

# MRIaggr

un package pour la gestion et le traitement de données  
multivariées d'imagerie

Brice Ozenne

Équipe Biostatistique Santé, LBBE, CNRS UMR 5558, France  
Service de Biostatistiques, Hospices Civils de Lyon, France

4<sup>ième</sup> rencontres R, Grenoble - 26 juin 2015



Hospices de Lyon



# Contexte de l'imagerie médicale

## Étude typique

3 niveaux de mesure :

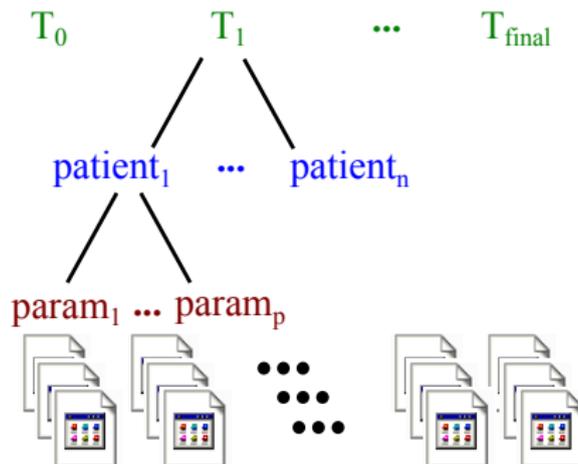
- Temps d'observation
- Patient
- Paramètre IRM

(étude I-know)

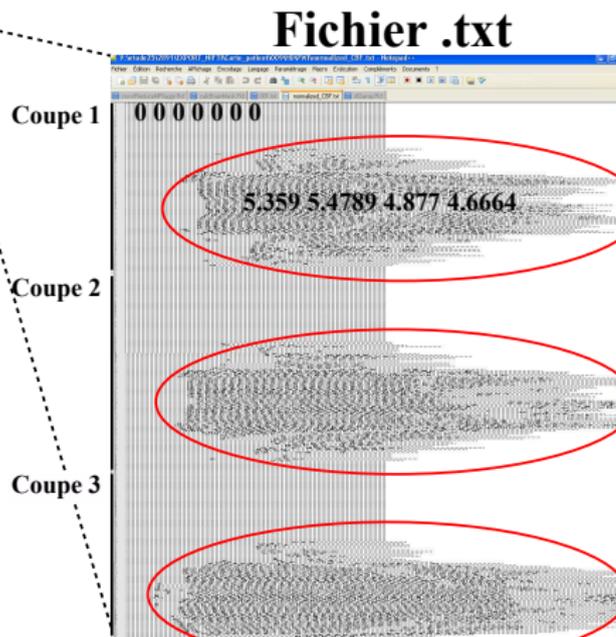
4 temps

63 patients

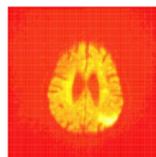
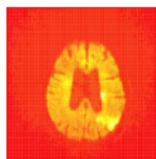
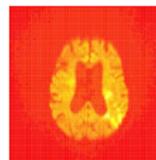
11 paramètres



## Aperçu d'un fichier - format .txt



### Image



1 valeur = 1 voxel (pixel 3D)

## Imagerie médicale - Enjeux

### Volume de données

(I-know)

- Nombreux fichiers de données
- Taille des données

2296 fichiers

2,80 Go

n>327680/fichier

# Imagerie médicale - Enjeux

## Volume de données

### Normalisation des données

(I-know)

- formats spécifiques
- nb de paramètres IRM variables
- résolution spatiale variable

nifti

entre 17 et 31

de  $128 \times 128 \times 15$

à  $256 \times 256 \times 20$

# Imagerie médicale - Enjeux

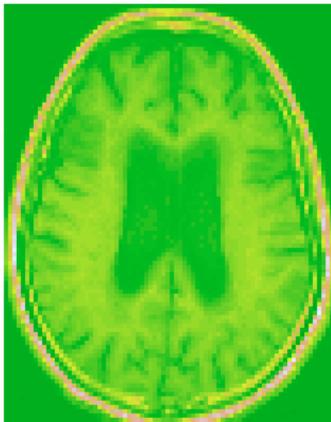
Volume de données

Normalisation des données

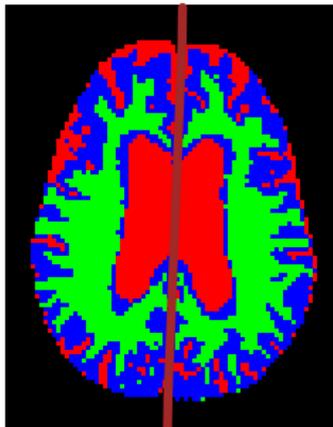
Prétraitement des données

- Identification des structures cérébrales
- Normalisation des paramètres

**T1**



**structures**



■ fond

■ substance grise

■ substance blanche

■ liquide cérébro-spinal

■ plan sagittal moyen

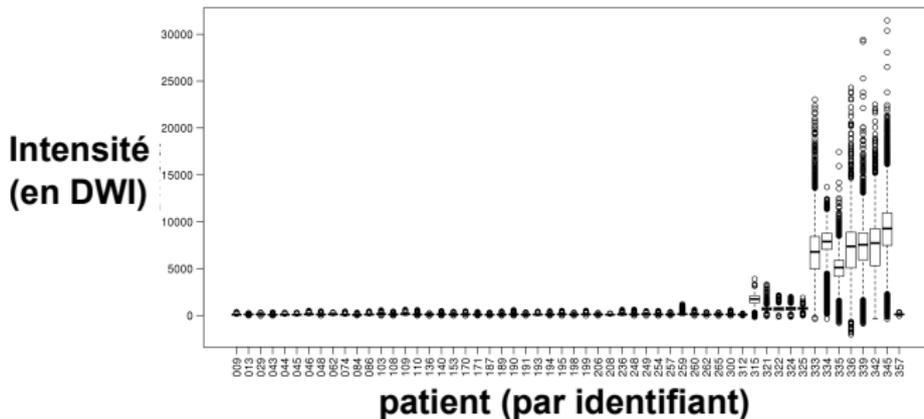
# Imagerie médicale - Enjeux

Volume de données

Normalisation des données

Prétraitement des données

- Identification des structures cérébrales
- Normalisation des paramètres



## Limites des outils sous

Des packages d'analyse existent ...

(<http://cran.r-project.org/web/views/MedicalImaging.html>)

... Mais la gestion de données reste difficile :

- stockage univarié :
  - 1 array / paramètre / patient
  - incompatible avec les fonctions usuelles
- outils de pré-traitement dispersés
  - packages *EImage*, *mmand*, *RNiftyReg*, *fslR*, *mritc* ...
  - ou inexistant

## Vers un cadre de travail plus convivial

Définition d'un **nouveau type d'objet**  :

- agrégeant l'ensemble de l'information patient
- facilitant le prétraitement des données
- permettant la visualisation des données et résultats
- compatible avec les fonctions statistiques usuelles

## Quelques définitions de POO

### Objet

- Entité informatique représentant un **concept** réel  
ex : *cerveau d'un patient*

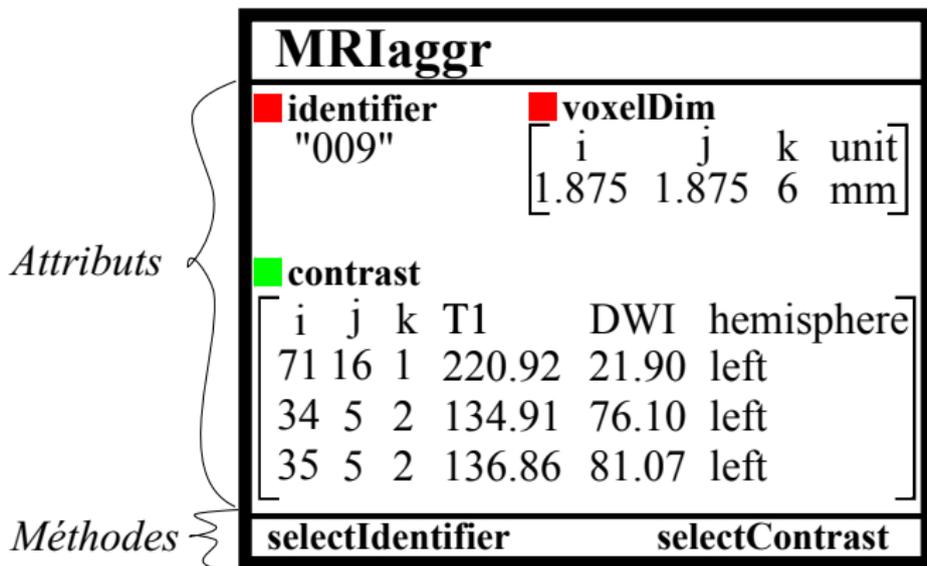
### Attributs

- **Caractéristiques** intrinsèques à l'objet :  
fixes : *identifiant patient*  
ajustables : *localisation de la lésion*

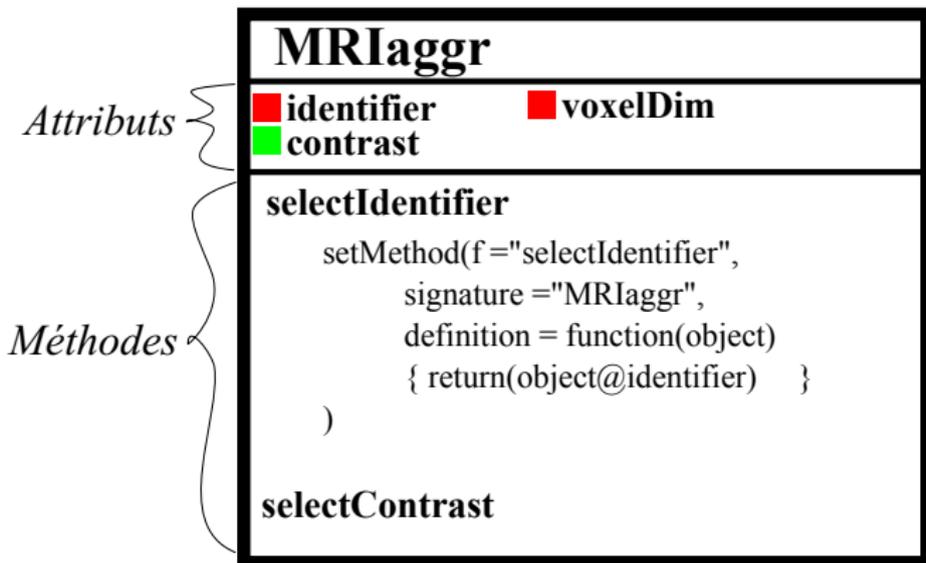
### Méthodes

- **fonctions** spécialement conçues pour traiter les objets :  
multiplot : affichage graphique  
selectClinic : sélection des données cliniques  
calcHemisphere : identification des hémisphères

## Illustration



## Illustration



```
R> selectIdentif(MRIaggr)R> selectContrast(MRIaggr,num=2)
```

```
"009"
```

```

          i j k T1      DWI    ...
432 34 5 2 134.91  76.10 ...
433 35 5 2 136.86  81.07 ...

```

## Synthèse des fonctionnalités

Préfixes associés aux méthodes :

- **read/write** : lecture/écriture de fichiers d'image
- **const** : construction d'objets MRlaggr
- **calc** : prétraitement des données  
update.objet=TRUE : mise à jour automatique de l'objet
- **select** : extraction d'informations
- **alloc/suppr** : mis à jour des données
- **plot** : affichage graphique

## Import et conversion en MRlaggr

```
R> path <- system.file("nifti",package = "MRlaggr")
```

```
R> T1 <- readMRI(file=file.path(path,"T1_t0.nii"),  
+               format="nifti")
```

```
R> Patient1 <- constMRlaggr(T1,  
+                            identifier="001",  
+                            param="T1")
```

```
Merging : (1) T1
```

```
R> summary(Patient1)
```

```
# image dimensions (i,j,k) : 78x100x3 voxels  
# ...
```

## Import et conversion en MRlagger

Similaire en multiparamétrique :

```
R> T2 <- readMRI(file=file.path(path,"T2_GRE_t0.nii"),
+               format="nifti")
R> Lesion <- readMRI(file=file.path(path,"MASK_DWI_t0.nii"),
+                   format="nifti")

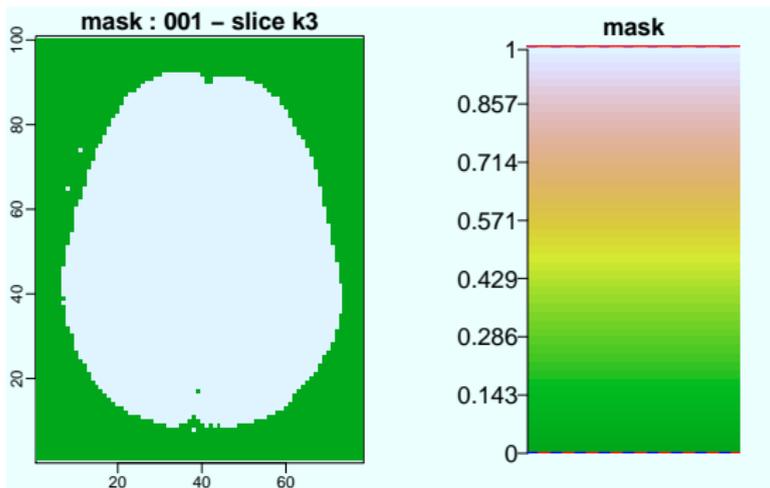
R> Patient1 <- constMRIaggr(list(T1,T2,Lesion),
+                            identifier="001",
+                            param=c("T1","T2","Lesion")
+ )
```

Merging : (1) T1 (2) T2 (3) MASK

## Exemple de prétraitement

Définition du masque cérébral (k-means) :

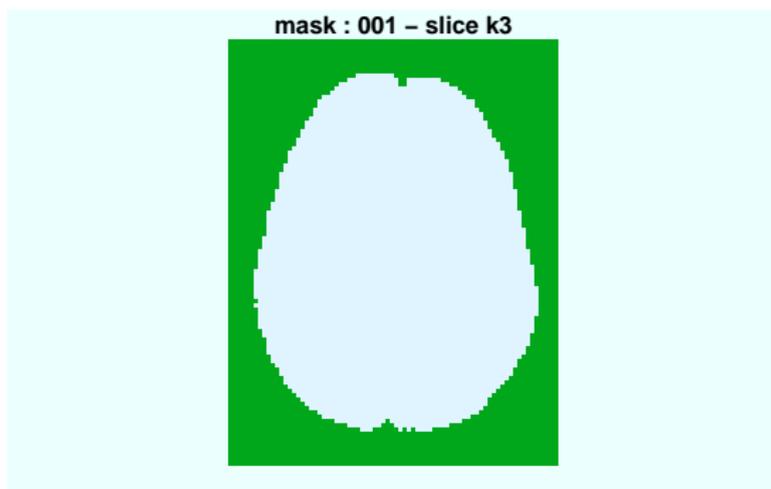
```
R> calcBrainMask(Patient1,param="T2",  
+               update.object=TRUE)  
allocContrast [MRIaggr] : Cartography "mask"  
                    has been allocated
```



## Exemple de prétraitement

Lissage du masque cérébral (critères de voisinage spatial) :

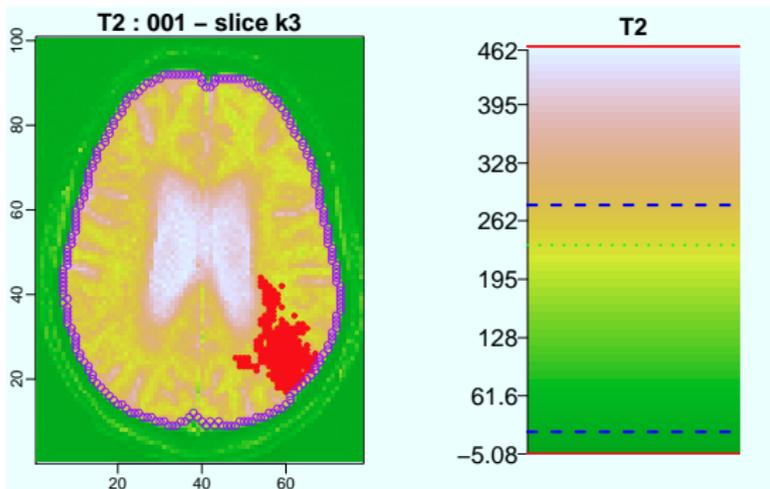
```
R> calcSmoothMask(Patient1,update.object=TRUE,  
+                 overwrite=TRUE)  
allocContrast [MRIaggr] : Cartography "mask"  
                        has been updated
```



## Affichage graphique

Affichage du T2, de la lésion et du masque cérébral :

```
R> multiplot(Patient1,param="T2",num=3,
+           index1="Lesion",as.logical=TRUE,
+           index2=list(coords="mask",outline=TRUE)
+           )
```



## Construction d'objet MRlaggr

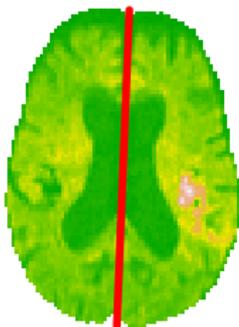
Élimination des voxels hors cerveau :

```
R> Patient1r <- constReduceMRIaggr(Patient1,  
+                                 mask="mask")
```

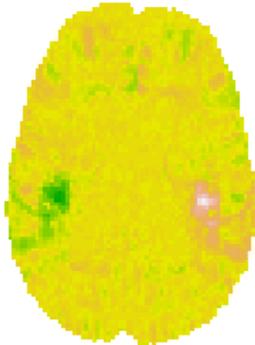
```
R> object.size(Patient1r)/object.size(Patient1)  
0.664266703682022 bytes
```

## Autres pré-traitements possibles

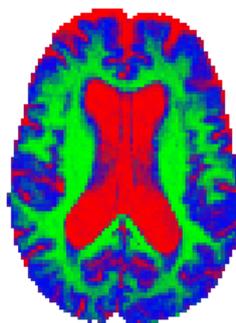
**calcHemisphere**



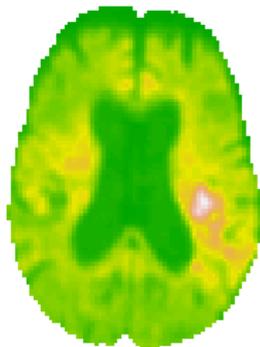
**calcContralateral**



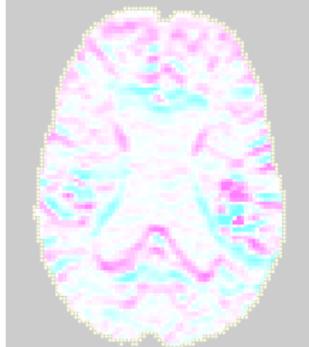
**calcTissueType**



**calcFilter - Gaussian**



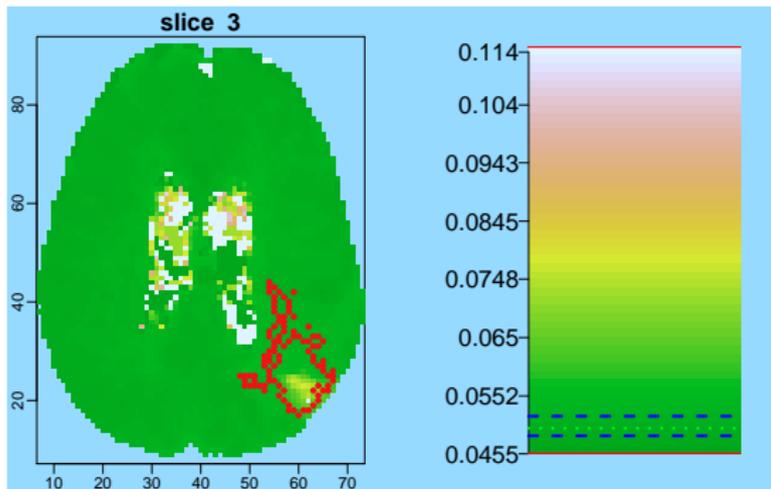
**calcFilter - Sobel**



## Utilisation des fonctions stat usuelles

```
R> data.df <- selectContrast(Patient1r,param=c("TTP","Lesion"))
```

```
R> glm_DWI <- glm(Lesion ~ TTP,data=data.df,family=binomial)
```



## Discussion

### Apports du package MRlaggr

- unifie le stockage des données
- propose un pipeline de pré-traitement (🕒 10 min)
- facilite l'affiche graphique
- possibilité d'inclure des résultats externes

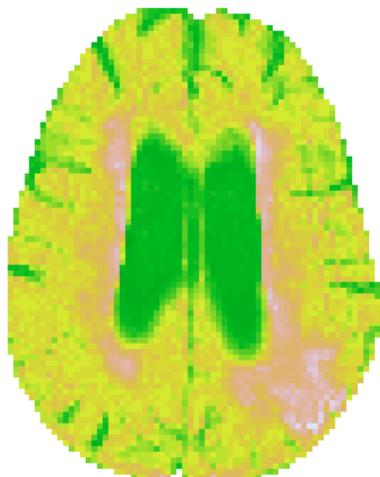
### À venir

- interaction avec RNiftiReg pour la co-registation
- ajouter des fonctions de segmentation (en cours) :
  - segmentation de lésion (OZENNE et al. 2015)
  - lissage par morceaux (STORATH et WEINMANN 2014)

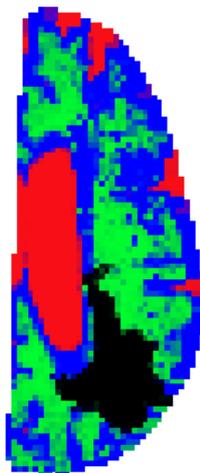
## OZENNE et al. 2015

Segmentation de la lésion avec régularisation de forme :

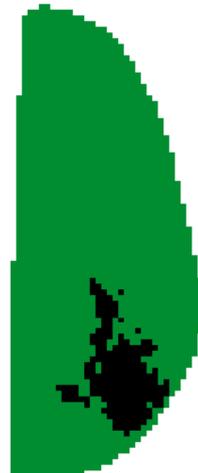
T2 FLAIR



Segmentation



Lésion

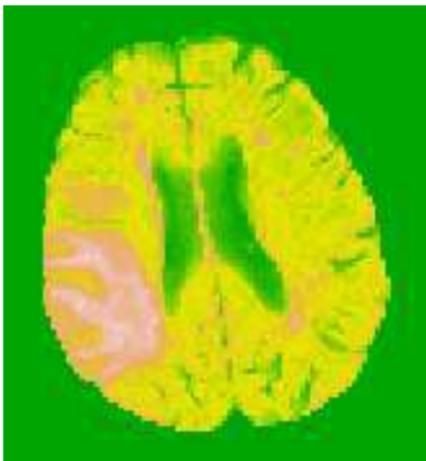
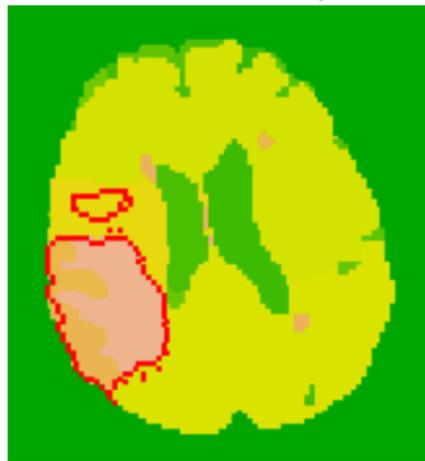


## STORATH et WEINMANN 2014

Résout le problème de Potts :

$$u^* = \operatorname{argmin}_u \gamma \|\nabla u\|_0 + \|u - f\|_2^2$$

T2 FLAIR

modèle de Potts 2D ( $\gamma = 0.75$ )

# Bibliographie I

OZENNE, B. et al. (2015). « Spatially regularized mixture model for lesion segmentation with application to stroke patients ». Dans : *Biostatistics (In Press)*.  
URL : <http://www.ncbi.nlm.nih.gov/pubmed/25745872>.

STORATH, M. et A. WEINMANN (2014). « Fast partitioning of vector-valued images ». Dans : *SIAM Journal on Imaging Sciences* 7.3, p. 1826–1852. URL :  
<http://bigwww.epfl.ch/preprints/storath1402p.pdf>.