

JIVE: Joint and Individual Vairation Explained

Stephen Coleman

13/11/2019

Summary

JIVE (Lock et al. 2013) produces a decomposition of multiple datasets in three terms: a low-rank approximation of variation for the integrated analysis, low-rank approximations of variation for the individual datasets and residual noise. It can be thought of as a flavour of factor analysis model, similarly to iCluster.

The authors believe that there will exist some **joint structure** in the data that is common to all data, but that there will also be **individual structure** unique to each dataset. The individual structure may be artifacts of the process used to generate the data or of biological interest. This dataset-level variation can interfere with finding global signal, just as joint structure can obscure important signal that is individual to a data type.

In the authors own words:

“Analysis of individual structure provides a way to identify potentially useful information that exists in one data type, but not others. Accounting for individual structure also allows for more accurate estimation of what is common between data types.” Lock et al. (2013)

Selling points of JIVE according to the original paper:

- may be used regardless of whether the dimension of a dataset exceeds the sample size;
- is applicable to datasets with more than two data types; and
- has a simple algebraic interpretation.

The authors also state that JIVE can identify joint structure not found by existing methods, but they would say that.

Basically, this method describes the data as a sum of global factors, local factors and local noise. If one thinks of it in this way (i.e. related to factor analysis), JIVE avoids computation and considers only the matrix that would normally be considered the product of the loadings matrix and the factors matrix. It is a dimension reduction method that can help with clustering, but is not itself a clustering method (such as MOFA, another method we’ll describe someday).

Terminology

From here on out I will use my preferred terminology of dataset or context-specific features being described as **local** and joint features being described as **global**.

The model

Data transform

Consider L datasets denoted X_1, \dots, X_L for some $L \geq 2$. Each dataset **must** have a common number of columns representing n objects. Each dataset may have a unique number of rows (let p_l be the number of rows for the l th dataset X_l). Let:

$$p = \sum_{l=1}^L p_l$$

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_L \end{bmatrix}$$

In this case X is a $p \times n$ matrix. Lock et al. (2013) recommend therwmean-centreing the data by row, and scaleing the individual datasets by their individual variation to avoid dominance by anyone dataset. Let:

$$X_l^{scaled} = \frac{X_l}{\|X_l\|}$$

$$\|X_l\|^2 = \sum_{i,j} x_{lij}$$

$$\therefore \|X_l^{scaled}\| = 1 \forall l$$

Now:

$$X^{scaled} = \begin{bmatrix} X_1^{scaled} \\ \vdots \\ X_L^{scaled} \end{bmatrix}$$

and each dataset contributes equally to the total variation of the concatenated matrix, X .

Model

For this section let X_1, \dots, X_L be data matrices scaled as described above. Let J_1, \dots, J_L and A_1, \dots, A_L be the joint and individual structure matrices. Then the full model is given by:

$$X_1 = J_1 + A_1 + \epsilon_1$$

$$\vdots$$

$$X_L = J_L + A_L + \epsilon_L$$

where ϵ_l are $p_l \times n$ error matrices with independet entries and an expectation of 0. Let J be the stacked joint structure matrices. This model imposes rank constraints:

Let:

$$rank(J) = r < rank(X)$$

$$rank(A_l) = r_l < rank(X_l) \text{ for } l \in \{1, \dots, L\}$$

A further constrain that the joint and individual structure matrices are orthogonal is imposed:

$$JA_l^T = 0_{p \times p} \text{ for } l \in \{1, \dots, L\}$$

The purpose of this is to ensure that patterns in the samples that are informing the global structure are unrelated to those responsible for the local structure.

Estimating J and A_l

For fixed ranks r, r_1, \dots, r_L , the global and local structure captured in J, A_1, \dots, A_L is estimated by minimising the sum of squared error for the given ranks. Let:

$$R = \begin{bmatrix} R_1 \\ \vdots \\ R_L \end{bmatrix} = \begin{bmatrix} X_1 - J_1 - A_1 \\ \vdots \\ X_L - J_L - A_L \end{bmatrix}.$$

The minimisation process is achieved iteratively by repeating two steps until convergence is achieved:

- Given the current J , find A_1, \dots, A_L to minimise $\|R\|$; and
- Given the current A_1, \dots, A_L , find J to minimise $\|R\|$.

The joint structure J minimizing $\|R\|$ is equal to the first r terms in the singular value decomposition (SVD) of X with individual structure removed.

R example

I include some examples of using the `r.jive` package (O'Connell and Lock 2016). I apply it to the simulated data from the original paper, the example real data from the same and then the Yeast data case study from the original MDI paper (Kirk et al. 2012).

```
#!/usr/bin/env Rscript

# install.packages("r.jive")
library(r.jive)
library(magrittr)

# set a random seed
set.seed(1)

# Load data that were simulated as in Section 2.4 of Lock et al., 2013,
# with rank 1 joint structure, and rank 1 individual structure for each dataset
data(SimData)

# Data on breast cancer (BRCA) tumor samples from The Cancer Genome Atlas
data(BRCA_data)

# Read in yeast data from MDI paper
time_course_data <- read.csv("../Data/Yeast/Granovskaia_timecourse_normalised_reduced.csv",
  row.names = 1
)

harbison_data <- read.csv("../Data/Yeast/harbison_marina.csv", row.names = 1)
ppi_data <- read.csv("../Data/Yeast/ppi.csv", row.names = 1)

# Ensure object order is the same in all datasets
harbison_data_order <- match(row.names(harbison_data), row.names(time_course_data))
ppi_data_order <- match(row.names(ppi_data), row.names(time_course_data))

harbison_data_ordered <- harbison_data[harbison_data_order, ]
ppi_data_ordered <- ppi_data[ppi_data_order, ]
```

```
# Create an list of datasets for JIVE input
yeast_data <- list(
  t(time_course_data),
  t(harbison_data_ordered),
  t(ppi_data_ordered)
)
```

First we look at the simulated example from Lock et al. (2013) and the breast cancer tumour samples from the TGCNA (also an example in the original paper). We run JIVE using the permutation selection method proposed in the paper and visualise the results. The `jive` function does have another selection method based upon the BIC. The authors of the `r.jive` package claim that the accuracy of the permutation estimated ranks are generally better, but that the BIC is less computationally intensive. By default the ranks are selected via permutation, with row-orthogonality enforced between the joint and individual estimates and also between each individual estimate. Previously orthogonality was only enforced between joint and individual estimates, but the authors find that also enforcing the individual estimates to be orthogonal to each other improves convergence and robustness of the results to rank mispecification.

See the package vignette for details. The BRCNA example is taken from there and does recreate successfully.

I have run these examples separately and thus input the converged rank structure to reduce the time taken for this markdown document to compile.

```
# Using default method ("perm")
# sim_results <- jive(SimData)
sim_results <- jive(SimData, rankJ = 1, rankA = c(1, 1), method = "given")

## Running JIVE algorithm for ranks:
## joint rank: 1 , individual ranks: 1 1
## JIVE algorithm converged after 41 iterations.

summary(sim_results)

## $Method
## [1] "given"
##
## $Ranks
##      Source Rank
## [1,] "Joint" "1"
## [2,] "Data1" "1"
## [3,] "Data2" "1"
##
## $Variance
##             Data1 Data2
## Joint      0.360 0.120
## Individual 0.373 0.619
## Residual   0.267 0.261

# Using BIC rank selection
# BIC_result <- jive(SimData, method="bic")
BIC_result <- jive(SimData, rankJ = 1, rankA = c(1, 2), method="given")

## Running JIVE algorithm for ranks:
## joint rank: 1 , individual ranks: 1 2
## JIVE algorithm converged after 27 iterations.

summary(BIC_result)
```

```

## $Method
## [1] "given"
##
## $Ranks
##      Source  Rank
## [1,] "Joint" "1"
## [2,] "Data1" "1"
## [3,] "Data2" "2"
##
## $Variance
##          Data1 Data2
## Joint      0.222 0.209
## Individual 0.515 0.552
## Residual   0.264 0.239

# With some real data
# BRCA_results <- jive(Data)
BRCA_results <- jive(Data, rankJ = 2, rankA = c(28, 26, 25), method = "given")

```

```

## Running JIVE algorithm for ranks:
## joint rank: 2 , individual ranks: 28 26 25
## JIVE algorithm converged after 95 iterations.

summary(BRCA_results)

```

```

## $Method
## [1] "given"
##
## $Ranks
##      Source      Rank
## [1,] "Joint"     "2"
## [2,] "Expression" "28"
## [3,] "Methylation" "26"
## [4,] "miRNA"      "25"
##
## $Variance
##          Expression Methylation miRNA
## Joint      0.187      0.251 0.163
## Individual 0.423      0.219 0.298
## Residual   0.390      0.530 0.539

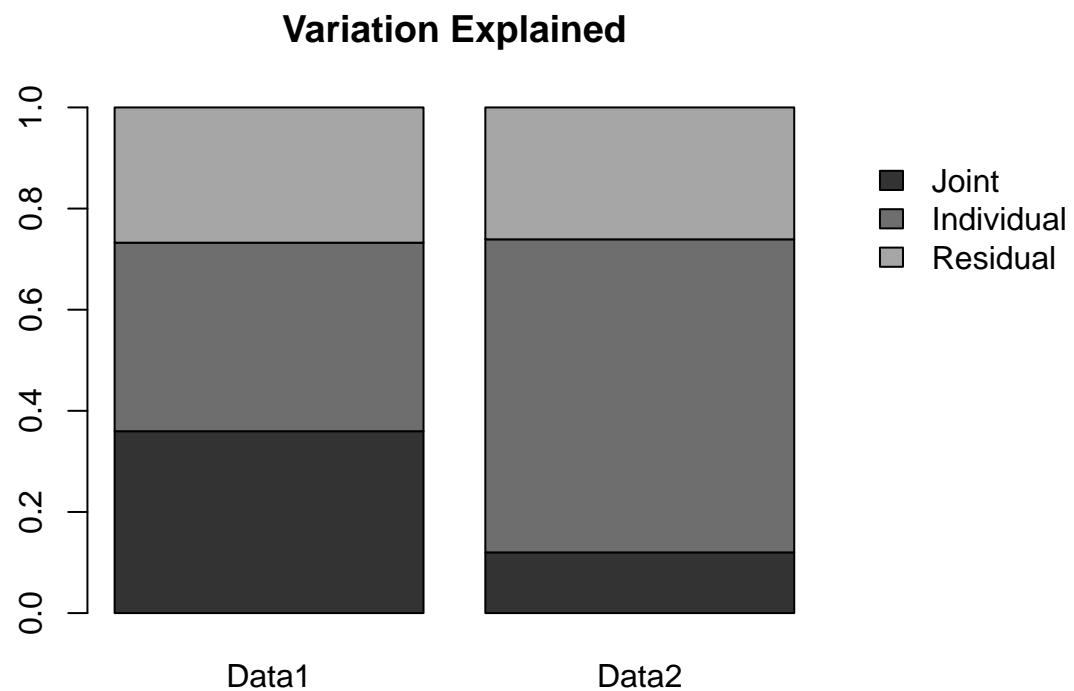
```

The results of JIVE can then be visualised. First for the simulation.

```

# Visualize results
showVarExplained(sim_results)

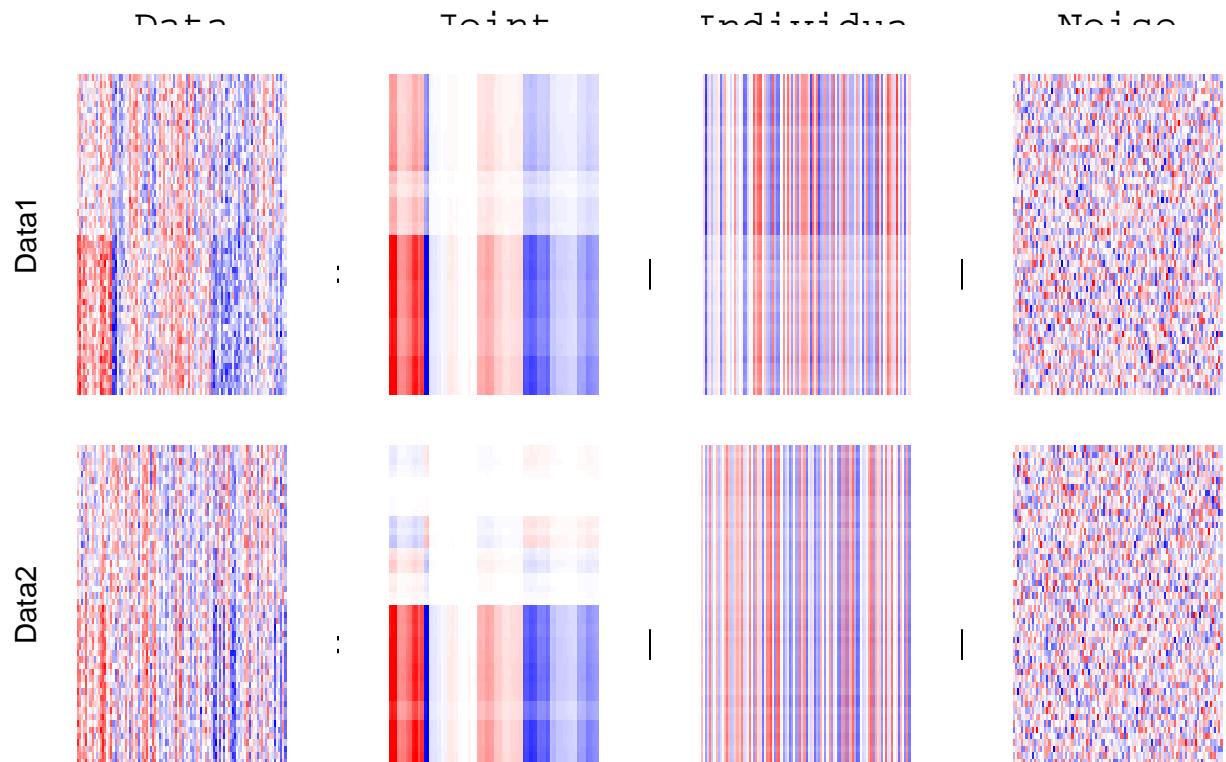
```



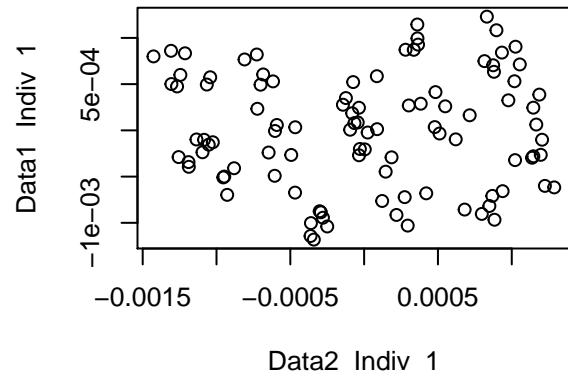
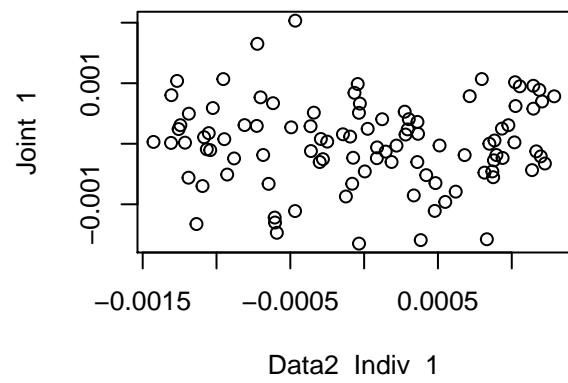
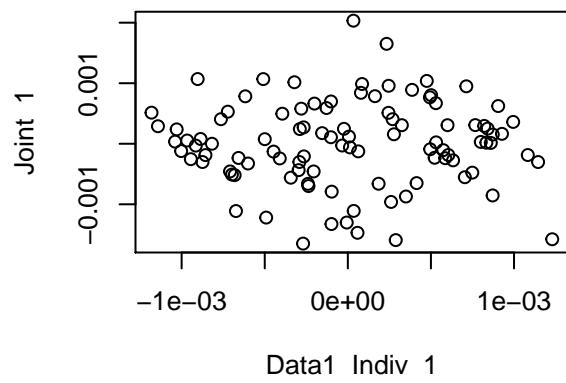
```
# showVarExplained is also called by the "jive" S3 class default plot method
```

```
# show heatmaps
showHeatmaps(sim_results)
```

22	2	13	5	16	1	19	10
23	3	14	6	17	8	20	11
24	4	15	7	18	9	21	12



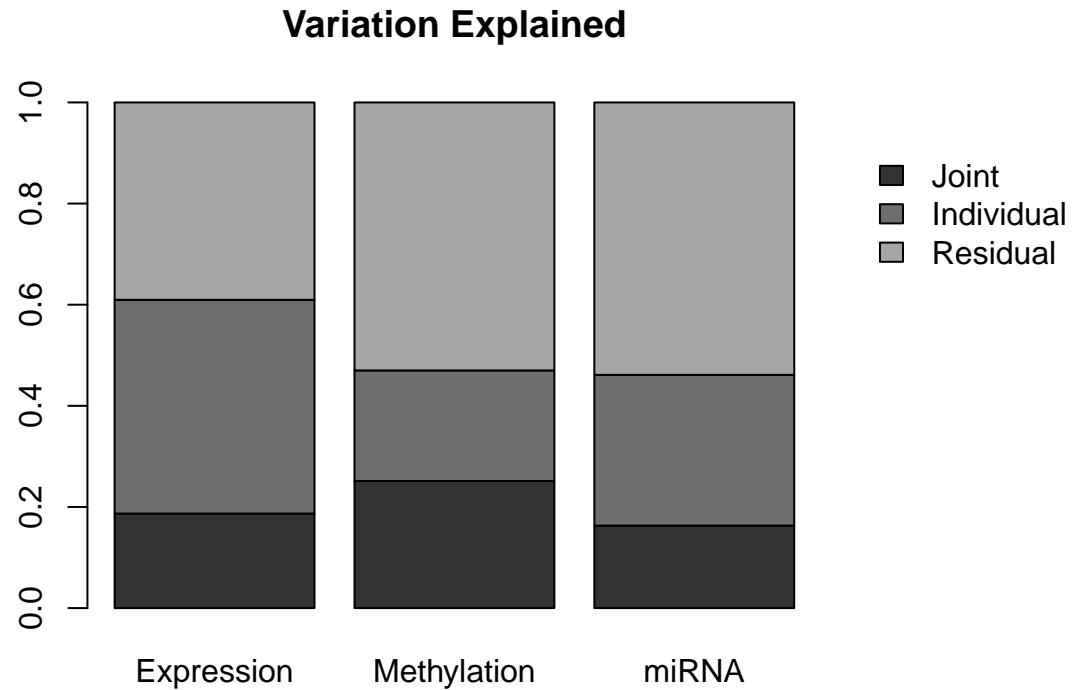
```
# show PCA plots  
showPCA(sim_results, 1, c(1, 1))
```



These can be seen to match the results shown in the paper.

Then a more detailed trawl through the results of the BRCA data analysis.

```
# Visualisation
# Display a barchart of the amount of variation explained by joint and
# individual estimates in each data source.
showVarExplained(BRCA_results)
```

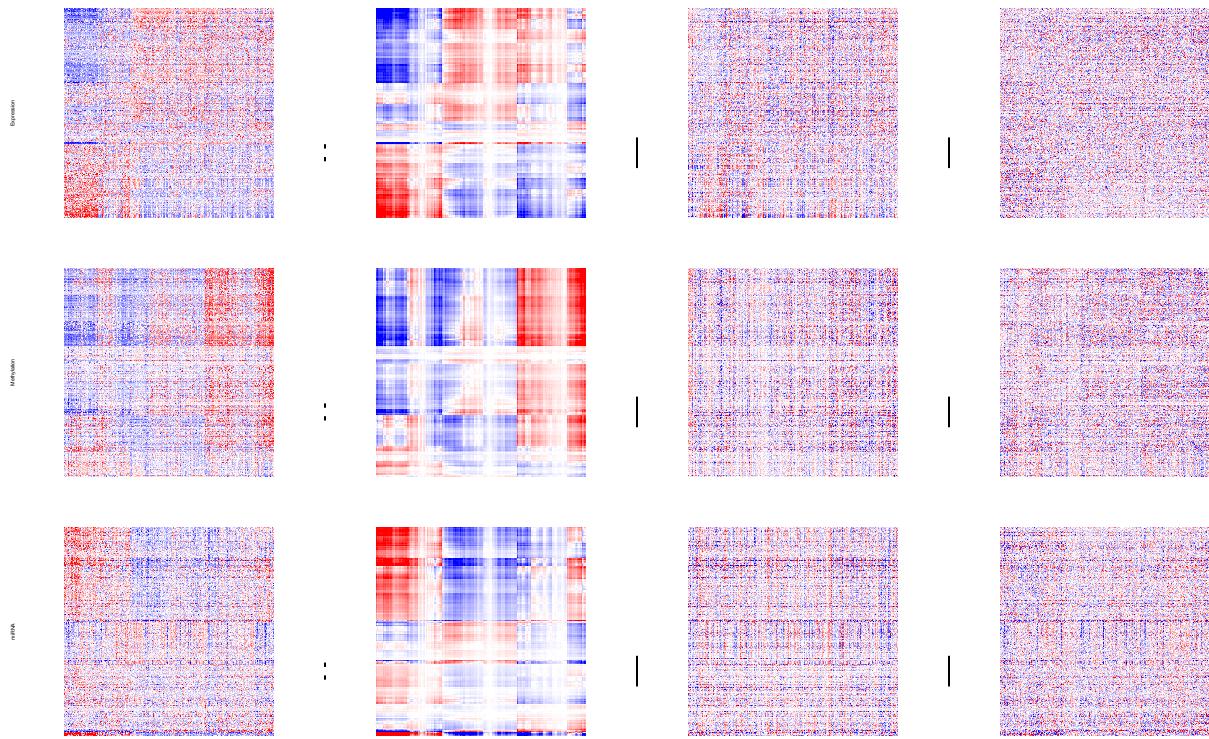


Note that the variation explained by joint structure is higher than that for individual structure for methylation data, despite the much higher rank of individual structure (rank 26 individual vs. rank 2 joint for methylation).

Display the JIVE estimates in the form of low-rank matrix approximations the rows and columns of all matrices are ordered by complete linkage clustering of the joint structure.

```
showHeatmaps(BRCA_results)
```

29	2	17	6	21	1	25	13
30	3	18	7	22	10	26	14
31	4	19	8	23	11	27	15
32	5	20	9	24	12	28	16

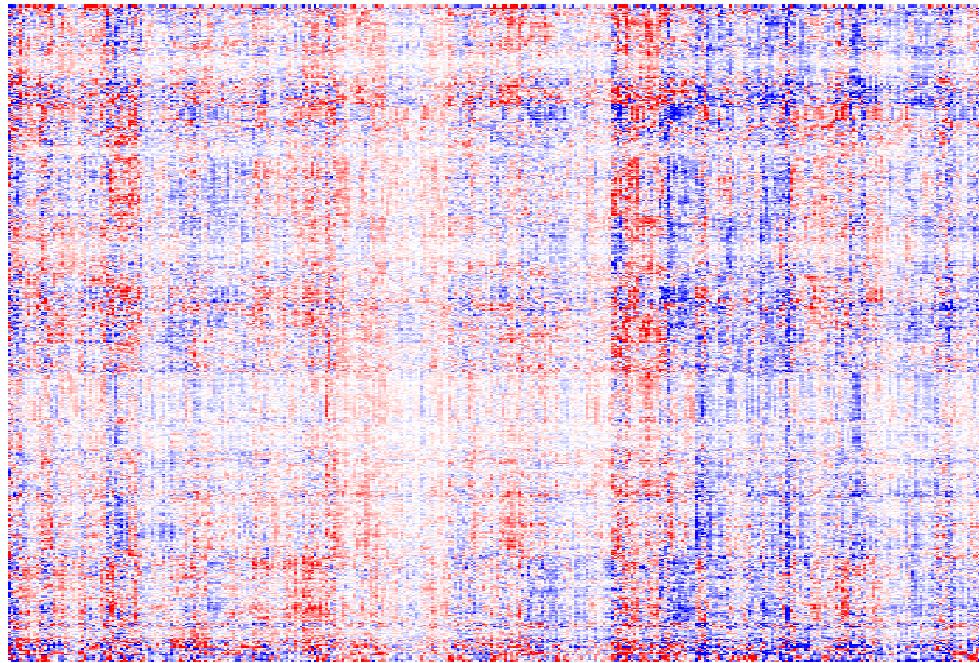


In addition, the `showHeatmaps` function includes options to specify how to order rows and columns, and which matrices to display. For example, we can order by the individual methylation structure (data source 2) and show only this heatmap.

```
showHeatmaps(BRCA_results, order_by = 2, show_all = FALSE)
```

3	1
4	2

Methylation



One factor appears to be a mean effect, distinguishing those samples with # relatively high methylation genome-wide from those with relatively low # methylation.

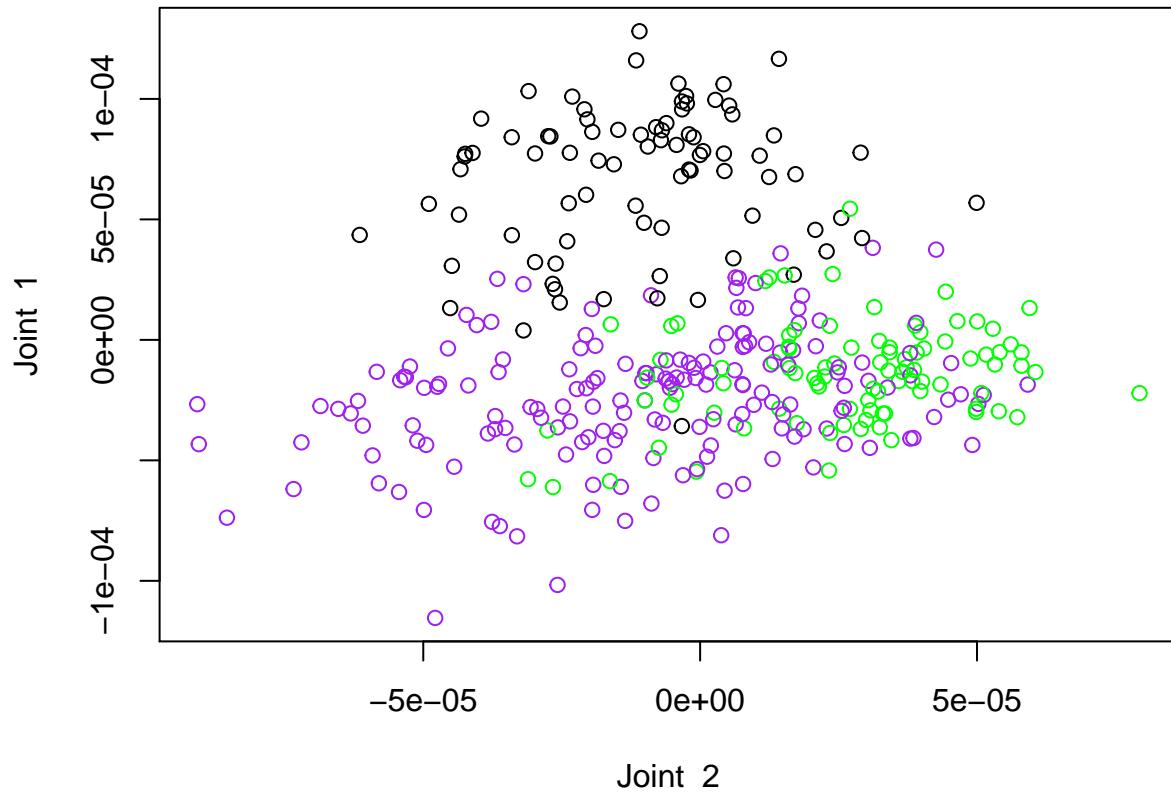
To further examine the biological relevance of the estimated joint structure,

we consider the “point cloud” view provided by the showPCA function. This

shows the patterns in the column space that maximize variability of joint or

individual structure, analogous to principal components.

```
Colors <- rep("black", 348)
Colors[clusts == 2] <- "green"
Colors[clusts == 3] <- "purple"
showPCA(BRCA_results, n_joint = 2, Colors = Colors)
```

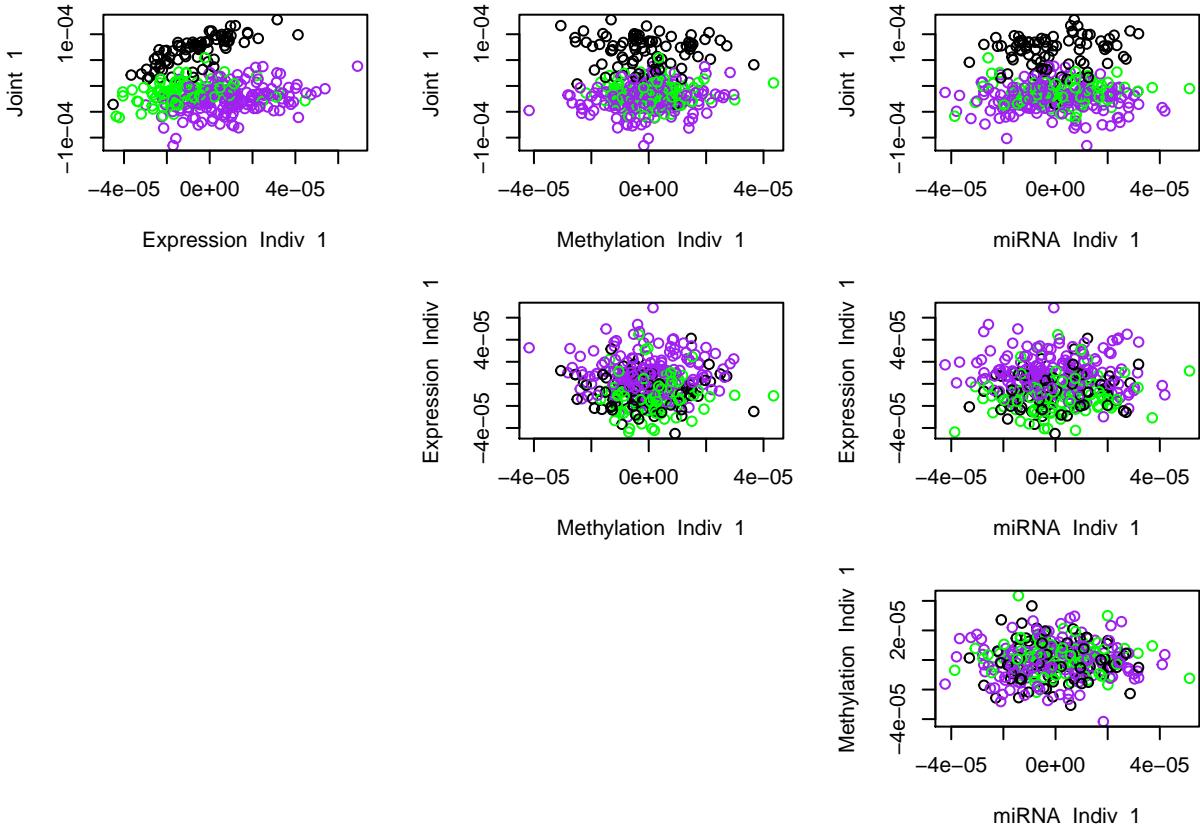


Joint 2

We see that the estimated joint corresponds well to the three previously identified clusters. Specifically, one pattern distinguishes Basal-like tumor samples (cluster 1) from other samples; among the remaining samples a subgroup of Luminal A tumors with a low fraction of genomic alteration and improved clinical prognosis (cluster 2) is distinguished.

For a broader view, we show the first component of joint structure with the first component of each of the three individual structures.

```
showPCA(BRCA_results, n_joint = 1, n_indiv = c(1, 1, 1), Colors = Colors)
```



A clustering effect is not apparent in the individual components shown, besides a slight distinction between clusters 2 and 3 in the expression individual component. This suggests that the coordinated expression, methylation, and miRNA activity in BRCA tumors is primarily driven by the cluster effects mentioned above, whereas the activity specific to each data source is driven by other biological components.

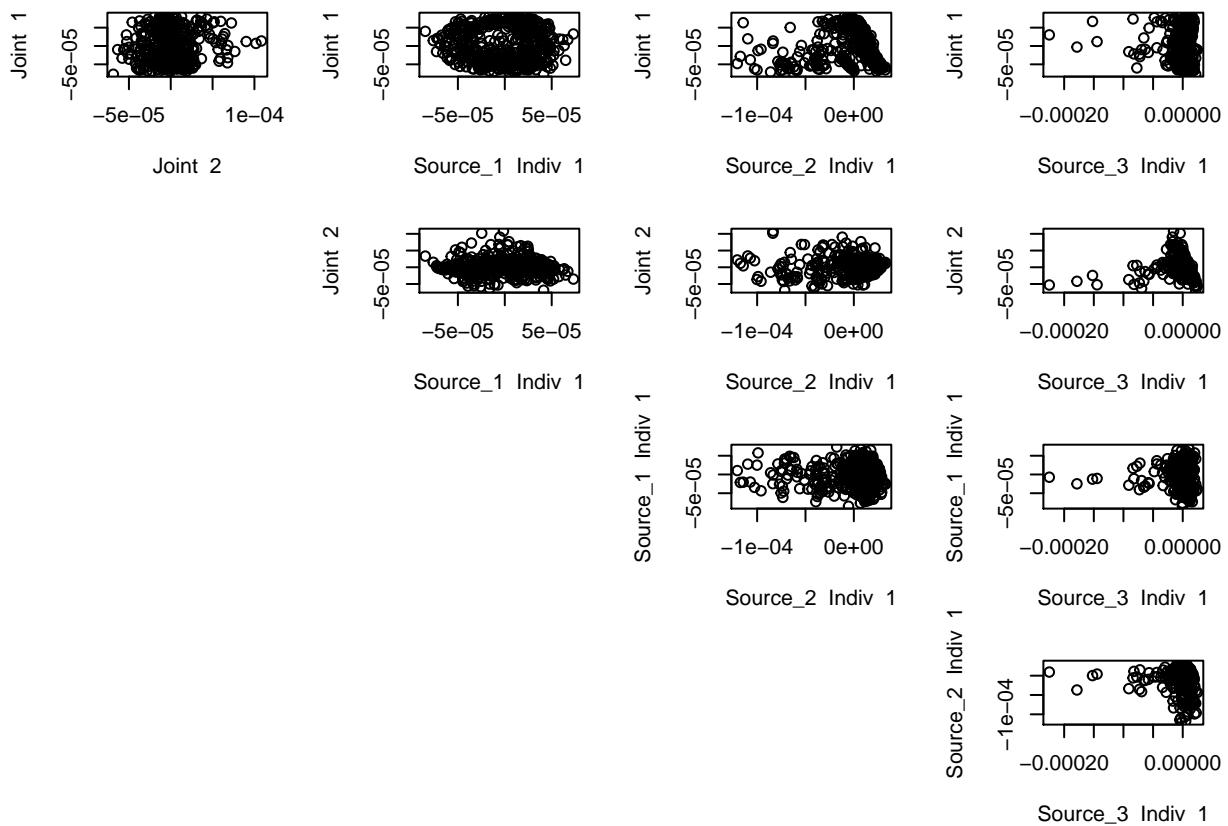
We look at JIVE applied to the Yeast gene datasets used in the original MDI paper.

I applied JIVE previously and know that the converged model uses the rank structure given below and thus use it to reduce run-time.

```
# Not run: original command to find the rank structure
# yeast_jive <- jive(yeast_data)

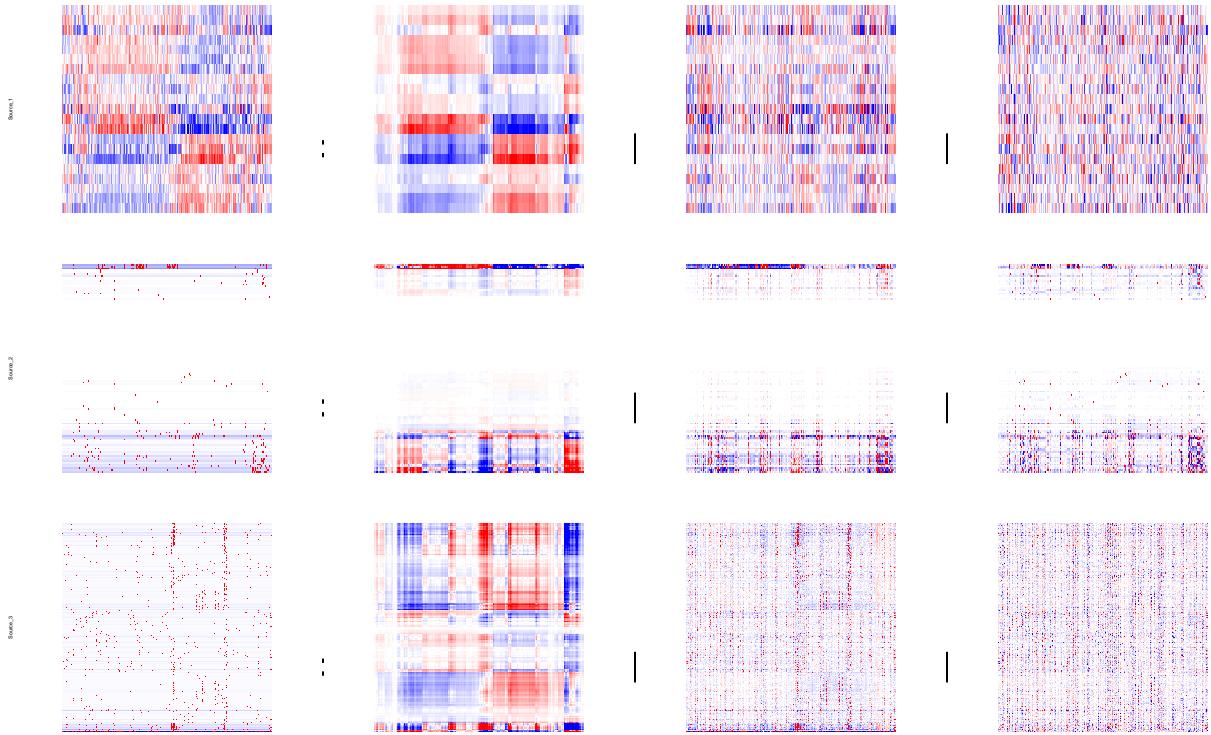
yeast_jive <- jive(yeast_data,
  rankJ = 2,
  rankA = c(3, 6, 49),
  method = "given"
)

## Running JIVE algorithm for ranks:
## joint rank: 2 , individual ranks: 3 6 49
## JIVE algorithm converged after 249 iterations.
showPCA(yeast_jive, 2, c(1, 1, 1))
```

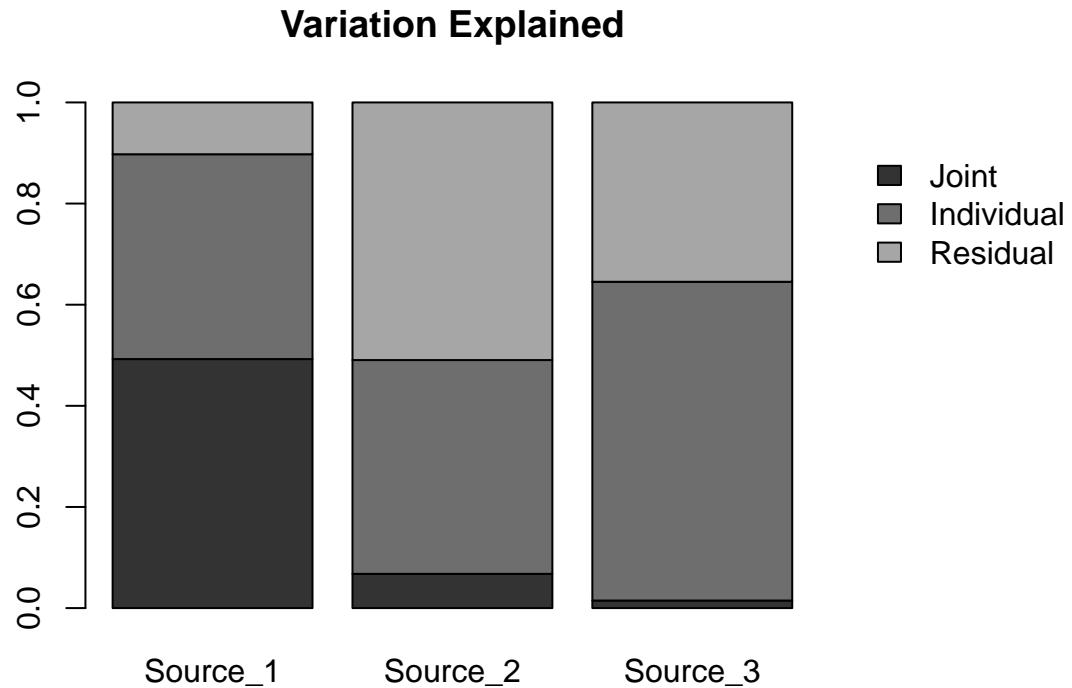


```
showHeatmaps(yeast_jive)
```

29	2	17	6	21	1	25	13
30	3	18	7	22	10	26	14
31	4	19	8	23	11	27	15
32	5	20	9	24	12	28	16



```
showVarExplained(yeast_jive)
```



References

- Kirk, Paul, Jim E Griffin, Richard S Savage, Zoubin Ghahramani, and David L Wild. 2012. “Bayesian Correlated Clustering to Integrate Multiple Datasets.” *Bioinformatics* 28 (24): 3290–7.
- Lock, Eric F, Katherine A Hoadley, James Stephen Marron, and Andrew B Nobel. 2013. “Joint and Individual Variation Explained (Jive) for Integrated Analysis of Multiple Data Types.” *The Annals of Applied Statistics* 7 (1): 523.
- O’Connell, Michael J, and Eric F Lock. 2016. “R. JIVE for Exploration of Multi-Source Molecular Data.” *Bioinformatics* 32 (18): 2877–9.