

scPCA: A toolbox for sparse contrastive principal component analysis in R

Philippe Boileau¹, Nima S. Hejazi^{1, 2}, and Sandrine Dudoit^{2, 3, 4}

1 Graduate Group in Biostatistics, University of California, Berkeley **2** Center for Computational Biology, University of California, Berkeley **3** Department of Statistics, University of California, Berkeley **4** Division of Epidemiology and Biostatistics, School of Public Health, University of California, Berkeley

DOI: [10.21105/joss.02079](https://doi.org/10.21105/joss.02079)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Charlotte Soneson](#) ↗

Reviewers:

- [@fabian-s](#)
- [@LTLA](#)

Submitted: 28 January 2020

Published: 07 February 2020

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Summary

Data pre-processing and exploratory data analysis are crucial steps in the data science life-cycle, often relying on dimensionality reduction techniques to extract pertinent signal. As the collection of large and complex datasets becomes the norm, the need for methods that can successfully glean pertinent information from among increasingly intricate technical artifacts is greater than ever. What's more, many of the most historically reliable and commonly used methods have demonstrably poor performance, or even fail outright, in reducing the dimensionality of large and noisy datasets in a stable, interpretable, and relevant manner.

Principal component analysis (PCA) is one such method. Although popular for its interpretable results and ease of implementation, PCA's performance on high-dimensional data often leaves much to be desired. Its performance has been characterized as unstable on large datasets (Johnstone & Lu, 2009), and it has been shown to often emphasize unwanted variation (e.g., batch effects) in lieu of the signal of interest.

Consequently, modifications of PCA have been developed to remedy these issues. Namely, sparse PCA (SPCA) (Zou, Hastie, & Tibshirani, 2006) was created to increase the stability and interpretability of the principal component loadings in high dimensions, while contrastive PCA (cPCA) (Abid, Zhang, Bagaria, & Zou, 2018) leverages control data to adjust for unwanted effects and capture relevant information.

Although SPCA and cPCA have proven useful in resolving individual shortcomings of PCA, neither is capable of tackling the issues of stability and relevance simultaneously. The scPCA R package implements sparse contrastive PCA (scPCA) (Boileau, Hejazi, & Dudoit, 2019), a combination of these methods, drawing on cPCA to remove unwanted effects and on SPCA to sparsify the principal component loadings. In both simulation studies and data analysis, Boileau et al. (2019) provided practical demonstrations of scPCA's ability to extract stable, interpretable, and uncontaminated signal from high-dimensional biological data. Such demonstrations included the re-analysis of several publicly available protein expression, microarray gene expression, and single-cell transcriptome sequencing datasets.

As the scPCA software package was specially designed for use in disentangling biological signal from technical noise in high-throughput sequencing data, a free and open-source software implementation has been made available via the Bioconductor Project (Gentleman, Carey, Huber, Irizarry, & Dudoit, 2006; Gentleman et al., 2004; Huber et al., 2015) for the R language and environment for statistical computing (R Core Team, 2020). The scPCA package also implements cPCA, previously unavailable in the R language, in two flavors: (1) the semi-automated version of Abid et al. (2018) and (2) the automated version formulated by Boileau et al. (2019). In order to interface seamlessly with data structures common in computational

biology, the `scPCA` package integrates fully with the `SingleCellExperiment` container class (Lun & Risso, 2019), utilizing the class to store the `cPCA` and `scPCA` representations generated via the `reducedDims` accessor method. Finally, to facilitate parallel computation, the `scPCA` package contains parallelized versions of each of its core subroutines, making use of the infrastructure provided by the `BiocParallel` package. In order to effectively utilize parallelization, one need only set `parallel = TRUE` in a call to the `scPCA` package, after having registered a particular parallelization backend, as per the `BiocParallel` documentation.

Acknowledgments

Philippe Boileau's contribution to this work was supported by the Fonds de recherche du Québec - Nature et technologies (B1X).

References

- Abid, A., Zhang, M. J., Bagaria, V. K., & Zou, J. (2018). Exploring patterns enriched in a dataset with contrastive principal component analysis. *Nature Communications*, 9(1), 2134. doi:[10.1038/s41467-018-04608-8](https://doi.org/10.1038/s41467-018-04608-8)
- Boileau, P., Hejazi, N. S., & Dudoit, S. (2019). Exploring high-dimensional biological data with sparse contrastive principal component analysis. *bioRxiv*. doi:[10.1101/836650](https://doi.org/10.1101/836650)
- Gentleman, R., Carey, V., Huber, W., Irizarry, R., & Dudoit, S. (2006). *Bioinformatics and computational biology solutions using R and Bioconductor*. Springer Science & Business Media.
- Gentleman, R., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., et al. (2004). Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5(10), R80. doi:[10.1186/gb-2004-5-10-r80](https://doi.org/10.1186/gb-2004-5-10-r80)
- Huber, W., Carey, V. J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B. S., Bravo, H. C., et al. (2015). Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods*, 12(2), 115. doi:[10.1038/nmeth.3252](https://doi.org/10.1038/nmeth.3252)
- Johnstone, I. M., & Lu, A. Y. (2009). On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 104(486), 682–693. doi:[10.1198/jasa.2009.0121](https://doi.org/10.1198/jasa.2009.0121)
- Lun, A., & Risso, D. (2019). *SingleCellExperiment: S4 classes for single cell data*. doi:[10.18129/B9.bioc.SingleCellExperiment](https://doi.org/10.18129/B9.bioc.SingleCellExperiment)
- R Core Team. (2020). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from <https://www.R-project.org/>
- Zou, H., Hastie, T., & Tibshirani, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2), 265–286. doi:[10.1198/106186006X113430](https://doi.org/10.1198/106186006X113430)