



---

Max Planck Institute for Molecular Genetics  
Computational Diagnostics Group @ Dept. Vingron  
Ihnestrasse 63-73, D-14195 Berlin, Germany  
<http://compdiag.molgen.mpg.de/>

---



## User's guide to the Package *dcoex* 0.9

Dennis Kostka<sup>1</sup>, Dominic Fabian<sup>2</sup>, and Rainer Spang

Max Planck Institute for Molecular Genetics  
Computational Diagnostics  
Department for Computation Molecular Biology  
Ihnestr. 63-73, D-14195 Berlin, Germany

Technical Report  
Nr. 2006/01

---

<sup>1</sup>dennis.kostka@molgen.mpg.de

<sup>2</sup>dominic.fabian@molgen.mpg.de

## Abstract

This is the vignette of the package *dcoex*. Motivation: Standard analysis routines for microarray data aim at differentially expressed genes. In this package an algorithm for finding sets of differentially co-expressed genes is implemented.

# Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Introduction</b>                                     | <b>2</b> |
| <b>2</b> | <b>Finding groups of differentially expressed genes</b> | <b>3</b> |
| 2.1      | Basic functionality . . . . .                           | 3        |
| 2.2      | Tuning the set size . . . . .                           | 3        |
| 2.3      | Starting point heuristic . . . . .                      | 4        |

# Chapter 1

## Introduction

For detection and interpretation of molecular disease mechanisms it is of great interest to identify the genes involved. Common Microarray analysis strategies look for up- or down-regulated genes. Not all changes in gene expression are detectable by this approach. In contrast to differential expression, differential co-expression aims to detect groups of genes with changes in their covariance structure. This cannot be done in a gene-wise fashion, one (in principle) needs to take into account all possible subsets of genes. To screen this large number of candidate groups effectively, an additive model for scoring differential co-expression is employed. `dcoex` implements an algorithm for finding differentially co-expressed groups.

## Chapter 2

# Finding groups of differentially expressed genes

### 2.1 Basic functionality

```
> set.seed(12342324)
> library(dcoex)

R Package to provide an algorithm to find
groups of differentially co-expressed genes

> data("normal", "bcrabl", package = "dcoex")
> normal = normal[sample(1:dim(normal)[1], 500), ]
> bcrabl = bcrabl[sample(1:dim(normal)[1], 500), ]
> res <- dcoex(bcrabl, normal)

> paste(res$dcoex.genes)

[1] "69" "84" "98" "131" "170" "177" "212" "224" "252" "260" "286" "348"
[13] "369" "493"
```

### 2.2 Tuning the set size

```
> for (al in c(0.1, 0.6, 0.9)) {
+   res = dcoex(normal, bcrabl, NULL, al)
+   cat(paste("alpha: ", al, "\t#genes: ", length(res$dcoex.genes),
+     "\ttscore: ", round(res$residual, 4), "\n"), sep = "")
+ }
```

|        |     |         |    |        |        |
|--------|-----|---------|----|--------|--------|
| alpha: | 0.1 | #genes: | 49 | score: | 0.3482 |
| alpha: | 0.6 | #genes: | 12 | score: | 0.1751 |
| alpha: | 0.9 | #genes: | 6  | score: | 0.1318 |

## 2.3 Starting point heuristic

```
> for (al in c(0.1, 0.6, 0.9)) {
+   start = dcoex.kmeans(normal, , subsample = 500)
+   res = dcoex(normal, bcrabl, as.matrix(as.numeric(start)),
+     al)
+   cat(paste("alpha: ", al, "\t#genes: ", length(res$dcoex.genes),
+     "\tscore: ", round(res$residual, 4), "\n"), sep = "")
+ }
```

|        |     |         |    |        |        |
|--------|-----|---------|----|--------|--------|
| alpha: | 0.1 | #genes: | 49 | score: | 0.3482 |
| alpha: | 0.6 | #genes: | 12 | score: | 0.1751 |
| alpha: | 0.9 | #genes: | 2  | score: | 0.0607 |