DeCovarT, a multidimensional probalistic model for the deconvolution of heterogeneous transcriptomic samples

Bastien Chassagnol[1]

LPSM (Laboratoire de Probabilités, Statistiques et Modélisation), Sorbonne Université, 4, place Jussieu, 75252 PARIS, FRANCE
Institut De Recherches Servier (IDRS), FRANCE

Yufei Luo

Institut De Recherches Servier (IDRS), FRANCE

Grégory Nuel

LPSM (Laboratoire de Probabilités, Statistiques et Modélisation), CNRS 8001, Sorbonne Université, 4, place Jussieu, 75252 PARIS, FRANCE

Etienne Becht [2]

Institut De Recherches Internationales Servier (IRIS), FRANCE

Abstract

Although bulk transcriptomic analyses have greatly contributed to a better understanding of complex diseases, their sensibility is hampered by the highly heterogeneous cellular compositions of biological samples. To address this limitation, computational deconvolution methods have been designed to automatically estimate the frequencies of the cellular components that make up tissues, typically using reference samples of physically purified populations. However, they perform badly at differentiating closely related cell populations.

We hypothesised that the integration of the covariance matrices of the reference samples could improve the performance of deconvolution algorithms. We therefore developed a new tool, DeCovarT, that integrates the structure of individual cellular transcriptomic network to reconstruct the bulk profile. Specifically, we inferred the ratios of the mixture components by a standard maximum likelihood estimation (MLE) method, using the Levenberg-Marquardt algorithm to recover the maximum from the parametric convolutional distribution of our model. We then consider a reparametrisation of the log-likelihood to explicitly incorporate the simplex constraint on the ratios. Preliminary numerical simulations suggest that this new algorithm outperforms previously published methods, particularly when individual cellular transcriptomic profiles strongly overlap.

2012 ACM Subject Classification Applied computing → Computational transcriptomics, Biological networks; Mathematics of computing → Multivariate statistics

Keywords and phrases Cellular Deconvolution Bulk Transcriptome gLasso Optimisation under simplex constraint Levenberg-Marquardt

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Supplementary Material The source code used to prepare this document, notably the DeCovarT algorithm used to infer the ratios of transcriptomic samples; can be found as follows :
Software (Source Code): `https://github.com/bastienchassagnol/DeCovarT` [3]
Image (Boxplot and Heatmap of bootstrapped estimates, in a bivariate and high-dimensional framework, compared to 8 standard deconvolution methods.): `https://github.com/bastienchassagnol/DeCovarT/figs`

[1] Corresponding author
[2] Corresponding author and supervisor

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:12
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Dataset (Bootstrapped estimation datasets): `https://github.com/bastienchassagnol/DeCovarT/data`

1 Introduction

The analysis of the bulk transcriptome provided new insights on the mechanisms underlying disease development. However, such methods ignore the intrinsic cellular heterogeneity of complex biological samples, by averaging measurements over several distinct cell populations. Failure to account for changes of the cell composition is likely to result in a loss of *specificity* (genes mistakenly identified as differentially expressed, while they only reflect an increase in the cell population naturally producing them) and *sensibility* (genes expressed by minor cell populations are amenable being masked by highly variable expression from major cell populations).

Accordingly, a range of computational methods have been developed to estimate cellular fractions, but they perform poorly in discriminating cell types displaying high phenotypic proximity. Indeed, most of them assume that purified cell expression profiles are fixed observations, omitting the variability and intrinsically interconnected structure of the transcriptome. For instance, the gold-standard deconvolution algorithm *CIBERSORT* ([13]) applies nu-support vector regression (ν-SVR) to recover the minimal subset of the most informative genes in the purified signature matrix. However, this machine learning approach assumes that the transcriptomic expressions are independent.

In contrast to these approaches, we hypothesised that integrating the pairwise covariance of the genes into the reference transcriptome profiles could enhance the performance of transcriptomic deconvolution methods. The generative probabilistic model of our algorithm, *DeCovarT* (Deconvolution using the Transcriptomic Covariance), implements this integrated approach.

2 Model

First, we introduce the following notations:

- $(\boldsymbol{y} = (y_{gi}) \in \mathbb{R}_+^{G\times N}$ is the global bulk transcriptomic expression, measured in N individuals.
- $\boldsymbol{X} = (x_{gj}) \in \mathcal{M}_{\mathbb{R}^{G\times J}}$ the signature matrix of the mean expression of G genes in J purified cell populations.
- $\boldsymbol{p} = (p_{ji}) \in]0,1[^{J\times N}$ the unknown relative proportions of cell populations in N samples

As in most traditional deconvolution models, we assume that the total bulk expression can be reconstructed by summing the individual contributions of each cell population weighted by its frequency, as stated explicitly in the following linear matricial relationship (Equation (1)):

$$\boldsymbol{y} = \boldsymbol{X} \times \boldsymbol{p} \tag{1}$$

In addition, we consider unit simplex constraint on the cellular ratios, $\boldsymbol{p}$ (Equation (2)):

$$\begin{cases} \sum_{j=1}^{J} p_j = 1 \\ \forall j \in \widetilde{J} \quad p_j \geq 0 \end{cases} \tag{2}$$

2.1 Standard linear deconvolution model

However, in real conditions with technical and environmental variability, strict linearity of the deconvolution does not usually hold. Thus, an additional error term is usually considered, and without further assumption on the distribution of this error term, the usual approach to retrieve the best of parameters is by minimising the squared error term between the mixture expressions predicted by the linear model and the actual observed response. This optimisation task is achieved through the ordinary least squares (OLS) approach (Equation (3)),

$$\hat{\boldsymbol{p}}_i^{\mathrm{OLS}} \equiv \arg\min_{\boldsymbol{p}_i} ||\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i||^2 = \arg\min_{\boldsymbol{p}_i} ||\boldsymbol{X}\boldsymbol{p}_i - \boldsymbol{y}_i||^2 = \sum_{g=1}^{G} \left(y_{gi} - \sum_{j=1}^{J} x_{gj} p_{ji} \right) \tag{3}$$

If we additionally assume that the stochastic error term follows a *homoscedastic* zero-centred Gaussian distribution and that the value of the observed covariates (here, the purified expression profiles) is determined (see the corresponding graphical representation in Figure 1a and the set of equations describing it Equation (4)),

$$y_{gi} = \sum_{j=1}^{J} x_{gj} p_{ji} + \epsilon_i, \quad y_{gi} \sim \mathcal{N}\left(\sum_{j=1}^{J} x_{gj} p_{ji}, \sigma_i^2 \right), \quad \epsilon_i \sim \mathcal{N}(0, \sigma_i^2) \tag{4}$$

then, the MLE is equal to the OLS, which, in this framework, is given explicitly by Equation (5):

$$\hat{\boldsymbol{p}}_i^{\mathrm{OLS}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}_i \tag{5}$$

This result, known as the Gaussian-Markow theorem, is reported along with all the mandatory assumptions, in theorem Theorem 1, in Appendix A.1.

2.2 Motivation of using a probalistic convolution framework

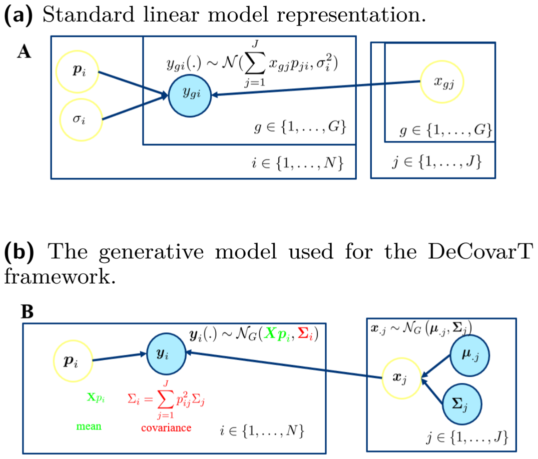
In contrast to standard linear regression models, we relax in the DeCovarT modelling framework the *exogeneity* assumption, by considering the set of covariates $\boldsymbol{X}$ as random variables rather than fixed measures, in a process close to the approach of DSection algorithm and DeMixt algorithms. However, to our knowledge, we are the first to weaken the independence assumption between observations by explicitly considering a multivariate distribution and integrating the intrinsic covariance structure of the transcriptome of each purified cell population.

To do so, we conjecture that the G-dimensional vector $\boldsymbol{x}_j$ characterising the transcriptomic expression of each cell population follows a multivariate Gaussian distribution, given by Equation (6):

$$\mathrm{Det}(2\pi\boldsymbol{\Sigma}_j)^{-\frac{1}{2}} \exp\left(-\frac{1}{2} (\boldsymbol{x}_j - \boldsymbol{\mu}_{.j}) \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_{.j})^\top \right) \tag{6}$$

and parametrised by:

- $\boldsymbol{\mu}_{.j}$, the mean purified transcriptomic expression of cell population j
- $\boldsymbol{\Sigma}_j$, the *positive-definite* (see Definition Definition 2) covariance matrix of each cell population. Precisely, we retrieve it from inferring its inverse, known as the precision matrix, through the gLasso [9] algorithm. We define $\boldsymbol{\Theta}_j \equiv \boldsymbol{\Sigma}_j^{-1}$ the corresponding *precision matrix*, whose inputs, after normalisation, store the partial correlation between



(a) Standard linear model representation.

A
p_i
σ_i
$y_{gi}(.) \sim \mathcal{N}(\sum_{j=1}^{J} x_{gj}p_{ji}, \sigma_i^2)$
y_{gi}
x_{gj}
$g \in \{1,\dots,G\}$
$g \in \{1,\dots,G\}$
$i \in \{1,\dots,N\}$
$j \in \{1,\dots,J\}$

(b) The generative model used for the DeCovarT framework.

B
$\boldsymbol{y}_i(.) \sim \mathcal{N}_G(\boldsymbol{X}\boldsymbol{p}_i, \boldsymbol{\Sigma}_i)$
$\boldsymbol{p}_i$
$\boldsymbol{y}_i$
$\mathrm{X}p_i$
mean
$\Sigma_i = \sum_{j=1}^{J} p_{ij}^2 \Sigma_j$
covariance
$i \in \{1,\dots,N\}$
$\boldsymbol{x}_{.j} \sim \mathcal{N}_G(\boldsymbol{\mu}_{.j}, \boldsymbol{\Sigma}_j)$
$\boldsymbol{x}_j$
$\boldsymbol{\mu}_{.j}$
$\boldsymbol{\Sigma}_j$
$j \in \{1,\dots,J\}$

Figure 1 We use the standard graphical convention of graphical models, as depicted in RevBayes webpage. For identifiability reasons, we conjecture that all variability proceeds from the stochastic nature of the covariates.

two genes, conditioned on all the others. Notably, pairwise gene interactions whose corresponding off-diagonal terms in the precision matrix are null are considered statistically spurious, and discarded.

To derive the log-likelihood of our model, first we *plugged-in* the mean and covariance parameters $\zeta_j = \left(\boldsymbol{\mu}_{.j}, \boldsymbol{\Sigma}_j\right)$ estimated for each cell population in the previous step. Then, setting $\boldsymbol{\zeta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\boldsymbol{\mu} = (\boldsymbol{\mu}_{.j})_{j \in \widetilde{J}} \in \mathcal{M}_{G\times J}$, $\boldsymbol{\Sigma} \in \mathcal{M}_{G\times G}$ the known parameters and $\boldsymbol{p}$ the unknown cellular ratios, we show that the conditional distribution of the observed bulk mixture, conditioned on the individual purified expression profiles and their ratios in the sample, $\boldsymbol{y}|(\boldsymbol{\zeta}, \boldsymbol{p})$, is the convolution of pairwise independent multivariate Gaussian distributions. Using the *affine invariance* property of Gaussian distributions, we can show that this convolution is also a multivariate Gaussian distribution, given by Equation (7).

$$\boldsymbol{y}|(\boldsymbol{\zeta}, \boldsymbol{p}) \sim \mathcal{N}_G(\boldsymbol{\mu}\boldsymbol{p}, \boldsymbol{\Sigma}) \text{ with } \boldsymbol{\mu} = (\boldsymbol{\mu}_{.j})_{j \in \widetilde{J}}, \quad \boldsymbol{p} = (p_1, \dots, p_J) \text{ and } \boldsymbol{\Sigma} = \sum_{j=1}^{J} p_j^2 \boldsymbol{\Sigma}_j \tag{7}$$

. The DAG associated to this modelling framework is shown in Figure Figure 1b).

In the next section, we provide an explicit formula of the log-likelihood of our probabilistic framework, its gradient and hessian, which in turn can be used to retrieve the MLE of our distribution.

2.3 Derivation of the log-likelihood

From Equation (7), the conditional log-likelihood is readily computed and given by Equation (8):

$$\ell_{\boldsymbol{y}|\boldsymbol{\zeta}}(\boldsymbol{p}) = C + \log\left(\operatorname{Det}\left(\sum_{j=1}^{J} p_j^2 \boldsymbol{\Sigma}_j\right)^{-1}\right) - \frac{1}{2}(\boldsymbol{y} - \boldsymbol{p}\boldsymbol{\mu})^\top \left(\sum_{j=1}^{J} p_j^2 \boldsymbol{\Sigma}_j\right)^{-1} (\boldsymbol{y} - \boldsymbol{p}\boldsymbol{\mu}) \tag{8}$$

2.4 First and second-order derivation of the unconstrained DeCovarT log-likelihood function

The stationary points of a function and notably maxima, are given by the roots (the values at which the function crosses the x-axis) of its gradient, in our context, the vector: $\nabla\ell : \mathbb{R}^J \rightarrow \mathbb{R}^J$ evaluated at point $\nabla\ell(\boldsymbol{p}) :]0,1[^J \rightarrow \mathbb{R}^J$. Since the computation is the same for any cell ratio p_j, we give an explicit formula for only one of them (Equation (9)):

$$\begin{aligned}\frac{\partial \ell_{\boldsymbol{y}|\boldsymbol{\zeta}}(\boldsymbol{p})}{\partial p_j} &= \tfrac{\partial \log(\mathrm{Det}(\boldsymbol{\Theta}))}{\partial p_j} - \tfrac{1}{2}\left[\tfrac{\partial(\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p})^\top}{\partial p_j}\boldsymbol{\Theta}(\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p}) + (\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p})^\top\tfrac{\partial\boldsymbol{\Theta}}{\partial p_j}(\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p}) + (\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p})^\top\boldsymbol{\Theta}\tfrac{\partial(\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p})}{\partial p_j}\right] \\ &= -\mathrm{Tr}\left(\boldsymbol{\Theta}\tfrac{\partial\boldsymbol{\Sigma}}{\partial p_j}\right) - \tfrac{1}{2}\left[-\boldsymbol{\mu}_{.j}^\top\boldsymbol{\Theta}(\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p}) - (\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p})^\top\Theta\tfrac{\partial\Sigma}{\partial p_j}\Theta(\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p}) - (\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p})^\top\boldsymbol{\Theta}\boldsymbol{\mu}_{.j}\right] \\ &= -2p_j\,\mathrm{Tr}\left(\boldsymbol{\Theta}\boldsymbol{\Sigma}_j\right) + (\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p})^\top\boldsymbol{\Theta}\boldsymbol{\mu}_{.j} + p_j(\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p})^\top\boldsymbol{\Theta}\Sigma_j\boldsymbol{\Theta}(\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p})\end{aligned} \tag{9}$$

Since the solution to $\nabla\left(\ell_{\boldsymbol{y}|\boldsymbol{\zeta}}(\boldsymbol{p})\right) = 0$ is not closed, we had to approximate the MLE using iterated numerical optimisation methods. Some of them, such as the Levenberg–Marquardt algorithm, require a second-order approximation of the function, which needs the computation of the Hessian matrix. Deriving once more Equation (9) yields the Hessian matrix, $\mathbf{H} \in \mathcal{M}_{J\times J}$ is given by:

$$\begin{aligned}\mathbf{H}_{i,i} = \frac{\partial^2\ell}{\partial^2 p_i} &= -2\,\mathrm{Tr}\left(\boldsymbol{\Theta}\boldsymbol{\Sigma}_i\right) + 4p_i^2\,\mathrm{Tr}\left(\left(\boldsymbol{\Theta}\boldsymbol{\Sigma}_i\right)^2\right) - 2p_i(\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p})^\top\boldsymbol{\Theta}\boldsymbol{\Sigma}_i\boldsymbol{\Theta}\boldsymbol{\mu}_{.\boldsymbol{i}} - \boldsymbol{\mu}_{.i}^\top\boldsymbol{\Theta}\boldsymbol{\mu}_{.\boldsymbol{i}} - \\ &\quad 2p_i(\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p})^\top\boldsymbol{\Theta}\boldsymbol{\Sigma}_i\boldsymbol{\Theta}\boldsymbol{\mu}_{.i} - (\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p})^\top\boldsymbol{\Theta}\left(4p_i^2\boldsymbol{\Sigma}_i\boldsymbol{\Theta}\boldsymbol{\Sigma}_i - \boldsymbol{\Sigma}_i\right)\boldsymbol{\Theta}(\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p}), \quad i \in \widetilde{J} \\ \mathbf{H}_{i,j} = \frac{\partial^2\ell}{\partial p_i\partial p_j} &= 4p_jp_i\,\mathrm{Tr}\left(\boldsymbol{\Theta}\boldsymbol{\Sigma}_j\boldsymbol{\Theta}\boldsymbol{\Sigma}_i\right) - 2p_i(\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p})^\top\boldsymbol{\Theta}\boldsymbol{\Sigma}_i\boldsymbol{\Theta}\boldsymbol{\mu}_{.\boldsymbol{j}} - \boldsymbol{\mu}_{.i}^\top\boldsymbol{\Theta}\boldsymbol{\mu}_{.\boldsymbol{j}} - \\ &\quad 2p_j(\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p})^\top\boldsymbol{\Theta}\boldsymbol{\Sigma}_j\boldsymbol{\Theta}\boldsymbol{\mu}_{.i} - 4p_ip_j(\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p})^\top\boldsymbol{\Theta}\boldsymbol{\Sigma}_i\boldsymbol{\Theta}\boldsymbol{\Sigma}_j\boldsymbol{\Theta}(\boldsymbol{y}-\boldsymbol{\mu}\boldsymbol{p}), \quad (i,j) \in \widetilde{J}^2, i \neq j\end{aligned} \tag{10}$$

in which the coloured sections pair one by one with the corresponding coloured sections of the gradient, given in Equation (9). Matrix calculus can largely ease the derivation of complex algebraic expressions, thus we remind in Appendix (*Matrix calculus*) relevant matrix properties and derivations [3].

However, the explicit formulas for the gradient and the hessian matrix of the log-likelihood function, given in Equation (9) and Equation (10) respectively, do not take into account the simplex constraint assigned to the ratios. While some optimisation methods use heuristic methods to solve this problem, we consider alternatively a reparametrised version of the problem, detailed comprehensively in Appendix Appendix A.3.

3 Simulations

3.1 Simulation of a convolution of multivariate Gaussian mixtures

To assert numerically the relevance of accounting the correlation between expressed transcripts, we designed a simple toy example with two genes and two cell proportions. Hence, using the simplex constraint (Equation (2)), we only have to estimate one free unconstrained parameter, θ_1, and then uses the mapping function Equation (12) to recover the ratios.

[3] The numerical consistency of these derivatives was asserted with the **numDeriv** package, using the more stable Richardson's extrapolation ([6]).

We simulated the bulk mixture, $\boldsymbol{y} \in \mathcal{M}_{G\times N}$, for a set of artificial samples $N = 500$, with the following generative model:

- We have tested two levels of cellular ratios, one with equi-balanced proportions ($\boldsymbol{p} = (p_1, p_2 = 1-p_1) = (\frac{1}{2}, \frac{1}{2})$ and one with highly unbalanced cell populations: $\boldsymbol{p} = (0.95, 0.05)$.
- Then, each purified transcriptomic profile is drawn from a multivariate Gaussian distribution. We compared two scenarios, playing on the mean distance of centroids, respectively $\mu_{.1} = (20, 22), \mu_{.2} = (22, 20)$ and $\mu_{.2} = (20, 40), \mu_{.2} = (40, 20))$ and building the covariance matrix, $\boldsymbol{\Sigma} \in \mathcal{M}_{2\times 2}$ by assuming equal individual variances for each gene (the diagonal terms of the covariance matrix, $\text{Diag}(\boldsymbol{\Sigma_1}) = \text{Diag}(\boldsymbol{\Sigma_1}) = \boldsymbol{I}_2$) but varying the pairwise correlation between gene 1 and gene 2, $\mathbb{C}\text{ov}\,[x_{1,2}]$, on the following set of values: $\{-0.8, -0.6, \dots, 0.8\}$ for each of the cell population.
- As stated in Equation (1), we assume that the bulk mixture, $\mathbf{y}_{.i}$ could be directly reconstructed by summing up the individual cellular contributions weighted by their abundance, without additional noise.

3.2 Iterated optimisation

The extremum, and by extension the MLE, is a root of the gradient of the log-likelihood. However, in our generative framework, the inverse function cancelling the gradient of Equation Equation (8) is non-closed. Instead, iterated numerical optimisation algorithms that consider first or second-order approximations of the function to optimise are used to approximate the roots.

The *Levenberg-Marquardt (LM)* algorithm bridges the gap between between the steepest descent method (first-order) and the Newton-Raphson method (second-order) by inflating the diagonal terms of the Hessian matrix. Far from the endpoint, a second-order descent is favoured for its faster convergence pace, while the steepest approach is privileged close to the extremum since it allows careful refinement of the step size. Specially, we used the LM implementation of R package **marqLevAlg** to infer the ratios $\hat{\boldsymbol{p}}$ from the bootstrap simulations, since it includes an additional convergence criteria, the relative distance to the maximum (RDM), that sets apart extrema from spurious saddle points.

3.3 Results

We compared the performance of DeCovarT algorithm with the outcome of a quadratic algorithm that specifically addresses the unit simplex constraint: the negative least squares algorithm (NNLS, [8]).

Even with a limited toy example including two cell populations characterised only by two genes, we observe that the overlap was a good proxy of the quality of the estimation: the less the overlap between the two cell distributions, the better the quality of the estimation Figure 2.

The package used to generate the simulations and infer ratios from virtual or real biological mixtures with the DeCovarT algorithm is implemented on my personal Github account DeCovarT.

4 Perspectives

The new deconvolution algorithm that we implemented, DeCovarT, is the first one based on a multivariate generative model while complying explicitly the simplex constraint. Hence, it

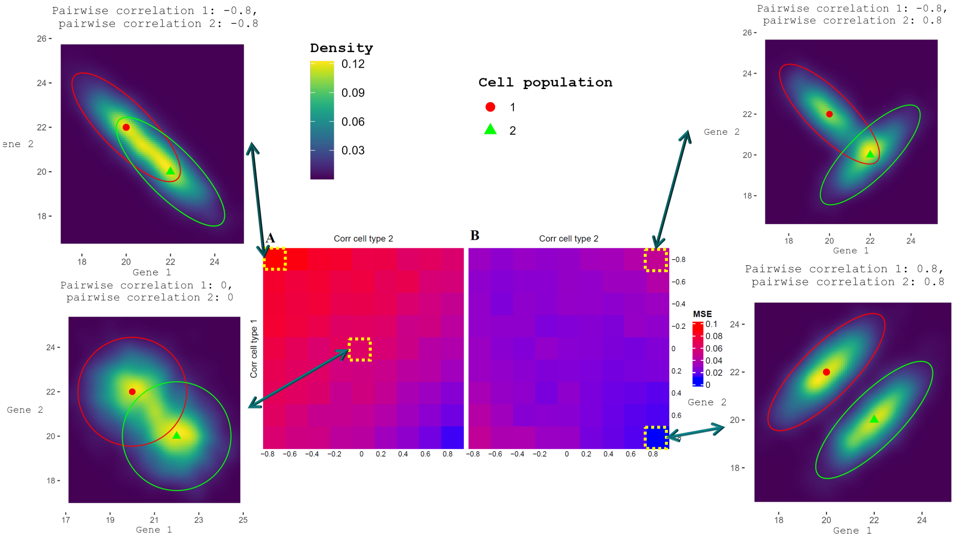


Figure 2 We used the package **ComplexHeatmap** to display the mean square error (MSE) of the estimated cell ratios, comparing the NNLS output, as implemented in the deconRNASEQ algorithm ([7]), in Panel **A**, with our newly implemented DeCovarT algorithm, in Panel **B**. The lower the MSE, the least noisy and biased the estimates. In addition, we added the two-dimensional density plot for the intermediate scenario, for which each population is parameterised by a diagonal covariance matrix, and the most extreme scenarios (those with the highest correlation between genes). The ellipsoids represent for each cell population the 95% confidence region and the red spherical icon and the green triangular icon represent respectively the centroids (average expression of gene 1 and gene 2) of cell population 1 and cell population 2.

provides a strong basis to further derive statistical tests to assert whether the proportion of a given cell population differs significantly between two distinct biological conditions.

However, we still need to assert its performance in an extended simulation framework. In a numerical setting, we could first increase the dimensionality of our purified datasets by using more realistic parametrisations, using the mean and sparse covariance parameters inferred from purified cellular datasets. Then, we need to evaluate our algorithm in a real-world experience, with both blood and tumoral samples. The Kassandra project would be a good place to start, since the purified database collects a compendium of 9,404 cellular transcriptomic profiles, annotated into 38 blood cellular populations and the performance of Kassandra's algorithm was benchmarked in $N = 517$ samples in 6 public datasets with both flow cytometry annotations and bulk RNA-seq expression, against 8 different standard deconvolution algorithms: 5 reference profile deconvolution algorithms: EPIC [14], CIBERSORT [13], CIBERSORTx [12], quanTIseq [5] and ABIS [11], and 3 marker-based deconvolution algorithms [4]: MCPcounter [2], xCell [1] and Scaden [10].

Finally, the gLasso algorithm used to derive each purified cell accuracy matrix, like any penalty regularisation approach, is subject to *parameter shrinkage*. Notably, in our setting, shrinkage leads to systematically underestimate the non-zero partial correlations of the precision matrix. A way to circumvent this problem is to only use the *support* (the non-null inputs) output of the gLasso and use the associated topological constraints within a standard

[4] Contrary to algorithms based on signature references, marker-based algorithms make the strong asssumption that any discriminant gene, referred to as *marker* is uniquely expresssed in a cell population.

MLE approach to fine-tune the inputs of the precision matrix. One way of doing so would be to infer a directed Gaussian Graphical Model (GGM), however, except in really specific topological configurations, such as chordal graphs, there is no current direct equivalence between the space of undirected Markov graphs, as returned by gLasso, and directed Bayesian graphs ([4]).

References

1 Aran, Dvir and Hu, Zicheng and others. xCell: Digitally Portraying the Tissue Cellular Heterogeneity Landscape. *Genome Biology*, 2017. doi:10.1186/s13059-017-1349-1.

2 Etienne Becht, Nicolas A. Giraldo, Laetitia Lacroix, Bénédicte Buttard, Nabila Elarouci, Florent Petitprez, Janick Selves, Pierre Laurent-Puig, Catherine Sautès-Fridman, Wolf H. Fridman, and Aurélien de Reyniès. Estimating the population abundance of tissue-infiltrating immune and stromal cell populations using gene expression. *Genome Biology*, 2016. doi:10.1186/s13059-016-1070-5.

3 Bastien Chassagnol, Yufei Luo, Grégory Nuel, and Etienne Becht. DeCovarT, a R package for the robust deconvolution of cell mixtures in transcriptomic samples, 2023. URL: https://github.com/bastienchassagnol/DeCovarT.

4 Joachim Dahl, Vwani Roychowdhury, and Lieven Vandenberghe. Maximum likelihood estimation of gaussian graphical models: numerical implementation and topology selection. *SIAM journal*, 2005.

5 Finotello, Francesca and Mayer, Clemens and others. Molecular and Pharmacological Modulators of the Tumor Immune Contexture Revealed by Deconvolution of RNA-seq Data. *Genome Medicine*, 2019. doi:10.1186/s13073-019-0638-6.

6 Bengt Fornberg. Numerical differentiation of analytic functions. *ACM Trans. Math. Softw.*, 7(4):512–526, 1981. doi:10.1145/355972.355979.

7 Gong, Ting and Szustakowski, Joseph D. DeconRNASeq: A Statistical Framework for Deconvolution of Heterogeneous Tissue Samples Based on mRNA-Seq Data. *Bioinformatics (Oxford, England)*, 2013. doi:10.1093/bioinformatics/btt090.

8 Karen H. Haskell and Richard J. Hanson. An algorithm for linear least squares problems with equality and nonnegativity constraints. *Mathematical Programming*, 1981. doi:10.1007/BF01584232.

9 Mazumder, Rahul and Hastie, Trevor. The Graphical Lasso: New Insights and Alternatives. *Electronic Journal of Statistics*, 2011. doi:10.1214/12-EJS740.

10 Kevin Menden, Mohamed Marouf, Sergio Oller, Anupriya Dalmia, Daniel Sumner Magruder, Karin Kloiber, Peter Heutink, and Stefan Bonn. Deep learning–based cell composition analysis from tissue expression profiles. *Science Advances*, 2020. doi:10.1126/sciadv.aba2619.

11 Gianni Monaco, Bernett Lee, Weili Xu, Seri Mustafah, You Yi Hwang, Christophe Carré, Nicolas Burdin, Lucian Visan, Michele Ceccarelli, Michael Poidinger, Alfred Zippelius, João Pedro de Magalhães, and Anis Larbi. RNA-Seq Signatures Normalized by mRNA Abundance Allow Absolute Deconvolution of Human Immune Cell Types. *Cell Reports*, 2019. doi:10.1016/j.celrep.2019.01.041.

12 Aaron M. Newman, Chloé B. Steen, Chih Long Liu, Andrew J. Gentles, Aadel A. Chaudhuri, Florian Scherer, Michael S. Khodadoust, Mohammad S. Esfahani, Bogdan A. Luca, David Steiner, Maximilian Diehn, and Ash A. Alizadeh. Determining cell type abundance and expression from bulk tissues with digital cytometry. *Nature Biotechnology*, 2019. doi:10.1038/s41587-019-0114-2.

13 Newman, Aaron and Liu, Chih and others. Robust Enumeration of Cell Subsets from Tissue Expression Profiles. *Nature methods*, 2015. doi:10.1038/nmeth.3337.

14 Racle, Julien and de Jonge, Kaat and others. Simultaneous Enumeration of Cancer and Immune Cell Types from Bulk Tumor Gene Expression Data. *eLife*, 2017. doi:10.7554/eLife.26476.

A Theoretical details

A.1 Standard linear assumptions

Without noise or confusing variable, the solution to Equation (1) is *determined*, provided that the number of genes equals at least the number of cell types and that no purified cellular expression profile can be rewritten as a linear combination of the other cell populations, in other terms, that the reference matrix $\boldsymbol{X}$ is invertible. When the number of genes exceeds the number of cell populations, the problem is then *overdetermined*.

The general approach to solve an overdetermined system of equations consists of adding an unobserved error term, as considered in standard *linear regression*. The estimate returned by the ordinary least squares (OLS) approach is the one minimising the squared distance between the observed response variables, noted $\boldsymbol{y_i}$, and the predicted values, $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$ (Equation (5)):

$$\hat{\boldsymbol{p}}_i^{\text{OLS}} \equiv \arg\min_{\boldsymbol{p}_i} ||\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i||^2 = \arg\min_{\boldsymbol{p}_i} ||\boldsymbol{X}\boldsymbol{p}_i - \boldsymbol{y}_i||^2 = \sum_{g=1}^{G} \left(y_{gi} - \sum_{j=1}^{J} x_{gj} p_{ji} \right) \tag{11}$$

An alternative method consists of deriving a *generative model* that models the variability of the measured observations by drawing them from parametric probability density functions. The set of parameters, $\boldsymbol{p}$, that described the best the observations under that probabilistic framework, is termed the *maximum likelihood estimate* (MLE). Formally, the MLE maximises the likelihood or the log-likelihood of the response variable, conditioned on the set of independent covariates: $\ell(\boldsymbol{p}|\boldsymbol{y}, \boldsymbol{X})$. Under the assumptions listed in Theorem Theorem 1, the OLS is equal to the MLE estimate of a linear model :

▶ **Theorem 1** **(Gauss-Markov theorem).** *If the following assumptions hold,*

1. ***Strong exogeneity:*** *The cell type-specific expression profiles are not random variables but rather fixed and constant observations, underlying implicitly that cell populations do no interact:* $\forall i \in \widetilde{J}, \forall j \in \widetilde{J},\ i \neq j, \quad \mathbb{C}ov[\boldsymbol{x_{.i}}, \boldsymbol{x_{.j}}] = 0$.
2. ***Gaussian-Markov noise:*** *This hypothesis assumes an independent white Gaussian noise, of null mean and variance not depending on the gene (**homoscedasticity**), for the distribution of the error term. Formally, it is likewise to adding a Gaussian distributed error term,* $y_{gi} = \sum_{j=1}^{J} x_{gj} p_{ji} + \epsilon_{gi}$, $\epsilon_{gi} \sim \mathcal{N}\left(0, \sigma_{gi}^2\right)$, *from which, directly injecting the exogeneity and homoscedasticity properties, we deduce the univariate Gaussian nature of the distribution of each transcript:*
   $y_{gi} \sim \mathcal{N}\left(\sum_{j=1}^{J} x_{gj} p_{ji}, \sigma_i^2\right)$
3. ***Independence:*** *From the aforementioned Gaussian-Markov and exogeneity assumptions, we readily deduce that the gene expressions of the bulk measures are independent:* $\forall j \in \widetilde{G}, \forall k \in \widetilde{G}, j \neq k, \quad \mathbb{C}ov[y_{ji}, y_{ki}] = 0$.
4. ***Completeness:*** *We assume no additional latent variable, such as a non-surveyed cell population, that would contribute to the variability of the bulk mixture.*

*then, the MLE estimate is equal to the OLS estimate that is readily computed by the **Normal equations** (Equation* (5)*). Additionally, under the Gauss-Markov assumptions, the MLE is the unique BLUE (best linear unbiased estimator), i.e. the unbiased estimator with the lowest variance.*

A.2 Algebra Calculus Memo

In Equation (7), we coerce each covariance matrix to be positive-definite Definition 2

▶ **Definition 2.** *A symmetric real matrix $\boldsymbol{A}$ of rank G is said to be positive-definite if $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} > 0$ for all non-zero vectors $\boldsymbol{x}$ in $\mathbb{R}^G$.*

To better understand this constraint, consider a multivariate Gaussian distribution in which any of the individual features are pairwise independent. Such a distribution would be parametrised by a covariance matrix with only diagonal terms. Consider now that some of these terms are negative, this would imply that the individual variance for some of the covariates is negative, which is physically impossible and would make it impossible to define a proper probability distribution.

Basic linear algebra formulas, implying the transpose and the inverse of a matrix, and highly relevant to ease the derivation of complex functions involving matrices, are reported in Note 3:

▶ Note 3. First, we introduce below some properties associated to the determinant and trace operators. For a squared matrix A of rank G with defined inverse variance A^{-1} and a constant p, the following properties hold:

(a) $\operatorname{Det}(p\boldsymbol{A}) = p^G \operatorname{Det}(\boldsymbol{A})$ **(b)** $\operatorname{Tr}(p\boldsymbol{A}) = p \operatorname{Tr}(\boldsymbol{A})$ **(c)** $\operatorname{Det}(A^{-1}) = \frac{1}{\operatorname{Det}(A)}$

Then, we introduce some properties practical with the transpose operator, which switch the row and column indexes. Given two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, the following properties hold when computing their transpose:

(a) $(\boldsymbol{A}^\top)^\top = \boldsymbol{A}$ **(b)** $(\boldsymbol{A}\boldsymbol{B})^\top = \boldsymbol{B}^\top \boldsymbol{A}^\top$ **(c)** $\left(\boldsymbol{A}^{-1}\right)^\top = \boldsymbol{A}^{-1}$[5]

Another useful equality, given two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathbb{R}^G$ and $\boldsymbol{A}$ a symmetric matrix [6] of rank G: $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} = \boldsymbol{y}^\top \boldsymbol{A}\boldsymbol{x}$

The trace operator is additionally invariant under *cyclic permutation*. For instance, for three matrices with matching dimensions, we have $\operatorname{Tr}(\boldsymbol{ABC}) = \operatorname{Tr}(\boldsymbol{CAB}) = \operatorname{Tr}(\boldsymbol{BCA})$. ⌟

The most useful algebra calculus formulas used to derive respectively first order (Equation (9)) and second order (Equation (10)) of the log-likelihood function (Equation (8)) are reported in Note 4 and Note 5.

▶ Note 4. Given two invertible, positive-definite matrices (see Definition in Definition 2) $\boldsymbol{A}$ and $\boldsymbol{B}$, with respective inverses $\boldsymbol{A}^{-1}$ and $\boldsymbol{B}^{-1}$, $A = \boldsymbol{A}(p)$ and $B = \boldsymbol{B}(p)$ being functions of a scalar variable p, the following properties hold:

(a) $\frac{\partial \operatorname{Det}(\boldsymbol{A})}{\partial p} = \operatorname{Det}(\boldsymbol{A}) \operatorname{Tr}\left(\boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial p}\right)$ **(b)** $\frac{\partial \boldsymbol{UAV}}{\partial p} = \boldsymbol{U} \frac{\partial \boldsymbol{A}}{\partial p} \boldsymbol{V}$

(c) $\frac{\partial \boldsymbol{A}^{-1}}{\partial p} = -\boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial p} \boldsymbol{A}^{-1}$

From equation a), we can readily compute $\frac{\partial \log(\operatorname{Det}(\boldsymbol{A}))}{\partial p} = \operatorname{Tr}\left(\boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial p}\right)$ using the chain rule applied to a logarithmic function. Finally, using the algebra properties (Note 3), we deduce directly that: $\frac{\partial \log\left(\operatorname{Det}(\boldsymbol{A}^{-1})\right)}{\partial p} = -\operatorname{Tr}\left(\boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial p}\right)$.

Finally, setting $\boldsymbol{A} = \boldsymbol{D} = -\boldsymbol{x}$ as vectors in $\mathbb{R}^G$, $\boldsymbol{b} = \boldsymbol{e} = \boldsymbol{y}$ and $\boldsymbol{C} = \boldsymbol{\Theta}$ a symmetric matrix, we have:

$$\frac{\partial (\boldsymbol{y} - \boldsymbol{x}p)^\top \Theta (\boldsymbol{y} - \boldsymbol{x}p)}{\partial p} = -2(\boldsymbol{y} - \boldsymbol{x}p)^\top \Theta \boldsymbol{x} = -2\boldsymbol{x}^\top \Theta (\boldsymbol{y} - \boldsymbol{x}p)$$

[5] with $\boldsymbol{A}$ a symmetric matrix.

[6] if a matrix is symmetric, then by definition, $\boldsymbol{A}^\top = \boldsymbol{A}$

.

▶ Note 5. Given an invertible matrix $\boldsymbol{A}$ depending on a variable p, the following calculus formulas hold:
(a) $\frac{\partial^2 \boldsymbol{A}^{-1}}{\partial p_i \partial p_j} = \boldsymbol{A}^{-1} \left(\frac{\partial \boldsymbol{A}}{\partial p_i} \boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial p_j} - \frac{\partial^2 \boldsymbol{A}}{\partial p_i \partial p_j} + \frac{\partial \boldsymbol{A}}{\partial p_j} \boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial p_i} \right) \boldsymbol{A}^{-1}$ (b) $\frac{\partial \operatorname{Tr}(\boldsymbol{A})}{\partial p_i} = \operatorname{Tr} \left(\frac{\partial \boldsymbol{A}}{\partial p_i} \right)$

Combining the derivative formula for the inverse of a matrix, given in Note 4 with the linearity of the trace operator and the chain rule formula yields in particular:

$$\frac{\partial^2 \log \left(\operatorname{Det}(\boldsymbol{A}^{-1})\right)}{\partial^2 p} = - \operatorname{Tr} \left[\boldsymbol{A}^{-1} \frac{\partial^2 \boldsymbol{A}}{\partial^2 p_i} \right] + \operatorname{Tr} \left[\left(\boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial p_i} \right)^2 \right]$$

.

Additionally, we have using the invariance of the trace operator under cyclic permutation:

$$(\boldsymbol{y}-\boldsymbol{\mu p})^\top \boldsymbol{\Theta \Sigma}_i \boldsymbol{\Theta \Sigma}_j \boldsymbol{\Theta}(\boldsymbol{y}-\boldsymbol{\mu p})=(\boldsymbol{y}-\boldsymbol{\mu p})^\top (\boldsymbol{\Theta \Sigma}_i \boldsymbol{\Theta \Sigma}_j \boldsymbol{\Theta})^\top (\boldsymbol{y}-\boldsymbol{\mu p})=(\boldsymbol{y}-\boldsymbol{\mu p})^\top \boldsymbol{\Theta \Sigma}_j \boldsymbol{\Theta \Sigma}_i \boldsymbol{\Theta}(\boldsymbol{y}-\boldsymbol{\mu p})$$

This result can be used to guarantee that the Hessian matrix, whose expression is given in Equation (10), is indeed symmetric.

A.3 First and second-order derivation of the constrained DeCovarT log-likelihood function

To reparametrise the log-likelihood function (Equation (8)) in order to explicitly handling the unit simplex constraint (Equation (2)), we consider the following mapping function: $\boldsymbol{\psi} : \boldsymbol{\theta} \to \boldsymbol{p} \mid \quad \boldsymbol{\theta} \in \mathbb{R}^{J-1}, \boldsymbol{p} \in]0,1[^J$ (Equation (12)):

1. (12)

$$\boldsymbol{p} = \boldsymbol{\psi}(\boldsymbol{\theta}) = \begin{cases} p_j = \frac{e^{\theta_j}}{\sum_{k<J} e^{\theta_k} + 1}, \; j < J \\ p_J = \frac{1}{\sum_{k<J} e^{\theta_j} + 1} \end{cases}$$

2. $\boldsymbol{\theta} = \boldsymbol{\psi}^{-1}(\boldsymbol{p}) = \left(\ln \left(\frac{p_j}{p_J} \right) \right)_{j \in \{1,\dots,J-1\}}$

that is a C^2-diffeomorphism, since $\boldsymbol{\psi}$ is a bijection between $\boldsymbol{p}$ and $\boldsymbol{\theta}$ twice differentiable. Its Jacobian, $\mathbf{J}_{\boldsymbol{\psi}} \in \mathcal{M}_{J \times (J-1)}$ is given by Equation (13):

$$\mathbf{J}_{i,j} = \frac{\partial p_i}{\partial \theta_j} = \begin{cases} \frac{e^{\theta_i} B_i}{A^2}, \quad i = j, \, i < J \\ \frac{-e^{\theta_j} e^{\theta_i}}{A^2}, \quad i \neq j, \, i < J \\ \frac{-e^{\theta_j}}{A^2}, \quad i = J \end{cases} \tag{13}$$

with i indexing vector-valued $\boldsymbol{p}$ and j indexing the first-order order partial derivatives of the mapping function, $A = \sum_{j'<J} e^{\theta_{j'}} + 1$ the sum over exponential (denominator of the mapping function) and $B = A - e^{\theta_i}$ the sum over ratios minus the exponential indexed with the currently considered index i.

The Hessian (which fortunately is symmetric for each component j, as expected according to the Schwarz's theorem) of the vectorial mapping function $\boldsymbol{\psi(\theta)}$ is a third-order tensor of rank $(J-1)(J-1)J$, given by Equation (14):

$$\frac{\partial^2 p_i}{\partial k \partial j} = \begin{cases} \frac{e^{\theta_i} e^{\theta_l}\left(-B_i+e^{\theta_i}\right)}{A^3}, (i < J) \wedge ((i \neq j) \oplus (i \neq k)) \quad (a) \\ \frac{2e^{\theta_i} e^{\theta_j} e^{\theta_k}}{A^3}, (i < J) \wedge (i \neq j \neq k) \quad (b) \\ \frac{e^{\theta_i} e^{\theta_j}\left(-A+2e^{\theta_j}\right)}{A^3}, (i < J) \wedge (j = k \neq i) \quad (c) \\ \frac{B_i e^{\theta_i}\left(B_i-e^{\theta_i}\right)}{A^3}, (i < J) \wedge (j = k = i) \quad (d) \\ \frac{e^{\theta_j}\left(-A+2e^{\theta_j}\right)}{A^3}, (i = J) \wedge (j = k) \quad (e) \\ \frac{2e^{\theta_j} e^{\theta_k}}{A^3}, (i = J) \wedge (j \neq k) \quad (f) \end{cases} \tag{14}$$

with i indexing $\boldsymbol{p}$, j and k respectively indexing the first-order and second-order partial derivatives of the mapping function with respect to $\boldsymbol{\theta}$. In line (a), $\oplus$ refers to the Boolean XOR operator, $\wedge$ to the AND operator and $l = \{j, k\} \setminus i$.

To derive the log-likelihood function in Equation (9), we reparametrise $\boldsymbol{p}$ to $\boldsymbol{\theta}$, using a standard *chain rule formula* (see Appendix A.2). Considering the original log-likelihood function, Equation (8), and the mapping function, Equation (12), the differential at the first order and at the second order is given by Equation (15) and Equation (16), respectively defined in $\mathbb{R}^{J-1}$ and $\mathcal{M}_{(J-1)\times(J-1)}$:

$$\left[\frac{\partial \ell_{\boldsymbol{y}|\boldsymbol{\zeta}}}{\partial \theta_j}\right]_{j<J} = \sum_{i=1}^{J} \frac{\partial \ell_{\boldsymbol{y}|\boldsymbol{\zeta}}}{\partial p_i} \frac{\partial p_i}{\partial \theta_j} \tag{15}$$

$$\left[\frac{\partial \ell^2_{\boldsymbol{y}|\boldsymbol{\zeta}}}{\partial \theta_k \theta_j}\right]_{j<J,\, k<J} = \sum_{i=1}^{J} \sum_{l=1}^{J} \left(\frac{\partial p_i}{\partial \theta_j} \frac{\partial^2 \ell_{\boldsymbol{y}|\boldsymbol{\zeta}}}{\partial p_i \partial p_l} \frac{\partial p_l}{\partial \theta_k}\right) + \sum_{i=1}^{J} \left(\frac{\partial \ell_{\boldsymbol{y}|\boldsymbol{\zeta}}}{\partial p_i} \frac{\partial^2 p_i}{\partial \theta_k \theta_j}\right) \quad (d) \tag{16}$$