pyPLNmodels: getting started.

Bastien Batardière, Joon Kwon and Julien Chiquet

Université Paris-Saclay, AgroParisTech, INRAE, UMR MIA Paris-Saclay

June 9, 2023

Modelling

- Dataset:
  - $\boldsymbol{Y}$: $n \times p$ count matrix
  - $\boldsymbol{X}$: $n \times d$ covariates
- Parameter $\theta = (\beta, \Sigma)$
  - $\beta \in \mathbb{R}^{d \times p}$ regression parameter
  - $\Sigma = CC^\top, C \in \mathbb{R}^{p \times q}$ covariance matrix.
- Model:

$$\begin{aligned} W_i \sim& \mathcal{N}(0, I_q) \\ Z_i =& \beta^\top X_i + CW_i \\ (Y_{ij} \mid Z_{ij}) \sim& \mathcal{P}(\exp(Z_{ij})) \end{aligned}$$

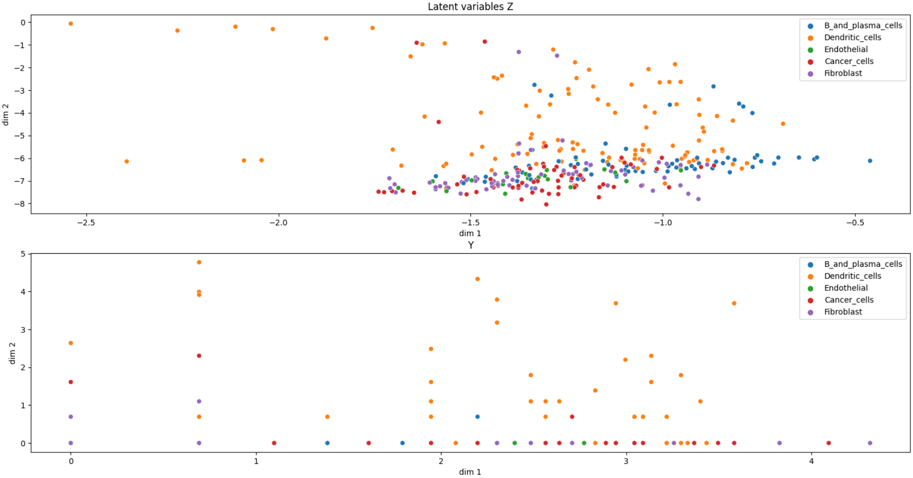
- When $p = q$: unrestricted model: Pln class in pyPLNmodels
- When $p >> q$: Σ has low rank: PlnPCA class in pyPLNmodels

- Model:

$$\begin{aligned} W_i \sim & \mathcal{N}(0, I_q) \\ Z_i = & \beta^\top X_i + C W_i \\ (Y_{ij} \mid Z_{ij}) \sim & \mathcal{P}(\exp(Z_{ij})) \end{aligned}$$

- Goal of the package: retrieve back the latent representation Z_i:
  - $W_i | Y_i$ is enough for PlnPCA
  - $Z_i | Y_i$ for Pln
- Problem: both $W_i | Y_i$ and $Z_i | Y_i$ have unknown distribution.
- $\Longrightarrow$ gaussian variational approximation.
  - PlnPCA:$W_i | Y_i \sim \mathcal{N}(M_i, \text{diag}(S_i)), M_i, S_i \in \mathbb{R}^q$
  - Pln:$Z_i | Y_i \sim \mathcal{N}(M_i, \text{diag}(S_i)), M_i, S_i \in \mathbb{R}^p$

First dimension against second dimension in dimension p = 5



PC 1 agains PC 2 in dimension p = 200.