Received: 9 August 2019 | Accepted: 1 April 2021

DOI: 10.1111/rssc.12494

ROYAL STATISTICAL SOCIETY
Series C Applied Statistics

ORIGINAL ARTICLE

# Sparse reduced-rank regression for exploratory visualisation of paired multivariate data

**Dmitry Kobak**[1] | **Yves Bernaerts**[1,2] | **Marissa A. Weis**[1] | **Federico Scala**[3] | **Andreas S. Tolias**[3] | **Philipp Berens**[1,4]

[1]Institute for Ophthalmic Research, University of Tübingen, Tübingen, Germany

[2]International Max Planck, Research School for Intelligent Systems, Germany

[3]Department of Neuroscience, Baylor College of Medicine, Houston, Texas, USA

[4]Department of Computer Science, University of Tübingen, Tübingen, Germany

**Correspondence**
Dmitry Kobak and Philipp Berens, Institute for Ophthalmic Research, University of Tübingen, Tübingen, Germany
Emails: dmitry.kobak@uni-tuebingen.de and philipp.berens@uni-tuebingen.de

**Funding information**
German Ministry of Education and Research, Grant/Award Number: FKZ 01GQ1601; German Research Foundation, Grant/Award Number: 390727645 and BE5601/4-1; National Institutes of Health, Grant/Award Number: U19MH114830

## Abstract

In genomics, transcriptomics, and related biological fields (collectively known as *omics*), combinations of experimental techniques can yield multiple sets of features for the same set of biological replicates. One example is Patch-seq, a method combining single-cell RNA sequencing with electrophysiological recordings from the same cells. Here we present a framework based on sparse reduced-rank regression (RRR) for obtaining an interpretable visualisation of the relationship between the transcriptomic and the electrophysiological data. We use elastic net regularisation that yields sparse solutions and allows for an efficient computational implementation. Using several Patch-seq datasets, we show that sparse RRR outperforms both sparse full-rank regression and non-sparse RRR, as well as previous sparse RRR approaches, in terms of predictive performance. We introduce a *bibiplot* visualisation in order to display the dominant factors determining the relationship between transcriptomic and electrophysiological properties of neurons. We believe that sparse RRR can provide a valuable tool for the exploration and visualisation of paired multivariate datasets.

**KEYWORDS**
Patch-seq, reduced-rank regression

# 1 | INTRODUCTION

Since the days of Ramón y Cajal, neuroscientists have classified neurons into cell types, which are often considered the fundamental building blocks of neural circuits (Masland, 2004). Classically, these types have been defined based on their electrophysiology or anatomy, but due to the recent rise of single-cell transcriptomics, a definition of cell types based on genetics is becoming increasingly popular (Poulin et al., 2016). For example, single-cell RNA sequencing has been used to establish a census of neurons in the retina (Macosko et al., 2015; Shekhar et al., 2016), the cortex (Tasic et al., 2016, 2018; Zeisel et al., 2015), the whole brain (Saunders et al., 2018) and the entire nervous system (Zeisel et al., 2018) of mice. Despite this success, it has proven difficult to integrate the obtained cell type taxonomy based on the transcriptome with information about physiology and anatomy (Tripathy et al., 2017; Zeng & Sanes, 2017) and it remains unclear to what extent neural types are discrete or show continuous variation (Harris et al., 2018; Zeng & Sanes, 2017).

A recently developed technique called Patch-seq (Cadwell et al., 2016, 2017; Földy et al., 2016; Fuzik et al., 2016) allows to isolate and sequence RNA content of cells characterised electrophysiologically and/or morphologically (Figure 1a), opening the way to relate gene expression patterns to physiological characteristics on the single-cell level (Lipovsek et al., 2021). Patch-seq experiments are laborious and low throughput, resulting in multimodal datasets with a particular statistical structure: a few dozen or hundreds of cells are characterised with expression levels of many thousands of genes as well as dozens of electrophysiological measurements (Figure 1a). Integrating and properly visualising genetic and physiological information in this $n \ll p$ regime requires specialised statistical techniques that could isolate a subset of relevant genes and exploit information about the relationships within both data modalities to increase statistical power.

Here we extended the sparse reduced-rank regression (sRRR) method of Chen and Huang (2012) and used it to obtain an interpretable and intuitive visualisation of the relationship between high-dimensional single-cell transcriptomes and electrophysiological information obtained using techniques like Patch-seq. We used five existing Patch-seq datasets (Cadwell et al., 2016; Fuzik et al., 2016; Gouwens et al., 2020; Scala et al., 2019, 2020) with sample sizes ranging from $n = 44$ to $n = 3395$ to demonstrate and validate our method (Table 1). Our sparse RRR method outperformed the sparse RRR of Chen and Huang (2012) on our data in terms of cross-validated $R^2$.

Our code in Python is available at https://github.com/berenslab/patch-seq-rrr.

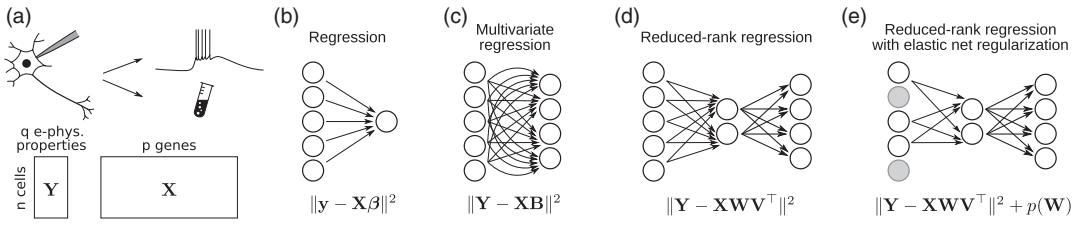



**FIGURE 1** (a) Schematic illustration of a Patch-seq experiment: electrophysiological activity is recorded by patch clamping, followed by RNA extraction and sequencing. Below: data matrices after computational characterisation of electrophysiological properties ($\mathbf{Y}$) and estimation of gene counts ($\mathbf{X}$). (b–e). Schematic illustrations and loss functions for several regression methods. (b) Simple regression. (c) Multivariate regression. (d) Reduced-rank regression. (e) Regularised reduced-rank regression. Grey circles denote predictors that are left out of the sparse model

**TABLE 1** Patch-seq datasets used in this study. All recordings were done in the mouse neocortex

| Name | Citation | Cells ($n$) | Genes ($p$) | Features ($q$) | Description |
|---|---|---|---|---|---|
| M1 | Scala et al. (2020) | 1213 | 1000 | 16 | Motor cortex, all layers/types |
| V1 | Gouwens et al. (2020) | 3395 | 1252 | 55 | Visual cortex, interneurons (L1–L6) |
| L4 | Scala et al. (2019) | 102 | 1000 | 13 | Layer 4 *Sst* interneurons |
| L1 | Cadwell et al. (2016) | 44 | 3000 | 11 | Layer 1 interneurons |
| S1 | Fuzik et al. (2016) | 80 | 1384 | 80 | Layer 1/2 neurons |

# 2 | RESULTS

## 2.1 | Patch-seq data

A Patch-seq experiment yields two paired data matrices (Figure 1a): an $n \times p$ matrix $\mathbf{X}$ containing expression levels of $p$ genes for each of the $n$ cells, and an $n \times q$ matrix $\mathbf{Y}$ containing $q$ electrophysiological properties of the same $n$ cells. We assume that both matrices are centred, that is column means have been subtracted.

To illustrate the structure of such datasets and motivate the development of sparse RRR for exploratory visualisation, we use principal component analysis (PCA) on the M1 dataset, one of the largest existing Patch-seq datasets (Scala et al., 2020). It contains $n = 1213$ neurons from the primary motor cortex of adult mice and spans all types of neurons, both excitatory and inhibitory (Table 1). Each cell was described by $q = 16$ electrophysiological properties and we used the $p = 1000$ most variable genes that were selected in the original publication. Note that there are over 40 thousand coding and non-coding genes in the mouse genome that were detected in at least one cell in this particular dataset. It is, however, a common practice to select a smaller set of genes for downstream analysis (Luecken & Theis, 2019), as most detected genes have low average expression, low variance, and are likely not informative. We log-transformed all gene counts and standardised the columns of $\mathbf{X}$ and $\mathbf{Y}$ matrices (see Methods for more details).

PCA in the transcriptomic space (Figure 2a) revealed that PC1, in this case, was an experimental artefact largely driven by the variability in the sequencing quality between cells (correlation between PC1 and the log number of detected genes was 0.90), whereas PC2 captured a biologically meaningful difference between the excitatory and the inhibitory cells. In contrast, PCA in the electrophysiological space (Figure 2b) separated major classes of neurons with different firing properties, such as *Pvalb*- (red), *Sst*- (orange) and *Vip*- (purple) expressing interneurons. Thus, there appears to be no direct relationship between the leading PCs of the two modalities. The aim of RRR is to uncover such relationships.

The visualisation in Figure 2b is known as *biplot* (Gabriel, 1971). Lines represent correlations between each electrophysiological property and PC1/PC2: the horizontal coordinate of each line's tip shows the correlation with PC1 and the vertical coordinate shows the correlation with PC2. The circle, sometimes called *correlation circle*, shows the maximum attainable correlation. The scaling between the scatter plot and the lines/circle is arbitrary. Following Gabriel (1971), we standardise both PCs and scale the lines/circle by an arbitrary factor of 3 (so that most points in the scatter plot are contained within the circle). We do not show a biplot in the transcriptomic space (Figure 2a) because the PCA in the gene space is not sparse, making the biplot practically impossible to display and interpret as it

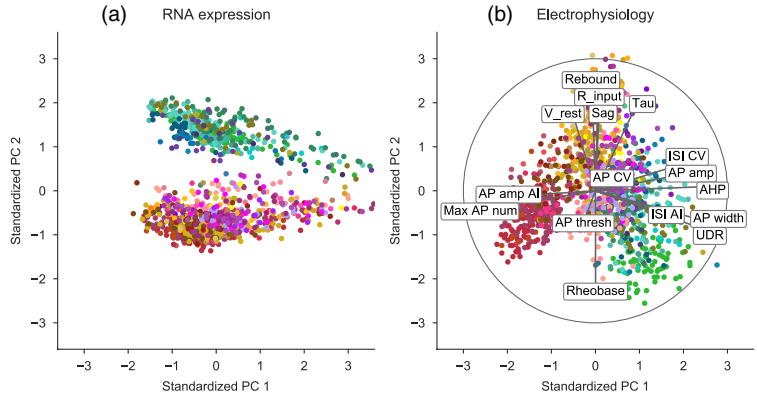



**FIGURE 2** (a) Principal component analysis (PCA) of the transcriptomic data in the M1 dataset (Scala et al., 2020). Colour denotes transcriptomic type (cold colours: excitatory neurons; warm colours: inhibitory neurons). Both PCs were standardised. (b) PCA biplot of the electrophysiological data in the same dataset. Grey lines show correlations of individual electrophysiological features with PC1 and PC2. The circle (*correlation circle*) shows maximal possible correlations. The relative scaling of the scatter plot and the lines/circle is arbitrary. The label positions were automatically adjusted by simulating spring repulsive forces between them until they stopped overlapping

would have to show all 1000 genes from $\mathbf{X}$. This motivates the sparsity constraint that we impose on RRR.

## 2.2 | Reduced-rank regression

To relate gene expression patterns to electrophysiological properties, one could use the transcriptomic data to explain any given electrophysiological property, for example action potential threshold. This is a *regression* problem: the expression level of each gene is an explanatory variable and the action potential threshold is the response variable (Figure 1b). To predict multiple electrophysiological properties at the same time, one can combine individual regressions into a *multivariate regression* problem where the response is a multivariate vector (Figure 1c). The loss function of multivariate linear regression (known as ordinary least squares, OLS) is

$$\mathcal{L}_{\text{OLS}} = \|\mathbf{Y} - \mathbf{XB}\|^2 \tag{1}$$

and its well-known solution is given by

$$\hat{\mathbf{B}}_{\text{OLS}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}. \tag{2}$$

Here and below all matrix norms are Frobenius norms. The intercept is omitted because both $\mathbf{X}$ and $\mathbf{Y}$ are assumed to be centered.

Different electrophysiological properties tend to be strongly correlated and so one can construct a more parsimonious model where gene expression is predicting $r < q$ latent factors that in turn predict all $q$ electrophysiological properties together (Figure 1d). These latent factors form a bottleneck in the linear mapping and allow exploiting correlations between the predicted electrophysiological properties to reduce the number of model parameters and to decrease overfitting. This approach is called *reduced-rank regression* (RRR) (Izenman, 1975; Velu & Reinsel, 2013). Its loss function is

$$\mathcal{L}_{\text{RRR}} = \|\mathbf{Y} - \mathbf{X}\mathbf{W}\mathbf{V}^\top\|^2, \tag{3}$$

where $\mathbf{W}$ and $\mathbf{V}$ each have $r \le \min(p, q)$ columns. Without loss of generality, it is convenient to require that $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$. The product $\mathbf{W}\mathbf{V}^\top$ forms the matrix of regression coefficients that has rank $r$.

This decomposition allows to interpret $\mathbf{W}$ as a mapping that transforms $\mathbf{X}$ into $r$ latent variables and $\mathbf{V}$ as a mapping that transforms the latent variables into $\mathbf{Y}$ (Figure 1e). As a result, RRR can be viewed not only as a prediction method, but also as a dimensionality reduction method, allowing visualisation and exploration of the paired dataset. Latent factors $\mathbf{X}\mathbf{W}$ can be interpreted as capturing low-dimensional genetic variability that is predictive of electrophysiological variability, while $\mathbf{Y}\mathbf{V}$ can be interpreted as low-dimensional electrophysiological variability that can be predicted from the genetic variability.

RRR can be directly solved by applying singular value decomposition (SVD) to the results of multivariate regression. Indeed, the RRR loss can be decomposed into the OLS loss and the low-rank loss:

$$\mathcal{L}_{\text{RRR}} = \|\mathbf{Y} - \mathbf{X}\mathbf{W}\mathbf{V}^\top\|^2 = \|\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}}_{\text{OLS}}\|^2 + \|\mathbf{X}\hat{\mathbf{B}}_{\text{OLS}} - \mathbf{X}\mathbf{W}\mathbf{V}^\top\|^2, \tag{4}$$

The first term corresponds to the variance of $\mathbf{Y}$ that is unexplainable by any linear model. The minimum of the second term can be obtained by computing the SVD of $\mathbf{X}\hat{\mathbf{B}}_{\text{OLS}}$. The right singular vectors corresponding to the $r$ largest singular values give $\hat{\mathbf{V}}$, and $\widehat{\mathbf{W}} = \hat{\mathbf{B}}_{\text{OLS}}\hat{\mathbf{V}}$.

Reduced-rank regression with the ridge penalty $\lambda\|\mathbf{W}\mathbf{V}\|^2 = \lambda\|\mathbf{W}\|^2$ has the same analytic solution, but $\hat{\mathbf{B}}_{\text{OLS}}$ should be replaced with $\hat{\mathbf{B}}_{\text{ridge}} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{Y}$.

## 2.3 | Reduced-rank regression with elastic net penalty

As there are over 20 thousand genes in a mouse genome (with 1000–5000 typically retained for analysis) while the typical sample size $n$ of a Patch-seq dataset is on the order of 100–1000, the regression problems discussed above are in the $n < p$ regime and need to be regularised. Here we use elastic net regularisation, which combines $\ell_1$ (lasso) and $\ell_2$ (ridge) penalties (Zou & Hastie, 2005). Elastic net enforces sparsity and performs feature selection: only a small subset of genes are selected into the model while all other genes get zero regression coefficients (Figure 1e). Our elastic net RRR extends a previously suggested sparse RRR (Chen & Huang, 2012) that used the lasso penalty on its own. The elastic net penalty has well-known advantages compared to the pure lasso penalty, for example it allows to select more than $n$ predictors and can outperform lasso when predictors are strongly correlated (Zou & Hastie, 2005).

The loss function of our regularised RRR is:

$$\mathcal{L}_{\text{sRRR}} = \frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\mathbf{W}\mathbf{V}^\top\|^2 + \lambda\left(\alpha\sum_{i=1}^{p}\|\mathbf{W}_{i\cdot}\|_2 + (1-\alpha)\|\mathbf{W}\|^2/2\right) \quad \text{s.t.} \quad \mathbf{V}^\top\mathbf{V} = \mathbf{I}. \tag{5}$$

The $\ell_2$ penalty is only applied to the matrix $\mathbf{W}$ because $\mathbf{V}$ is constrained to have a fixed $\ell_2$ norm. Also, following Chen and Huang (2012), we chose not to apply $\ell_1$ penalty to $\mathbf{V}$ and impose sparsity constraint only on the gene selection (see Discussion). We used the same parametrisation of the penalty as in the popular `glmnet` library (Friedman et al., 2010): $\alpha$ controls the trade-off between the lasso ($\alpha = 1$) and the ridge ($\alpha = 0$) while $\lambda$ controls the overall regularisation strength. Following Chen and Huang (2012), the lasso penalty term $\sum_{i=1}^{p}\|\mathbf{W}_{i\cdot}\|_2 = \sum_{i=1}^{p}\sqrt{\sum_{j=1}^{r}W_{ij}^2}$ computes the sum of $\ell_2$ norms of each row of

$\mathbf{W}$. This is known as *group lasso* (Yuan & Lin, 2006), because it is the $\ell_1$ norm of the vector of row $\ell_2$ norms; it encourages the entire rows of $\mathbf{W}$, and not just its individual elements, to be zeroed out, corresponding to some of the genes being left out of the model entirely. See Discussion about this choice.

This optimisation problem is biconvex and can be solved with an iterative alternating approach: in turn, we fix $\mathbf{V}$ and find the optimal $\mathbf{W}_{\text{opt}}$ and then fix $\mathbf{W}$ and find the optimal $\mathbf{V}_{\text{opt}}$ until convergence. For a fixed $\mathbf{V}$, the least-squares term can be re-written as

$$\begin{aligned}\|\mathbf{Y}-\mathbf{X}\mathbf{W}\mathbf{V}^\top\|^2 &= \mathrm{tr}(\mathbf{Y}^\top\mathbf{Y})+\mathrm{tr}(\mathbf{V}\mathbf{W}^\top\mathbf{X}^\top\mathbf{X}\mathbf{W}\mathbf{V}^\top)-2\mathrm{tr}(\mathbf{V}\mathbf{W}^\top\mathbf{X}^\top\mathbf{Y})\\ &= \mathrm{const}+\mathrm{tr}(\mathbf{V}^\top\mathbf{Y}^\top\mathbf{Y}\mathbf{V})+\mathrm{tr}(\mathbf{W}^\top\mathbf{X}^\top\mathbf{X}\mathbf{W})-2\mathrm{tr}(\mathbf{W}^\top\mathbf{X}^\top\mathbf{Y}\mathbf{V})\\ &= \mathrm{const}+\|\mathbf{Y}\mathbf{V}-\mathbf{X}\mathbf{W}\|^2,\end{aligned} \tag{6}$$

meaning that for a fixed $\mathbf{V}$, the loss is equivalent to

$$\mathcal{L}_{\text{sRRR}}\,|\,\mathbf{V} \sim \frac{1}{2n}\|\mathbf{Y}\mathbf{V}-\mathbf{X}\mathbf{W}\|^2+\lambda\left(\alpha\sum_{i=1}^{p}\|\mathbf{W}_{i\cdot}\|_2+(1-\alpha)\|\mathbf{W}\|^2/2\right). \tag{7}$$

This is the loss of multivariate elastic net regression of $\mathbf{Y}\mathbf{V}$ on $\mathbf{X}$, and so the optimal $\mathbf{W}_{\text{opt}}$ can be obtained using the `glmnet` library (Friedman et al., 2010) (using `family = "mgaussian"` option for row-wise lasso penalty) which has readily available interfaces for Matlab, Python and R.

For a fixed $\mathbf{W}$, the loss does not depend on the penalty terms and the least-squares term can be written as

$$\begin{aligned}\|\mathbf{Y}-\mathbf{X}\mathbf{W}\mathbf{V}^\top\|^2 &= \|\mathbf{Y}\|^2+\mathrm{tr}(\mathbf{V}\mathbf{W}^\top\mathbf{X}^\top\mathbf{X}\mathbf{W}\mathbf{V}^\top)-2\mathrm{tr}(\mathbf{Y}^\top\mathbf{X}\mathbf{W}\mathbf{V}^\top)\\ &= \mathrm{const}-2\mathrm{tr}(\mathbf{Y}^\top\mathbf{X}\mathbf{W}\mathbf{V}^\top).\end{aligned} \tag{8}$$

This is an example of the orthogonal Procrustes problem (Gower & Dijksterhuis, 2004). Maximising the trace $\mathrm{tr}(\mathbf{Y}^\top\mathbf{X}\mathbf{W}\mathbf{V}^\top)$ is achieved by the thin SVD of $\mathbf{Y}^\top\mathbf{X}\mathbf{W}$. If the $r$ left and right singular vectors are stacked in columns of $\mathbf{L}$ and $\mathbf{R}$, respectively (we order them by singular values, in decreasing order), then $\mathbf{V}_{\text{opt}}=\mathbf{L}\mathbf{R}^\top$. We provide a short proof in the Appendix.

Given that the loss function is biconvex but possibly not jointly convex in $\mathbf{V}$ and $\mathbf{W}$, it can be important to choose a reasonable initialisation. We initialised $\mathbf{V}$ by the $r$ leading right singular vectors of $\mathbf{X}^\top\mathbf{Y}$ and found this strategy to work well.

## 2.4 | Relaxed elastic net

It has been argued that elastic net or even the lasso penalty on its own can lead to an over-shrinkage with non-zero coefficients shrinking too much (Zou & Hastie, 2005). There have been several suggestions in the literature on how to mitigate this effect (Efron et al., 2004; Meinshausen, 2007; Zou & Hastie, 2005). *Relaxed lasso* (Meinshausen, 2007) performs lasso (setting $\alpha=1$ and $\lambda=\lambda_1$) and then, using only the terms with non-zero coefficients, performs another lasso with a different penalty ($\alpha=1$, $\lambda=\lambda_2$; usually $\lambda_2<\lambda_1$). If $\lambda_2=0$, then this has also been called *LARS-OLS hybrid* (Efron et al., 2004).

Similar two-stage procedures for the elastic net penalty are not as established. We obtained a strong improvement in predictive performance if—after RRR with elastic net penalty with coefficients $\lambda$ and $\alpha$—we take the genes with non-zero coefficients and run RRR again using $\alpha=0$ (i.e. pure ridge) and the same value of $\lambda$ (see Figure 3 below). This procedure does not introduce any additional tuning

parameters but substantially outperforms pure elastic net RRR on our data, as we show below. We called it *relaxed elastic net*, following the relaxed lasso terminology (Meinshausen, 2007). The solution of the first round of sparse RRR we call *naïve*, following Zou and Hastie (2005).

A similar approach was used by De Mol et al. (2009) who performed elastic net using $\lambda = \lambda_1$ and some small fixed value of $\alpha = \alpha_1$, selected all genes with non-zero coefficients, and did pure ridge ($\alpha = 0$) regression with $\lambda = \lambda_2$ on this gene subset. This approach also has two hyperparameters that need to be selected using cross-validation, but requires a manual choice of $\alpha_1$ for the first elastic net. If $\alpha$ is also treated as an adjustable hyperparameter, then it becomes a more flexible generalisation of our approach with three hyperparameters.

## 2.5 | Cross-validation

We used cross-validation (CV) to select the values of $r$, $\lambda$, and $\alpha$ that maximise the predictive performance of the sparse RRR model. The cross-validation estimates of $R^2$ are shown in Figure 3 for the

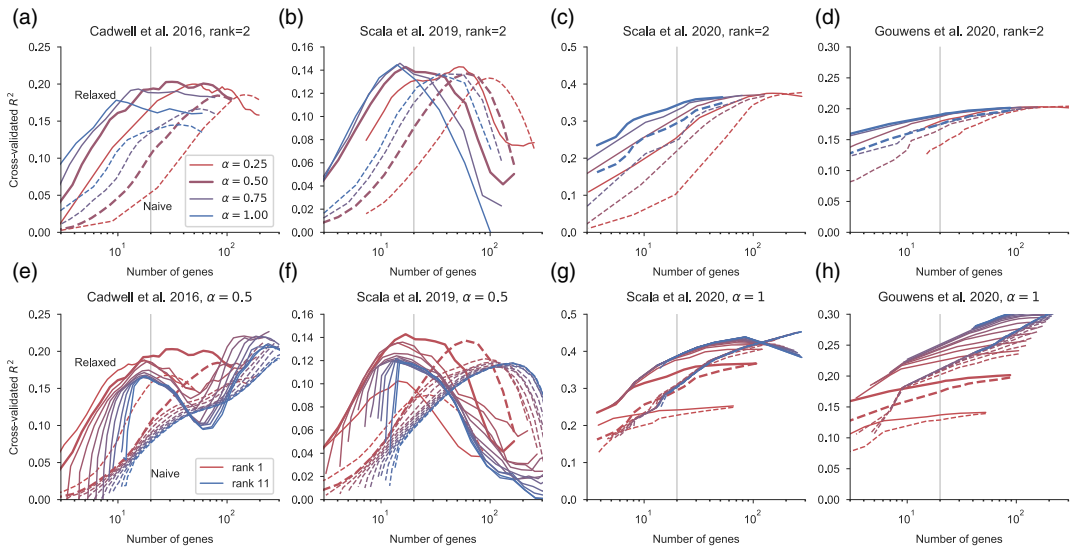



**FIGURE 3** (a) Cross-validation performance of sparse RRR with $r = 2$ on the L1 dataset, depending on $\alpha$ (colour-coded, see legend) and $\lambda$. Horizontal axis shows the average number of selected genes obtained for each $\lambda$. Dashed lines: naive sparse RRR. Solid lines: relaxed sparse RRR. The vertical line at 20 selected genes indicates our parameter choice. Thick lines highlight $\alpha = 0.5$. The standard deviation over all CV folds and repetitions at $\alpha = 0.5$ and $\lambda$ value yielding ~20 genes was 0.05 for the naive and 0.12 for the relaxed estimator (note that each fold had only 4 samples in the test set). (b) The same for the L4 dataset. The standard deviation over all CV folds and repetitions at $\alpha = 0.5$ and $\lambda$ value yielding ~20 genes was 0.04 for the naive and 0.07 for the relaxed estimator (here each fold had only ~10 samples in the test set). (c) The same for the M1 dataset. Here thick lines highlight $\alpha = 1$. The standard deviation over all CV folds and repetitions at $\alpha = 1$ and $\lambda$ value yielding ~20 genes was 0.01 for the naive and for the relaxed estimators (here each fold had ~120 samples in the test set). (d) The same for the V1 dataset. Here thick lines highlight $\alpha = 1$. The standard deviation over all CV folds and repetitions at $\alpha = 1$ and $\lambda$ value yielding ~20 genes was 0.01 for the naive and for the relaxed estimators (here each fold had ~340 samples in the test set). (e) Cross-validation performance with $\alpha = 0.5$ depending on the rank (colour-coded, see legend) on the L1 dataset. Thick lines highlight $r = 2$. (f) The same for the L4 dataset. (g) The same for the M1 dataset (here using $\alpha = 1$). (h) The same for the V1 dataset (here using $\alpha = 1$)

L1, L4, M1 and V1 datasets. We used 10 times repeated 11-fold CV for the L1 data ($n = 44$), 10 times repeated 10-fold CV for the L4 data ($n = 102$), and non-repeated 10-fold CV for the M1 ($n = 1321$) and V1 ($n = 3395$) data. See Methods for the preprocessing details. The test-set $R^2$ for each CV fold was computed as

$$R^2 = 1 - \frac{\| \mathbf{Y}_{\text{test}} - \mathbf{X}_{\text{test}} \widehat{\mathbf{W}} \widehat{\mathbf{V}}^\top \|^2}{\| \mathbf{Y}_{\text{test}} \|^2}, \tag{9}$$

where $\mathbf{X}_{\text{test}}$ and $\mathbf{Y}_{\text{test}}$ were centred using the corresponding training-set means. We averaged the resulting $R^2$ across all folds and repetitions.

When using rank $r = 2$ (Figure 3a–d), we found that $\alpha = 0.5$ outperformed $\alpha = 1$ on the L1 dataset, suggesting that adding an additional ridge penalty to the sparse RRR model of Chen and Huang (2012) can be helpful. At the same time, $\alpha = 0.5$ and $\alpha = 1$ performed equally well on the L4 and V1 datasets, while $\alpha = 1$ outperformed other values on the M1 dataset. Overall, the differences in predictive performance in the $\alpha \in [0.5, 1]$ range were moderate. For the downstream analysis, we used $\alpha = 0.5$ for the L4 and L1 datasets, and $\alpha = 1$ for the M1 and V1 datasets. We recommend $\alpha = 0.5$ as a default setting.

The optimal $\lambda$ corresponded to ∼30 selected genes for the L1 dataset, ∼15 selected genes for the L4 dataset, and ∼100 selected genes for the M1 and the V1 datasets (Figure 3a–d), but the performance was comparably good in the range of ∼10–100 genes. For the downstream analysis, we always chose the value of $\lambda$ yielding 20 selected genes. Selecting many more genes than that would make visualisation difficult (see below).

The optimal value of rank was $r = 2$ for the L1 and L4 datasets and $r \approx 10$ for the M1 and V1 datasets (Figure 3e–h). Lower ranks had worse performance due to underfitting, whereas higher ranks led to a drop in performance due to overfitting. Note that the full rank ($r = 11$, $r = 13$, $r = 16$ and $r = 55$ for the L1, L4, M1 and V1 datasets, respectively) corresponds to the standard multivariate elastic net regression. We verified that for the full rank, our algorithm yields the same solution as `glmnet` does on its own. The much better performance of $r = 2$ compared to the full rank on the L1 and L4 datasets shows that $r$ can act as a regularisation parameter, making sparse RRR outperform sparse full-rank regression. At the same time, for the M1 and V1 datasets, there was almost no difference in performance for any $r \geq 5$ and the full-rank model performed almost as well, suggesting little overfitting due to the larger sample sizes. Still, even in this case, the RRR framework allows to order individual components by their importance (explained variance) and to make low-dimensional visualisations (see below).

Finally, in all datasets, the relaxed version of sparse RRR strongly outperformed the naive version, at least in the range of 10–50 selected genes, which is the range needed for interpretable visualisations (see below). If a much higher number of genes were selected into the model, the relaxed version performed worse than the naive version, suggesting that the second stage of our relaxed approach was overfitting. For the L1 dataset with the smallest sample size, we observed non-monotonic dependency of the relaxed performance on the number of genes (Figure 3e), suggesting that the relaxed estimator can occupy different positions on the bias/variance trade-off depending on the $\lambda$. However, for the low number of selected genes, the relaxed version had superior performance across all datasets, all ranks, and all values of $\alpha$.

We did not use nested CV above because our CV performed almost no hyperparameter optimisation (Cawley & Talbot, 2010): in the downstream analysis, the rank was fixed to $r = 2$, $\lambda$ was fixed to yield 20 genes, leaving only the four-value $\alpha$ grid for hyperparameter optimisation. As a sanity check, we implemented nested CV with 10 outer folds to measure $R^2$ and 10 inner folds to find the value of $\alpha \in \{0.25, 0.5, 0.75, 1.0\}$ that yielded the highest $R^2$ with $\lambda$ corresponding to 20 genes and rank $r = 2$.

This procedure yielded the following values for the four datasets: 0.17 (L1 dataset), 0.13 (S1 dataset), 0.32 (M1 dataset) and 0.19 (V1 dataset). These estimates were nearly identical to the ones shown in Figure 3.

Note that sparse RRR of Chen and Huang (2012) corresponds to the naive (non-relaxed) version with $\alpha = 1$. In the regime when the model selects a few dozen genes, it was strongly outperformed by our relaxed sparse RRR estimator.

## 2.6 | Bibiplot visualisation

We applied our sparse RRR approach with $r = 2$ and $\lambda$ chosen to yield 20 selected genes to the L1, L4, M1 and V1 datasets (used $\alpha$ values: 0.5, 0.5, 1.0 and 1.0, respectively). For each of the datasets, we visualised the results with a pair of biplots, a graphical technique that we suggest to call a *bibiplot*.

To construct a biplot in the transcriptomic space, we use the bottleneck representation $\mathbf{XW}$ for the scatter plot and show lines for all genes that are selected by the model (even though other genes can also have non-zero correlations with $\mathbf{XW}$). The biplot in the electrophysiological space is constructed using $\mathbf{YV}$ and shows all available electrophysiological properties. If $R^2$ of the model is high, then the two scatter plots will be similar to each other. Comparing the directions of variables between the two biplots can suggest which electrophysiological variables are associated with which genes.

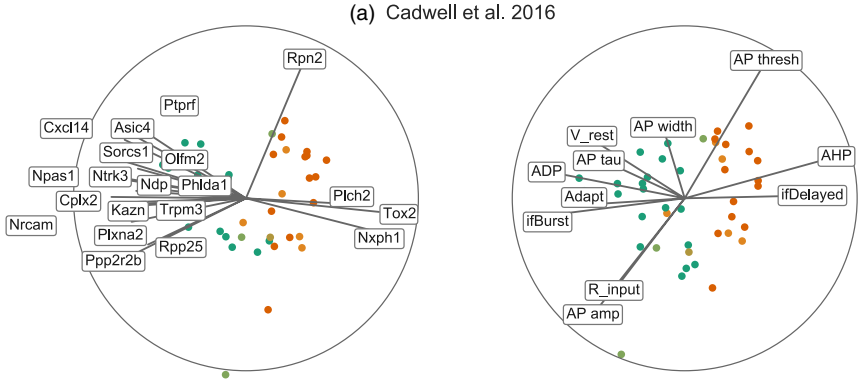
The L1 dataset encompasses two types of interneurons from layer 1 of mouse cortex: neurogliaform cells (NGC) and single bouqet cells (SBC). Accordingly, the first RRR component captured the difference between the two cell types (Figure 4a). The second RRR component had only one gene strongly associated with it (Figure 4a) and contributed only a very small increase in cross-validated $R^2$, as one can see comparing the cross-validated values for $r = 1$ and $r = 2$ at 20 selected genes (Figure 3e). We conclude that the second RRR component in this dataset is only weakly detectable.

In the L4 dataset (Figure 4b), the most salient feature in the bibiplot is the separation between the cells recorded in the visual and the somatosensory cortices. The selected genes here are pointing in all directions, and indeed the second component contributed a substantial increase in $R^2$ (Figure 3f). This suggests that both components are biologically meaningful. See Scala et al. (2019) for a more in-depth analysis using sparse RRR.

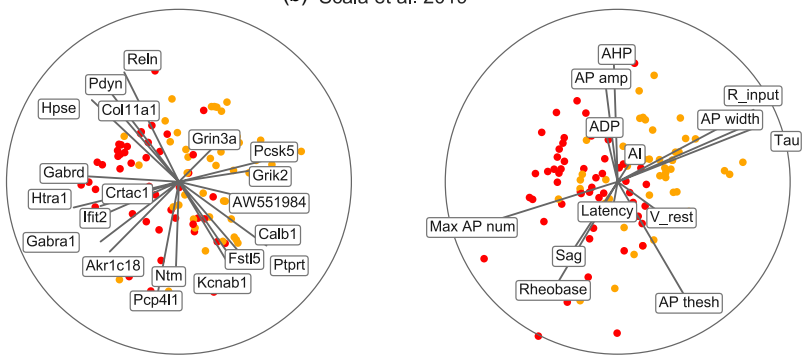
The M1 dataset was much larger than the previous two and included a much more diverse selection of neuron types. As a result, the $R^2$ values were substantially higher and the model needed to use rank $r \geq 5$ to reach its optimal performance. Here we nevertheless used $r = 2$ because it allows the same kind of visualisation as for the other datasets (Figure 4c). See Scala et al. (2020) for a more in-depth analysis using sparse RRR with rank $r = 5$. Two-dimensional bibiplot separated major classes of neurons, such as *Pvalb*, *Sst*, *Vip* and *Lamp5* expressing interneurons (red/orange/purple/salmon), and excitatory cells (green). Moreover, some selected genes were directly related to ion channel dynamics, such as the calcium channel subunit genes *Cacna1e* and *Cacna2d1* or the potassium channel-interacting protein gene *Kcnip2*. The same was true in the V1 dataset (Figure 4d), where, for example a potassium channel gene *Kctd8* was among those selected by the model.

It is worth noting that the RRR biplot in the electrophysiological space for the M1 data (Figure 4c, right) was very similar to the PCA biplot (Figure 2b). This indicates that the sparse RRR model explained the dominant modes of variation among the dependent variables. We observed the same in other datasets analysed here, even though in principle PCA and RRR components of the $\mathbf{Y}$ matrix can be very different.

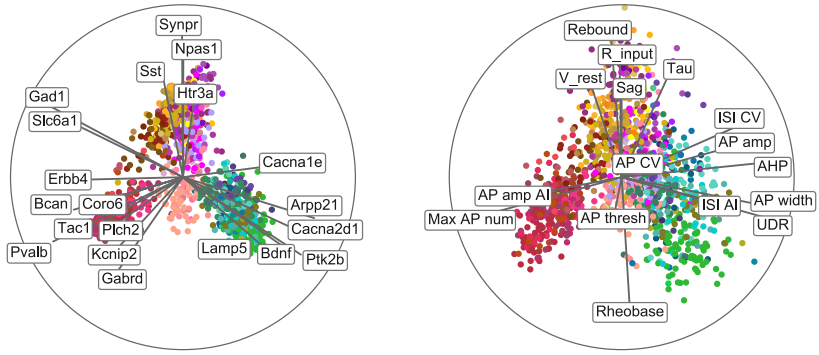
(a) Cadwell et al. 2016




(b) Scala et al. 2019




(c) Scala et al. 2020




(d) Gouwens et al. 2020

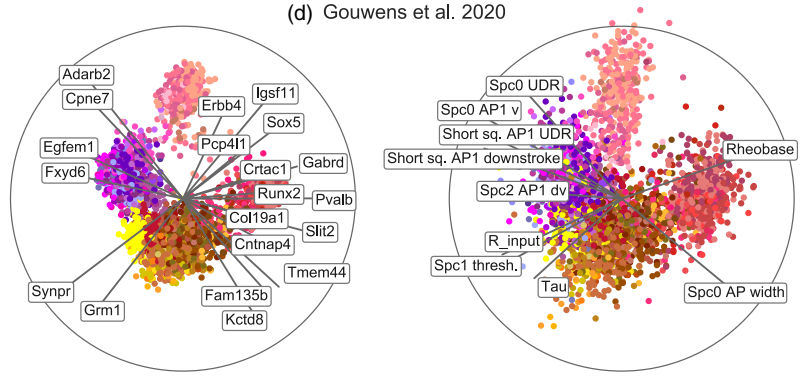


**FIGURE 4** (a) Sparse RRR biplot of the transcriptomic space (left) and electrophysiological space (right) in the L1 dataset. Colour codes cell type (orange: neurogliaform cells, NGC; green: single bouqet cells, SBC). Only 20 genes selected by the model are shown on the left. See Figure 2b for the details of biplot visualisation. As there, label positions were automatically adjusted to prevent overlap. (b) The same for the L4 dataset. Colour denotes cortical area (orange: visual cortex; red: somatosensory cortex). (c) The same for the M1 dataset. Colour denotes transcriptomic type (see Figure 2). (d) The same for the V1 dataset. Colour denotes transcriptomic type. For this dataset, only a subset of electrophysiological features are shown on the right to reduce the clutter

## 2.7 | Comparison to sparse CCA and PLS

Reduced-rank regression does not directly aim to maximise the correlation between $\mathbf{Xw}$ and $\mathbf{Yv}$, where $\mathbf{w}$ and $\mathbf{v}$ are corresponding columns of $\mathbf{W}$ and $\mathbf{V}$, even though high correlation is needed to achieve high $R^2$. Nevertheless, one can ask what is the cross-validated estimate of this correlation in each pair of RRR components. We used the same cross-validation scheme to measure these out-of-sample correlations (Figure 5)[1]. With the hyperparameters used above, the correlations in the L1 dataset were 0.69 for component 1 and 0.35 for component 2 (Figure 5a). In the L4 dataset, they were 0.65 and 0.54, respectively (Figure 5b). In the M1 dataset, they were 0.91 and 0.77 (Figure 5c); in the V1 dataset—0.92 and 0.83 (Figure 5d).

A statistical method that directly maximises correlation between $\mathbf{Xw}$ and $\mathbf{Yv}$ is called canonical correlation analysis (CCA). A number of different methods for sparse CCA have been suggested in the last decade (Chen et al., 2012b; Chu et al., 2013; Gao et al., 2017; Hardoon & Shawe-Taylor, 2011; Lykou & Whittaker, 2010; Mai & Zhang, 2019; Parkhomenko et al., 2009; Suo et al., 2017; Waaijenborg et al., 2008; Wiesel et al., 2008; Wilms & Croux, 2015; Witten & Tibshirani, 2009; Witten et al., 2009), of which the sparse CCA of Witten et al. (2009) is arguably the most well-known (judging by the number of citations in Google Scholar at the time of writing). We used the same cross-validation procedure as described above to measure the out-of-sample performance of their algorithm, using the original R implementation in package `PMA`. We found that this method performed worse than our sparse RRR: correlations for all four datasets and both components (the first and the second) were similar or lower than with sparse RRR, with only one exception (Figure 5, orange lines).

A likely explanation is that the method of Witten et al. (2009) is 'over-regularised'. To see this, note that RRR maximises explained variance in $\mathbf{Y}$, that is correlation between $\mathbf{Xw}$ and $\mathbf{Yv}$, times the standard deviation of $\mathbf{Yv}$. Another related method is called partial least squares (PLS): it maximises the covariance between $\mathbf{Xw}$ and $\mathbf{Yv}$, that is correlation, times the standard deviation of $\mathbf{Yv}$, times the standard deviation of $\mathbf{Xw}$. Both RRR and PLS can be seen as particular regularised versions of CCA, because they bias $\mathbf{w}$ and $\mathbf{v}$ toward the high-variance directions in $\mathbf{X}$ and $\mathbf{Y}$, somewhat similar to the ridge penalty. The method of Witten et al. (2009) maximises covariance (and so could in fact have been called 'sparse PLS' and not 'sparse CCA'), which may provide too strong $\ell_2$ regularisation. More recent sparse CCA methods (Mai & Zhang, 2019; Suo et al., 2017) have not been benchmarked here.

The method of Witten et al. (2009) can also be used to construct a bibiplot, even though the original paper did not discuss such visualisations. However, we found that for the M1 dataset such bibiplot was less informative than the one built with our method (Figure 6). We tuned the regularisation

[1]In some related previous work (González et al., 2008, 2009), cross-validated correlations were computed by pooling test set points across all cross-validation splits. We observed that this procedure can sometimes yield biased results; we compute test-set correlation within each test set, and then average across CV splits.

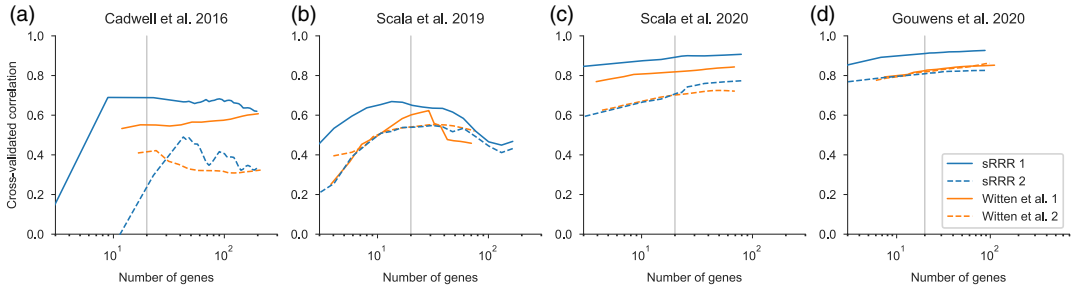



**FIGURE 5** (a) Cross-validation estimates of correlations between the transcriptomic and the electrophyiological reduced-rank regression (RRR) components with $r = 2$ in the L1 dataset, depending on $\lambda$. Horizontal axis shows the average number of selected genes obtained for each $\lambda$. Here we used $\alpha = 0.5$. Solid blue line: RRR component 1. Dashed blue line: RRR component 2. Solid and dashed orange lines: sparse CCA method of Witten et al. (2009), components 1 and 2. (b) The same for the L4 dataset. (c) The same for the M1 dataset. (d) The same for the V1 dataset

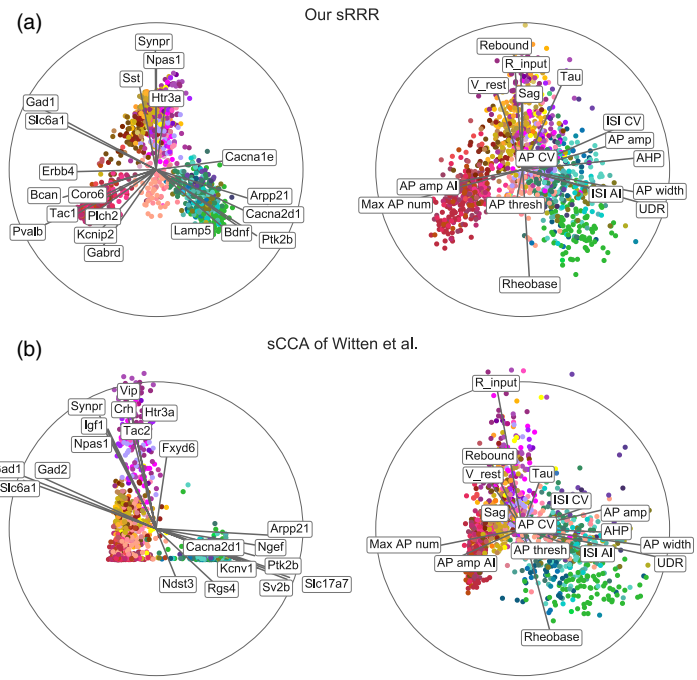



**FIGURE 6** (a) The same bibiplot as shown in Figure 4c for the M1 dataset. (b) The analogous bibiplot constructed using the method of Witten et al. (2009), with regularisation parameter tuned to yield 20 selected genes for both components together

penalty to yield 20 selected genes for two components together, but the resulting set had no genes associated with the fast-spiking interneurons (red colour). This is likely due to deflation procedure of Witten et al. (2009) leading to correlated components, as mentioned in the original paper. For the M1 dataset with 20 selected genes, correlation between the first and the second component in the gene space was 0.33. Using our method, the same correlation was 0.00.

## 2.8 | Gene selection stability

Instability is a general feature of all sparse models, especially when $n \ll p$ (Xu et al., 2011). We used bootstrapping to estimate gene selection stability in our datasets. On each of the 100 iterations, we drew a bootstrap sample of $n$ cells with repetitions and fit the sparse RRR model with the same parameters as above. This allows to measure how often each gene is selected into the model.

As expected, we found that the larger the sample size the more stable the gene selection was. In the L1 dataset ($n = 44$), the most reliably selected gene was selected only 67% of times. In the L4 dataset ($n = 102$), 89% of times. In the M1 dataset ($n = 1213$), there were 10 genes that were selected over 90% of times, with four genes getting into the model on every bootstrap iteration. Finally, in the V1 dataset ($n = 3395$), 12 genes were selected at least 90% of times, with eight genes getting selected on every iteration.

We also found that elastic net with $\alpha = 0.5$ typically led to a more stable model than the pure lasso with $\alpha = 1$ (with $\lambda$ values appropriately adjusted to select 20 genes). We quantified the overall gene selection stability by computing the mean and the standard deviation of the bootstrap selection fraction across the top 20 most often selected genes. This average stability was $0.24 \pm 0.10$ with $\alpha = 1.0$ versus $0.34 \pm 0.12$ with $\alpha = 0.5$ in the L1 dataset; $0.48 \pm 0.12$ versus $0.55 \pm 0.19$ in the L4 dataset; $0.80 \pm 0.20$ versus $0.85 \pm 0.19$ in the M1 dataset; and $0.90 \pm 0.13$ versus $0.83 \pm 0.20$ in the V1 dataset. The difference was not large, but observed in three out of four datasets. See Discussion for more considerations about model stability.

## 2.9 | Preprocessing choices

All of the analysis shown above was done after selecting 1000–3000 most variable genes and standardising the predictors. Putting these two preprocessing steps inside the cross-validation loop yielded practically the same results (Figure 7a–c; magenta lines).

Omitting the gene selection step and performing sparse RRR directly on all detected genes (this number can exceed 40000, counting both coding and non-coding genes) and/or omitting the standardisation step led to lower cross-validated $R^2$ values in the L1 and L4 datasets but to exactly the same performance in the M1 dataset (Figure 7). This suggests that feature selection and standardisation can be useful heuristics when the sample size is low, but are not needed for larger sample sizes.

## 2.10 | Sparse RRR with $r \neq 2$

For the L1 and L4 datasets, cross-validation suggested $r = 2$ as the optimal rank, conveniently allowing us to use two-dimensional scatter plots for visualisation. For the M1 dataset we used $r = 2$ for visualisation, despite cross-validation suggesting that a higher rank could achieve better predictive performance. In this case, one can use a higher rank and display several biplots for different pairs of components, or alternatively perform separate RRR analyses on different subsets of the data. For this, we refer to our parallel publication describing the M1 dataset using $r = 5$ (Scala et al., 2020). Importantly, we found that the first two components of the rank-5 model were very similar to the two components of the rank-2 model shown here, reassuringly suggesting that the choice of $r$ does not strongly affect the leading components.

We observed an opposite case when we applied sparse RRR to the S1 dataset with $n = 80$ inhibitory (all *Cck* from layers 1/2) and excitatory neurons from mouse somatosensory cortex (Fuzik et al., 2016). The first RRR component strongly separated excitatory and inhibitory neurons (Figure 8), which is not surprising given the large differences in gene expression and in firing patterns between these two classes of neurons. However, subsequent RRR components did not carry much signal in this

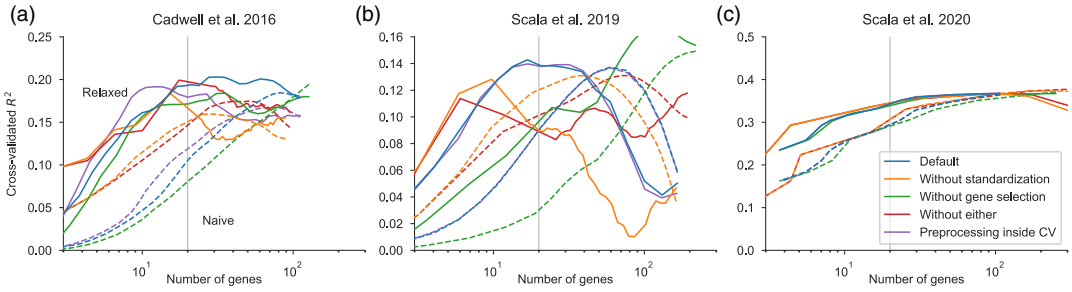



**FIGURE 7** (a) Cross-validated $R^2$ in the L1 dataset with $r = 2$, $\alpha = 0.5$, and different values of $\lambda$. Solid lines: relaxed version, dashed lines: naive version. Colours code different preprocessing choices. Blue: default approach used in other figures. Orange: without standardising the columns of $\mathbf{X}$. Green: without selecting most variable columns of $\mathbf{X}$. Red: without either. Purple: gene selection and standardisation done within the CV loop on each training set separately (and applied to the test set). (b) The same for the L4 dataset. (c) The same for the M1 dataset. Note that the V1 dataset is not analysed here, because for that dataset we used data that were already preprocessed by the original authors

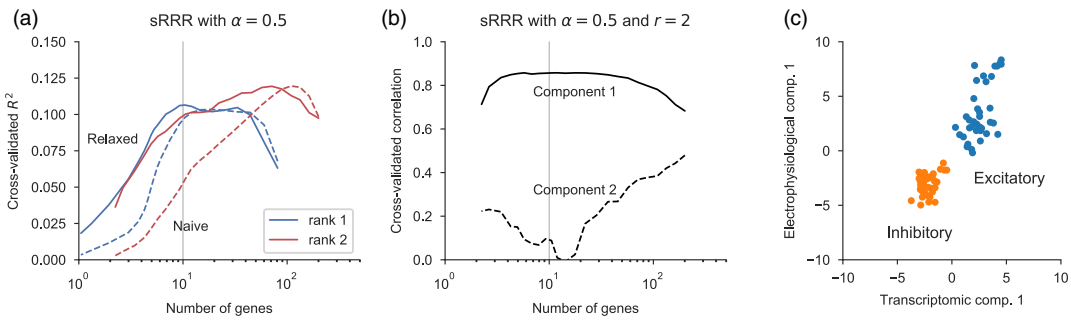



**FIGURE 8** (a) Cross-validation estimates of $R^2$ in sparse RRR with $r = 1$ and $r = 2$ ($\alpha = 0.5$) in the S1 dataset. Horizontal axis shows the average number of selected genes obtained for each $\lambda$. (b) Cross-validation estimates of correlations between the transcriptomic and the electrophysiological RRR components with $r = 2$ and $\alpha = 0.5$. (c) The RRR component using $r = 1$ and $\alpha = 0.5$ in the transcriptomic space (horizontal axis) and in the electrophysiological space (vertical axis). The value of $\lambda$ was chosen to yield 10 selected genes

dataset. The RRR model with $r = 1$ and $\alpha = 0.5$ outperformed the model with $r = 2$ for low number of selected genes (Figure 8a), while the correlation in the second component pair was close to zero (Figure 8b). This suggests effectively a one-dimensional shared subspace in this dataset.

## 3 | DISCUSSION

We proposed sRRR as a tool for interpretable data exploration and visualisation of Patch-seq recordings as an example of paired multivariate datasets in general. It allows to visualise the variability across cells in transcriptomic and electrophysiological modalities in a consistent way, and to find a sparse set of genes explaining electrophysiological variability. We used cross-validation to tune the hyperparameters and to estimate the out-of-sample performance of the model. The method has already been used in our parallel publications (Scala et al., 2019, 2020).

## 3.1 | Comparison to other regression methods

Our method directly builds up on the sparse RRR of Chen and Huang (2012) who added the lasso penalty to the RRR loss function. We extended this approach by using an elastic net penalty that combines lasso and ridge regularisation. This adds flexibility to the method and indeed we showed that in some cases non-zero ridge penalty is beneficial for the predictive performance (Figure 3) and for model stability. In addition, we introduced the relaxed elastic net approach to mitigate the over-shrinkage bias associated with naive elastic net or naive lasso solutions (De Mol et al., 2009; Efron et al., 2004; Meinshausen, 2007; Zou & Hastie, 2005). The sparse RRR method of Chen and Huang (2012) corresponds to our naive RRR with $\alpha = 1$, which in our experiments performed much worse than the relaxed version (Figure 3).

Furthermore, on some of the datasets sRRR outperformed sparse full-rank regression (which is directly available, for example in the popular `glmnet` library), suggesting that reduced-rank constraint is not only useful for visualisation but can also provide additional regularisation and reduces overfitting.

Elastic net regularisation has two parameters, $\alpha$ and $\lambda$, and cross-validation sometimes indicated that $\alpha$ can be varied in some range without affecting the model performance (Figure 3). This allows the researcher to control the trade-off between a sparser solution and a more comprehensive gene selection. If there is a set of genes that are highly correlated among each other, then large $\alpha$ will tend to select only one of them, whereas small $\alpha$ will tend to assign similar weights to all of them. We recommend using $\alpha = 0.5$ as a reasonable default compromise.

We followed Friedman et al. (2010), Chen and Huang (2012) and others in imposing sparsity only in the predictor space: the lasso penalty is applied only to $\mathbf{W}$ but not to $\mathbf{V}$ and so feature selection only happens on the columns of $\mathbf{X}$ but not of $\mathbf{Y}$. This arguably makes sense for the Patch-seq data considered in this manuscript but may be different for other kinds of datasets. See Chen et al. (2012a) for a discussion of sRRR with sparsity in both $\mathbf{X}$ and $\mathbf{Y}$.

## 3.2 | Comparison to other dimensionality reduction methods

Reduced rank-regression is closely related to two other classical dimensionality reduction methods analysing two paired data matrices (also called *two-view* data): CCA and PLS. They can be understood as looking for projections with maximal correlation (CCA) or maximal covariance (PLS) between $\mathbf{X}$ and $\mathbf{Y}$, whereas RRR looks for projections with maximal explained variance in $\mathbf{Y}$. In recent years, multiple approaches to sparse CCA (Chen et al., 2012b; Chu et al., 2013; Gao et al., 2017; Hardoon & Shawe-Taylor, 2011; Lykou & Whittaker, 2010; Mai & Zhang, 2019; Parkhomenko et al., 2009; Suo et al., 2017; Waaijenborg et al., 2008; Wiesel et al., 2008; Wilms & Croux, 2015; Witten & Tibshirani, 2009; Witten et al., 2009) and sparse PLS (Chun & Keleş, 2010; Lê Cao et al., 2008, 2011) have been suggested in the literature. Here, we chose sparse RRR at the core of our framework, because for the Patch-seq data it seems more meaningful to predict electrophyiological properties from transcriptomic information instead of treating them symmetrically, as genes give rise to physiological function. In addition, sparse RRR allows a mathematically simple formulation for rank $r > 1$ (using group lasso), whereas sparse PLS/CCA methods cited above are typically iterative: after extracting the $k$-th component, matrices $\mathbf{X}$ and $\mathbf{Y}$ are deflated and the algorithm is repeated to extract the $(k{+}1)$-th component, resulting in a cumbersome procedure that is often difficult to analyse mathematically.

That said, by comparing our sparse RRR method with sparse PLS/CCA method of Witten et al. (2009), we found that our method can be competitive as a CCA variant, at least for the kind of datasets studied here. On the other hand, the method of Witten et al. (2009) can also be used to construct a

bibiplot of Patch-seq data, using the same approach as developed here. The original papers (Witten & Tibshirani, 2009; Witten et al., 2009) did not discuss any such visualisations.

Sparse (and non-sparse) CCA and PLS have been applied to biological datasets in order to integrate multi-omics data (González et al., 2008, 2009, 2012; Lê Cao et al., 2008, 2009), with the recent `mixOmics` package for R providing a convenient implementation for several of these methods (Rohart et al., 2017). We believe that our sparse RRR can be a useful addition to this array of multi-omics statistical techniques.

This series of multi-omics papers always used two separate plots for visualising the CCA/PLS results within each modality: a *sample plot*, also called a *units plot* (in our case this would be a scatter plot of Patch-seq cells), and a *variable plot*, also called a *correlation circle plot* (in our case, this would be a scatter plot of selected genes or electrophysiological properties, together with the correlation circle). We found it convenient to combine these two plots into a single biplot (Gabriel, 1971). This allows to use two biplots (what we called a bibiplot) instead of four separate plots.

In ecology, RRR has been long used for dimensionality reduction and data visualisation, under the name of redundancy analysis (RDA) (Ramette, 2007; Ter Braak, 1994). This field uses biplots similar to the ones developed in this manuscript (Braak & Looman, 1994), sometimes combining both biplots into one single plot. A recently suggested sparse RDA (Csala et al., 2017) has similar aim to our work; their method can be seen as a variant of sparse PLS.

## 3.3 | Limitations and outlook

Following Friedman et al. (2010) and Chen and Huang (2012), we used group lasso that induces row-wise sparsity in $\mathbf{W}$. This means that the same set of genes is selected for all RRR components. For $r = 2$, as used in this manuscript, the same set of genes influences the first and the second component, which has both advantages and disadvantages. Our sparse RRR algorithm is easy to modify for the standard lasso case: using $\sum\|W_{i\cdot}\|_1$ in Equation (5) instead of $\sum\|W_{i\cdot}\|_2$ would induce element-wise sparsity. In this case, the loss in Equation (7) can be minimised separately for each column of $\mathbf{W}$ (e.g. also using `glmnet`). Using this approach, different genes can be selected for different RRR components and the same value of $\lambda$ can yield different number of selected genes for different components. However, when using relaxed elastic net and performing RRR again, all components will get non-zero contributions from all selected genes. Further work would be needed to formulate a relaxed version of the element-wise sparse RRR that would preserve element-wise sparsity. Empirically, we found that for the datasets considered here, the performance of the element-wise sparse RRR without relaxation was similar to the performance of the naive row-wise sparse RRR but worse than the performance of the relaxed row-wise sparse RRR.

One important caveat is that the list of selected genes should not be interpreted as definite. There are two reasons for that. First, the model performance (Figure 3) was often unaffected in some range of parameters corresponding to selecting from ∼10 to ∼100 genes, meaning that the choice of regularisation strength in this interval remains an analyst's call. Second, even for fixed regularisation parameters, a somewhat different set of genes may be selected every time the experiment is repeated. We used bootstrapping to directly estimate gene selection stability of sparse RRR, and found that the larger the sample size, the more stable the model was. There is an interplay between these two factors. Stronger $\ell_1$ regularisation leads to a sparser model with less selection reliability. Weaker $\ell_1$ regularisation leads to a less sparse model with higher selection reliability. We stress that such instability is an inherent feature of *all* sparse methods (Xu et al., 2011).

Single-cell RNA-seq data are notoriously noisy, with non-perfect sensitivity due to Poisson detection noise, and high variability due to variation in sequencing depths and other technical factors (Lause et al., 2020). These problems can be particularly strong in Patch-seq data where samples are collected manually, inducing additional variability (Lipovsek et al., 2021). Such data quality issues can make it difficult to detect an underlying biological signal using statistical methods. Here we used bootstrapping and

cross-validation to show that our method was nevertheless able to extract reliable and useful biological information.

In conclusion, we believe that sparse RRR can be a valuable tool for exploration and visualisation of paired datasets. We expect that our method can be relevant beyond the scope of Patch-seq data. For example, spatial transcriptomics (Lein et al., 2017) combined with two-photon imaging may allow characterising the transcriptome and physiology of individual cells in the intact tissue, yielding large multi-modal datasets. Similarly, other types of multi-omics data where single-cell or bulk transcriptomic data are combined with some other type of measurements (e.g. chemical, medical, or even behavioural), may benefit from interpretable visualisation techniques such as the one introduced here.

# 4 | METHODS

## 4.1 | Data preprocessing

### 4.1.1 | L1 dataset

We used read counts table from the original publication (Cadwell et al., 2016). In this dataset, there are $n = 51$ interneurons (from 53 sequenced interneurons, 2 were excluded in the original publication as contaminated), $p = 15{,}074$ genes identified by the authors as detected, and $q = 11$ electrophysiological properties. We excluded all cells for which at least one electrophysiological property was not estimated, resulting in $n = 44$. We restricted the gene pool to the $p = 3000$ most variable genes, the same ones identified in the original publication. We used the expert classification of cells into two classes performed in the original publication for annotating cell types. Out of $n = 44$ cells, only 35 cells were classified unambiguously (score 1 or score 5 on the scale from 1 to 5); the remaining 9 cells received intermediate scores. When performing cross-validation with gene selection in the CV loop, we used the gene selection procedure from Kobak and Berens (2019).

### 4.1.2 | S1 dataset

We used UMI counts table from the original publication (Fuzik et al., 2016). In this dataset, there are $n = 83$ cells, $p = 24{,}378$ genes after excluding ERCC spike-ins, and $q = 89$ electrophysiological properties. Out of 83 sequenced cells, we were only able to match $n = 80$ to the electrophysiological data. We used only $q = 80$ electrophysiological properties for which the data were available for all these cells (the fact that $n = q = 80$ is coincidental). We selected $p = 1{,}384$ genes with average expression above 0.5 (before standardisation) for the RRR analysis.

### 4.1.3 | L4 dataset

We used read counts table from the original publication (Scala et al., 2019). In this dataset, there are $n = 110$ Patch-seq neurons; we used the same $n = 102$ as in the original publication (after excluding low quality cells). We used the same $p = 1000$ genes selected in the original publication, and the same $q = 13$ electrophysiological properties.

### 4.1.4 | M1 dataset

We used read counts table from the original publication (Scala et al., 2020). In this dataset, there are $n = 1320$ Patch-seq neurons; we used the same $n = 1213$ as in the original publication (after excluding low quality cells). We used the same $p = 1000$ genes selected in the original publication, and the same $q = 16$ electrophysiological properties.

### 4.1.5 | V1 dataset

We used preprocessed data (after gene selection, normalisation, log-transformation, etc.) from https://github.com/AllenInstitute/coupledAE-patchseq (Gala et al., 2020) and further $z$-scored the electrophysiological features. This is a subset of the data from Gouwens et al. (2020); the entire dataset was not available at the time of writing.

### 4.1.6 | Preprocessing

For the full-length datasets (L1, L4, M1) we performed sequencing depth normalisation by converting the counts to counter per million (CPM). For the UMI-based dataset (S1), we divided the values for each cell by the cell sum over all genes (sequencing depth) and multiplying the result by the median sequencing depth size across all cells. In both cases we then log-transformed the data using $\log_2(x+1)$ transformation. Finally, we standardised all gene expression values and all electrophysiological properties to zero mean and unit variance.

## 4.2 | Data availability

All datasets were either downloaded following the links in the original publications or provided by the authors. All datasets can be found at https://github.com/berenslab/patch-seq-rrr. Our full analysis code in Python is also available there.

**ACKNOWLEDGEMENTS**

We thank Rickard Sandberg, Cathryn Cadwell, and Jiaolong Xiang for discussions, Cathryn Cadwell, Janos Fuzik and their co-authors for making their data available and Shreejoy Tripathy for comments and help with data processing. This work was funded by the German Ministry of Education and Research (FKZ 01GQ1601), the German Research Foundation (EXC 2064 project number 390727645, BE5601/4-1) and the National Institute Of Mental Health of the National Institutes of Health under Award Number U19MH114830. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

## AUTHORS' CONTRIBUTIONS

PB and DK conceptualised the project, DK developed the statistical method and wrote the software, DK, YB and MW performed computational experiments, FS performed Patch-seq experiments under

the supervision of AT and helped analysing the data, PB supervised the project. DK and PB wrote the paper with input from all authors.

## ORCID

*Dmitry Kobak* http://orcid.org/0000-0002-5639-7209
*Philipp Berens* https://orcid.org/0000-0002-0199-4727

## REFERENCES

Braak, C.J.F.T. & Looman, C.W.N. (1994) Biplots in reduced-rank regression. *Biometrical Journal*, 36 (8), 983–1003.

Cadwell, C.R., Palasantza, A., Jiang, X., Berens, P., Deng, Q., Yilmaz, M. et al. (2016) Electrophysiological, transcriptomic and morphologic profiling of single neurons using patch-seq. *Nature Biotechnology*, 34 (2), 199.

Cadwell, C.R., Scala, F., Li, S., Livrizzi, G., Shen, S., Sandberg, R. et al. (2017) Multimodal profiling of single-cell morphology, electrophysiology, and gene expression using Patch-seq. *Nature Protocols*, 12 (12), 2531.

Cawley, G.C. & Talbot, N.L.C. (2010) On over-fitting in model selection and subsequent selection bias in performance evaluation. *The Journal of Machine Learning Research*, 11, 2079–2107.

Chen, L. & Huang, J.Z. (2012) Sparse reduced-rank regression for simultaneous dimension reduction and variable selection. *Journal of the American Statistical Association*, 107 (500), 1533–1545.

Chen, K., Chan, K.-S. & Stenseth, N.C. (2012a) Reduced rank stochastic regression with a sparse singular value decomposition. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74 (2), 203–221.

Chen, X., Han, L. & Carbonell, J. (2012b) Structured sparse canonical correlation analysis. In: *Artificial Intelligence and Statistics*. pp. 199–207.

Chu, D., Liao, L.-Z., Ng, M.K. & Zhang, X. (2013) Sparse canonical correlation analysis: New formulation and algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35 (12), 3050–3065.

Chun, H. & Keleş, S. (2010) Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72 (1), 3–25.

Csala, A., Voorbraak, F.P.J.M., Zwinderman, A.H. & Hof, M.H. (2017) Sparse redundancy analysis of high-dimensional genetic and genomic data. *Bioinformatics*, 33 (20), 3228–3234.

De Mol, C., Mosci, S., Traskine, M. & Verri. A. (2009) Regularized method for selecting nested groups of relevant genes from microarray data. *Journal of Computational Biology*, 16 (5), 677–690.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R. et al. (2004) Least angle regression. *The Annals of Statistics*, 32 (2), 407–499.

Földy, C., Darmanis, S., Aoto, J., Malenka, R.C., Quake, S.R. & Südhof. T.C. (2016) Single-cell rnaseq reveals cell adhesion molecule profiles in electrophysiologically defined neurons. *Proceedings of the National Academy of Sciences*, 113 (35), E5222–E5231.

Friedman, J., Hastie, T. & Tibshirani, R. (2010) Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33 (1), 1.

Fuzik, J., Zeisel, A., Máté, Z., Calvigioni, D., Yanagawa, Y., Szabó, G. et al. (2016) Integration of electrophysiological recordings with single-cell RNA-seq data identifies neuronal subtypes. *Nature Biotechnology*, 34 (2), 175.

Gabriel, K.R. (1971) The biplot graphic display of matrices with application to principal component analysis. *Biometrika*, 58 (3), 453–467.

Gala, R., Budzillo, A., Baftizadeh, F., Miller, J.A., Gouwens, N.W., Arkhipov, A. et al. (2020) Consistent cross-modal identification of cortical neurons with coupled autoencoders. *Nature Computational Science*, 1, 120–127.

Gao, C., Ma, Z. & Zhou, H.H. (2017) Sparse CCA: Adaptive estimation and computational barriers. *The Annals of Statistics*, 45 (5), 2074–2101.

González, I., Déjean, S., Martin, P.G.P. & Baccini, A. (2008) CCA: An R package to extend canonical correlation analysis. *Journal of Statistical Software*, 23 (12), 1–14.

González, I., Déjean, S., Martin, P.G.P., Gonçalves, O., Besse, P. & Baccini, A. (2009) Highlighting relationships between heterogeneous biological data through graphical displays based on regularized canonical correlation analysis. *Journal of Biological Systems*, 17 (02), 173–199.

González, I., Lê Cao, K.-A., Davis, M.J. & Déjean, S. (2012) Visualising associations between paired 'omics' data sets. *BioData Mining*, 5 (1), 19.

Gouwens, N.W., Sorensen, S.A., Baftizadeh, F., Budzillo, A., Lee, B.R., Jarsky, T. et al. (2020) Integrated morphoelectric and transcriptomic classification of cortical gabaergic cells. *Cell*, 183 (4), 935–953.

Gower, J.C. & Dijksterhuis, G.B. (2004) *Procrustes problems*, volume 30. Oxford: Oxford University Press on Demand.

Hardoon, D.R. & Shawe-Taylor, J. (2011) Sparse canonical correlation analysis. *Machine Learning*, 83 (3), 331–353.

Harris, K.D., Hochgerner, H., Skene, N.G., Magno, L., Katona, L. Bengtsson, C. et al. (2018) Classes and continua of hippocampal CA1 inhibitory neurons revealed by single-cell transcriptomics. *PLoS Biology*, 16 (6), e2006387.

Izenman, A.J. (1975) Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5 (2), 248–264.

Kobak, D. & Berens, P. (2019) The art of using t-SNE for single-cell transcriptomics. *Nature Communications*, 10, 5416.

Lause, J., Berens, P. & Kobak, D. (2020) Analytic Pearson residuals for normalization of single-cell RNA-seq UMI data. *bioRxiv*.

Lê Cao, K.-A., Rossouw, D., Robert-Granié, C., & Besse, P. (2008) A sparse PLS for variable selection when integrating omics data. *Statistical Applications in Genetics and Molecular Biology*, 7 (1).

Lê Cao, K.-A., Martin, P.G.P., Robert-Granié, C. & Besse, P. (2009) Sparse canonical methods for biological data integration: Application to a cross-platform study. *BMC Bioinformatics*, 10 (1), 34.

Lê Cao, K.-A., Boitard, S. & Besse, P. (2011) Sparse PLS discriminant analysis: Biologically relevant feature selection and graphical displays for multiclass problems. *BMC Bioinformatics*, 12 (1), 253.

Lein, E., Borm, L.E. & Linnarsson, S. (2017) The promise of spatial transcriptomics for neuroscience in the era of molecular cell typing. *Science*, 358 (6359), 64–69.

Lipovsek, M., Bardy, C., Cadwell, C.R., Hadley, K., Kobak, D. & Tripathy, S.J. (2021) Patch-seq: Past, present, and future. *Journal of Neuroscience*, 41 (5), 937–946.

Luecken, M.D. & Theis, F.J. (2019) Current best practices in single-cell RNA-seq analysis: A tutorial. *Molecular Systems Biology*, 15 (6), e8746.

Lykou, A. & Whittaker, J. (2010) Sparse CCA using a lasso with positivity constraints. *Computational Statistics & Data Analysis*, 54 (12), 3144–3157.

Macosko, E.Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M. et al. (2015) Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161 (5), 1202–1214.

Mai, Q. & Zhang X. (2019) An iterative penalized least squares approach to sparse canonical correlation analysis. *Biometrics*, 75 (3), 734–744.

Masland, R.H. (2004) Neuronal cell types. *Current Biology*, 14 (13), R497–R500.

Meinshausen, N. (2007) Relaxed lasso. *Computational Statistics & Data Analysis*, 52 (1), 374–393.

Parkhomenko, E., Tritchler, D. & Beyene, J. (2009) Sparse canonical correlation analysis with application to genomic data integration. *Statistical Applications in Genetics and Molecular Biology*, 8 (1), 1–34.

Poulin, J.-F., Tasic, B., Hjerling-Leffler, J., Trimarchi, J.M. & Awatramani, R. (2016) Disentangling neural cell diversity using single-cell transcriptomics. *Nature Neuroscience*, 19 (9), 1131.

Ramette, A. (2007) Multivariate analyses in microbial ecology. *FEMS Microbiology Ecology*, 62 (2), 142–160.

Rohart, F., Gautier, B., Singh, A. & Lê Cao, K.-A. (2017) mixOmics: An R package for 'omics feature selection and multiple data integration. *PLoS Computational Biology*, 13 (11), e1005752.

Saunders, A., Macosko, E.Z., Wysoker, A., Goldman, M., Krienen, F.M., de Rivera, H. et al. (2018) Molecular diversity and specializations among the cells of the adult mouse brain. *Cell*, 174 (4), 1015–1030.

Scala, F., Kobak, D., Shan, S., Bernaerts, Y., Laturnus, S., Cadwell, C.R. et al. (2019) Layer 4 of mouse neocortex differs in cell types and circuit organization between sensory areas. *Nature Communications*, 10, 4174.

Scala, F., Kobak, D., Bernabucci, M., Bernaerts, Y., Cadwell, C.R., Castro, J.R. et al. (2020) Phenotypic variation of transcriptomic cell types in mouse motor cortex. *Nature*, 1–7. https://www.nature.com/articles/s41586-020-2907-3

Shekhar, K., Lapan, S.W., Whitney, I.E., Tran, N.M., Macosko, E.Z., Kowalczyk, M. et al. (2016) Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics. *Cell*, 166 (5), 1308–1323.

Suo, X., Minden, V., Nelson, B., Tibshirani, R. & Saunders, M. (2017) Sparse canonical correlation analysis. *arXiv*.

Tasic, B., Menon, V., Nguyen, T.N., Kim, T.K., Jarsky, T., Yao, Z. et al. (2016) Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. *Nature Neuroscience*, 19 (2), 335.

Tasic, B., Yao, Z., Graybuck, L.T., Smith, K.A. Nguyen, T., Bertagnolli, D. et al. (2018) Shared and distinct transcriptomic cell types across neocortical areas. *Nature*, 563 (7729), 72.

Ter Braak, C.J.F. (1994) Canonical community ordination. Part i: Basic theory and linear methods. *Ecoscience*, 1 (2), 127–140.

Tripathy, S.J., Toker, L., Li, B., Crichlow, C.-L., Tebaykin, D., Mancarci, B.O. et al. (2017) Transcriptomic correlates of neuron electrophysiological diversity. e1005814. *PLoS Computational Biology*, 13 (10),

Velu, R. & Reinsel, G.C. (2013) *Multivariate reduced-rank regression: theory and applications*, volume 136. Berlin: Springer Science & Business Media.

Waaijenborg, S., de Witt Hamer, P.C.V. & Zwinderman, A.H. (2008) Quantifying the association between gene expressions and DNA-markers by penalized canonical correlation analysis. *Statistical Applications in Genetics and Molecular Biology*, 7 (1).

Wiesel, A., Kliger, M. & Hero, III, A.O. (2008) A greedy approach to sparse canonical correlation analysis. *arXiv*.

Wilms, I. & Croux, C. (2015) Sparse canonical correlation analysis from a predictive point of view. *Biometrical Journal*, 57 (5), 834–851.

Witten, D.M. & Tibshirani, R.J. (2009) Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical Applications in Genetics and Molecular Biology*, 8 (1), 1–27.

Witten, D.M., Tibshirani, R. & Hastie, T. (2009) A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10 (3), 515–534.

Xu, H., Caramanis, C. & Mannor, S. (2011) Sparse algorithms are not stable: A no-free-lunch theorem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (1), 187–193.

Yuan, M. & Lin, Y. (2006) Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68 (1), 49–67.

Zeisel, A., Muñoz-Manchado, A.B., Codeluppi, S., Lönnerberg, P., La Manno, G., Juréus, A. et al. (2015) Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, 347 (6226), 1138–1142.

Zeisel, A., Hochgerner, H., Lonnerberg, P., Johnsson, A., Memic, F., van der Zwan, J. et al. (2018) Molecular architecture of the mouse nervous system. *Cell*, 174 (4), 999–1014.

Zeng, H. & Sanes, J.R. (2017) Neuronal cell-type classification: Challenges, opportunities and the path forward. *Nature Reviews Neuroscience*, 18 (9), 530.

Zou, H. & Hastie T. (2005) Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67 (2), 301–320.

**How to cite this article:** Kobak D, Bernaerts Y, Weis MA, Scala F, Tolias AS, Berens P. Sparse reduced-rank regression for exploratory visualisation of paired multivariate data. *J R Stat Soc Series C*. 2021;00:1–21. https://doi.org/10.1111/rssc.12494

## APPENDIX

### PROCRUSTES PROBLEM

Given $\mathbf{A}$, the Procrustes problem is to maximise $\mathrm{tr}(\mathbf{AV}^\top)$ subject to $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$ (Gower & Dijksterhuis, 2004). Let us denote by $\mathbf{A} = \mathbf{LQR}^\top = \tilde{\mathbf{L}}\tilde{\mathbf{Q}}\mathbf{R}^\top$ the thin and the full SVD of $\mathbf{A}$. Now we have:

$$\mathrm{tr}(\mathbf{AV}^\top) = \mathrm{tr}(\tilde{\mathbf{L}}\tilde{\mathbf{Q}}\mathbf{R}^\top\mathbf{V}^\top) = \mathrm{tr}(\tilde{\mathbf{Q}}\mathbf{R}^\top\mathbf{V}^\top\tilde{\mathbf{L}}) = \mathrm{tr}(\tilde{\mathbf{Q}}\mathbf{H}) = \sum q_i H_{ii} \le \sum q_i = \mathrm{tr}(\mathbf{Q}).$$

Here $\mathbf{H} = \mathbf{R}^\top\mathbf{V}^\top\tilde{\mathbf{L}}$ is a matrix with orthonormal rows as can be verified directly, and so it must have all its elements not larger than one. It follows that the whole trace is not larger than the sum of singular values of $\mathbf{A}$. Using $\mathbf{V} = \mathbf{LR}^\top$ yields exactly this value of the trace, hence it is the optimum.