013)	0.227 (0.010)	0.181 (0.002)	0.097 (0.002)
$voomNSC_2$	0.111 (0.010)	0.226 (0.018)	0.192 (0.003)	0.097 (0.002)
$voomNSC_3$	0.112 (0.012)	0.233 (0.012)	0.184 (0.002)	0.092 (0.002)

Values are misclassification errors, calculated from 50 repetitions and expressed as mean (standard error).

Table 4.2. Sparsity results of classifiers for real datasets

Classifier	Cervical	Alzheimer	Renal Cell Cancer	Lung Cancer
NSC	194.18 (27.40)	333.06 (19.04)	1989.00 (7.32)	1685.22 (47.73)
$PLDA_1$	290.44 (40.01)	10.81 (9.31)	606.82 (112.40)	1339.90 (112.54)
$PLDA_2$	126.66 (29.13)	228.97 (22.53)	1640.47 (81.59)	1060.84 (70.93)
$voomNSC_1$	56.28 (10.94)	48.06 (10.78)	178.26 (8.18)	85.04 (39.34)
$voomNSC_2$	59.16 (13.60)	140.32 (20.22)	700.90 (114.63)	122.44 (33.22)
$voomNSC_3$	63.34 (13.94)	30.02 (8.10)	208.22 (42.35)	54.18 (34.97)

Values are the number of genes selected in each model, calculated from 50 repetitions and expressed as mean (standard error).

In lung cancer dataset, SVM and RF methods are again the most accurate classifiers with 5.2-6.2% test set errors, respectively. $PLDA_1$ performed as the less accurate algorithm with 26.2% misclassification error. The performance of other classifiers was quite similar and lies between 9.2% and 10.0%. NSC and PLDA classifiers gave substantially less sparse solutions than voomNSC classifiers. Number of selected genes was approximately 1685 genes for NSC, 1340 and 1061 genes for $PLDA_1$ and $PLDA_2$, between 54 and 122 genes for voomNSC classifiers.

4.3. Computational Cost of Classifiers

Along with the accuracy and sparsity results, we calculated the computational costs of each classifier to see whether the developed algorithms are applicable to real datasets. We used a workstation with the properties of Xeon E5-1650, 3.20 GHz CPU, 64GB memory, 12 cores. Performance results are given in Table 4.3. All classifiers seem to be practical for cervical and Alzheimer miRNA datasets. These classifiers are able to fit models less than 2.15 seconds, for these two datasets. Both sample size and number of features are relatively higher in renal cell and lung cancer datasets. This increase affects the computational performance of classifiers, mostly random forests and support vector machines. In overall, DLDA and DQDA classifiers are the fastest among these classifiers. VoomDDA classifiers computational performance is also considerable which is between 0.16 and 5.07 seconds in all datasets.

4.4. VoomNSC Classifiers in Diagnostic Biomarker Discovery Problems

In this section, we detail voomNSC classifier in identifying the potential diagnostic biomarkers for real datasets. For this purpose, we did not apply splitting and use all samples of each dataset. We normalized each data using TMM normalization method. We applied near zero filtering and filtered 91 miRNAs from cervical dataset, 1,226 genes from renal cell cancer dataset and 1,171 genes from the lung cancer dataset. Alzheimer dataset was already filtered form the authors of that study (80). Variance filtering is applied for renal cell and lung cancer datasets. For this purpose, rlog transformation is applied and a total of 2,000 genes having maximal variances are selected.

Table 4.3. Computational costs of classifiers for real datasets

Classifier	Cervical	Alzheimer	Renal Cell Cancer	Lung Cancer
DLDA	<0.01	<0.01	0.07	0.06
DQDA	<0.01	<0.01	0.08	0.05
NBLDA	0.63	0.29	0.94	0.60
NSC	0.34	0.23	2.58	1.60
$PLDA_1$	1.23	0.76	20.76	16.70
$PLDA_2$	1.49	0.96	29.17	21.65
RF	1.41	0.88	116.82	94.20
SVM	2.14	1.06	7.02	5.63
$voomDLDA_1$	0.16	0.27	1.65	1.30
$voomDLDA_2$	0.19	0.18	1.32	1.00
$voomDLDA_3$	0.22	0.31	3.50	2.74
$voomDQDA_1$	0.18	0.19	2.11	1.34
$voomDQDA_2$	0.17	0.17	1.21	1.30
$voomDQDA_3$	0.22	0.26	3.58	2.57
$voomNSC_1$	0.21	0.30	2.69	2.02
$voomNSC_2$	0.20	0.26	2.41	1.75
$voomNSC_3$	0.27	0.50	5.07	3.47

Values are given in seconds.

In cervical dataset, voomNSC algorithm identified the optimal model with the threshold value of 1.858. Using this threshold value voomNSC selected only 14 miRNAs, which is able to assign the samples into one of the two class subtypes (i.e. tumor or non-tumor). Two cases are misclassified that leads to a misclassification error of 3.4%. In Alzheimer dataset, optimum threshold value was 3.313 and 3 miRNAs are selected with a misclassification error of 18.6%. Optimum threshold value was 4.358 in renal cell cancer dataset. A total of 87 genes are selected with 8.5% misclassification error. Finally, the threshold value was optimum at 5.360 in lung cancer dataset. In this dataset, only 6 genes are selected with 8.5% misclassification error. These results are summarized with the selected features in Table 4.4. Heatmap plots are given for these selected genes in Figures 4.39 - 42.

Table 4.4. Summary of voomNSC models and selected genes in real datasets

Classifier	**Misclassification Error**	**Number of Features**	**Selected Features**
Cervical	2/58	14	*miR-1, miR-10b*, miR-147b, miR-183*, miR-200a*, miR-204, miR-205, miR-21*, miR-31*, miR-497*, miR-542-5p, miR-944, Candidate-5, Candidate-12-3p*
Alzheimer	13/70	3	*miR-367, miR-756, miR-1786*
Renal Cell Cancer	87/1,020	87	*SLC6A3, RHCG, CA9, ATP6V0A4, CLDN8, TMEM213, FOXI1, SLC4A1, PVALB, KLK1, DMRT2, ATP6V0D2, PTGER3, HEPACAM2, CLCNKB, BSND, LCN2, PLA2G4F, SLC17A3, ATP6V1G3, RHBG, SLC9A4, GCGR, CLCNKA, NR0B2, CFTR, SCEL, ATP6V1B1, NDUFA4L2, FGF9, ENPP3, TMPRSS2, WBSCR17, HAPLN1, ACSM2A, FLJ42875, C6orf223, SLC26A7, ACSM2B, LRP2, FBN3, CNTN6, UGT2A3, EPN3, CALCA, SLC22A11, KLK4, STAP1, LOC389493, FOXI2, CLRN3, HS6ST3, HAVCR1, PART1, EBF2, PCSK6, SLC28A1, SFTPB, OXGR1, CLNK, C16orf89, HSD11B2, TRIM50, ACMSD, CXCL14, VWA5B1, KLK15, INPP5J, LRRTM1, SYT7, HGFAC, FAM184B, C1orf186, KLK3, GPRC6A* *KBTBD12, HCN2, C9orf84, GCOM1, PCDH17, PDZK1IP1, KRTAP5-8, ODAM, RGS5, CTNNA2, GGT1, KDR*
Lung Cancer	96/1,118	6	*DSG3, CALML3, KRT5 , SERPINB13, DSC3, LASS3*

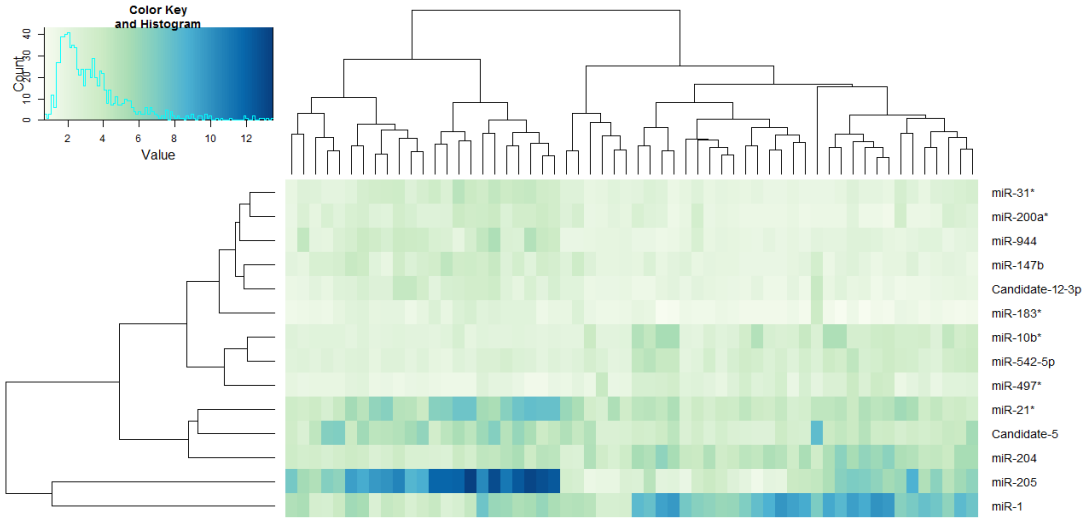

Figure 4.39. Heatmap plot for the selected miRNAs in cervical dataset

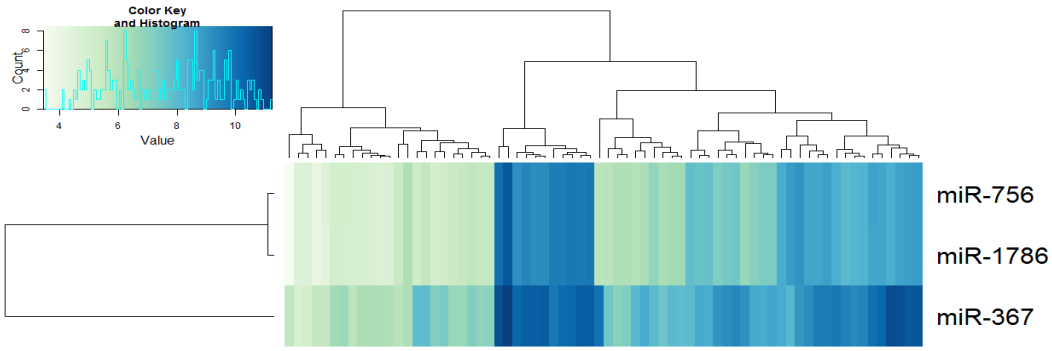

Figure 4.40. Heatmap plot for the selected miRNAs in Alzheimer dataset

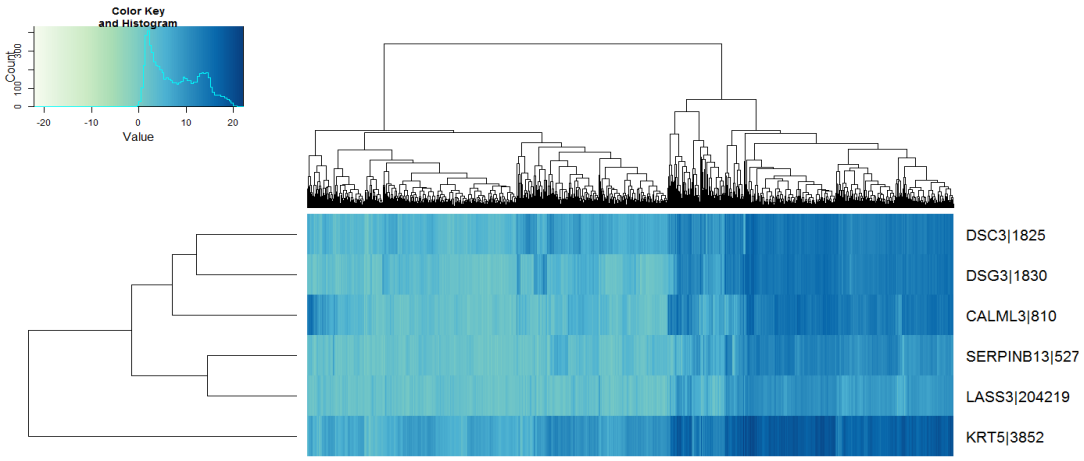

Figure 4.41. Heatmap plot for the selected genes in lung cancer dataset

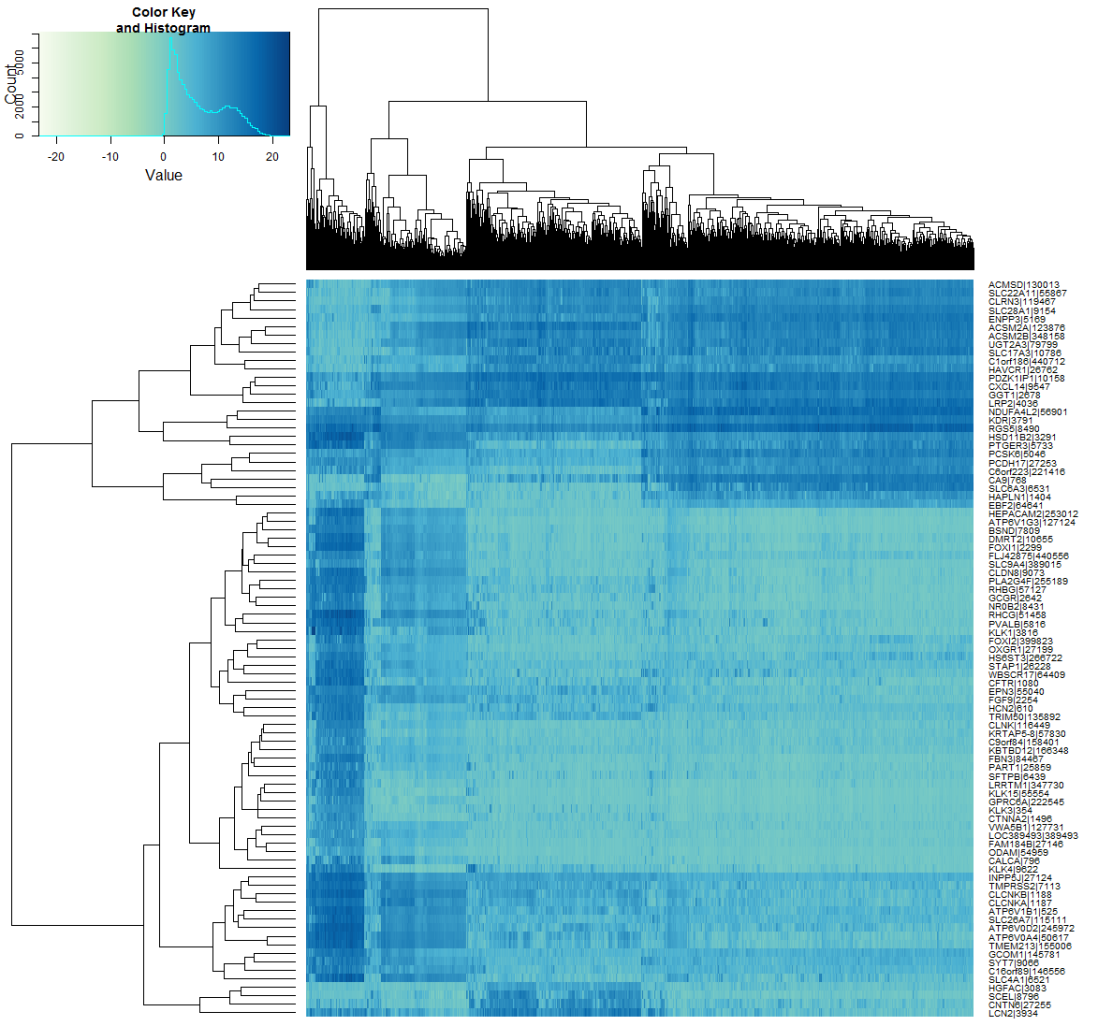

Figure 4.42. Heatmap plot for the selected genes in renal cell cancer dataset

These plots display the expression levels of corresponding features (miRNAs or genes) for each sample in pixels. Horizontal rows correspond to features, while the vertical columns correspond to samples. The colors describe the amount of expression from white (large negative) to dark blue (large positive). Hierarchical clustering algorithm is applied to cluster both samples and features within each other.

5. DISCUSSION

In this study, we presented a sparse classifier voomNSC for classification of RNA-Seq data. We successfully coupled the voom method and the NSC classifier together with using weighted statistics, thus extended voom method for classification studies and made NSC algorithm available for RNA-Seq data. We also proposed two non-sparse classifiers, which are the extensions of DLDA and DQDA algorithms for RNA-Seq classification. Law et al. (1) entered voom method into differential expression analysis and gene set testing. The authors mentioned that using precision weights with appropriate statistical algorithms increase the power of used methods. We extended this method for classification analysis and obtained very accurate and sparse results.

We designed a comprehensive simulation study, also used four real miRNA and mRNA sequencing datasets to assess the performance of developed approaches and compare their performances with other classification algorithms. We obtained successful results in both simulated and real dataset. Particularly, voomNSC sparse classifier is able to find the minimal subset of genes in an RNA-Seq data and provides fast, accurate and sparse solutions for RNA-Seq classification.

We both compared our results with the count based RNA-Seq classifiers and the microarray based classifiers after rlog transformation. Count based classifiers are the only developed approaches in the literature. Using microarray based classifiers; we found the opportunity to see the effect of voom method on classification studies. We selected rlog transformation for microarray based classifiers, since it accounts for the differences in library size, also stabilizes the variances more correctly than simple logarithmic transformation. In simulation results, the provided precision weights of voom method led to obtain both accurate and sparser models than microarray based classifiers. $PLDA_2$ give compatible results with voomNSC classifiers in classification accuracy. However voomNSC provides sparser models, which is crucial for more simple, interpretable and lower variance models. In real datasets, the accuracy of the classifiers was comparable with each other. However, again the voomNSC classifiers provided the sparsest solutions.

Our approaches are mostly superior in providing sparser and compatibly accurate models to both PLDA and NBLDA, also to microarray versions of DLDA,

DQDA and NSC. $PLDA_2$ and voomNSC classifiers give comparable results in model accuracy. We believe that this superiority originates from the robustness of voom method. Voom method matches the observed and expected mean-variance relationships perfectly with each other. Instead of modeling the mean and variance relationship of the data, PLDA and NBLDA aim to specify the exact probability distribution of counts. We made use of the normal distribution whose mathematical theory is more tractable than the count distributions (1). Precision weights also provide advantages such as working with samples with different sequencing depths, or down-weighting the low quality samples.

Dispersion has a direct effect on PLDA classifier. The reason may be that the PLDA algorithm uses a Poisson model which assumes the mean and dispersion are equal each other. Applying a power transformation enhances its performance. Thus, we recommend users to perform PLDA classifier always with power transformation, since RNA-Seq data is mostly overdispersed, because of the presence of biological replicates in most datasets. Overdispersion has significant effect on this classifier and should be taken into account before building models. NBLDA classifier (5) converges to PLDA algorithm, when the dispersion parameter approximates to zero. This classifier performed well for overdispersed cervical datasets, however does not perform as well as $PLDA_2$ classifier or voomNSC classifier in other scenarios. This may be originated due to the absence of sparsity option of this classifier. We leave sparse NBLDA classification as a topic for further research.

In slightly overdispersed datasets, RF performs as well as the sparse classifiers. Moreover, this classifier performed very well in lung and renal cell cancer datasets. The reason may be arised from the bootstrap property of this algorithm. Likely to its microarray classification performance, SVM algorithm performed very accurate results in real datasets. Mukherjee et al. (91) mentions that this high accuracy arises from the strong mathematical background of SVM algorithm. The idea of margin overcomes the problem of overfitting and make SVM algorithm capable to work in high-dimensional settings. This is also true for RNA-Seq datasets, since rlog transformation makes this data hierarchically closer to microarrays.

Increasing the class number decreases the overall accuracy. This may arise from the decrease of assignment probability of a sample in this condition. Moreover,

we reach a conclusion that the effect of sample size and number of genes on misclassification errors is highly dependent on the dispersion parameter. Decreasing the number of genes and samples leads to an increase in the misclassification error, unless the data is overdispersed.

Normalization had little impact on voomDDA classifiers in simulation results. However, we observed that performing voomNSC algorithm without any normalization provides less sparse results in Alzheimer, lung and renal cell cancer datasets. This may be arised from the very large differences in library sizes (e.g. 2.6 to 100.6 million in Alzheimer dataset). In this case, deseq median ratio or TMM methods may be applied before model building to obtain sparser results. In other cases, all three voomNSC classifiers provided very similar results in both model accuracy and sparsity.

We also demonstrated the use of voomNSC algorithm in diagnostic biomarker discovery problems. In cervical dataset, voomNSC identified 14 miRNAs as biomarkers with misclassifying 2 out of 58 samples. Witten et al. (58) applied NSC algorithm in her study and identified 41 miRNAs. A total of 9 miRNAs detected by the voomNSC algorithm including miR-200a*, miR-204, miR-205, miR-1, miR-147b, miR-31*, miR-944, miR-21*, and miR-10b* were commonly identified with the authors (Figure 5.1). Moreover, voomNSC also used Candidate-5, Candidate-12-3p, miR-183*, miR-497*, and miR-542-5p in the prediction. Witten et al. (58) misclassified 4 out of 58 samples. Thus, our algorithm is superior to their procedure in both accuracy and sparsity for classifying this dataset.

Leidinger et al. (80) identified 12 miRNAs in classifying Alzheimer data and obtained 7% misclassification errors. In our study, we detected 3 miRNAs and obtained 18.6% misclassification error. Any of the selected miRNAs were common with each other. VoomNSC performed less accurate, however sparser solutions than their procedure.

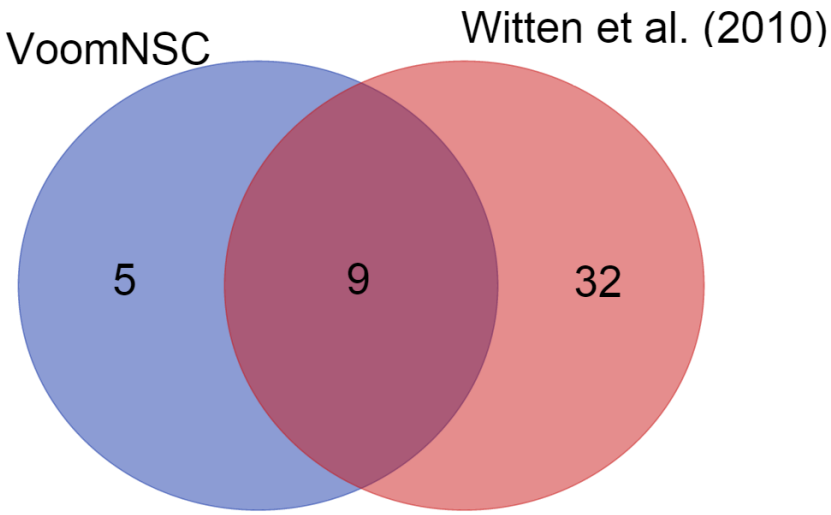

Figure 5.1. A Venn-diagram displaying the number of selected miRNAs from voomNSC algorithm and Witten et al. (58)

6. CONCLUSION

RNA-Seq has revolutionized as the premier technique in gene-expression profiling, all classification algorithms proposed for microarrays can be extended to RNA-Seq based gene-expression studies. RNA-Seq technique can detect novel transcripts, which may be a significant biomarker in an interested condition. This technique is less noisy than microarrays, thus may enhance the predictive performance of classification results. This study may contribute to other studies in proposing the voom extensions of powerful machine learning classifiers including support vector machines, random forests, etc. We also recommend extending this approach for other types of statistical analysis such as clustering analysis. These generalizations may allow users to analyze both microarray and RNA-Seq data with similar workflows and provide comparable results.

Proposed approaches can be used for all types of RNA-Seq based gene-expression classification studies such as cancer classification, development of RNA-Seq based diagnostic assays, identification of the types of species, separation of developmental differences, cellular responses against stressors, or diverse phenotypes, etc. However, they can easily be extended to other sequencing datasets including ChIP-seq, metagenome sequencing, Circularized Chromosome Conformation Capture (4C) sequencing, etc.

For most of the nonsparse classifiers used in this study, feature selection should be performed to obtain more accurate results. SVM and RF have the capability to deal with the high dimensional data. Sparse classifiers are able to detect the relevant subset of features, and performed better in the scenarios with high number of genes. We do not recommend a feature selection for nonsparse classifiers. A filtering at beginning may be useful to eliminate the noninformative genes.

Besides the prediction purpose, voomNSC classifier can be used just to identify the potential diagnostic biomarkers for an interested condition. In this way, a small subset of genes, that is relevant with the class conditions, can be detected. These genes can then be investigated for further analysis, such as discovering the other genes which have interactions with these genes. Shrunken differences of these selected genes may be related with specific class conditions.

We conclude that PLDA algorithm with power transformation and voomNSC classifiers may be the sparse methods of choice, if one aims to obtain accurate models for RNA-Seq classification. SVM and RF algorithms are the overall winners in nonsparse classifiers. When sparsity is the measure of interest, voomNSC classifiers should be the preferred classifiers. Along with its accurate and sparse performance, voomNSC method is fast and applicable to even very large types of RNA-Seq dataset.

For the applicability of the proposed approaches, we developed voomDDA, a user-friendly web-based platform. VoomDDA can be accessed from http://www.biosoft.hacettepe.edu.tr/voomDDA/ . Users can upload their raw data to this platform, and apply all necessary pre-processing, diagnostic biomarker discovery and classification steps online. In pre-processing step, users can filter the redundant genes from their data using near-zero filtering and variance filtering modules and normalize their data using deseq median ratio and TMM methods. For classification, they can select voomDLDA, voomDQDA or voomNSC methods. For sparse voomNSC method, the web-tool also displays the selected genes, and various plots to display their interactions.

REFERENCES

1. Law, C.W., Chen, Y., Shi, W. and Smyth, G.K. (2014). voom: Precision weights unlock linear model analysis tools for RNA-Seq read counts. *Genome Biology*; 15:R29.
2. Witten, D.M. (2011). Classification and clustering of sequencing data using a poisson model. *The Annals of Applied Statistics*; 5(4): 2493-518.
3. Zararsiz, G., Goksuluk, D., Korkmaz, S., Eldem, V., Goksuluk, I.P., Unver, T., et al. (2014). Classification of RNA-Seq Data via Bagging Support Vector Machines. *bioRxiv*; doi: **http://dx.doi.org/10.1101/007526** .
4. Díaz-Uriarte, R. and de Andrés S.A. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*; 7(3): doi:10.1186/1471-2105-7-3 .
5. Dong, K., Zhao, H., Wan, X., and Tong, T. (2015). NBLDA: Negative Binomial Linear Discriminant Analysis for RNA-Seq Data. *arXiv*:1501.06643 .
6. Tibshirani, R., Hastie, T., Narasimhan, B. and Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *PNAS*; 99(10): 6567–72.
7. Brown, M.P., Grundy, W.N., Lin, D., Cristianini, N., Sugnet, C.W., Furey, T.S., et al. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc Natl Acad Sci*; 97(1): 262-7.
8. Dudoit, S., Fridlyand, J. and Speed, T.P. (2002). Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data. *Journal of the American Statistical Association*; 97(457): 77-87.
9. Rapaport, F., Zinovyev, A., Dutreix, M., Barillot, E. and Vert, J.P. (2007). Classification of microarray data using gene networks. *BMC Bioinformatics*; 8(35): doi:10.1186/1471-2105-8-35 .

10. Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology*; 11(R106): doi:10.1186/gb-2010-11-10-r106 .
11. Cloonan, N., Forrest, A.R.R., Kolle, G., Gardiner, B.B.A., Faulkner, G.J., Brown, M.K., et al. (2008). Stem cell transcriptome profiling via massive-scale mRNA sequencing. *Nature Methods*; 5: 613–9.
12. Han, X., Wu, X., Chung, W.Y., Li, T., Nekrutenko, A., Altman, N.S. et al. (2009). Transcriptome of embryonic and neonatal mouse cortex by high-throughput RNA sequencing. *Proc Natl Acad Sci*; 1274, 106: 12741–6.
13. Parikh, A., Miranda, E.R., Katoh-Kurasawa, M., Fuller, D., Rot, G., Zagar, L., et al. (2010). Conserved developmental transcriptomes in evolutionarily divergent species. *Genome Biol*; 11: R35.
14. Perkins, T.T., Kingsley, R.A., Fookes, M.C., Gardner, P.P., James, K.D., Yu, L. et al. (2009). A strand-specific RNA-seq analysis of the transcriptome of the typhoid bacillus Salmonella Typhi. *PLoS Genet*; 5:e1000569.
15. Marioni, J.C., Mason, C.E., Mane, S.M., Stephens, M. and Gilad, Y. (2008). RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Res;* 18:1509-17.
16. Wang, L., Feng, Z., Wang, X., Wang, X. and Zhang, X. (2010). DEGseq: an R package for identifying differentially expressed genes from RNA-seq data. *Bioinformatics*; 26: 136-8.
17. Li, J., Witten, D.M., Johnstone, I.M. and Tibshirani, R. (2012). Normalization, testing, and false discovery rate estimation for RNA-sequencing data. *Biostatistics*; 13(3): 523-38. doi: 10.1093/biostatistics/kxr031.
18. Nagalakshmi, U., Wang, Z., Waern, K., Shou, C., Raha, D., Gerstein, M., et al. (2008). The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science*; 320(5881): 1344-9. doi: 10.1126/science.1158441 .

19. Robinson, M.D., McCarthy, D.J. and Smyth, G.K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*; 26(1): 139–40. doi: 10.1093/bioinformatics/btp616.
20. Ritchie, M.E., Phipson, B., Wu, D., Hu, Y., Law, C., Shi, W. et al. (2015). limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.*; 43(7): e47. doi: 10.1093/nar/gkv007 .
21. Yu, D., Huber, W., and Vitek, O. (2013). Shrinkage estimation of dispersion in Negative Binomial models for RNA-seq experiments with small sample size. *Bioinformatics*; 29(10): 1275–82. doi:10.1093/bioinformatics/btt143.
22. Zararsiz, G., Goksuluk, D., Korkmaz, S., Eldem, V., Goksuluk, I.P., Unver, T., et al. (2014). MLSeq: Machine learning interface for RNA-Seq data. R package version 1.2.0.
23. Love, M.I., Huber, W. and Anders, S. (2015). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*; 15(550). doi:10.1186/s13059-014-0550-8 .
24. Libbrecht, M.W., and Noble W.S. (2015). Machine learning applications in genetics and genomics. *Nature Review Genetics*; 16: 321–332.
25. Karlsen, F. (2004). The potential of RNA as a target for molecular diagnostics in cervical carcinoma screening. Proceedings of the 21st International Papillomavirus Conference: 20-26 February 2014 - Mexico City.
26. Wilkerson, M.D., Schallheim, J.M., Hayes, N., Roberts, P.J., Bastien, R.R.L., Mullins, M., et al. (2013). Prediction of lung cancer histological types by RT-qPCR gene expression in FFPE specimens. *The Journal of Molecular Diagnostics*; 15(4): 485-97.
27. van Rooij, E., Purcell, A.L., and Levin, A.A. (2012). Developing miRNA therapeutics. *Circulation Research*; 110: 496-507.
28. Cheng W., Zhang, Z., and Wang, J. (2013). Long noncoding RNAs: new players in prostate cancer. *Cancer Lett.* ; 339(1): 8-14.

29. Hui, P. (2012). Next Generation Sequencing: Chemistry, Technology and Applications. *Top Curr Chem*. DOI: 10.1007/128_2012_329.
30. Srinivasan, S., and Batra, J. (2014). Four Generations of Sequencing- Is it Ready for the Clinic Yet? *Next Generat Sequenc & Applic*; 1(1).
31. van Dijk, E.L., Auger, H., Jaszczyszyn, Y., and Thermes, C. (2014). *Trends in Genetics*; 30(9): 418-26.
32. Box, J.F. (1980). R. A. Fisher and the Design of Experiments,. 1922-1926. *The American Statistician;* 34(1): 1–7. doi:10.2307/2682986. JSTOR 2682986.
33. Aubert, J. (2012). Access: 03.06.2015, Statistical challenges in RNA-Seq data analysis. **http://www.cnrs.fr/inee/recherche/fichiers/EPEGE/Communications/Julie_AUBERT.pdf**
34. Williams, A.G., Thomas, S., Wyman, S.K., and Holloway, A.K. (2014). RNA-seq Data: Challenges in and Recommendations for Experimental Design and Analysis. *Curr Protoc Hum Genet*; 83. doi: 10.1002/0471142905.hg1113s83.
35. Ching, T., Huang, S., and Garmira, L.X. (2014). Power analysis and sample size estimation for RNA-Seq differential expression. *RNA;* 20: 1–13.
36. Hart, S.N., Therneau, T.M., Zhang, Y., Poland, G.A., and Kocher, J.P. (2013). Calculating Sample Size Estimates for RNA Sequencing Data. *Journal of Computational Biology;* 20(12): 970-8.
37. Guo, Y., Zhao, S., Li, C.I., Sheng, Q., and Shyr, Y. (2014). RNAseqPS: A Web Tool for Estimating Sample Size and Power for RNAseq Experiment. *Cancer Informatics*; 6: 1-5.
38. Busby, M.A., Stewart, C., Miller, C., Grzeda, K., and Marth, G. (2013). Scotty: A Web Tool For Designing RNA-Seq Experiments to Measure Differential Gene Expression. *Bioinformatics*; 10.1093/bioinformatics/btt015.

39. Andrews S. (2010). Access: 03.06.2015, FastQC: a quality control tool for high throughput sequence data, **http://www.bioinformatics.babraham.ac.uk/projects/fastqc** .
40. Anders, S., Pyl, P.T., and Huber, W. (2015) HTSeq - a Python framework to work with high-throughput sequencing data. *Bioinformatics*; 31(2):166-9.
41. Morgan, M., Anders, S., Lawrence, M., Aboyoun, P., Pages, H., and Gentleman, R. (2009). ShortRead: a bioconductor package for input, quality assessment and exploration of high-throughput sequence data. *Bioinformatics*; 25: 2607-8.
42. Schmieder, R. and Edwards, R. (2011). Quality control and preprocessing of metagenomic datasets. *Bioinformatics*; 27: 863-4.
43. Hannon Lab. (2015). Access: 03.06.2015, **http://hannonlab.cshl.edu/fastx_toolkit/index.html**
44. Shrestha, R.K., Lubinsky, B., Bansode, V.B., Moinz, M.B.J., McCormack, G.P., and Travers, S.A. (2014). QTrim: a novel tool for the quality trimming of sequence reads generated using the Roche/454 sequencing platform. *BMC Bioinformatics*; 15(33). doi:10.1186/1471-2105-15-33 .
45. Li, H., and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler Transform. *Bioinformatics*; 25: 1754-60.
46. Langmead, B., and Salzberg, S.L. (2012). Fast gapped-read alignment with Bowtie 2. *Nat Methods*; 9(4): 357–9.
47. Kim, D., Pertea, D., Trapnell, C., Pimentel, H., Kelley, R., and Salzberg, S.L. (2013). TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biology*; 14(4): R36. doi: 10.1186/gb-2013-14-4-r36.
48. Wang, K., Sing, D., Zeng, Z., Coleman, S.J., Huang, Y., Savich, G.L., et al. (2010). MapSplice: Accurate mapping of RNA-seq reads for splice junction discovery. *Nucleic Acids Research*.; 38(18): e178. doi: 10.1093/nar/gkq622.

49. Dobin, A., Davis, J.A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., et al. (2012). STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*; 29(1): 15-21. 10.1093/bioinformatics/bts635.
50. Haas, B.J., Papanicolaou, A., Yassour, M., Grabherr, M., Blood, P.D., Bowden, J., et al. (2013). De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. *Nat. Protoc.;* 8(8): 1494–512.
51. Schulz, M.H., Zerbino, D.R., Vingron, M., and Birney, E. (2012). Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels. *Bioinformatics*; 28(8), 1086–92.
52. Xie, Y., Wu, G., Tang, J., Luo, R., Patterson, J., Liu, S., et al. (2014). SOAPdenovo-Trans: de novo transcriptome assembly with short RNASeq reads. *Bioinformatics*; 30(12), 1660–6.
53. Robertson, G., Schein, J., Chiu, R., Corbett, R., Field, M., Jackman, S.D., et al. (2010). De novo assembly and analysis of RNA-seq data. *Nat. Methods*; 7(11), 909–12.
54. Li, B., Dewey, C.N. (2011). RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics*; 12(323). doi:10.1186/1471-2105-12-323 .
55. Davidson, N.M., and Oshlack, A. (2014). Corset: enabling differential gene expression analysis for *de novo* assembled transcriptomes. *Genome Biology*; (15)410. doi:10.1186/s13059-014-0410-6 .
56. Liao, Y., Smyth, G.K., and Shi, W. (2013). featureCounts: an efficient general-purpose program for assigning sequence reads to genomic features. *Bioinformatics*. doi: 10.1093/bioinformatics/btt656.
57. Quinlan, A.R., and Hall, I.M. (2010). BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*; 26(6): 841-2.
58. Witten, D., Tibshirani, R., Gu, S.G,, Fire, A. and Lui W.O. (2010). Ultra-high throughput sequencing-based small RNA discovery and discrete statistical biomarker analysis in a collection of cervical tumours and matched controls. *BMC Biology*; 8(58). doi:10.1186/1741-7007-8-58 .

59. Fang, Z., Martin, J., and Wang, Z. (2012). Statistical methods for identifying differentially expressed genes in RNA-Seq experiments. *Cell & Bioscience*; 2(26).
60. Bullard, J.H., Purdom,E., Hansen, K.D. and Dudoit, S. (2010). Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments, *BMC Bioinformatics*; 11(94).
61. Robinson, M.D., and Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*; 11(R25).
62. Agresti, A. (2002). *Categorical Data Analysis*. NJ: Wiley.
63. Tan, K.M., Petersen, A. and Witten, D.M. (2015). Classification of RNA-seq data. Datta, S. and Nettleton, E. (Ed.) *Statistical Analysis of Next Generation Sequencing Data* (pp. 219–46), Springer.
64. Robinson, M.D., and Smyth, G.K. (2008). Small-sample estimation of negative binomial dispersion, with applications to SAGE data. *Biostatistics*; 9(2): 321-32.
65. Oshlack, A., and Wakefield, M.J. (2009). Transcript length bias in RNA-seq data confounds systems biology. *Biol Direct*; 4(14). doi: 10.1186/1745-6150-4-14 .
66. Casella, G., and Berger, R.L. (2002). *Statistical Inference Pacific Grove*, CA: Duxbury Press.
67. Finotella F., and Di Camillo, B. (2015). Measuring differential expression with RNA-seq: challenges and strategies for data analysis. *Brief Funct Genomics*;14(2):130-42.
68. Huber, W. (2011). Differential expression for RNA-Seq data. Access: 03.06.2015, **http://www.bioconductor.org/help/course-materials/2011/CSAMA/Wednesday/Morning% 20Talks/110629-brixen-deseq-huber.pdf**
69. Tibshirani, R. (1988). Estimating transformations for regression via additivity and variance stabilization. *Journal of the American Statistical Association*; 83: 394–405.

70. Zwiener, I., Frisch, B., and Binder, H. (2014). Transforming RNA-Seq Data to Improve the Performance of Prognostic Gene Signatures. *Plos One*; 9(1): e85150. doi:10.1371/journal.pone.0085150 .
71. Pang, H. and Tong, T. (2012). Recent Advances in Discriminant Analysis for High-dimensional Data Classification. *J. Biomet. Biostat.*; 3(e106). doi: 10.4172/2155-6180.1000e106 .
72. Alon, U., Barkai, M., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., et al. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *PNAS*; 96(12): 6745–50.
73. Cleveland, W.S. (1979). Robust locally weighted regression and smoothing scatter plots. *J Am Stat Assoc*; 74: 829-36.
74. Tusher, V.G., Tibshirani, T., and Chu, G. (2000). Significance analysis of microarrays applied to the ionizing radiation response. *PNAS*; 98(9): 5116–21.
75. Witten, D. (2013). PoiClaClu: Classification and clustering of sequencing data based on a Poisson model. R package version 1.0.2. **http://CRAN.R-project.org/package=PoiClaClu**
76. Anscombe, F.J. (1948). The transformation of Poisson, binomial, and negative-binomial data. *Biometrika*; 35: 246-54.
77. Hastie, T., Tibshirani, R., Narasimhan, B., and Chu, G. (2014). pamr: Pam: prediction analysis for microarrays. R package version 1.55. **http://CRAN.R-project.org/package=pamr**
78. Maechler, M. (2015). sfsmisc: Utilities from Seminar fuer Statistik ETH Zurich. R package version 1.0-27. **http://CRAN.R-project.org/package=sfsmisc**
79. Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. *Journal of Statistical Software*; 28(5).
80. Leidinger, P., Backes, C., Deutscher, S., Schmitt, K., Mueller, S.C., Frese, K., et al. (2013). A blood based 12-miRNA signature of Alzheimer disease patients. *Genome Biology*; 14:R78. doi:10.1186/gb-2013-14-7-r78

81. Saleem, M., Shanmukha, A., Ngonga Ngomo, A.C., Almeida, J.S., Decker, H.F. and Deus, H.F. (2013). Linked cancer genome atlas database. I-SEMANTICS '13 - Proceedings of the 9th International Conference on Semantic Systems: 04-06 September 2013 - Graz (p. 129-134).
82. Goyal, R., Gersbach, E., Yang, X.J., and Rohan, S.M. (2013). Differential Diagnosis of Renal Tumors with Clear Cytoplasm. Clinical Relevance of Renal Tumor Subclassification in the Era of Targeted Therapies and Personalized Medicine. *Arch Patol Lab Med*; 137: 467-80.
83. R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. **http://www.R-project.org**/.
84. RStudio (2012). RStudio: Integrated development environment for R (Version 0.98.1103). Boston, MA. Retrieved May 20, 2012. **http://www.rstudio.org/** .
85. Revolution Analytics and Steve Weston (2014). doSNOW: Foreach parallel adaptor for the snow package. R package version 1.0.12. **http://CRAN.R-project.org/package=doSNOW .**
86. Revolution Analytics and Steve Weston (2014). doParallel: Foreach parallel adaptor for the parallel package. R package version 1.0.8. **http://CRAN.R-project.org/package=doParallel .**
87. Revolution Analytics (2014). doMC: Foreach parallel adaptor for the multicore package. R package version 1.3.3. **http://CRAN.R-project.org/package=doMC .**
88. Revolution Analytics and Steve Weston (2014). foreach: Foreach looping construct for R. R package version 1.4.2. **http://CRAN.R-project.org/package=foreach .**
89. Eddelbuettel, D., Lucas, A., Tuszynski, J., Bengtsson, H., Urbanek, S. and Frasca, M. et al. (2014). digest: Create Cryptographic Hash Digests of R Objects. R package version 0.6.8. **http://CRAN.R-project.org/package=digest .**

90. Chang, W., Cheng, J., Allaire, J.J., Xie, Y., and McPherson, J. (2015). shiny: Web Application Framework for R. R package version 0.11.1. **http://CRAN.R-project.org/package=shiny** .
91. Mukherjee, S., Tamayo, P., Mesirov, J., Slonim, D., Verri, A., and Poggio, T. (1999). Support vector machine classification of microarray data. *Technical Report CBCL Paper 182/AI Memo 1676 MIT*.

SUPPLEMENTARY FILE - 1:

USER GUIDE of VOOMDDA WEB APPLICATION

To provide the applicability of the developed approaches, a user-friendly web application is proposed. VoomDDA is an interactive platform, which can be accessed from http://www.biosoft.hacettepe.edu.tr/voomDDA/ . VoomDDA includes the sparse voomNSC, non-sparse voomDLDA and voomDQDA algorithms accompanied with several interactive plots (Figure S1.1).

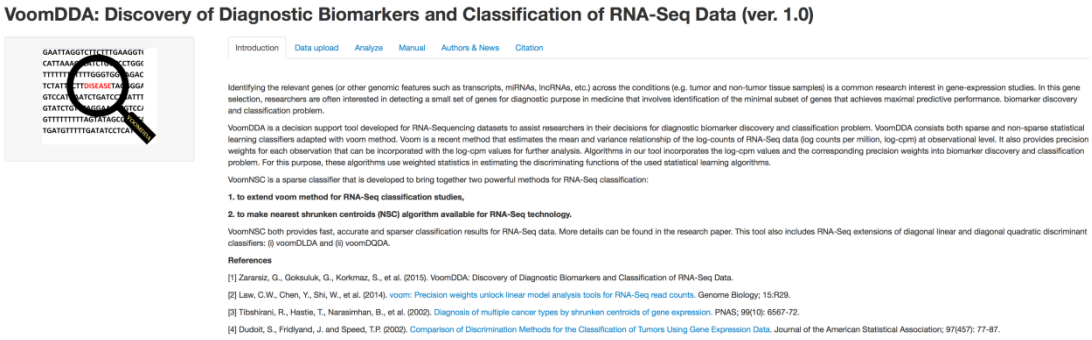

Figure S1.1. Introduction page of VoomDDA web-tool

To start analysis, users are required to upload their data as described in the 'Data upload' tab. Two example datasets are provided for researchers to test the tool and prepare their data in a suitable format. After uploading the data, the data will appear on the screen in a matrix form (Figure S1.2). Next, users can click on the 'Analyze Tab', pre-process their data and perform the classification analysis. VoomDDA consists two filtering (near-zero and variance filtering), and three normalization (none, deseq median ratio, TMM) modules. A data summary, training summary and predictions will appear in the screen based on selected analysis modules (Figure S1.3). If voomNSC is the selected classification model, selected subset of genes and a heatmap plot will be provided, as well in the same screen (Figure S1.4). A detailed tutorial is given in the Manual tab of the application.

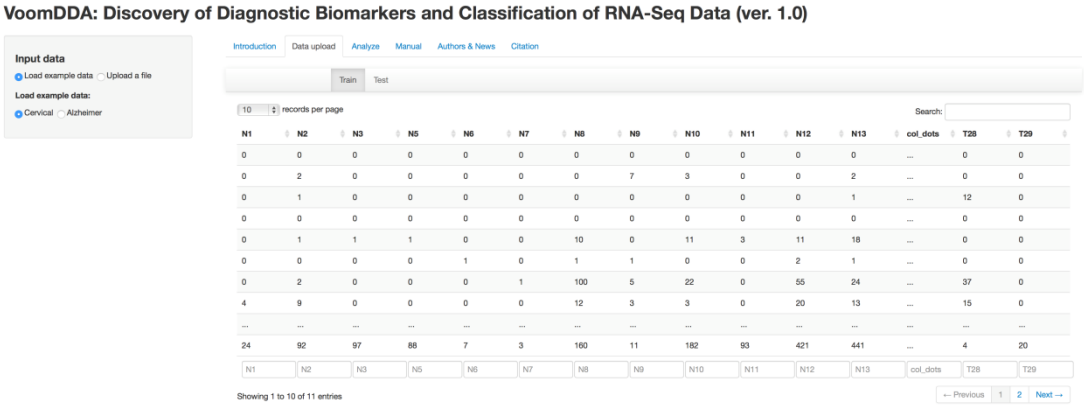

Figure S1.2. Data upload screen of VoomDDA web application

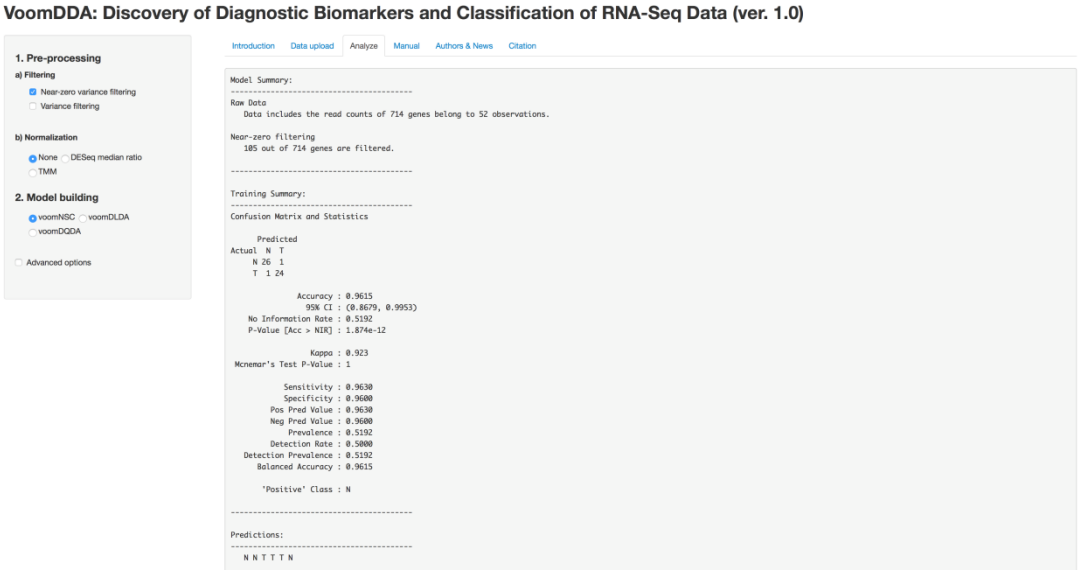

Figure S1.3. Analyze module of VoomDDA web application. Users can select the appropriate pre-processing method and the classifier on the left side and obtain data summary, training summary and predictions on the right side.

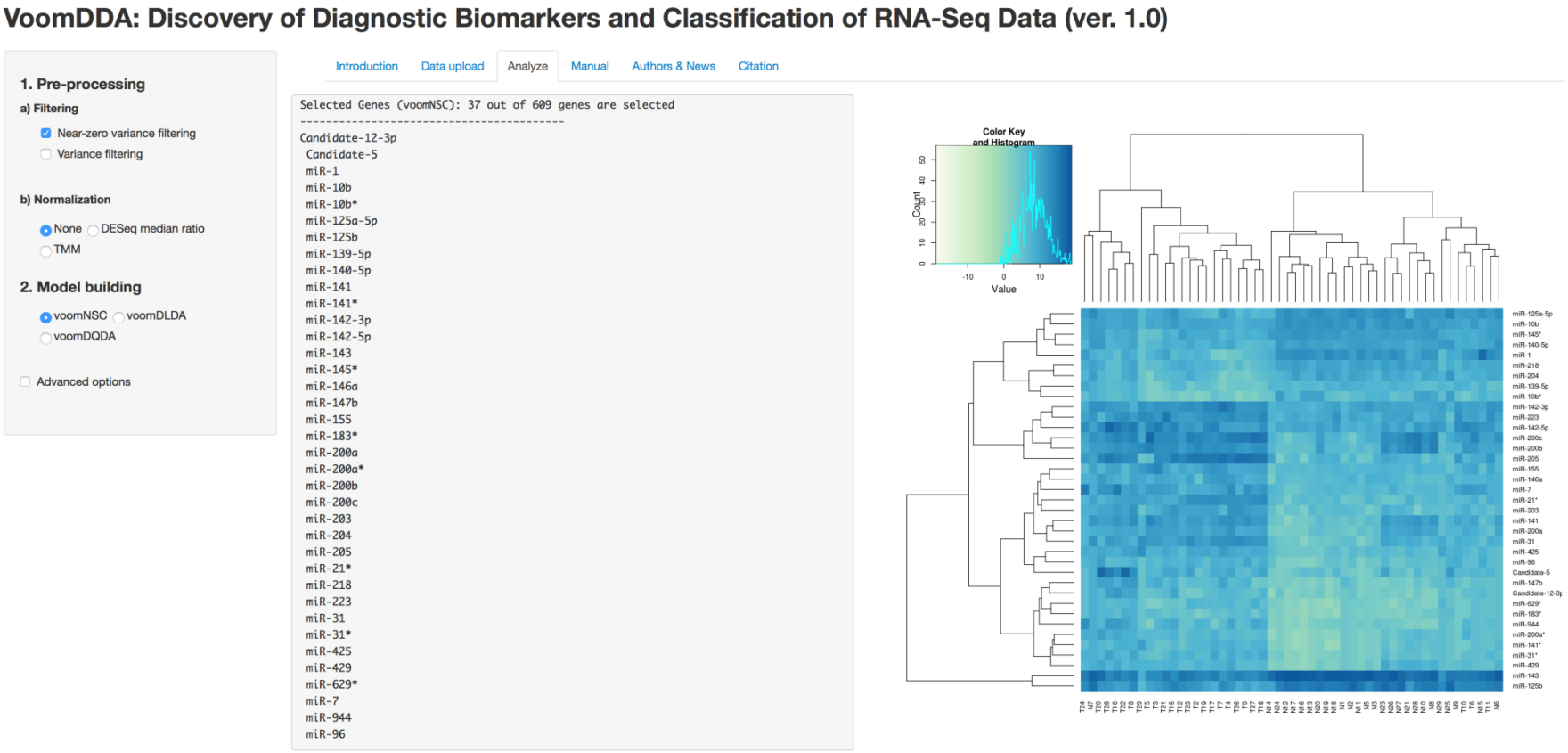

Figure S1.4. Selected genes and a heatmap plot for voomNSC algorithm