Contributions to marker detection and survival analysis in oncology

Mathilde Sautreuil

February 12th, 2021

Context

Precision medicine

- Not a new concept: Hippocratic origins
- Discovery of the DNA double helix in 1950
- Consequence: Omics data available for patients
- Objective: Adapt the treatments according to the molecular portrait of patients

➙ High-dimensional data: need of new statistical and bioinformatics methods

- Critical role in oncology
  1. Diagnostic (subtypes of cancers)
  2. Pronostic (survival, relapse and progression)
  3. Response of a patient to a treatment

Objectives of the PhD thesis

Marker detection

1. Identify the genes implied in the ccRCC
   - Differential analysis: statistical tests taking into account the characteristics of gene expression data
     → But false positives
   - Learning methods
     ➙ Detect the genes by using differential analysis and learning methods
2. Identify the genes impacting the survival duration
   - Regularization and screening methods, but the issue of selection stability
     ➙ Study of these methods by quantifying their stability

Survival prediction

1. Prediction of a patient's survival in a high-dimensional framework
   - Different models → Classical: Cox model
   - Interactions and non linearity
     ➙ Study of neural networks in high-dimension

Objectives of the PhD thesis

Marker detection

1. Identify the genes implied in the ccRCC
   - Differential analysis: statistical tests taking into account the characteristics of gene expression data
     → But false positives
   - Learning methods
     ➙ Detect the genes by using differential analysis and learning methods
2. Identify the genes impacting the survival duration
   - Regularization and screening methods, but the issue of selection stability
     ➙ Study of these methods by quantifying their stability

Survival prediction

1. Prediction of a patient's survival in a high-dimensional framework
   - Different models → Classical: Cox model
   - Interactions and non linearity
     ➙ Study of neural networks in high-dimension

Objectives of the PhD thesis

Marker detection

1. Identify the genes implied in the ccRCC
   - Differential analysis: statistical tests taking into account the characteristics of gene expression data
     → But false positives
   - Learning methods
     ➙ Detect the genes by using differential analysis and learning methods
2. Identify the genes impacting the survival duration
   - Regularization and screening methods, but the issue of selection stability
     ➙ Study of these methods by quantifying their stability

Survival prediction

1. Prediction of a patient's survival in a high-dimensional framework
   - Different models → Classical: Cox model
   - Interactions and non linearity
     ➙ Study of neural networks in high-dimension

Outline

Survival analysis in Oncology

Survival analysis

➙ Study of elapsed time until an interest event (death or recovery)

Objectives

1. Marker detection
   - Which factors have an impact on the survival of patients ?
2. Survival prediction
   - Computing the risk of death

Framework and notations

- Random variable to predict: Survival time T of a patient
- Explanatory variables: factors X (e.g. transcriptomic data)

→ Survival time T can be censored

Right censorship: Observed times are less than the survival duration

Notations

- n: number of individuals, p: number of variables
- T_i the survival time for individual i
- C_i the censoring time for individual i
- $X_i = (X_{i1}, \ldots, X_{ip})^T$: vector of variables for individual i
- We observe for individual i:

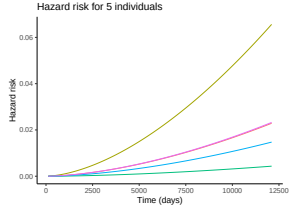
$$Y_i = \min(T_i, C_i) \qquad \delta_i = \begin{cases} 1 & \text{if } T_i \leq C_i \\ 0 & \text{otherwise} \end{cases}$$

Models in survival analysis

Cox model:

$$\lambda(t|X_i) = \alpha_0(t)\exp(\beta_0^T X_i)$$

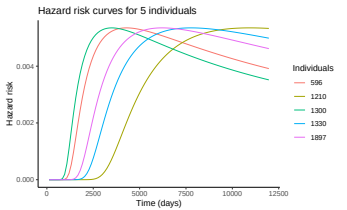
➞ a proportional hazards model



AFT model:

$$\lambda(t|X_i) = \exp(\beta^T X_i)\alpha_0(t\exp(\beta^T X_i))$$

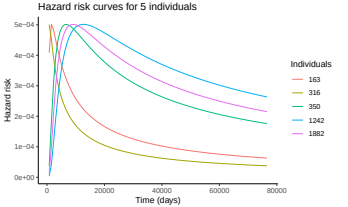
➞ Covariates accelerate the risk curve



AH model:

$$\lambda(t|X_i) = \alpha_0(t\exp(\beta^T X_i))$$

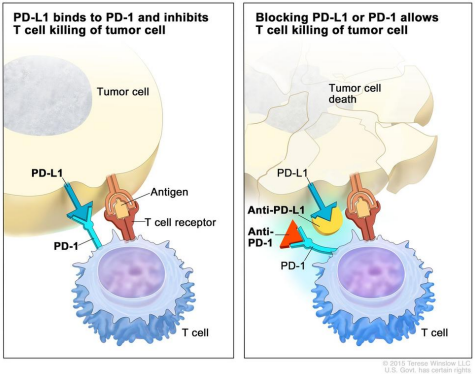
➞ More irregular behaviour



Detection of markers implied in one type of cancer

Real dataset: clear cell renal cell carcinoma (ccRCC)

- Collaboration with Dr Diana Tronik-Le Roux of St Louis Hospital/CEA.
- Immunotherapy: to understand the role of checkpoints in the blocking of immune action against tumour cells



- Some therapies target checkpoints CTLA-4 and PDL-1
  - ➞ Only 30% of patients respond positively

Detection of markers implied in one type of cancer

Analysis: clear cell renal cell carcinoma (ccRCC)

- 44 immune-checkpoints identified in ccRCC + 3 control genes
- Expression level in tumour vs control (TCGA database)

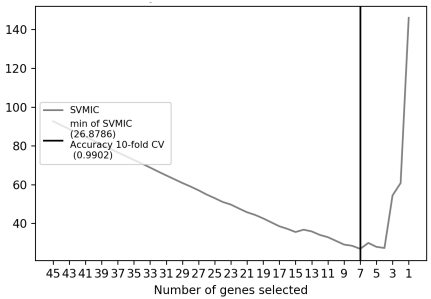
➙ Differential analysis + RFE-SVM

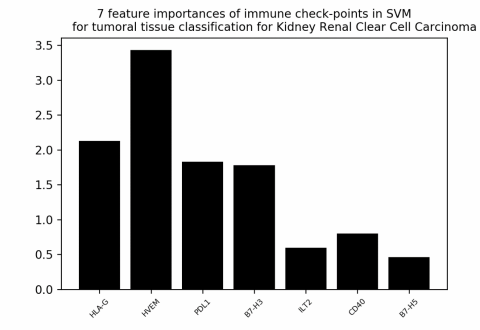
- Differential analysis: DESeq2
  - 39 ICs considered as differentially expressed with BH correction

  ➙ False positives
- RFE-SVM: Remove recursively genes which are the less important in the classification task

  ➙ Subset of optimal genes: 7 IC (HLA-G, HVEM, PD-L1, B7-H3, ILT2, CD40, B7-H5)

Detection of markers implied in one type of cancer





- HLA-G: IC the most important, but its value is not the higher

→ Target HLA-G/ILT potential strategy in the case of no response to anti-PD1/PDL-1

Diana Tronik-Le Roux, Mathilde Sautreuil, Mahmoud Bentriou, et al. (2020), Comprehensive landscape of immune-checkpoints uncovered in clear cell renal cell carcinoma reveals new and emerging therapeutic targets, Cancer Immunology, Immunotherapy

Marker detection for survival analysis in high-dimension

➙ Determine covariates with strong impact

- Cox model: $\lambda(t|X_i) = \alpha_0(t)\exp(\beta_0^T X_i)$
- $\hat{\beta} = \text{argmax}_\beta \mathcal{L}(\beta)$ with $\mathcal{L}(\beta) = \sum_{i=1}^{n}\left(\beta^T X_i\right) - \sum_{i=1}^{n}\delta_i \log\left(\sum_{l\in R_i}\exp\left(\beta^T X_l\right)\right)$
  - where $\mathcal{L}(\beta)$ the Cox partial log-likelihood, R_i the individuals at risk at time t_i and δ_i the censorship indicator

In high-dimension

- Estimation of β not consistent
- To have a better interpretability and to solve the optimization problem:

➙ Adding a penalty term to the minimization of the opposite of the partial log-likelihood

Regularization methods

$$\arg\min_\beta \left\{ -\mathcal{L}(\beta) + \lambda \sum_{j=1}^{p} ||\beta_j||_q \right\}$$

λ: regularization parameter

- Lasso method
- Adaptive-Lasso method
- Ridge method
- Elastic-Net method

Marker detection for survival analysis in high-dimension

➙ Determine covariates with strong impact

- Cox model: $\lambda(t|X_i) = \alpha_0(t)\exp(\beta_0^T X_i)$
- $\hat{\beta} = \text{argmax}_\beta \mathcal{L}(\beta)$ with $\mathcal{L}(\beta) = \sum_{i=1}^{n}\left(\beta^T X_i\right) - \sum_{i=1}^{n}\delta_i \log\left(\sum_{l\in R_i}\exp\left(\beta^T X_l\right)\right)$
  - where $\mathcal{L}(\beta)$ the Cox partial log-likelihood, R_i the individuals at risk at time t_i and δ_i the censorship indicator

In high-dimension

- Estimation of β not consistent
- To have a better interpretability and to solve the optimization problem:

➙ Adding a penalty term to the minimization of the opposite of the partial log-likelihood

Regularization methods

$$\arg\min_\beta \left\{ -\mathcal{L}(\beta) + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

λ the regularization parameter

- Lasso method
- Adaptive-Lasso method
- Ridge method
- Elastic-Net method

Marker detection for survival analysis in high-dimension

➙ Determine covariates with strong impact

- Cox model: $\lambda(t|X_i) = \alpha_0(t)\exp(\beta_0^T X_i)$
- $\hat{\beta} = \text{argmax}_\beta \mathcal{L}(\beta)$ with $\mathcal{L}(\beta) = \sum_{i=1}^{n}\left(\beta^T X_i\right) - \sum_{i=1}^{n}\delta_i \log\left(\sum_{l\in R_i}\exp\left(\beta^T X_l\right)\right)$
  - where $\mathcal{L}(\beta)$ the Cox partial log-likelihood, R_i the individuals at risk at time t_i and δ_i the censorship indicator

In high-dimension

- Estimation of β not consistent
- To have a better interpretability and to solve the optimization problem:

➙ Adding a penalty term to the minimization of the opposite of the partial log-likelihood

Regularization methods

$$\arg\min_{\beta}\left\{-\mathcal{L}(\beta) + \lambda\sum_{j=1}^{p}\frac{|\beta_j|}{|\hat{\beta}_j^{ll}|}\right\}$$

λ: regularization parameter

- Lasso method
- Adaptive-Lasso method
- Ridge method
- Elastic-Net method

Marker detection for survival analysis in high-dimension

➙ Determine covariates with strong impact

- Cox model: $\lambda(t|X_i) = \alpha_0(t)\exp(\beta_0^T X_i)$
- $\hat{\beta} = \text{argmax}_\beta \mathcal{L}(\beta)$ with $\mathcal{L}(\beta) = \sum_{i=1}^{n}\left(\beta^T X_i\right) - \sum_{i=1}^{n}\delta_i \log\left(\sum_{l\in R_i}\exp\left(\beta^T X_l\right)\right)$
  - where $\mathcal{L}(\beta)$ the Cox partial log-likelihood, R_i the individuals at risk at time t_i and δ_i the censorship indicator

In high-dimension

- Estimation of β not consistent
- To have a better interpretability and to solve the optimization problem:

➙ Adding a penalty term to the minimization of the opposite of the partial log-likelihood

Regularization methods

$$\arg\min_\beta \left\{ -\mathcal{L}(\beta) + \lambda \sum_{j=1}^{p} |\beta_j|^2 \right\}$$

λ: regularization parameter

- Lasso method
- Adaptive-Lasso method
- Ridge method
- Elastic-Net method

Marker detection for survival analysis in high-dimension

➙ Determine covariates with strong impact

- Cox model: $\lambda(t|X_i) = \alpha_0(t)\exp(\beta_0^T X_i)$
- $\hat{\beta} = \text{argmax}_\beta \mathcal{L}(\beta)$ with $\mathcal{L}(\beta) = \sum_{i=1}^{n}\left(\beta^T X_i\right) - \sum_{i=1}^{n}\delta_i \log\left(\sum_{l\in R_i}\exp\left(\beta^T X_l\right)\right)$
  - where $\mathcal{L}(\beta)$ the Cox partial log-likelihood, R_i the individuals at risk at time t_i and δ_i the censorship indicator

In high-dimension

- Estimation of β not consistent
- To have a better interpretability and to solve the optimization problem:

➙ Adding a penalty term to the minimization of the opposite of the partial log-likelihood

Regularization methods

$$\arg\min_{\beta}\left\{-\mathcal{L}(\beta) + \lambda_1\sum_{j=1}^{p}|\beta_j| + \lambda_2\sum_{j=1}^{p}|\beta_j|^2\right\}$$

λ_1, λ_2: regularization parameters

- Lasso method
- Adaptive-Lasso method
- Ridge method
- Elastic-Net method

Marker detection for survival analysis in high-dimension

Screening methods

- Methods developed to solve stability problems of regularization methods
- (I)SIS (Saldana and Feng, 2018; Fan and Lv, 2017)
- PSIS (Zhao and Li, 2012)
- coxCS (Hong et al., 2018)

Principles

1. A pre-selection is made by computing a score for each covariate individually
2. Covariates are sorted and the covariates with the higher scores are chosen
3. A regularization method (Lasso) is applied to this pre-selection

Marker detection for survival analysis in high-dimension

Screening methods: SIS

- SIS: Sure Independance Screening
  1. Compute a score: marginal utility

$$u_m = \max_{\beta_m} \left(\sum_{i=1}^{n} (\delta_i \beta_m x_{im}) - \sum_{i=1}^{n} \delta_i \log \left(\sum_{j \in R(y_i)} \exp(\beta_m x_{jm}) \right) \right)$$

  2. Covariates with a score $> \gamma$ are selected
  3. Lasso procedure is applied on the selected covariates

Marker detection for survival analysis in high-dimension

Screening methods: SIS, ISIS

- SIS: Sure Independance Screening
  1. Compute a score: marginal utility

$$u_m = \max_{\beta_m} \left(\sum_{i=1}^{n} (\delta_i \beta_m x_{im}) - \sum_{i=1}^{n} \delta_i \log \left(\sum_{j \in R(y_i)} \exp(\beta_m x_{jm}) \right) \right)$$

  2. Covariates with a score $> \gamma$ are selected
  3. Lasso procedure is applied on the selected covariates

- ISIS: Iterative version of SIS
  - Application of SIS procedure
  - For SIS selection set
    1. Compute a new score: conditional utility
    2. Covariates with a score $> \gamma$ are selected
    3. Lasso procedure is applied on the selected covariates

    → Repeat until convergence

Marker detection for survival analysis in high-dimension

Screening methods: SIS, ISIS, PSIS

- PSIS: Steps similar to SIS procedure
  1. Score $\rightarrow$ to take into account the False Postives
     - ➙ score $= \mathrm{I_j}(\hat{\beta}_\mathrm{j})^{1/2}|\hat{\beta}_\mathrm{j}|$
  2. Covariates with a score $> \gamma$ are selected
  3. Lasso procedure is applied on the selected covariates

Marker detection for survival analysis in high-dimension

Screening methods: SIS, ISIS, PSIS, coxCS

- PSIS: Steps similar to SIS procedure
  1. Score → to take into account the False Postives
     - ➞ score $= \mathrm{I_j}(\hat{\beta}_\mathrm{j})^{1/2}|\hat{\beta}_\mathrm{j}|$
  2. Covariates with a score $> \gamma$ are selected
  3. Lasso procedure is applied on the selected covariates

- coxCS: Biological knowledge
  - Biological selection
  - For pre-selection set
    1. Compute a score
    2. Covariates with a score $> \gamma$ are selected
    3. Lasso procedure is applied on the selected covariates

Marker detection for survival analysis in high-dimension

Regularization and screening methods

1. Studying the stability of both methods with similarity indexes and other criteria:
   - What is the level of stability of regularization methods?
   - Do screening methods solve the stability problem?
   - Can biological knowledge improve stability?
   - And what is the quality of the selection?
   - → Illustration on ccRCC dataset
2. Discovering new potential markers impacting the survival for the ccRCC

Stability study on the ccRCC dataset

3 approaches

1. Immune-Checkpoints (p = 48)
2. Differential expressed genes (p = 11 289)
3. All the genes (p = 17 789)

Procedure

- Run on 100 different seeds
- Selected genes for each seed
- Compute the similarity between the seeds: Sørensen index
- Compute the validity of model: AIC

Indexes

Sørensen Index

$$\text{Sor} = \frac{\frac{1}{S-1}\sum_{j=1}^{N}(s_j - 1)}{\frac{1}{S}\sum_{i=1}^{S} n_i}, \quad 0 \leq \text{Sor} \leq 1$$

- N the number of selected genes, S the number of seeds, s_j the number of seeds where the gene j is selected and n_i the number of selected genes in the seed i

➙ Variation of the composition of genes between seeds

Indexes

Sørensen Index

$$\text{Sor} = \frac{\frac{1}{\text{S}-1}\sum_{\text{j}=1}^{\text{N}}(\text{s}_\text{j}-1)}{\frac{1}{\text{S}}\sum_{\text{i}=1}^{\text{S}}\text{n}_\text{i}}, \quad 0 \leq \text{Sor} \leq 1$$

- N the number of selected genes, S the number of seeds, s_j the number of seeds where the gene j is selected and n_i the number of selected genes in the seed i

➙ Variation of the composition of genes between seeds

AIC (Akaike Information Criterion)

- Evaluate the quality of a model
  - ➙ Quality of the adjustement and complexity of the model
- Compute the mean and the standard deviation of the AIC for the 100 seeds

Indexes

Sørensen Index

$$\text{Sor} = \frac{\frac{1}{S-1}\sum_{j=1}^{N}(s_j - 1)}{\frac{1}{S}\sum_{i=1}^{S} n_i}, \quad 0 \leq \text{Sor} \leq 1$$

- N the number of selected genes, S the number of seeds, s_j the number of seeds where the gene j is selected and n_i the number of selected genes in the seed i

➞ Variation of the composition of genes between seeds

AIC (Akaike Information Criterion)

- Evaluate the quality of a model
  - ➞ Quality of the adjustement and complexity of the model
- Compute the mean and the standard deviation of the AIC for the 100 seeds

Mean and standard deviation of the number of selected genes

Studying of stability (Summary)

On the ccRCC dataset

1. Stability of regularization methods in high-dimension?
   - ➙ Poor: Sørensen index ↘ with the dimension
2. Screening methods solve the problem of stability in high-dimension?
   - ➙ Partially true: SIS and ISIS → better results than regularization methods
   - ➙ Sørensen indexes ↗ with the dimension
   - ➙ False for PSIS (similar to regularization methods)
3. Biological knowledge improves the stability?
   - ➙ Unclear: coxCS → worst results
4. And about the quality of the selection?
   - ➙ Good values of AIC for the regularization methods (despite their bad stability)
   - ➙ Values of AIC for SIS and ISIS close to the values of regularization methods
   - ➙ PSIS and coxCS worse → due to the number of parameters?

Studying of stability (Summary)

On the ccRCC dataset

1. Stability of regularization methods in high-dimension?
   - ➙ Poor: Sørensen index $\searrow$ with the dimension
2. Screening methods solve the problem of stability in high-dimension?
   - ➙ Partially true: SIS and ISIS $\rightarrow$ better results than regularization methods
   - ➙ Sørensen indexes $\nearrow$ with the dimension
   - ➙ False for PSIS (similar to regularization methods)
3. Biological knowledge improves the stability?
   - ➙ Unclear: coxCS $\rightarrow$ worst results
4. And about the quality of the selection?
   - ➙ Good values of AIC for the regularization methods (despite their bad stability)
   - ➙ Values of AIC for SIS and ISIS close to the values of regularization methods
   - ➙ PSIS and coxCS worse $\rightarrow$ due to the number of parameters?

Studying of stability (Summary)

On the ccRCC dataset

1. Stability of regularization methods in high-dimension?
   - ➙ Poor: Sørensen index $\searrow$ with the dimension
2. Screening methods solve the problem of stability in high-dimension?
   - ➙ Partially true: SIS and ISIS $\rightarrow$ better results than regularization methods
   - ➙ Sørensen indexes $\nearrow$ with the dimension
   - ➙ False for PSIS (similar to regularization methods)
3. Biological knowledge improves the stability?
   - ➙ Unclear: coxCS $\rightarrow$ worst results
4. And about the quality of the selection?
   - ➙ Good values of AIC for the regularization methods (despite their bad stability)
   - ➙ Values of AIC for SIS and ISIS close to the values of regularization methods
   - ➙ PSIS and coxCS worse $\rightarrow$ due to the number of parameters?

Studying of stability (Summary)

On the ccRCC dataset

1. Stability of regularization methods in high-dimension?
   - ➙ Poor: Sørensen index $\searrow$ with the dimension
2. Screening methods solve the problem of stability in high-dimension?
   - ➙ Partially true: SIS and ISIS $\rightarrow$ better results than regularization methods
   - ➙ Sørensen indexes $\nearrow$ with the dimension
   - ➙ False for PSIS (similar to regularization methods)
3. Biological knowledge improves the stability?
   - ➙ Unclear: coxCS $\rightarrow$ worst results
4. And about the quality of the selection?
   - ➙ Good values of AIC for the regularization methods (despite their bad stability)
   - ➙ Values of AIC for SIS and ISIS close to the values of regularization methods
   - ➙ PSIS and coxCS worse $\rightarrow$ due to the number of parameters?

Studying of stability (Summary)

On the ccRCC dataset

1. Stability of regularization methods in high-dimension?
   - ➙ Poor: Sørensen index $\searrow$ with the dimension
2. Screening methods solve the problem of stability in high-dimension?
   - ➙ Partially true: SIS and ISIS $\rightarrow$ better results than regularization methods
   - ➙ Sørensen indexes $\nearrow$ with the dimension
   - ➙ False for PSIS (similar to regularization methods)
3. Biological knowledge improves the stability?
   - ➙ Unclear: coxCS $\rightarrow$ worst results
4. And about the quality of the selection?
   - ➙ Good values of AIC for the regularization methods (despite their bad stability)
   - ➙ Values of AIC for SIS and ISIS close to the values of regularization methods
   - ➙ PSIS and coxCS worse $\rightarrow$ due to the number of parameters?

Issue: Sørensen index $\rightarrow$ more stable for the nested selection scenarii

Discovering new markers for the ccRCC

Potential genes impacting patients' survival

1. Immune-Checkpoints
   - B7-H3 (CD276): good biomarker → to validate biologically
   - HLA-G: good biomarker, sharing information with others genes as ILT2
     - Alternative to treatments PD1/PDL1 (in case of no response)
2. Differential expressed genes/All genes

   ➙ Identify of potential genes → to explore more precisely
   - CHEK2: known to have an impact in the breast cancer
   - CKAP4: implied in the immune system (ccRCC: immunogene cancer)
   - CUBN: validated by Gremel et al. (2017)
   - FBXL5:
     - playing a role in the immune system
     - implied in the chronic renal diseases
     - linked to 2 immune-checkpoints

Survival prediction in high-dimension

Objective

- Prediction of survival of patients according to patient features

Survival prediction: CoxL1

1. Applying Lasso method to select variables
2. Coefficients of selected variables are estimated from the Cox partial likelihood
3. Baseline hazard function $\alpha_0(t)$ is estimated from the Ramlau-Hansen kernel estimator

Deep learning more and more popular in the biomedical field

→ Study of neural networks to predict the survival duration in comparison to coxL1 (reference model)

Neural networks in survival analysis

Our objective: Explore the potential of neural networks to predict survival duration of patients from genomic data

- Approach not recent (Faraggi and Simon (1995)): only considered for small numbers of input data
- In high-dimension?

2 strategies based on neural networks

1. Based on Cox partial log-likelihood (Faraggi and Simon, 1995)
   - ➙ Study in high-dimension by Ching et al. in 2018 (Cox-nnet)
2. Based on discrete time model (Biganzoli et al., 1998)
   - ➙ Lee et al. (2018); Sautreuil et al. (2019)

Cox-nnet (Ching et al., 2018)

Neural network based on Cox model

➙ Principle: Output layer is the regression part of the Cox model

$$\exp(\beta^{\mathrm{T}} \mathrm{G}(\mathrm{W}\mathrm{X}_{\mathrm{i}} + \mathrm{b}))$$

- X_{i} is replaced by the output of the hidden layer: $\mathrm{G}(\mathrm{W}\mathrm{X}_{\mathrm{i}} + \mathrm{b})^{\mathrm{T}}$
- W is the weight matrix, b is the bias term for each hidden neuron and G is the activation function

➙ Parameter estimation from the Cox partial log-likelihood:

$$\mathcal{L}(\beta, \mathrm{W}, \mathrm{b}) = \sum_{\mathrm{i}=1}^{\mathrm{n}} \theta_{\mathrm{i}} - \sum_{\mathrm{i}=1}^{\mathrm{n}} \delta_{\mathrm{i}} \log \left(\sum_{\mathrm{l} \in \mathrm{R}_{\mathrm{i}}} \exp\left(\theta_{\mathrm{l}}\right) \right),$$

- δ_{i} : censorship indicator and $\theta_{\mathrm{i}} = \beta^{\mathrm{T}} \mathrm{G}(\mathrm{W}\mathrm{X}_{\mathrm{i}} + \mathrm{b})$

➙ Drawback: Need to estimate $\alpha_0(\mathrm{t})$ separately

- $\alpha_0(\mathrm{t})$ is estimated from the Ramlau-Hansen kernel estimator and bandwidth selected by Goldenshluger-Lepski method (Guilloux et al., 2016)

NNsurv (Sautreuil et al., 2019)

Adaptation of Biganzoli et al. (1998) to the high-dimension (implementation with Keras library)

- Introducing L time intervals $A_l =]t_{l-1}, t_l]$ to which belong survival times

➙ Discrete hazard rate function is defined as the survival conditional probability:

$$h_{il} = P(T_i \in A_l | T_i > t_{l-1})$$

- Introducing the death indicator:

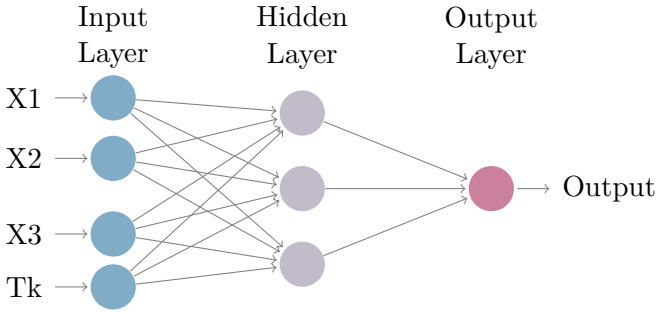
$$d_{il} = \begin{cases} 1 & \text{if } A_l \text{ contains the interest event for the uncensored individuals} \\ 0 & \text{otherwise} \end{cases}$$

➙ Parameter estimation from total log-likelihood (used as cross-entropy error function):

$$\mathcal{L}(W) = -\sum_{i=1}^{n}\sum_{l=1}^{l_i} d_{il}\log(\widehat{h}_{il}) + (1-d_{il})\log(1-\widehat{h}_{il}).$$

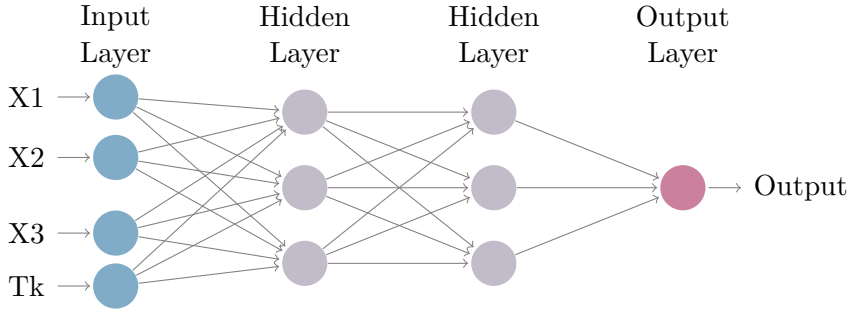
- $\widehat{h}_{il} = \hat{h}_l(X_i, W)$ with W the weight matrix and bias

Version 1: NNsurv



- Inputs:
  - T_k : Mid-point of intervals A_l (p variables of each individual duplicated for each time interval)
- Configurations:
  - Cross-validation procedure to select the hyperparameters
  - Number of neurons in the hidden layer: $H = \sqrt{p}$
  - Batch size, early stopping, optimization methods and dropout

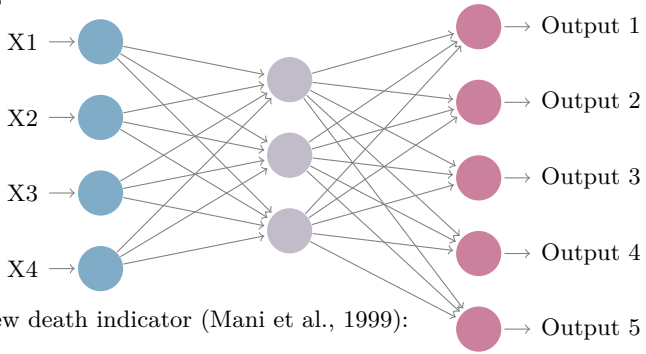
Version 2: NNsurv deep



- Inputs:
  - T_k : Mid-point of intervals A_l (p variables of each individual duplicated for each time interval)
- Configurations:
  - Cross-validation procedure to select the hyperparameters
  - Number of neurons in the hidden layer: $H = \sqrt{p}$
  - Batch size, early stopping, optimization methods and dropout

Version 3: NNsurvK (NNsurv with multivariate outputs)

- L outputs (for each time interval) (Liestbl et al., 1994)
  P = 4 and L = 5



- Introducing a new death indicator (Mani et al., 1999):

$$\widetilde{d}_{il} = \begin{cases} 0 & \text{for } 1 \leq l < l_i, \\ 1 & \text{for } l_i \leq l \leq L \text{ and individual i is uncensored,} \\ p_l = \frac{r_l}{n_l} & \text{for } l_i \leq l \leq L \text{ and individual i is censored} \end{cases}$$

- Fused-lasso regularization: penalize the deviation from proportional hazards

$$\mathcal{L} = -\sum_{i=1}^{n}\sum_{l=1}^{l_i} \widetilde{d}_{il}\log(\widehat{h}_{il}) + (1-\widetilde{d}_{il})\log(1-\widehat{h}_{il}) + \alpha\sum_{h=1}^{H}\sum_{l=1}^{L}\left(W_{hl} - W_{h(l-1)}\right)^2$$

Test cases

Objective

➙ Comparison of five different methods (CoxL1, Cox-nnet, NNsurv, NNsurv deep, NNsurvK) adapted to high-dimension on:

1. Simulated datasets
2. Real datasets

Metrics

- Concordance index (C_{td}) (Antolini et al., 2005): score indicating how well prediction corresponds to ranks of survival data
  - $C_{td} = 0.5 \rightarrow$ random process
  - ➙ The prediction is better when the value of C_{td} is closer to 1
- Integrated Brier Score (IBS): score computing the squared error between the predicted survival probability and the actual survival of patients at each time point
  - ➙ The prediction is better when the value of IBS is closer to 0

Simulated datasets

Simulation of survival data from Cox, AFT and AH (R package survMS)

https://gitlab-research.centralesupelec.fr/2017sautreuim/survms

- Based on Bender et al. (2005) for the Cox model and extended to two others survival models (AFT and AH)
- The expression of the survival time is written in a general way :

$$\mathrm{T} = \frac{1}{\psi_1(\mathrm{X})} \mathrm{H}_0^{-1} \left(\frac{\log(1-\mathrm{U})}{\psi_2(\mathrm{X})} \right) \tag{1}$$

$$(\psi_1(\mathrm{X}), \psi_2(\mathrm{X})) = \begin{cases} (1, \exp(\beta^{\mathrm{T}}\mathrm{X})) & \text{for the Cox model} \\ (\exp(\beta^{\mathrm{T}}\mathrm{X}), \exp(-\beta^{\mathrm{T}}\mathrm{X})) & \text{for the AH model} \\ (\exp(\beta^{\mathrm{T}}\mathrm{X}), 1) & \text{for the AFT model} \end{cases}$$

and $\mathrm{U} \sim \mathcal{U}[0,1]$

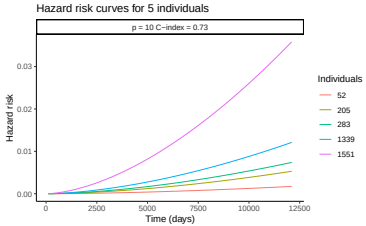
- $\psi(\mathrm{X})$: interactions and non-linear (perspectives)

Simulated datasets

Cox/Weibull model:

➙ Survival time following a Weibull distribution $\mathcal{W}(\alpha, \lambda)$

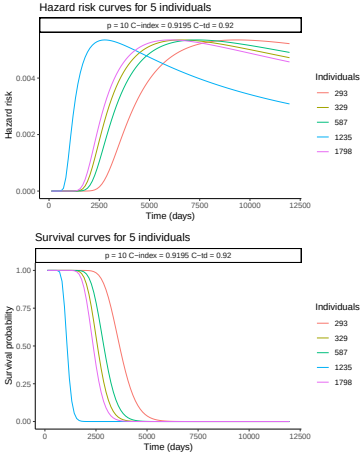
$$\mathrm{T} = \left(-\frac{1}{\lambda}\log(1-\mathrm{U})\exp(-\beta^{\mathrm{T}}\mathrm{X_i})\right)^{\frac{1}{\alpha}}$$



AFT/Log-normal model:

➙ Survival times following a Log-normal distribution $\mathcal{LN}(\alpha, \lambda)$

$$\mathrm{T} = \frac{\left(\exp(\sigma\Phi^{-1}(\mathrm{U})+\mu)\right)}{\exp((1/\sqrt{\mathrm{p}})\beta^{\mathrm{T}}\mathrm{X})}$$

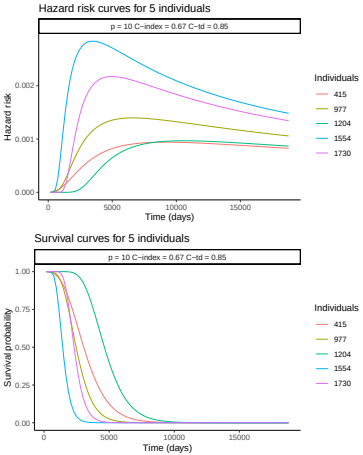


Simulated datasets

Shifted AFT/Log-normal model:

➙ Survival times following a Log-normal distribution $\mathcal{LN}(\alpha, \lambda)$

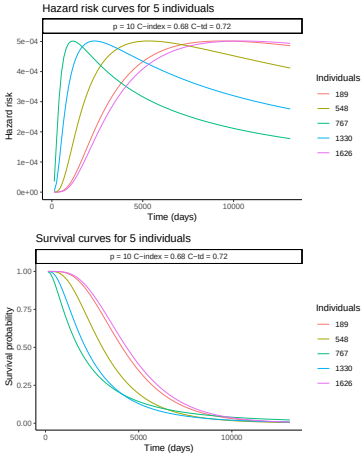
$$\mathrm{T} = \frac{\left(\exp(\sigma\Phi^{-1}(\mathrm{U}) + \mu) + \beta_2^{\mathrm{T}}\mathrm{X}\right)}{\exp((1/\sqrt{\mathrm{p}})\beta^{\mathrm{T}}\mathrm{X})}$$



AH/Log-normal model:

➙ Survival times following a Log-normal distribution $\mathcal{LN}(\alpha, \lambda)$

$$\mathrm{T} = \frac{1}{\exp(\beta^{\mathrm{T}}\mathrm{X_i})} \exp\left[\sigma\Phi^{-1}\left(\frac{\log(1-\mathrm{U})}{\exp(-\beta^{\mathrm{T}}\mathrm{X_i})}\right) + \mu\right]$$



Summary and conclusion - Simulated datasets

1. Cox/Weibull model
   - Methods based on Cox partial log-likelihood: best results
   - In high-dimension: best results for neural network based on Cox partial log-likelihood (cox-nnet)
2. AFT/Log-normal model
   - Neural networks based on discrete time model: best for n = 1 000
   - Cox-nnet: best for n = 200
3. Shifted AFT/Log-normal - AH/Log-normal model
   - Best results for neural networks based on discrete time model
   - CoxL1 et cox-nnet: good results

	n = 200			n = 1000		
methods \ p	10	100	1000	10	100	1000
Reference ($C^{\star}_{td}$)	0.8468	0.8589	0.8459	0.8468	0.8589	0.8459
NNsurv	0.8080	0.7764	0.5607	0.8404	0.8391	0.7098
NNsurv deep	0.8385	0.7746	0.6028	0.8463	0.8361	0.7021
NNsurvK	0.8197	0.5870	0.5610	0.8404	0.7990	0.6154
Cox-nnet	0.8448	0.7747	0.5916	0.8441	0.8410	0.6678
CoxL1	0.8449	0.5893	0.5168	0.8457	0.8381	0.5456

Conclusion

- In most situations: Cox-nnet
- Complex data: Deep version of neural network based on discrete time model

→ Neural networks: good performance for a cohort of a thousand patients with one hundred covariates

Results - Real dataset

- ccRCC: clear cell Renal carcinoma cancer (TCGA database)
  - ➙ 17 789 covariates (genomic) for 533 individuals and 67.8% censored individuals

		CoxL1	Cox-nnet	NNsurv Deep	NNsurv
KIRC	C_{td}	0.5115	0.5277	0.5741	0.5741
	IBS	0.2069	0.2076	0.2869	0.2491

Summary

- coxL1: best value of IBS
  - Only pertinent covariates (Lasso) → best prediction
- Low performance: Many covariates and high censorship rate

➙ Perspective: preliminary variable selection

Results - Real dataset

- Metabric: breast cancer (from UK and Canada project)
  - ➙ 800 covariates (RNA-seq and clinical) for 1981 individuals and 55% censored individuals

		CoxL1	Cox-nnet	NNsurv Deep	NNsurv
Metabric	C_{td}	0.6757	0.6676	0.6853	0.6728
	IBS	0.1937	0.1965	0.1972	0.2038

Summary

- ➙ Good performance of Neural networks: but marginaly higher than coxL1
- ➙ Confirm good performance of Neural networks for a cohort of a thousand patients with a hundred covariates

Marker detection for the ccRCC

Conclusion

- Identification of the genes implied in the ccRCC
  - ➙ Target HLA-G/ILT potential strategy in the case of no response to anti-PD1/PDL-1
- Identification of the genes impacting the survival duration
  1. Discovery of new markers
     - B7H3, CHEK2, CKAP4, CUBN, FBXL5
     - ➙ To validate biologically
  2. Stability study of regularization and screening methods
     - Screening methods: more stable
     - Regularization and screening methods: same quality of adjustement
     - ➙ Quantifying on only one real dataset
     - ➙ Sørensen index: ill-adapted

Perspective

- ➙ Extend the stability study to simulated datasets with other indexes

Marker detection for the ccRCC

Perspectives

1. Proposing a new index: Approximated F_{score} based on the number of hypothetical true covariates
   - Precision and Recall:

$$\text{Precision}(n^*) = \frac{n^* - \sum_{i=1}^{n^*}(1 - s_i)}{\sum_{i=1}^{N} s_i} = \frac{\sum_{i=1}^{n^*} s_i}{\sum_{i=1}^{N} s_i}$$

$$\text{Recall}(n^*) = \frac{n^* - \sum_{1 \leq i \leq n^*}(1 - s_i)}{n^*} = \frac{\sum_{i=1}^{n^*} s_i}{n^*}$$

   - $F_{score}(n^*) = 2\frac{\text{Precision}(n^*)\text{Recall}(n^*)}{\text{Precision}(n^*)+\text{Recall}(n^*)}$
2. Generating simulated datasets from Cox model
   - Use of R package survMS
3. Compute F_{score} to validate the selection (pertinent covariates are known)

Mathilde Sautreuil, Sarah Lemler, Paul-Henry Cournède, Benchmarking the Stability of Variable Selection Methods in the Cox Model, in process

Survival prediction

Conclusion

- In most situations:
  - Best neural network: Cox-nnet
  - ➙ Based on the Cox framework
  - ➙ Neural network enables to handle nonlinear effects and interactions
- In the most complex situations:
  - Best neural network: NNsurv deep (with several hidden layers)
  - ➙ Neural network enables to estimate directly the hazard risk
  - ➙ Handles better non-proportional risks and crossing survival curves

➙ NN: good performance for a cohort of a thousand patients with one hundred covariates

Survival prediction

Perspectives

- Publish the R survMS (survival Models Simulation) package on CRAN
- Study neural networks recently developed
  - Based on Cox model with time as covariate (Kvamme and Borgan, 2019)
  - Based on pseudo-observations (Zhao and Feng, 2020; Roblin et al., 2020)
  - Multi-task neural networks (Goncalves et al., 2020)
- Other models (e.g. mixture models (Bussy et al., 2019), random survival forest (Ishwaran et al., 2008))
- Variable selection before neural network
- Interpretability of neural networks

Thanks for your attention

References I

Antolini, L., Boracchi, P., and Biganzoli, E. (2005). A time-dependent discrimination index for survival data. Statistics in Medicine, 24(24), 3927–3944.

Bender, R., Augustin, T., and Blettner, M. (2005). Generating survival times to simulate Cox proportional hazards models. Statistics in Medicine, 24(11), 1713–1723.

Biganzoli, E., Boracchi, P., Mariani, L., and Marubini, E. (1998). Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. Statistics in Medicine, 17(10), 1169–1186.

Bussy, S., Guilloux, A., Gaïffas, S., and Jannot, A.-S. (2019). C-mix: A high-dimensional mixture model for censored durations, with applications to genetic data. Statistical Methods in Medical Research, 28(5), 1523–1539. PMID: 29658407.

Ching, T., Zhu, X., and Garmire, L. X. (2018). Cox-nnet: An artificial neural network method for prognosis prediction of high-throughput omics data. PLOS Computational Biology, 14(4), e1006076.

Fan, J. and Lv, J. (2017). Sure independence screening for ultrahigh dimensional feature space. Journal of the Royal Statistical Society: Series B (Statistical Methodology), pages 849–911.

Faraggi, D. and Simon, R. (1995). A neural network model for survival data. Statistics in Medicine, 14(1), 73–82.

Gerds, T. A. and Schumacher, M. (2006). Consistent Estimation of the Expected Brier Score in General Survival Models with Right-Censored Event Times. Biometrical Journal, 48(6), 1029–1040.

Goncalves, A., Soper, B., Nygård, M., Nygård, J. F., Ray, P., Widemann, D., and Sales, A. P. (2020). Improving five-year survival prediction via multitask learning across hpv-related cancers. PloS one, 15(11).

Gremel, G., Djureinovic, D., Niinivirta, M., Laird, A., Ljungqvist, O., Johannesson, H., Bergman, J., Edqvist, P.-H., Navani, S., Khan, N., Patil, T., Sivertsson, ., Uhlén, M., Harrison, D. J., Ullenhag, G. J., Stewart, G. D., and Pontén, F. (2017). A systematic search strategy identifies cubilin as independent prognostic marker for renal cell carcinoma. BMC cancer, 17(1), 9.

Guilloux, A., Lemler, S., and Taupin, M.-L. (2016). Adaptive estimation of the baseline hazard function in the Cox model by model selection, with high-dimensional covariates. Journal of Statistical Planning and Inference, 171, 38–62.

Hong, H. G., Kang, J., and Li, Y. (2018). Conditional screening for ultra-high dimensional covariates with survival outcomes. Lifetime Data Analysis, 24(1), 45–71.

References II

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., and Lauer, M. S. (2008). Random survival forests. The Annals of Applied Statistics, 2(3), 841–860. arXiv: 0811.1645.

Kvamme, H. and Borgan, . (2019). Continuous and Discrete-Time Survival Prediction with Neural Networks. arXiv:1910.06724 [cs, stat]. arXiv: 1910.06724.

Lee, C., Zame, W., Yoon, J., and van der Schaar, M. (2018). Deephit: A deep learning approach to survival analysis with competing risks. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32.

Liestbl, K., Andersen, P. K., and Andersen, U. (1994). Survival analysis and neural nets. Statistics in Medicine, 13(12), 1189–1200.

Mani, D. R., Drew, J., Betz, A., and Datta, P. (1999). Statistics and data mining techniques for lifetime value modeling. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '99, pages 94–103, San Diego, California, USA. Association for Computing Machinery.

Roblin, E., Cournède, P.-H., and Michiels, S. (2020). On the Use of Neural Networks with Censored Time-to-Event Data. In G. Bebis, editor, Proceedings of IMSCO 2020. LNBI.

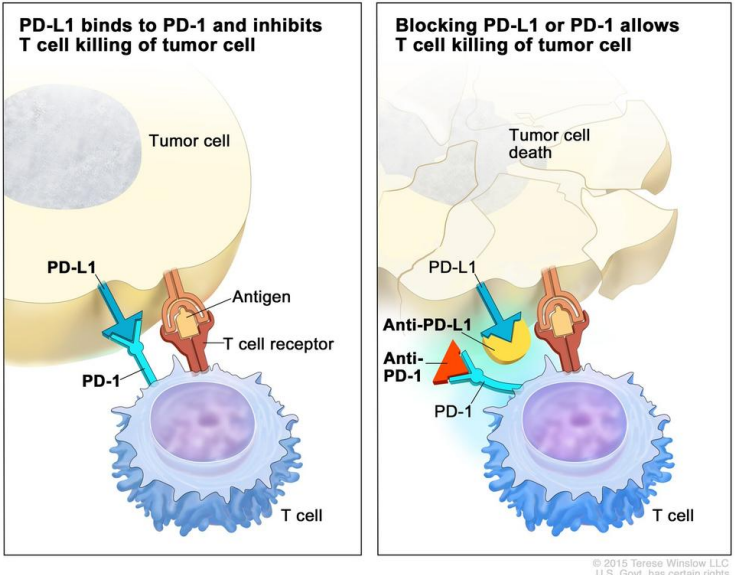
Saldana, D. F. and Feng, Y. (2018). SIS : An R Package for Sure Independence Screening in Ultrahigh-Dimensional Statistical Models. Journal of Statistical Software, 83(2).

Sautreuil, M., Lemler, S., and Cournede, P.-H. (2019). Réseaux de neurones pour l'analyse de survie en grande dimension.

Zhao, L. and Feng, D. (2020). DNNSurv: Deep Neural Networks for Survival Analysis Using Pseudo Values. arXiv:1908.02337 [cs, stat]. arXiv: 1908.02337.

Zhao, S. D. and Li, Y. (2012). Principled sure independence screening for Cox models with ultra-high-dimensional covariates. Journal of Multivariate Analysis, 105(1), 397–411.

Immune-checkpoints



Metrics

Concordance Index

- Concordance index (C_{td}): score indicating how well prediction corresponds to ranks of survival data

$$\widehat{C}^{td} = \frac{\sum_{i=1}^{n} \sum_{j \neq i} conc_{ij}^{td}}{\sum_{i=1}^{n} \sum_{j \neq i} comp_{ij}} \tag{2}$$

$$comp_{ij} = 1_{\{(t_i < t_j ; \delta_i = 1) \cup (t_i = t_j ; \delta_i = 1, \delta_j = 0)\}} \text{ et}$$
$$conc_{ij}^{td} = 1_{\{S(t_i|X_i) < S(t_j|X_j)\}} comp_{ij}.$$

- $\{(t_i, \delta_i, S(t_{(k)}, X_i); k = 1, \ldots, K)\}$, avec :
- t_i : observed time of individual i
- δ_i : censorship indicator
- $S(t_{(k)}, X_i)$: predicted survival function

Metrics

Integrated Brier Score

- Integrated Brier Score (IBS): score computing the squared error between the predicted survival probability and the actual survival of patients at each time point

$$\text{IBS} = \frac{1}{\tau}\int_0^\tau \widehat{\text{BS}}(t, \widehat{S})dt,$$

  - $\widehat{\text{BS}}(t, \widehat{S})$ is the expected Brier score

- To estimate the Brier Score from right-censored data:

$$\widehat{\text{BS}}(t, \widehat{S}) = \frac{1}{N}\sum_{i \in \widetilde{D}_N} \widehat{W}_i(t)(\widetilde{Y}_i(t) - \widehat{S}(t|X_i))^2$$

  - $\widetilde{Y}_i = 1_{\{Y_i > t\}}$ the observed status and N the number of samples in $\widetilde{D}_N$ (test)

Brier Score

Estimation from right-censored data

- Squared residuals are weighted using Inverse Probability of Censoring Weights (IPCW) (Gerds and Schumacher, 2006) given by:

$$\widehat{W}_i(t) = \frac{(1 - \widetilde{Y}_i(t))\Delta_i}{\widehat{G}(\widetilde{T}_{i-}|X_i)} + \frac{\widetilde{Y}_i(t)}{\widehat{G}(t|X_i)}, \tag{3}$$

  - $\widehat{G}(t|x) \approx P(C_i > t|X_i = x)$ estimate of the conditional survival function of the censoring times (e.g.: Kaplan-Meier estimate).

Studying of stability (Regularization methods)

		Lasso	Ridge	Adaptive Lasso	Elastic Net
Immune-Checkpoints	Sørensen index	0.9960	0.9975	0.9933	0.9940
	Jaccard index	0.73	0.80	0.60	0.62
	Fscore(n*) (n*=20)	0.8682	0.9525	0.703	0.898
	Number of selected genes	15.36 (2.83)	20 (fixed)	10.84 (3.53)	20 (fixed)
	AIC	1915.50 (4.33)	1919.42 (2.01)	1917.71 (11.06)	1932.69 (3.34)
Differential expressed genes	Sørensen index	0.9946	0.9500	0.9436	0.9501
	Jaccard index	0.65	0.58	0.14	0.64
	Fscore(n*) (n*=20)	0.739	0.8635	0.523	0.872
	Number of selected genes	11.72 (2.34)	20 (fixed)	7.84 (3.01)	20 (fixed)
	AIC	1867.35 (1.95)	1869.61 (2.91)	1862.47 (23.04)	1878.10 (4.45)
All genes	Sørensen index	0.9332	0.9940	0.8284	0.9755
	Jaccard index	0.12	0.62	0.05	0.28
	Fscore(n*) (n*=20)	0.7225	0.8605	0.4754	0.672
	Number of selected genes	17.70 (3.57)	20 (fixed)	8.65 (3.64)	20 (fixed)
	AIC	1873.43 (24.95)	1870.44 (8.33)	1870.42 (40.97)	1874.36 (5.60)

Studying of stability (Screening methods)

		SIS	ISIS	PSIS	coxCS
Immune-Checkpoints	Sørensen index	0.9708	0.6089	0.9983	0.9974
	Jaccard index	0.2495	0.798	0.8566	0.796
	Fscore(n*) (n*=20)	0.2277	0.3532	0.6004	0.5704
	Number of selected genes	2.57 (2.23)	4.29 (0.69)	8.58 (0.57)	7.98 (2.01)
	AIC	1953.37 (21.36)	1935.35 (3.66)	1961.22 (4.33)	1946.50 (9.92)
Differential expressed genes	Sørensen index	0.9905	0.382	0.9662	0.8885
	Jaccard index	0.5101	0.5841	0.2207	0.8127
	Fscore(n*) (n*=20)	0.3416	0.3519	0.8824	0.6188
	Number of selected genes	4.12 (1.57)	4.27 (1.02)	18.51 (5.11)	8.96 (1.47)
	AIC	1903.63 (7.78)	1895.25 (5.92)	1944.20 (5.27)	1960.39 (3.50)
All genes	Sørensen index	0.9962	0.8956	0.9610	0.9341
	Jaccard index	0.7222	0.6212	0.2492	0.1231
	Fscore(n*) (n*=20)	0.4496	0.424	0.7494	0.7814
	Number of selected genes	5.80 (1.04)	5.39 (1.50)	27.21 (9.18)	25.85 (14.44)
	AIC	1873.80 (0.71)	1880.01 (24.69)	1931.38 (12.55)	1937.71 (13.69)

Studying of stability (Simulated Datasets)

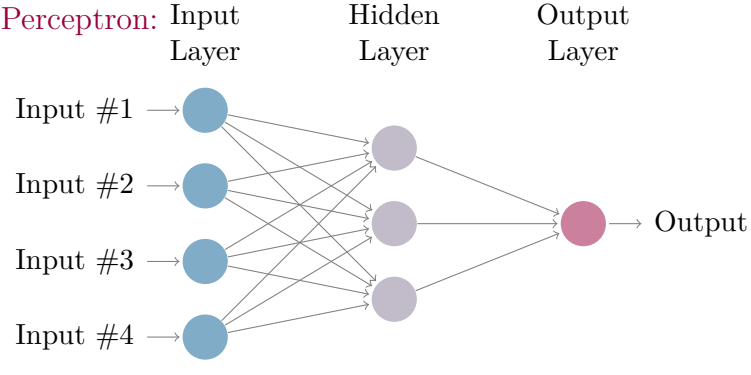
Methods	#Covariates	Sorensen	Jaccard	#Genes selected	Fscore(n*) (n*=20)	Fscore	AIC
Adaptive-Lasso	1000	0.9937	0.6122	25.26 (7.0534)	0.8506	0.384	5063.25 (30.4043)
	25000	0.9892	0.4788	4.84 (2.2862)	0.3897	0	5153.96 (27.0826)
coxCS1	1000	0.9989	0.9	27.03 (1.5005)	0.8505	0.7974	5154.83 (0)
	25000	0.9992	0.9251	26.85 (1.1135)	0.8538	0.8021	5180.22 (0)
coxCS2	1000	0.9997	0.9697	23.28 (0.8885)	0.9242	0.7985	5153.42 (0)
	25000	0.9997	0.9662	25.13 (0.6913)	0.8863	0.8482	5196.65 (0)
Elastic-Net	1000	0.9784	0.3115	21.97 (11.388)	0.823	0.3826	5038.4054 (510.0245)
	25000	0.9748	0.2788	1.43 (0.9348)	0.1335	0	5197.3719 (12.4215)
ISIS	1000	0.9995	0.9519	20 (0)	0.999	0.35	5071.75 (1.0458)
	25000	1	1	20 (0)	1	0.05	5003.24 (0)
lasso	1000	0.9939	0.6184	28 (6.9515)	0.8304	0.38	5058.89 (28.5348)
	25000	0.9869	0.4298	4.79 (2.5556)	0.3864	0	5156.06 (27.1983)
PSIS	1000	1	1	6 (0)	0.4615	0.4615	5151.49 (0)
	25000	1	1	4 (0)	0.3333	0.3333	5164.32 (0)
SIS	1000	1	1	20 (0)	1	0.35	5097.24 (0)
	25000	1	1	20 (0)	1	0.05	5034.58 (0)

Studying of stability (Simulated Datasets)

Methods	#Covariates	#Genes selected	Fscore(n*) (n*=20)	Fscore
coxCS2	1000	23.28 (0.8885)	0.9242	0.7985
	25000	25.13 (0.6913)	0.8863	0.8482
lasso	1000	28 (6.9515)	0.8304	0.38
	25000	4.79 (2.5556)	0.3864	0
SIS	1000	20 (0)	1	0.35
	25000	20 (0)	1	0.05

Neural networks

Multi-Layer Perceptron:



- Constituting by one input layer, at least one hidden layer, and one output layer
- Each neuron of layers plays the role of a non-linear regression between its inputs and output variables
- Coefficients of its regression are called weights and non-linear transformation of its combinaison is called activation function

Neural network based on discrete time model (Biganzoli et al., 1998)

Structure of neural network

- I have coded the neural network of Biganzoli et al. to adapt it to high-dimension
- I have coded the 2nd approach of neural network

➙ Use of keras library

- Due to large numbers of variables ➙ Overfitting
  - Biganzoli et al. had already proposed a regularization term: ridge
  - ➙ I implemented a cross-validation procedure to choose the regularization parameter λ
- Many configurations were tested:
  - Number of nodes in hidden layer ➙ $\sqrt{p}$
  - With or without dropout ➙ without
  - Optimization methods (adam, sgd) ➙ adam
  - Early stopping ➙ with n = 200, without n = 1000
  - Batch size ➙ chosen by cross-validation

Datasets

Simulated datasets from Cox model

- Considering the survival times distributed following Weibull distribution $\mathcal{W}(\alpha, \lambda)$, with:
  - Baseline function of the form $\alpha_0(t) = \alpha\lambda t^{\alpha-1}$
  - Inverse cumulative hazard function $H_0^{-1}(t) = \left(\frac{t}{\lambda}\right)^{1/\alpha}$
  - Survival time T of the Cox model:

$$T = \left(-\frac{1}{\lambda}\log(1-U)\exp(-\beta^T X_i)\right)^{\frac{1}{\alpha}} \tag{4}$$

- Setting $\alpha = 2.67$ and $\lambda = 7.5e^{-10}$ to have mean and variance close to real datasets
- Design matrix X simulated from an uniform distribution on $[-1, 1]$.
- Number of samples: 200 and 1000

Datasets

Simulated datasets from AFT model

- Considering the survival times distributed following Log-normal distribution $\mathcal{LN}(\alpha, \lambda)$, with:
  - Baseline function of the form $h_0(t) = \frac{\frac{1}{a\sqrt{2\pi}t} \exp\left[-\frac{(\log t - \lambda)^2}{2a^2}\right]}{1-\Phi\left[\frac{\log t - \lambda}{a}\right]}$
  - Inverse cumulative hazard function $H_0^{-1}(t) = \exp\left(\alpha\Phi^{-1}(1-\exp(-t)) + \lambda\right)$
  - Survival time T of AFT model:

$$T = \frac{1}{\exp((1/\sqrt{p})\beta^T X)} \left(\exp(\sigma\Phi^{-1}(U) + \mu)\right). \tag{5}$$

    - $\Phi(t)$: cumulative distribution function of normal distribution ($\mathcal{N}(0,1)$)
    - $(1/\sqrt{p})$: normalization term

- Setting $\alpha = 0.7$ and $\lambda = 7.71$ to have mean and variance close to real datasets
- Design matrix X simulated from an uniform distribution on $[0, 1]$.
- Number of samples: 200 and 1000

Datasets

Simulated datasets from modified AFT model

- Considering the survival times distributed following Log-normal distribution $\mathcal{LN}(\alpha, \lambda)$, with:
  - Baseline function of the form $h_0(t) = \frac{\frac{1}{a\sqrt{2\pi t}} \exp\left[-\frac{(\log t - \lambda)^2}{2a^2}\right]}{1-\Phi\left[\frac{\log t - \lambda}{a}\right]}$
  - Inverse cumulative hazard function $H_0^{-1}(t) = \exp\left(\alpha\Phi^{-1}(1-\exp(-t)) + \lambda\right)$
  - Survival time T of AFT model:

$$T = \frac{1}{\exp((1/\sqrt{p})\beta^T X)}\left(\exp(\sigma\Phi^{-1}(U) + \mu) + \beta_2^T X\right). \quad (6)$$

  - $\beta_2 \sim \mathcal{U}[-1.5, 1.5]$
  - $\Phi(t)$: cumulative distribution function of normal distribution ($\mathcal{N}(0,1)$)
  - $(1/\sqrt{p})$: normalization term
- Setting $\alpha = 0.7$ and $\lambda = 7.71$ to have mean and variance close to real datasets
- Design matrix X simulated from an uniform distribution on $[0, 1]$.
- Number of samples: 200 and 1000

Results - Cox/Weibull datasets

	n	200			1000		
Méthode	p	10	100	1000	10	100	1000
Référence	$C^{\star}_{td}$	0.7442	0.7428	0.7309	0.7442	0.7428	0.7309
	IBS*	0.0471	0.0549	0.0582	0.0471	0.0549	0.0582
NNsurv	C_{td}	0.7137	0.6224	0.5036	0.7398	0.7282	0.5700
	IBS	0.0980	0.0646	0.1359	0.0759	0.0537	0.1007
NNsurvK	C_{td}	0.6261	0.5135	0.5173	0.7312	0.6504	0.5699
	IBS	0.1310	0.1121	0.1137	0.1178	0.1011	0.1130
NNsurv deep	C_{td}	0.7225	0.5982	0.5054	0.7424	0.7236	0.5741
	IBS	0.0878	0.0689	0.1080	0.0591	0.0555	0.1185
NNsurvK deep	C_{td}	0.6178	0.4784	0.4112	0.7112	0.5772	0.4748
	IBS	0.1324	0.1122	0.1561	0.1179	0.1023	0.1260
Cox -nnet	C_{td}	0.7313	0.6481	0.5351	0.7427	0.7309	0.6110
	IBS	0.0688	0.0622	0.1402	0.0640	0.0498	0.0710
CoxL1	C_{td}	0.7292	0.5330	0.5011	0.7419	0.7243	0.5
	IBS	0.0715	0.0672	0.1175	0.0541	0.0509	0.0770

Results - AFT/Log-normale datasets

	n	200			1000		
Méthode	p	10	100	1000	10	100	1000
Référence	$C^{\star}_{td}$	0.9203	0.9136	0.9037	0.9203	0.9136	0.9037
	IBS*	0.0504	0.0604	0.0417	0.0504	0.0604	0.0417
NNsurv	C_{td}	0.9832	0.8349	0.5425	0.9851	0.9038	0.7426
	IBS	0.0265	0.0560	0.2577	0.0247	0.0188	0.0642
NNsurvK	C_{td}	0.9802	0.7118	0.5575	0.9856	0.8707	0.6049
	IBS	0.1425	0.1043	0.1468	0.1319	0.0820	0.0979
NNsurv deep	C_{td}	0.9786	0.8275	0.5576	0.9857	0.9060	0.7500
	IBS	0.0295	0.0561	0.1886	0.0261	0.0207	0.0631
NNsurvK deep	C_{td}	0.9791	0.6976	0.5694	0.9861	0.8716	0.6090
	IBS	0.1079	0.1049	0.1905	0.0984	0.0657	0.1334
Cox -nnet	C_{td}	0.9825	0.8558	0.5979	0.9844	0.9060	0.7085
	IBS	0.0122	0.0906	0.0959	0.0126	0.0374	0.0808
CoxL1	C_{td}	0.9867	0.7827	0.5091	0.9856	0.9028	0.5349
	IBS	0.0146	0.0965	0.0960	0.0077	0.0182	0.0827

Results - AH/Log-normale datasets

	n	200			1000		
Méthode	p	10	100	1000	10	100	1000
Référence	$C^{\star}_{td}$	0.7225	0.6857	0.7070	0.7225	0.6867	0.7070
	IBS*	0.0755	0.0316	0.0651	0.0755	0.0316	0.0651
NNsurv	C_{td}	0.6863	0.5971	0.5358	0.7084	0.6088	0.5654
	IBS	0.1247	0.0780	0.0859	0.0699	0.0347	0.0533
NNsurvK	C_{td}	0.6151	0.5258	0.5025	0.7107	0.6214	0.5159
	IBS	0.1267	0.1087	0.1396	0.1020	0.0459	0.0790
NNsurv deep	C_{td}	0.7042	0.5793	0.5325	0.7155	0.6450	0.5702
	IBS	0.1789	0.2529	0.1554	0.0602	0.0303	0.0484
NNsurvK deep	C_{td}	0.6067	0.4847	0.5025	0.7138	0.5570	0.5199
	IBS	0.1234	0.1058	0.1328	0.1048	0.0451	0.0558
Cox -nnet	C_{td}	0.7128	0.5812	0.5356	0.7097	0.6047	0.5720
	IBS	0.1342	0.2243	0.1609	0.0843	0.0875	0.0553
CoxL1	C_{td}	0.7042	0.5219	0.5112	0.7088	0.5597	0.5
	IBS	0.1350	0.2278	0.1614	0.0608	0.0408	0.0553

Results - shifted AFT/Log-normale datasets

	n	200			1000		
Méthode	p	10	100	1000	10	100	1000
Référence	$C^{\star}_{td}$	0.8468	0.8589	0.8459	0.8468	0.8589	0.8459
	IBS*	0.0294	0.0199	0.0305	0.0294	0.0199	0.0305
NNsurv	C_{td}	0.8080	0.7764	0.5607	0.8404	0.8391	0.7098
	IBS	0.0624	0.0775	0.0669	0.0532	0.0564	0.0651
NNsurvK	C_{td}	0.8197	0.5870	0.5610	0.8404	0.7990	0.6154
	IBS	0.0859	0.1003	0.1235	0.0771	0.0759	0.0856
NNsurv deep	C_{td}	0.8385	0.7746	0.6028	0.8463	0.8361	0.7021
	IBS	0.0487	0.0897	0.0759	0.0363	0.0312	0.0510
NNsurvK deep	C_{td}	0.7941	0.4673	0.5559	0.8394	0.7716	0.6011
	IBS	0.0838	0.0942	0.1237	0.0735	0.0744	0.0843
Cox -nnet	C_{td}	0.8448	0.7747	0.5916	0.8441	0.8410	0.6678
	IBS	0.0347	0.0717	0.0819	0.0323	0.0680	0.0622
CoxL1	C_{td}	0.8449	0.5893	0.5168	0.8457	0.8381	0.5456
	IBS	0.0354	0.0933	0.0818	0.0267	0.0429	0.0628

Results - Censored shifted AFT/Log-normale datasets

	n	200			1000		
Méthode	p	10	100	1000	10	100	1000
Référence	$C^{\star}_{td}$	0.8718	0.8917	0.8765	0.8718	0.8917	0.8765
	IBS*	0.0473	0.0569	0.0482	0.0473	0.0569	0.0482
NNsurv	C_{td}	0.8600	0.8086	0.5175	0.8697	0.8706	0.6990
	IBS	0.1064	0.1009	0.2866	0.1335	0.0673	0.1952
NNsurvK	C_{td}	0.8063	0.6810	0.5422	0.8591	0.7866	0.6063
	IBS	0.1704	0.1946	0.2856	0.1961	0.1550	0.1523
NNsurv deep	C_{td}	0.8431	0.7168	0.5463	0.8710	0.8739	0.7155
	IBS	0.1212	0.1268	0.1142	0.0869	0.0587	0.1013
NNsurvK deep	C_{td}	0.8193	0.5633	0.5217	0.8435	0.7466	0.5921
	IBS	0.1925	0.2038	0.2883	0.2018	0.1593	0.1520
Cox -nnet	C_{td}	0.8643	0.8038	0.5	0.8697	0.8730	0.7145
	IBS	0.0613	0.1233	0.1192	0.0529	0.0844	0.0961
CoxL1	C_{td}	0.8623	0.6107	0.5309	0.8694	0.8659	0.5160
	IBS	0.0602	0.1340	0.1394	0.0667	0.0799	0.1142

Results - Sparse shifted AFT/Log-normale datasets

	n	200			1000		
Méthode	p	10	100	1000	10	100	1000
Référence	$C^{\star}_{td}$	0.8673	0.8673	0.8673	0.8673	0.8673	0.8673
	IBS*	0.0284	0.0284	0.0284	0.0284	0.0284	0.0284
NNsurv	C_{td}	0.8684	0.8012	0.5902	0.8766	0.8646	0.7436
	IBS	0.1254	0.1129	0.0738	0.0621	0.1566	0.0622
NNsurvK	C_{td}	0.8648	0.5215	0.5581	0.8770	0.8511	0.6566
	IBS	0.1094	0.0987	0.0995	0.0899	0.0872	0.0835
NNsurv deep	C_{td}	0.8744	0.8062	0.5938	0.8761	0.8664	0.7284
	IBS	0.0474	0.0488	0.0739	0.0378	0.0304	0.0487
NNsurvK deep	C_{td}	0.8610	0.5100	0.5263	0.8746	0.8227	0.5835
	IBS	0.1099	0.0992	0.1091	0.0913	0.0848	0.0869
Cox -nnet	C_{td}	0.8742	0.7922	0.5832	0.8757	0.8683	0.6952
	IBS	0.0885	0.0773	0.1015	0.0532	0.0519	0.0699
CoxL1	C_{td}	0.8759	0.8686	0.8733	0.8739	0.8743	0.8726
	IBS	0.0904	0.0805	0.0754	0.0300	0.0291	0.0290

Results - Real datasets

- KIRC: Clear cell renal cell carcinoma (from TCGA database)
  - ➙ 17781 covariates (genomic) for 533 indivduals and 67% censored individuals
- Metabric: breast cancer (from UK and Canada project)
  - ➙ 800 covariates (genomic and clinical) for 1981 indivduals and 55% censored individuals

		Cox	Cox-nnet	NNsurv Deep	NNsurv	NNsurvK
KIRC	C_{td}	0.5115	0.5277	0.5741	0.5888	0.6076
	IBS	0.2069	0.2075	0.2869	0.?	0.4928
Metabric	C_{td}	0.6757	0.6676	0.6853	0.6728	0.6015
	IBS	0.1937	0.1965	0.1972	0.2038	0.43698

- NNsurv: Neural network based on discrete time model adapted to the high-dimension
- NNsurv deep: NNsurv with several hidden layers
- NNsurvK: Neural network based on discrete time model with modifications
- Cox-nnet: Neural network based on Cox partial log-likelihood adapted by Ching et al. to the high-dimension ➙ Estimation of $\alpha_0(t)$ to get the estimated survival duration
- CoxL1: Cox partial log-likelihood with Lasso procedure

Counting processes in the specific case of right censoring

Counting processes [Aalen, 1980] :

- $N_i(t) = 1_{\{X_i \leq t,\ \delta_i = 1\}}$ counting process
- $Y_i(t) = 1_{\{X_i \geq t\}}$ at-risk process