Tutorial thesis

Author: Ludivine Morvan

Institute:

Date: June 7, 2020

Version: 3.09

Victory won't come to us unless we go to it.

Contents

Chapter 1 Biology

1.0.1 Multiple Myeloma

(From article Morvan et al. (2020)) [[[Multiple myeloma (MM) is a bone marrow cancer that accounts for 10% of all hematological malignancies. It was reported that full-body FDG PET imaging provides prognostic information for both baseline and therapeutic follow-up of MM patients [an example of full-body FDG PET imaging is presented in Fig. 2b]. Quantitative imaging have a great importance for treatment protocol guidance.]]]

1.0.2 PET images

1.0.3 CT images

Chapter 2 Imaging

2.0.1 Radiomics

2.0.2 Mathematics morphology

2.0.3 Filters

Chapter 3 Datasets

Dataset	Input	task
Imajem	Images 3D (PET+CT) + clinical	Survival
Bologna	Images 3D (PET+CT) + clinical	Survival
Brats	Images 3D (MRI)	Segmentation + classification + survival
MNIST	Images 2D (handwritted digits)	classification

Table 3.1: List of the datasets

3.1 Imajem

Context: prospective multi-centric Multiple myeloma Carlier et al. (2017)

Number of patients: 134 in total

Input images: PET and CT images of whole body.

Input others: clinical data and radiomics and volumics

task: Survival (until 7 years). In days with censorship (...%)

masks: global polygon generated from Dosisoft and segmentation(majority vote between k-mean, 2.5suv and 40%)

3.2 Bologna

Context: prospective multi-centric Multiple myeloma Nanni et al. (2018)

Number of patients:

Input images: PET and CT images of whole body.

Input others: clinical data and radiomics and volumics

task: Survival. In days with censorship (...%)

masks: global polygon generated from Dosisoft

Chapter 4 Machine learning

4.1 Dimension reduction or selection

- PCA
- T-SNE
- Selection derived from Cox
- Selection derived from RSF

4.1.1 PCA

4.1.2 T-SNE

4.1.3 Selection derived from Cox

4.1.3.1 Lasso

4.1.3.2 Elatic Net

4.1.4 Selection derived from RSF

4.1.4.1 Variable importance (VIMP)

(From article Morvan et al. (2020)) [[[The *variable importance* (VIMP) measures for each variable, the increase in prediction error for the forest ensemble when random daughter nodes are assigned for this variable Ishwaran et al. (2018).]]]

4.1.4.2 Minimal depth

(From article Morvan et al. (2020)) [[[The *minimal depth* assesses the predictiveness of a variable by its depth relative to the root node of a tree Ishwaran et al. (2010).]]]

4.1.4.3 Variable Hunting (VH)

(From article Morvan et al. (2020)) [[[Variable-Hunting Ishwaran et al. (2010), was defined in the ultra-high dimensional problems, where minimal depth thresholding becomes ineffective. A forest is fit to a random number of variables and variables selected using minimal depth thresholding. Then, variables are added to the selected ones in order of minimal depth until the joint VIMP for the nested models stabilizes. This whole process is then repeated several times.]]]

4.2 Random Forest

Chapter 5 Deep Learning

5.1 Definitions

supervised/semi-supervised/unsupervised

Latent space, manifold

5.2 Possible layers

5.2.1 Convolutions

1. Basic convolution
2. Separable convolution

5.2.1.1 Basic convolution

5.2.1.2 Separable convolutions

5.2.2 Pooling

- Max-pooling
- Average pooling
- spacial pooling Fig.
- spacial pyramidal pooling (SPP)

5.2.2.1 Spatial pooling

Spatial pooling is represented in Fig. 5.1.

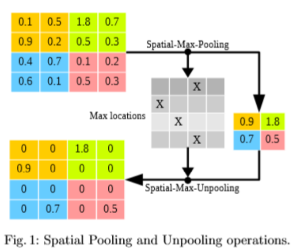

Figure 5.1: Spacial pooling

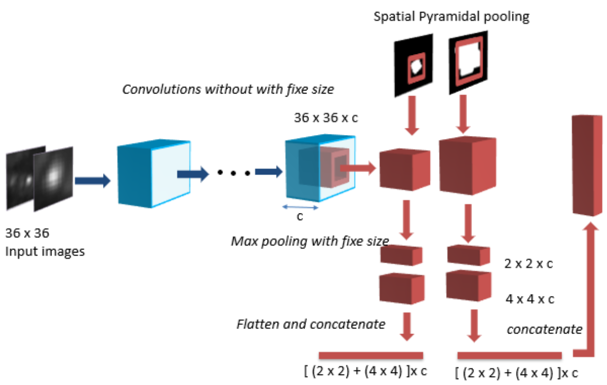

Figure 5.2: Figure of SPP from miccai 2020 article

5.2.2.2 Spatial Pyramidal Pooling

To handle different size images. (Fig.5.2)

5.2.3 normalisation

1. Batch normalisation
2. Instance normalisation

5.2.3.1 Batch normalisation

5.2.3.2 Instance normalisation

5.3 Losses

5.3.1 classification

5.3.2 regression

5.3.3 segmentation

5.3.4 survival

5.3.5 ranking

5.4 Others

5.4.1 Bilinear

5.4.2 Attention

(Fig. 5.3) Figs. 5.5 and 5.4 are examples of channel and spatial weight obtained from the

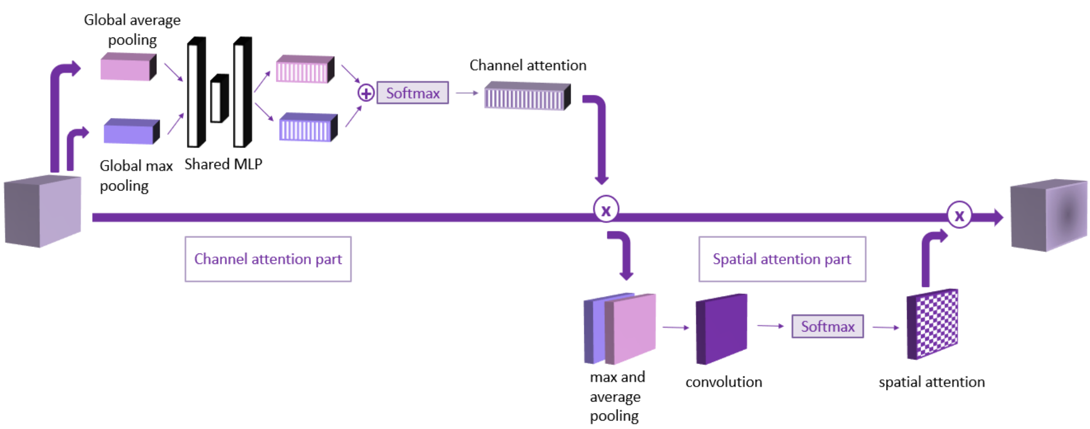

Figure 5.3: Attention block (figure from Miccai 2020 paper)

attention block.

Figure 5.4: Example of Channel attention weights (in columns)for a set of patients. Color varies from blue to red for increasing weight values.

Figure 5.5: For every 3-image block: left) Spatial attention weights. middle) Input image. right) Segmentation mask (used only for validation here).

5.4.2.1 Papers using it

- reconstruction Huang et al. (2019)
- segmentation and detection Tong et al. (2019)
- classification Herent et al. (2019) (with images)
- Survival Kaji DA (2019) (model clinical events in the intensive care unit, no images)

5.5 models

5.5.1 classification

5.5.2 regression

5.5.3 segmentation

u-net

5.5.4 survival

1. DeepSurv
2. DeepConvSurv
3. RankDeepSurv

5.5.5 ranking

1. RankDeepSurv [cf. survival losses]

Chapter 6 Metrics

6.1 For classification

6.2 For segmentation

6.3 For regression

6.4 For survival analysis

6.5 For ranking

Chapter 7 Survival

7.1 Definitions

7.2 Machine learning models

7.2.1 Kaplan Meier model

7.2.2 Cox model

7.2.3 Gradient boosting cox

7.2.4 Random Survival Forest

(From article Morvan et al. (2020)) [[[A *Random Forest* is a collection of N_{trees} decision trees $\mathcal{T} = \{f_1, \ldots, f_{N_{\text{trees}}}\}$ trained to predict a target value y given an input feature vector $\mathbf{x} \in \mathbb{R}^D$ composed of the concatenation of D features. The target y can be a categorical or continuous variable. One decision tree f consists of a series of nodes, each characterized by a binary split decision function

The *Random Survival Forest* is also an ensemble-tree method, introduced by Ishwaran in 2008 Ishwaran et al. (2018) to adapt the Random Forests to right-censored data and survival analysis. For survival, every input feature vector $\mathbf{x}_i$ is accompanied by a censorship θ_i, which is equal to 1 if an event occurred during the study period and 0 if not. The target value y_i is the expected ensemble mortality, an interpretation in terms of the expected total number of deaths and derived from the ensemble cumulative hazard function (CHF). It is estimated from the given time to event data τ_i for the individuals i in the training set.

As before, tree growing is done through randomized node optimization, generating several candidates for feature (axis aligned split) and threshold $\{\phi, th\}$. However, instead of an information theoretic criteria, the split function and threshold are chosen to maximize the survival difference between the individuals going to the two daughter nodes. In particular, we use the *log-rank* criteria: Considering $t_1 < \ldots < t_m$ the distinct times of events in the node h. $d_{k,l}$ and $Y_{k,l}$ respectively the numbers of event and the number of individuals at risk in the left daughter node without event at t_{k-1} ($d_{k,r}$ and $Y_{k,r}$ for the right daughter node). $Y_k = Y_{k,l} + Y_{k,r}$ and $d_k = d_{k,l} + d_{k,r}$.

$$L(\phi, th) = \frac{\sum_{k=1}^{m}(d_{k,l} - Y_{k,l})\frac{d_k}{Y_k}}{\sqrt{\sum_{k=1}^{m}\frac{Y_{k,l}}{Y_k}(1 - \frac{Y_{k,l}}{Y_k})\frac{Y_k - d_k}{Y_k - 1})d_k}} \tag{7.1}$$

to evaluate the best population separation. Finally, unlike class histogram or regression value in RF, each leaf in RSF stores the ensemble mortality and a survival curve. An example of a survival tree is presented in Fig. **??**. The mortality is calculated with the Nelson-Aalen estimator for the Cumulative Hazard Function (CHF):

$$\hat{H}_h(t) = \sum_{t_{k,h} \leq t} \frac{d_{k,h}}{Y_{k,h}}, \tag{7.2}$$

with k the index of the event time between 1 and m, $\hat{H}_h(t)$ the CHF at the node h and time t; while $d_{k,h}$ stands for the number of event at time $t_{k,h}$ and $Y_{k,h}$ is the number of individuals at risk at time $t_{k,h}$. It can be interpreted as the sum on each event time of rate of deaths. The ensemble mortality M_i of an individual i in the node h is the sum of CHF on each unique time:

$$M_i = \sum_{k=1}^{m} \hat{H}_h(t_k|X_i) \tag{7.3}$$

The ensemble mortality is the expected value for the CHF summed over time. It measures the number of deaths expected under a null hypothesis of similar survival behavior.]]]

7.3 Deep learning models

Chapter 8 Python programming

8.1 OS system

Check if a path exists

```
os.path.exists(pathCSV)
```

Check if a variable exists

```
try:
    myVar
except NameError:
    myVar = 1
```

8.2 Numpy/arrays

To load a numpy file

```
data=np.load(doss+"data.npy")
```

To save a numpy file

```
data.save(doss+"data.npy")
```

8.3 plot graphics

How to use matplotlib colors : Website

8.3.1 Box Plot

lstlisting plt.subplot(121) plt.boxplot([[1, 2, 3, 4, 5, 13], [6, 7, 8, 10, 10, 11, 12], [1, 2, 3]]) plt.ylim(0, 14) plt.title('boxplot avec sequence')

plt.subplot(122) plt.boxplot(numpy.array([[1, 2, 3], [2, 7, 8], [1, 3, 10], [2, 5, 12]])) plt.ylim(0, 14) plt.title('boxplot avec array 2d')

8.4 Images

8.4.0.1 Open an image

Figure 8.1: Matplotlib colors

```
from PIL import Image
im = Image.open("data/image.jpeg")
 plt.imshow(np.array(im))
```

```
from skimage import io
image = io.imread(os.path.join(PATH,patient,'image.tif')).T
```

Open a .nii.gz (Lien)

```
import nibabel as nib
img = nib.load(os.path.join(data_path, 'example4d.nii.gz'))
img = np.array(img.get_fdata()) # pour passer de nifti a array
```

Open a .mha

```
from medpy.io import load
image_data, image_header = load('../data/image.mha')
```

8.4.0.2 show an image

```
plt.imshow(image)\textit{(image should be 2D, RGB or not, array)}
```

8.4.0.3 Save an image

To save in Tiff

```
from skimage.external import tifffile
tifffile.imsave(im_Path, np_image.T)
```

8.5 Pandas

Read a csv

```
data = pd.read_csv(pathCSV, encoding='utf-8')
```

Take a value

```
data.iloc[0,1].values #iloc: take a value thanks to the index. .loc: take thanks to the name
```

Pass from dataframe to array

```
data.values
```

From array to dataframe

```
fff = pd.DataFrame(rtrain)
```

Save a dataFrame

```
data.to_csv(doss+"data.csv")
```

Change/obtain the row names

```
data.index
```

8.6 Tensorflow and Keras

8.6.1 Tensorboard

8.6.1.1 Sites

- https://itnext.io/how-to-use-tensorboard-5d82f8654496
- add hyperparams https://github.com/tensorflow/tensorboard/issues/46

8.6.1.2 Bases

Imports

```
from keras.callbacks import TensorBoard
import time
```

At the begining

```
NAME = mode + "-{}".format(int(time.time()))
tensorboard = TensorBoard(log_dir='Resultats/logs/{}'.format(NAME))
```

In the model

```
fitting = model.fit(xtrain, ytrain, validation_data=(xval, yval), callbacks=[tensorboard])
```

In prompt

```
tensorboard --logdir="D:\Documents\thèse\simple_model\Resultats\logs\CNN_small-1565598264"
```

8.6.1.3 Tensorboard

- to view all scalars in the same time: write * in te research barre

Chapter 9 R programming

9.1 Create a package

Follow the following tutorial

9.2 Open files

Open numpy

```
library ( reticulate )
np <- import("numpy")
mat <- np$load("fmat.npy")
```

Open csv

```
data <- read.csv2(path ,header=T,sep=',' ,dec='.')
#https :// pandas.pydata.org/pandas-docs/stable / reference / api/pandas.read_csv.html
```

9.3 Modify

Concatenate

```
a= rbind(b,c) #for rows
b= cbind(a,c) #for columns
```

Rename Columns

```
colnames(x) <- c("col1","col2")
```

9.4 PCA

```
library ("FactoMineR")
library (" factoextra ")
#http :// www.sthda.com/french/ articles /38-methodes-des-composantes-principales-dans-r-guide-pratique
    /73-acp-analyse-en-composantes-principales-avec-r-l-essentiel/
```

```
pca <- PCA(features, graph = FALSE,scale.unit = TRUE,ncp=100)
# graph : Show or not the graph. ncp: number of dim to keep. Si scale. unit = True normalisation
eig. val <- get_eigenvalue (pca) #Extraction des valeurs propres
fviz_eig (res.pca) # Visualisation des valeurs propres
```

```
get_pca_ind(res.pca), get_pca_var(res.pca) # Extraction des résultats pour les individus et les
    variables , respectivement .
fviz_pca_ind(res.pca), fviz_pca_var(res.pca) # visualisez les résultats des individus et des variables ,
    respectivement .
fviz_pca_biplot(res.pca) # Création dun biplot des individus et des variables
```

```
NewXtrain <- pca$ind$coord #to obtain the features in the new dimensions. number of features = ncp de
    PCA.
NewXtest = predict(pca,XTEST)$coord #to obtain the features in the new dimensions of a new dataset
```

To print cumulative information according to the number of kept dimensions.

```
eig.val <- get_eigenvalue(pca)
barplot(eig.val$cumulative.variance.percent [1:100], names.arg=c(1:100))
abline(h=80, col = "Red") # horizontal line
abline(v=56, col = "Red") # vertical line
```

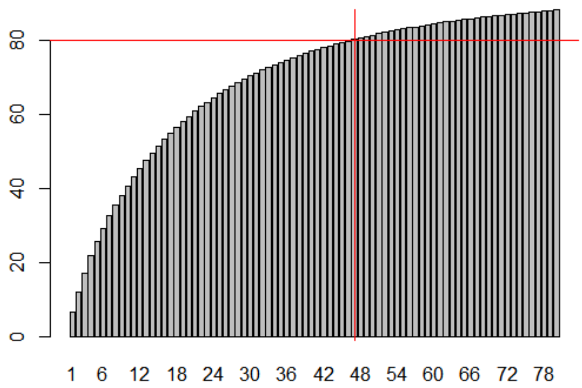

Figure 9.1: Cumulative information of PCA

Chapter 10 Latex programming

Aide mémoire

10.1 Design

Put a line

```
\headrule # full  line
```

10.2 special caracters

Alphabet *grec*

α	`\alpha`	ζ	`\zeta`	$\varkappa$	`\varkappa`	ϖ	`\varpi`	υ	`\upsilon`	$\digamma$	`\digamma`	Π	`\Pi`
β	`\beta`	η	`\eta`	λ	`\lambda`	ρ	`\rho`	ϕ	`\phi`	Γ	`\Gamma`	Σ	`\Sigma`
γ	`\gamma`	θ	`\theta`	μ	`\mu`	ρ	`\rho`	φ	`\varphi`	Δ	`\Delta`	Υ	`\Upsilon`
δ	`\delta`	ϑ	`\vartheta`	ν	`\nu`	σ	`\sigma`	χ	`\chi`	Θ	`\Theta`	Φ	`\Phi`
ϵ	`\epsilon`	ι	`\iota`	ξ	`\xi`	ς	`\varsigma`	ψ	`\psi`	Λ	`\Lambda`	Ψ	`\Psi`
ε	`\varepsilon`	κ	`\kappa`	π	`\pi`	τ	`\tau`	ω	`\omega`	Ξ	`\Xi`	Ω	`\Omega`

Symboles *alphanumériques*

$\aleph$	`\aleph`	$\gimel$	`\gimel`	ℓ	`\ell`	∂	`\partial`	$\forall$	`\forall`	$\exists!$	`\exists!`	$\neg$	`\neg`
$\beth$	`\beth`	$\daleth$	`\daleth`	∞	`\infty`	$\wp$	`\wp`	$\exists$	`\exists`	$\varnothing$	`\varnothing`	∇	`\nabla`

Lois *de composition*

$+$	`+`	$\times$	`\times`	$\star$	`\star`	$\sqcup$	`\sqcup`	$\oplus$	`\oplus`	$\lhd$	`\lhd`	$\amalg$	`\amalg`
$-$	`-`	$\cdot$	`\cdot`	$\circ$	`\circ`	$\sqcap$	`\sqcap`	$\otimes$	`\otimes`	$\rhd$	`\rhd`	$\wr$	`\wr`
$\pm$	`\pm`	$\div$	`\div`	$\cup$	`\cup`	$\amalg$	`\amalg`	$\wedge$	`\wedge`	$\rtimes$	`\rtimes`	$\bot$	`\bot`
$\mp$	`\mp`	$*$	`*`	$\cap$	`\cap`	$\vartriangle$	`\vartriangle`	$\vee$	`\vee`	$\ltimes$	`\ltimes`	$\top$	`\top`

Symboles *de relation*

Égalités.

$=$	`=`	$\neq$	`\neq`	$\propto$	`\propto`	$\not\propto$	`\not\propto`	$\simeq$	`\simeq`	$\not\simeq$	`\not\simeq`	$\approxeq$	`\approxeq`
$\coloneqq$	`\coloneqq`	$:\neq$	`\vcentcolon\neq`	$\asymp$	`\asymp`	$\not\asymp$	`\not\asymp`	$\approx$	`\approx`	$\not\approx$	`\not\approx`	$\cong$	`\cong`
$\equiv$	`\equiv`	$\not\equiv$	`\not\equiv`	$\sim$	`\sim`	$\not\sim$	`\not\sim`						

$\not\approxeq$	`\not\approxeq`
$\not\cong$	`\not\cong`

Inclusions.

$\subset$	`\subset`	$\subseteq$	`\subseteq`	$\subsetneq$	`\subsetneq`	$\subseteqq$	`\subseteqq`	$\subsetneqq$	`\subsetneqq`	$\sqsubset$	`\sqsubset`	$\sqsubseteq$	`\sqsubseteq`
$\not\subset$	`\not\subset`	$\nsubseteq$	`\nsubseteq`	$\supsetneq$	`\supsetneq`	$\nsubseteqq$	`\nsubseteqq`	$\supsetneqq$	`\supsetneqq`	$\not\sqsubset$	`\not\sqsubset`	$\not\sqsubseteq$	`\not\sqsubsete`
$\supset$	`\supset`	$\supseteq$	`\supseteq`	$\varsubsetneq$	`\varsubsetneq`	$\supseteqq$	`\supseteqq`	$\varsubsetneqq$	`\varsubsetneqq`	$\sqsupset$	`\sqsupset`	$\sqsupseteq$	`\sqsupseteq`
$\not\supset$	`\not\supset`	$\nsupseteq$	`\nsupseteq`	$\varsupsetneq$	`\varsupsetneq`	$\nsupseteqq$	`\nsupseteqq`	$\varsupsetneqq$	`\varsupsetneqq`	$\not\sqsupset$	`\not\sqsupset`	$\not\sqsupseteq$	`\not\sqsupsete`

Inégalités.

$<$	`<`	$>$	`>`	$\leq$	`\leq`	$\geq$	`\geq`	$\leqslant$	`\leqslant`	$\geqslant$	`\geqslant`	$\ll$	`\ll`

$\gg$	`\gg`

Flèches

$\to$	`\to`	$\implies$	`\implies`	$\nearrow$	`\nearrow`	$\nrightarrow$	`\nrightarrow`
$\mapsto$	`\mapsto`	$\iff$	`\iff`	$\swarrow$	`\swarrow`	$\mapstochar\nrightarrow$	`\mapstochar\nrightarrow`
$\hookrightarrow$	`\hookrightarrow`	$\centernot\implies$	`\centernot\implies` (pkg centernot)	$\rightleftarrows$	`\rightleftarrows`	$\rightsquigarrow$	`\rightsquigarrow`
$\twoheadrightarrow$	`\twoheadrightarrow`	$\centernot\iff$	`\centernot\iff` (pkg centernot)	$\rightrightarrows$	`\rightrightarrows`	$\dashrightarrow$	`\dashrightarrow`

Flèches extensible. `\xrightarrow{f}` donne $\xrightarrow{f}$ et `\xrightarrow[dessous]{dessus}` donne $\xrightarrow[dessous]{dessus}$.

$\xleftarrow{}$	`\xleftarrow`	$\xRightarrow{}$	`\xRightarrow`	$\xrightharpoondown{}$	`\xrightharpoondown`	$\xrightleftharpoons{}$	`\xrightleftharpoons`
$\xrightarrow{}$	`\xrightarrow`	$\xLeftarrow{}$	`\xLeftarrow`	$\xrightharpoonup{}$	`\xrightharpoonup`	$\xleftrightharpoons{}$	`\xleftrightharpoons`
$\xmapsto{}$	`\xmapsto`	$\xLeftrightarrow{}$	`\xLeftrightarrow`	$\xleftharpoondown{}$	`\xleftharpoondown`	$\xhookleftarrow{}$	`\xhookleftarrow`
$\xleftrightarrow{}$	`\xleftrightarrow`			$\xleftharpoonup{}$	`\xleftharpoonup`	$\xhookrightarrow{}$	`\xhookrightarrow`

Fonctions *usuelles*

`\ln`	$\ln$	`\cos`	$\cos$	`\arctan`	$\arctan$	`\deg`	$\deg$	`\hom`	$\hom$	`\varlimsup`	$\varlimsup$
`\exp`	$\exp$	`\sin`	$\sin$	`\sinh`	$\sinh$	`\det`	$\det$	`\lg`	$\lg$	`\projlim`	proj lim
`\lim`	$\lim$	`\tan`	$\tan$	`\cosh`	$\cosh$	`\dim`	$\dim$	`\log`	$\log$	`\varprojlim`	$\varprojlim$
`\max`	$\max$	`\cot`	$\cot$	`\tanh`	$\tanh$	`\ker`	$\ker$	`\liminf`	$\liminf$	`\injlim`	inj lim
`\sup`	$\sup$	`\arccos`	$\arccos$	`\coth`	$\coth$	`\arg`	$\arg$	`\varliminf`	$\varliminf$	`\varinjlim`	$\varinjlim$
`\min`	$\min$	`\arcsin`	$\arcsin$	`\inf`	$\inf$	`\gcd`	$\gcd$	`\limsup`	$\limsup$		

Pour définir de nouvelles fonctions : `\DeclareMathOperator{\Vect}{Vect}`

Délimiteurs

délimiteurs ouvrants et fermants

(x)	`(x)`	$\lvert x\rvert$	`\lvert x\rvert`	$\langle x\rangle$	`\langle x\rangle`		$/$	`/`	$\vert$ `|`
$[x]$	`[x]`	$\lVert x\rVert$	`\lVert x\rVert`	$\llbracket x\rrbracket$	`\llbracket x\rrbracket`	(stmaryrd)	$\backslash$	`\backslash`	$\Vert$ `|`
$\{x\}$	`\{x\}`	$\lfloor x\rfloor$	`\lfloor x\rfloor`	$\lceil x\rceil$	`\lceil x\rceil`				

délimiteurs médiants: $/$ `/`, $\vert$ `|`, $\backslash$ `\backslash`, $\Vert$ `\|`

Pour $]-1\,;1[$, utiliser la commande `\intervalleoo` ci-dessous.

Pour changer la taille : `\left`, `\right`, `\middle`, `\big`, `\bigl`, `\bigm`, `\bigr` (ainsi que Big, bigg et Bigg)

`\left` et `\right` sont certaines fois trop grands : $\left[\sum_i a_i\left|\sum_j x_{i,j}\right|^p\right]^{1/p}$ contre $\Big[\sum_i a_i\Big|\sum_j x_{i,j}\Big|^p\Big]^{1/p}$.

Grands *opérateurs*

`\int`	$\int \displaystyle\int$	`\iint`	$\iint \displaystyle\iint$	`\iiint`	$\iiint \displaystyle\iiint$	`\idotsint`	$\idotsint \displaystyle\idotsint$	`\oint`	$\oint \displaystyle\oint$
`\sum`	$\sum \displaystyle\sum$	`\coprod`	$\coprod \displaystyle\coprod$	`\bigcap`	$\bigcap \displaystyle\bigcap$	`\bigoplus`	$\bigoplus \displaystyle\bigoplus$	`\bigwedge`	$\bigwedge \displaystyle\bigwedge$
`\prod`	$\prod \displaystyle\prod$	`\bigcup`	$\bigcup \displaystyle\bigcup$	`\bigsqcup`	$\bigsqcup \displaystyle\bigsqcup$	`\bigotimes`	$\bigotimes \displaystyle\bigotimes$	`\bigvee`	$\bigvee \displaystyle\bigvee$

Utilisation de `\limits`, `\nolimits` et `\displaystyle` ainsi que \sideset

$$\sideset{^*}{'}\prod_{k=1}^{n}$$

Accents *mathématiques*

`\bar` $\bar{a}$ `\tilde` $\tilde{a}$ `\hat` $\hat{a}$ `\check` $\check{a}$ `\acute` $\acute{a}$ `\grave` $\grave{a}$ `\dot` $\dot{a}$ `\ddot` $\ddot{a}$ `\dddot` $\dddot{a}$ `\ddddot` $\ddddot{a}$ `\mathring` $\mathring{a}$
`\vec` $\vec{a}$ `\breve` $\breve{a}$

Flèches extensibles. `$\underbracket{ABC}_{bas}$` donne $\underbracket{ABC}_{bas}$ et `$\overbrace{ABC}^{haut}$` donne $\overbrace{ABC}^{haut}$.

`\overbrace`	$\overbrace{ABC\dots}$	`\underbrace`	$\underbrace{ABC\dots}$	`\widetilde`	$\widetilde{ABCDEFG}$
`\overline`	$\overline{ABC\dots}$	`\underline`	$\underline{ABC\dots}$	`\widehat`	$\widehat{ABCDEFG}$
`\overrightarrow`	$\overrightarrow{ABC\dots}$	`\underrightarrow`	$\underrightarrow{ABC\dots}$	`\widetriangle`	$\widetriangle{ABCDEFG}$
`\overleftarrow`	$\overleftarrow{ABC\dots}$	`\underleftarrow`	$\underleftarrow{ABC\dots}$	`\wideparen`	$\wideparen{ABCDEFG}$
`\overleftrightarrow`	$\overleftrightarrow{ABC\dots}$	`\underleftrightarrow`	$\underleftrightarrow{ABC\dots}$	`\widering`	$\widering{ABCDEFG}$
`\overbracket`	$\overbracket{ABC\dots}$	`\underbracket`	$\underbracket{ABC\dots}$		

Chapter 11 Software tips

11.1 Visual Studio code

11.1.1 ShortCut

ctl+k ctrl+à : Fold All

ctl+k ctrl+j : Unfold all

ctrl S puis ctrl p : Split

ctrl+shift+(: Fold

ctrl+shift+ ˆ : Unfold

F : walk in errors

shift + enter : Run a ligne in interactive terminal

ctrl+k puis ctrl+ T : Choose colors

ctrl+ù : Terminal

ctrl+shift+m : Problem

ctrl+B : Remove/add left barre

ctrl+K z : Zen mode

**ctrl+ ** : Side by side

ctrl K + ctrl S : Key map

11.2 Windows

11.2.1 ShortCut

CMD+P : Impression

Alt+tab : Navigation

Ctrl+tab : Navigation in onglets

Ctrl+del : Delete the entier word

ctrl + Fleche D/G : Deplace word by word

ctrl+G : I and U mis en forme

F2 : Rename

F5 : Actualise

alt + F4 : Close the window

ctrl + F4 : close the doc in software or onglet

alt+enter : Show properties of the doc

ctrl+o : open file in software

wind+fleche : diminish, increase, deplace window

wind+tab : new desk and see all windows

wind+i : parameters of the computer

wind+impr screen : register a screenshot

wind+shift+fleche G/D : deplace window on the other screen

wind+p : change affichage of the screen

11.3 Anaconda

11.3.1 In anaconda prompt

To list all the packages and versions

```
conda list
```

To list the available versions of a packages (ex. with numpy)

```
conda search -f numpy
```

To clean tarballs and cache

```
conda clean -a
```

11.4 Docker

To open pc-milcom

```
ssh ludivine@pc-milcom or ludivine@130.66.84.51
```

To Run command

```
>> docker run -it --rm -u $(id -u):$(id -g) -v /PATH bash
* -it     interactive
* --rm remove at the end
* -u $(id -u):$(id -g)
* -v /PATH bash
* ls2n/ludivine    name of image
* -- name ContainerName give a specific name to a container
* bash: launch to install things and modify
```

Install a package (Windows)

```
In the first CMD:
>> docker run --rm -it -v $PATH bash  (If there is a problem of permission, add: -u 0:0)
>> pip install packagename
In a second CMD
>> ssh name@pc-milcom
>> docker ps -a      ContainerID
>> docker commit ContainerID ls2n/ludivine
```

Useful commands (windows/linux)

```
>> who  who have an open session
>> nvidia -smi  information about the gpu
>> top
>> watch -n 1 nvidia-smi  information about the gpu in real time
>> docker image ls   see all docker images
```

Handle containers

```
docker container attach ContainerName (obtained with docker ps -a)  to see in the shell the running
    container
CTRL-p CTRL-q key combination Quit the container without stopping it .
CTRL-c Stop the container
Open PortNumber in the browser  watch the output of the container process in real time.
docker logs containerID  get acces to the logs
docker container exec -it containerName /bin/bash  run a bash shell inside a running container
```

11.5 Excel

11.5.1 Small hints

Fuse the text from cells A1 and B1

```
=A1&B1
```

Keep 4 first letters of the A1 cell

```
=LEFT(A1, 4)/ GAUCHE(A1,4)
```

11.5.2 Analyse data

- You can do a dynamic Table (tableau croisé dynamique) when you have several parameters, to compare the combination of parameters

11.6 Slicer 3D

11.7 Imajem Fidji

Bibliography

Carlier, Thomas, Clement Bailly, Rodolphe Leforestier, Cyrille Touzeau, Philippe Moreau, Francoise Bodere, and Caroline Bodet-Milin, “Prognostic added value of PET textural features at diagnosis in multiple myeloma,” *Journal of Nuclear Medicine*, 2017, *58* (supplement 1), 111.

et al. Kaji DA Zech JR, Kim JS, “An attention based deep learning model of clinical events in the intensive care unit.,” *PLoS One*, 2019, *14* (2:e0211057).

Herent, P., B. Schmauch, P. Jehanno, O. Dehaene, C. Saillard, C. Balleyguier, J. Arfi-Rouche, and S. Jégou, “Detection and characterization of MRI breast lesions using deep learning,” *Diagnostic and Interventional Imaging*, 2019, *100* (4), 219 – 225.

Huang, Q., D. Yang, P. Wu, H. Qu, J. Yi, and D. Metaxas, “MRI Reconstruction Via Cascaded Channel-Wise Attention Network,” in “2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)” April 2019, pp. 1622–1626.

Ishwaran, H., U. Kogalur, E. Blackstone, and M. Lauer, “Random Surival Forest,” *The annals of applied statistics*, 2018, *2* (3), 841–860.

Ishwaran, Hemant, Udaya B. Kogalur, Eiran Z. Gorodeski, Andy J. Minn, and Michael S. Lauer, “High-Dimensional Variable Selection for Survival Data,” *Journal of the American Statistical Association*, 2010, *105* (489), 205–217.

Morvan, L, T Carlier, B. Bailly C.and Jamet, C Bodet-Milin, P. Moreau, C Touzeau, F Kraeber-Bodere, and D Mateus, “Leveraging RSF and PET images for prognosis of multiple myeloma at diagnosis,” *International Journal of Computer Assisted Radiology and Surgery*, 2020, p. 129139.

Nanni, C., A. Versari, S. Chauvie, and et al., “Interpretation criteria for FDG PET/CT in multiple myeloma (IMPeTUs): final results. IMPeTUs (Italian myeloma criteria for PET USe).,” *European Journal of Nuclear Medicine and Molecular Imaging*, 2018, *45*, 712719.

Tong, Qianqian, Caizi Li, Weixin Si, Xiangyun Liao, Yaliang Tong, Zhiyong Yuan, and Pheng Ann Heng, “RIANet: Recurrent interleaved attention network for cardiac MRI segmentation,” *Computers in Biology and Medicine*, 2019, *109*, 290 – 302.