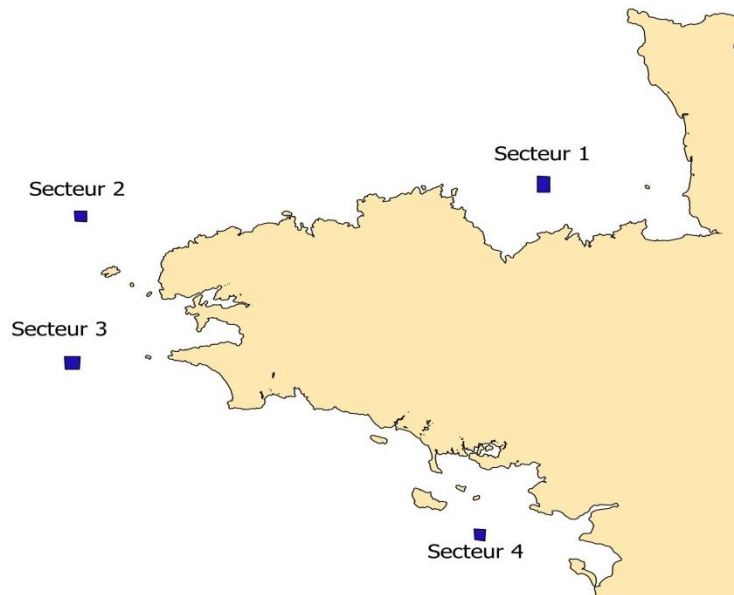


# Tutoriel d'extraction des données des fichiers satellite GeoTIFF

## Utilisation des logiciel QGIS et R



*Quatre secteurs ou zones d'extraction de la SST*



## Fiche documentaire

<b>Numéro d'identification du rapport :</b> RST.DYNECO n° 2018-01 <b>Diffusion :</b> libre <input checked="" type="checkbox"/> restreinte: <input type="checkbox"/> interdite : <input type="checkbox"/> <b>Validé par :</b> Adresse électronique : - chemin UNIX : - adresse WWW :	<b>date de publication</b> février 2018 <b>nombre de pages : 23</b> <b>bibliographie : oui</b> <b>illustration(s) : oui</b> <b>langue du rapport : français</b>	
<b>Titre et sous-titre du rapport :</b> <b>Tutoriel d'extraction des données des fichiers satellite GeoTIFF - Utilisation des logiciel QGIS et R</b>		
Contrat n° _____ Rapport intermédiaire <input type="checkbox"/> Rapport définitif <input type="checkbox"/> N° _____		
<b>Auteur(s) principal(aux) :</b> LAMPERT Luis	<b>Organisme / Direction / Service, laboratoire</b> IFREMER IFREMER/ODE/DYNECO-PELAGOS	
Collaborateur(s) :		
Cadre de la recherche : Programme : _____ Convention : _____ Projet : _____ Autres (préciser) : _____ Campagne océanographique : _____		
<b>Résumé :</b> Ce tutoriel aborde les méthodes d'extraction des données des images satellite au format GéoTIFF. Ils sont décrits les outils et méthodes pour la définition des zones d'extraction ainsi que les scripts en R pour leur extraction et mise en forme.		
<b>Abstract :</b> _____		
<b>Mots-clés :</b> Fichiers geotiff ; images satellite ; R ; QGIS		
<b>Rédacteur</b> Nom : Lampert Luis  Date : 1er février 2018  Visa	<b>Vérificateur</b> Anne-Laure Le Velly	<b>Approbateur</b> _____



# SOMMAIRE

---

<b>INTRODUCTION</b> .....	<b>1</b>
<b>1.- IMAGES DISPONIBLES POUR LES EXTRACTIONS</b> .....	<b>2</b>
1.1 SST .....	3
1.2 SSI .....	4
1.3 Vents .....	4
1.4 Couleur de l'eau .....	4
<b>2.- DEFINITION DU SECTEUR GEOGRAPHIQUE</b> .....	<b>7</b>
2.1 Avec QGIS seulement .....	7
2.2 Avec Google Earth.....	7
2.3 Au format .csv et .txt.....	8
<b>3.- UTILISATION DES SCRIPTS « R »</b> .....	<b>9</b>
3.1 Nettoyage et ouverture des packages.....	9
3.2 Création des zones d'extraction de valeurs.....	10
3.3 Préparation des données et fichiers .....	10
3.4 Boucle de calcul .....	11
3.5 Préparation des données et fichiers, partie 2.....	11
3.6 Boucle de calcul, partie 2.....	13
<b>ANNEXES</b> .....	<b>17</b>
Annexe 1 : script Sat_geotiff.r .....	20
Annexe 2 : script Sat_AELB_geotiff.r.....	14



# Tutoriel d'extraction des données des fichiers satellite GeoTIFF

---

## OBJECTIF

Ce tutoriel est créé afin de donner les procédures pour extraire des valeurs des images satellite stockées au format geotiff dans les disques de Dyneco. La méthode présentée met en application des outils libres de droits tels que QGIS et R. Il ne s'agit pas des cours de QGIS ou R, mais de leur application. L'utilisateur doit avoir une certaine pratique de ces deux outils pour les faire fonctionner.

## INTRODUCTION

L'intérêt des images satellite, avec une couverture spatiale étendue par rapport aux mesures stationnelles, ne se démontre plus. Selon les besoins en données environnementales, les images satellite peuvent nous renseigner sur l'évolution spatio-temporelle des paramètres tels que la chlorophylle, les MES, la houle, la température, l'irradiance solaire ou les vents. L'évolution des algorithmes sur les masses d'eau côtières a fortement évolué, et aujourd'hui ils nous permettent d'appréhender correctement leur évolution.

Ifremer a fortement contribué à la définition des algorithmes côtiers. OC5 est aujourd'hui le meilleur exemple concernant les caméras dites « couleur de l'eau », avec des paramètres chlorophylle, MES, turbidité, et ses paramètres dérivés. Un enregistrement quotidien est fait des images provenant des caméras telles que MODIS, VIIRS, PATHFINDER et traités en temps quasi-réel. Certains paramètres sont calculés avec OC5, d'autres proviennent d'autres plateformes, de l'Ifremer ou d'ailleurs.

On scindera ce tutoriel en trois parties :

1. une première, faisant l'état des lieux des images disponibles sur les disques de Dyneco,
2. une autre, pour définir les secteurs géographiques des extractions,
3. et finalement, les commandes sous R pour faire les extractions.

Des données brutes et traitées, issues des capteurs satellite, se trouvent stockées dans les différents disques Ifremer dont l'accès est libre. Certaines sont des produits incrémentés

quotidiennement et d'autres, sont traitées sur une période fixée par la nature de l'étude pour laquelle ont été extraites.

Voici, dans les lignes qui suivent, un état des lieux (non exhaustif) des données disponibles.

L'ensemble de ces données satellite se situent sur trois disques :

[\\tera10\amestz](#)

[\\iota1\guyasat2015](#)

[\\iota1\atcoast-public](#)

Pour plus de précisions, contacter Francis Gohin et/ou Luis Lampert.

## 1.- IMAGES DISPONIBLES POUR LES EXTRACTIONS

### 1.1 - Température de surface de la mer SST

Capteurs et plateformes concernant la température de surface de la mer (SST)

Données	Période	résolution	Origine	Commentaires
OSTIA	2006-2014	5km 1x2km si krigé	<a href="#">MetOffice</a> (GB)	Opérationnel. Intéressant pour la couverture et la bonne résolution mais série temporelle courte. Les données de 2006 sont disponibles à partir d'avril.
Pathfinder	1986-2012	4km	<a href="#">NOAA/NODC</a> AVHRR	Intéressant pour la série temporelle et la résolution mais couverture faible.
Odyssea	2010-2014	10km 1x2km si krigé	Cersat/Ifremer Multi capteur	Opérationnel. Intéressant pour la couverture et la bonne résolution mais série temporelle courte. Les données de 2010 sont disponibles à partir de septembre.
GRHSST	1986-2014	25km	Multi-capteurs	Opérationnel. Intéressant pour la couverture et la série temporelle, résolution faible mais suffisante pour cette première étude.

#### GRHSST

Les données se trouvent dans le disque [\\tera10\amestz\ghrsst](#)

La série s'étend de 1986 à 2013 avec des données journalières au format PNG et GéoTIF.

La couverture spatiale : Europe de l'ouest

#### OSTIA

Les données se trouvent sur le disque [\\tera10\amestz\ostia](#)

La série s'étend de 2006 à 2013 avec des données journalières au format PNG et GéoTIF krigés

La couverture spatiale : Europe de l'ouest



Des données concernant la Guyane se trouvent dans : [\\iota1\guyasat2015](#) portant sur les années 2006 à 2014.

### **ODYSSEA**

Les données se trouvent dans le disque [\\iota1\satcoast-public\atlantic\odyssea](#)

La série s'étend de 2010 à aujourd'hui avec des données journalières au format PNG et GéoTIF krigés.

La couverture spatiale : Europe de l'ouest.

Des données concernant la Guyane se trouvent dans : [\\iota1\guyasat2015](#) portant sur les années 2011 à 2014.

### **PATHFINDER**

Les données se trouvent sur le disque [\\iota1\satcoast-public\atlantic\pathfinder](#)

La série s'étend de 1986 à 2009 avec des données journalières au format PNG et GéoTIF krigées.

La couverture spatiale : Europe de l'ouest.

Une étude de synthèse sur les eaux du **Golfe de Gascogne** a été réalisée avec les données GRHSST 1986-2013. Elle se trouve dans [\\iota1\guyasat2015\GdG SST](#).

### **1.2.- Irradiance en surface de mer (SSI)**

Les données se trouvent dans le disque [\\iota1\satcoast-public\atlantic\SSI](#).

La série s'étend de 1983 à aujourd'hui avec des données journalières au format PNG et GéoTIF.

La couverture spatiale : Europe de l'ouest.

### **1.3.- Vents en surface de mer**

Les données se trouvent sur le disque [\\iota1\satcoast-public\atlantic\WIND](#).

La série s'étend de 1999 à aujourd'hui avec des données journalières au format PNG et GéoTIF sous forme de trois fichiers (direction U, V et vitesse).

La couverture spatiale : Europe de l'ouest.

### **1.4.- Données « couleur de l'eau » (Chla, MES, Turbidité, Kpar)**

Cette partie est la base du travail mené à l'Ifremer depuis des années par F.Gohin, qui a développé un algorithme « couleur de l'eau » pour les zones côtières (OC5).

Vous trouverez des images au niveau L3 pour la zone Ouest Europe et pour d'autres sites dans le monde en fonction des projets qui ont été réalisés.

Satellite	Période	résolution	Origine	Commentaires
SeaWiFS	1998-2010	1 km	<a href="#">OceanColor</a> NASA	Fin de mission le 10 décembre 2010
Viirs	2012-2014	1km	<a href="#">OceanColor</a> NASA	Toujours opérationnel à l'heure actuelle
Meris	2003-2011	1km/300m	<a href="#">ODESA</a> (ACRI)-ESA	Fin de mission le 7 avril 2012
Modis	2003-2014	1km	<a href="#">OceanColor</a> NASA	Toujours opérationnel à l'heure actuelle
OLCI	Depuis Fev 2016	300m	ESA	Non intégré encore à la chaîne Ifremer

Le tableau suivant décrit les produits L3 obtenus à l'Ifremer :

Paramètres	Unité	Algorithmes
Chlorophylle-a	µg/m <sup>3</sup>	OC5 IFREMER
Matières en Suspension non Algales (MES)	mg/m <sup>3</sup>	OC5 IFREMER
Turbidité	NTU	OC5 IFREMER
Kpar	m-1	OC5 IFREMER

La **chlorophylle\_a (CHLA)** a été calculée avec l'algorithme OC5. Il modifie l'algorithme de la NASA OC4 (Ocean Color 4 bands Algorithme) en corrigeant les effets des MES par les canaux 412 nm et 555 nm. Ce paramètre est exprimé en mg chla/m<sup>3</sup> ou en µg chla/L.

Les **matières en suspension (MES)** représentent la partie inorganique des particules en suspension dans l'eau de mer. Elles sont calculées à partir des réflectances dans les canaux 550 nm et 670 nm (Gohin et al. 2005). L'unité utilisée pour ce paramètre est mg/L.

La **turbidité (TURBI)** est calculée à partir des estimations de la chlorophylle (CHLA) et des matières en suspension (MES) par la formule :

$$\text{TURBI (NTU)} = 0.54 * (\text{MES}_{\text{minérale}} + 0.234 * \text{chl}^{0.57})$$

Bien que l'unité de turbidité la plus utilisée aujourd'hui soit FNU (Formazine Nephelometric Unit), nous avons exprimé la turbidité en NTU (Nephelometric Turbidity Unit) par souci d'homogénéité avec les données déjà acquises et des appareils de terrain encore utilisés de nos jours. Une relation linéaire simple peut être utilisée pour passer d'une unité à une autre : NTU = FNU/1.3. Cette relation est valable dans les eaux atlantiques de métropole.

#### 1.4.1 Produits avec une résolution de 1 km

**1.4.1.1 Ouest-Europe** : Tous ces produits se trouvent dans :

[\\iota1\satcoast-public\atlantic\meris](#) (2003-2012)

[\\iota1\satcoast-public\atlantic\modis](#) (2003-aujourd'hui)

[\\iota1\satcoast-public\atlantic\seawifs](#) (1998-2010)

[\\iota1\satcoast-public\atlantic\viirs](#) (2012- aujourd'hui)

Ils sont présents par produit aux formats PNG et GéoTIF. Parfois également au format NetCDF (NC).

#### **1.4.1.2 Données krigés journalières**

Des données krigés journalières sont enregistrées pour la Chla et les MES non-algales depuis 1998 et incrémentées quotidiennement.

Pour la Chla [\\iota1\satcoast-public\atlantic\analysis](#), où se trouvent les données journalières, hebdomadaires, les climatologies et les cartes d'erreurs.

Pour la MES [\\iota1\satcoast-public\atlantic\analysis\\_spim](#), où se trouvent les données journalières, hebdomadaires, les climatologies et les cartes d'erreurs.

#### **1.4.2 Produits avec une résolution de 300 m**

**Ouest-Europe** : Tous ces produits se trouvent dans :

[\\iota1\satcoast-public\roses\MERIS\\_300](#) (2003-2012)

Ils sont présents par produit aux formats PNG et GéoTIF.

#### **1.4.3 En dehors de la zone Ouest-Europe**

Pour la **Guyane Française**, toutes les données se trouvent dans [\\iota1\guyasat2015](#)

Pour d'autres régions du globe, il faut chercher dans les disques :

[\\tera10\amestez](#)

[\\iota1\satcoast-public](#)

## 2.- DEFINITION DU SECTEUR GEOGRAPHIQUE

Bien souvent nous aurons des fichiers des zones d'intérêt déjà disponibles sur internet, ou chez les partenaires associés, si l'on travaille avec les masses d'eau DCE, DCSMM ou Natura 2000. Mais parfois il faut définir soi-même la zone sur laquelle l'extraction sera faite. Nous verrons dans ces quelques lignes comment le faire avec QGIS directement ou en prenant des points géographiques sur GoogleEARTH.

Vous trouverez dans le répertoire de ce tutoriel des cours en PDF concernant QGIS où les principales fonctions sont expliquées.

### 2.1 – Avec QGIS uniquement

Les fonctionnalités de QGIS sont suffisantes pour réaliser les masques, puis les exporter pour que les scripts R le prennent en compte. La façon la plus simple est de travailler avec des formats « shape » (.shp).

- Ouvrir un projet QGIS existant, ou en créer un,
- Ouvrir « Couche/créer une couche/nouvelle couche shapefile », ou CTR+MAJ+N,
- Dans la boîte de dialogues qui s'ouvre, cocher « polygone »,
- Enregistrer la couche dans le répertoire de travail,
- Cliquer « éditer/ajouter une entité », puis dessiner les secteurs et leur attribuer des numéros (1, 2, 3...),
- Une fois fini, se mettre sur la couche créée et enregistrer la couche vectorielle sous... Garder « ESRI shapefile », choisir où l'enregistrer et lui donner un nom. C'est cette dernière qui servira pour importer sous R.

### 2.2 – Avec Google Earth et QGIS

- Ouvrir Google Earth et dessiner les secteurs (Ajouter/polygones) en lui donnant un nom (1, 2, 3...),
- Enregistrer le lieu sous/ au format .kml
- Sous QGIS faire couche/ajouter une couche/ajouter une couche vecteur/parcourir et cliquer sur le fichier .klm créé. Il s'ouvre dans QGIS,
- Se placer sur cette couche et faire projet/export DXF,
- Sur QGIS faire couche/ajouter une couche vecteur et ouvrir le fichier .dxf, il s'affiche avec le nom de « entities »,
- Il ne reste qu'à exporter cette couche au format shapefile comme dans le cas précédent.

Cette procédure est à utiliser au cas où vous recevrez des polygones de format .klm, sinon le faire avec QGIS.

### 2.3 – Au format .csv ou .txt avec QGIS

Il peut se présenter le cas d'obtenir les coordonnées des polygones au format .csv ou .txt. Dans ce cas, il est possible de créer une couche avec :

- couche/ajouter une couche/ajouter une couche de texte délimité,
- Ensuite il faut remplir la boîte de dialogues pour indiquer comment sont formatées les données,
- Une fois la couche chargée, suivre la même procédure que précédemment.

### 3.- UTILISATION DES SCRIPTS R

Le script qui est détaillé ci-dessous est réalisé pour utiliser des images au format GeoTIFF. Pour ceci, les packages « shapefile » et « raster » doivent être chargés. En annexes vous trouverez ce script (Sat\_geotiff.r) qui présente l'ensemble de procédures propres aux extractions des données des images GeoTIFF, puis un autre script, qui a été utilisé pour les extractions des données satellite sur les masses d'eau côtières DCE de Bretagne (Sat\_AELB\_geotiff.r).

#### EXEMPLE D'EXTRACTION DES DONNEES

On procédera par un exemple concret. Nous allons extraire les températures de surface de la mer (SST) de quatre secteurs en Atlantique, au large de la Bretagne (figure 3.1).

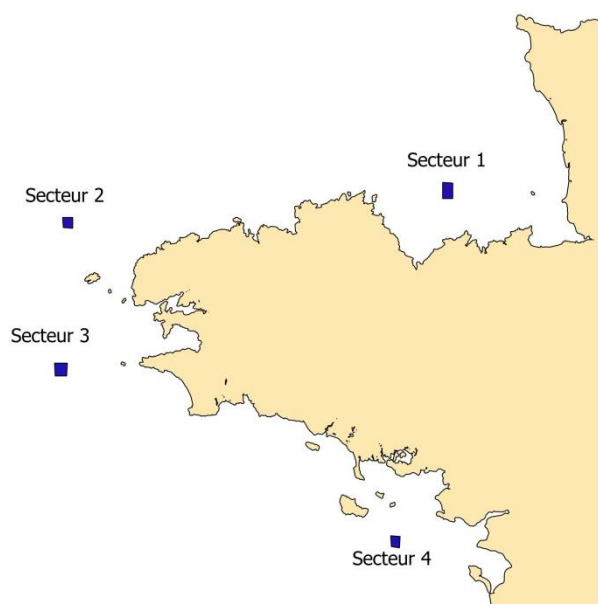


Figure 3.1 : quatre secteurs ou zones d'extraction de la SST

Pour l'exemple nous avons pris uniquement les SST des mois de juin, juillet et août 2007 produites par la plateforme Odyssea. Cet exemple se trouve dans le répertoire « TutoSAT » qui a la structure suivante (figure 3.2) :

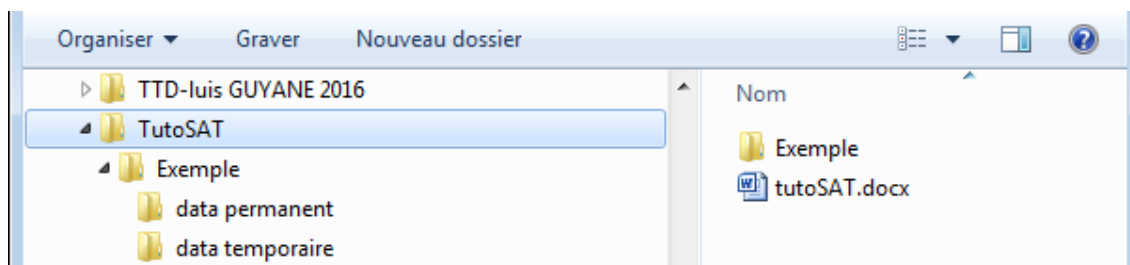


Figure 3.2 : Distribution des répertoires de ce tutoriel

Ce tutoriel se trouve sous la racine. Puis dans le répertoire « Exemple », deux nouveaux répertoires, « data permanent » et « data temporaire ». Dans data permanent se situent les fichiers shapefile « .shp » nécessaires au calcul. Dans le fichier temporaire sont placées les

images TIFF préalablement dézipées (avec IZArc par exemple) et le script Sat\_geotiff.r (voir figure 3.3).

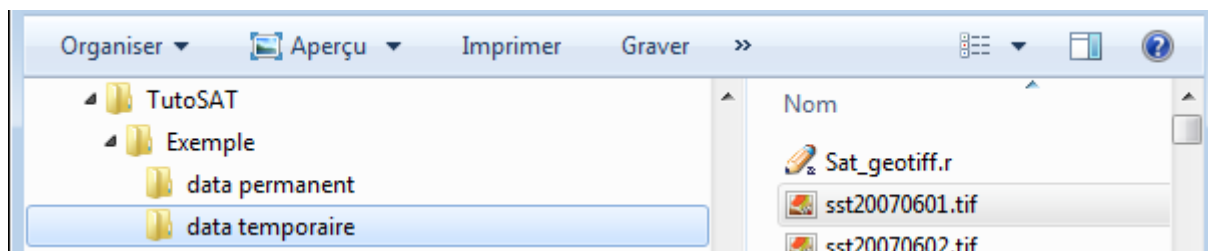


Figure 3.3 : Vue partielle du répertoire « data temporaire »

Mais passons à la description de chaque partie du script Sat\_geotiff.r :

### 3.1.- NETTOYAGE ET OUVERTURE DES PACKAGES

```

#####
#   EXTRACTION DONNEE DES IMAGES SATELLITE (.tif)
#
# Luis Lampert : 22/01/2018
# Modifications :
#####

# Objectif : Extraction des donnees des images satellite, puis creation des
# fichiers de resultats (moyennes, P90, max, min, etc...).
#-----

#####
#           P A R T I E 1
#   Fichier d'extraction brut
#####

#-----
# NETTOYAGE ET OUVERTURE DES PACKAGES
#-----
rm(list=ls()) # Effacement de toutes les données en mémoire
graphics.off() # Effacement de tous les graphiques en mémoire

# Ouverture des librerries necessaires
library(shapefiles)
library(raster)
    
```

Dans cette première phase il faut nettoyer l'écran et la mémoire pour ne pas trainer des objets qui pourraient interférer dans le calcul. Les packages spécifiques pour le traitement des images « raster » et les secteurs géoréférencés « shapefile » sont chargés en premier. Vérifier dans la fenêtre « R » qu'ils soient bien chargés.

### 3.2.- CREATIONS ZONES D'EXTRACTION DES VALEURS

```

#-----
# CREATIONS ZONES D'EXTRACTION DES VALEURS
#-----
# Nous aurons dans cet exemple quatre zones selon un decoupage fait avec
# QGIS.
shap=read.shapefile("X:\\TutoSAT\\Exemple\\data permanent\\exo")
#identification des 4 zones
    
```

```
z1=shap$shp$shp[[1]]$points
z2=shap$shp$shp[[2]]$points
z3=shap$shp$shp[[3]]$points
z4=shap$shp$shp[[4]]$points
```

A présent il faut charger le fichier shapefile créé précédemment avec QGIS, où se trouvent les secteurs pour l'extraction des données. Dans notre exemple il s'agit de 4 secteurs en Bretagne. Pour voir un résumé de l'objet « shap », tapez : **> str(shap)**, qui vous donnera la structure de la liste.

Dans les objets **z1** à **z4** nous stockons les quatre secteurs. Pour vérifier s'ils sont les bons secteurs, il faut regarder leur géométrie par la commande **>plot(z1, type='l')**.

### 3.3.- PREPARATION DES DONNEES ET FICHIERS

```
#-----
# PREPARATION DES DONNEES ET FICHIERS
#-----
# Creation du fichier de resultats.

# ***** [ATTENTION changer le nom du fichier de sortie chq fois] *****
con=file("SSTmoyenne_08.txt", open="w") #creation fichier resultats
cat("nomfich", "zone", "n.pix", "n.pix.ok", "NAs", "pour.pix.vid", "moy",
    "min", "max", file=con, "\n") # creation des labels des colonnes

# Identification des fichiers a traiter [ATTENTION au pattern a modifier]
fich.tif=list.files(path=".", pattern="sst200708")
length(fich.tif)
```

Il faut à présent créer un fichier de résultats vide, avec les noms de colonne qui correspondent au calcul. Puis créer la liste de fichiers que nous allons traiter dans ce batch. Dans notre cas, nous allons calculer les valeurs des mois de juin, juillet et août séparément. En réalité ce serait plus facile à faire ensemble, mais c'est juste pour le cas de l'exemple, car en général on peut lancer un run sur deux années complètes, mais ça prend quelques heures de calcul. Donc, ce qui est fait ici pour les mois, est transposable pour les années (voir script Sat\_AELB\_geotiff.r).

Il faut donc définir sous R, quelle est la façon de reconnaître les bons fichiers. Dans notre cas nous utilisons «pattern="sst200708" », il chargera tous les fichiers qui ont cette chaîne de caractères. Donc, seulement ceux du mois d'août.

ATTENTION : il faut faire très attention de changer dans cette partie le nom du fichier de sortie ("SSTmoyenne\_08.txt") et celui du pattern pour que ça coïncide (pattern="sst200708"). Donc, il faudra créer successivement les fichiers "SSTmoyenne\_08.txt", "SSTmoyenne\_07.txt" et "SSTmoyenne\_06.txt".

**Length(fich.tif)** nous donne la quantité de fichiers qui seront traités dans ce run.

Vérifiez que le fichier texte des résultats est bien enregistré dans le répertoire temporaire.

### 3.4.- BOUCLE DE CALCUL

```
#-----
# BOUCLE DE CALCUL
#-----
# Boucle fichier
for (j in 1:length(fich.tif)) {

# Charger fichier GEOTIFF
exo=raster(fich.tif[j])
```



```

nomfich=exo@data@names

#polygones des zones
polys= SpatialPolygons(list(Polygons(list(Polygon(z1)), 1),
                             Polygons(list(Polygon(z2)), 2),
                             Polygons(list(Polygon(z3)), 3),
                             Polygons(list(Polygon(z4)), 4)))

val=extract(exo, polys) # extraction des valeurs
for (i in 1:length(shap$shp$shp)) { val[[i]][val[[i]]<=0]=NA }
#BOUCLE ZONE
moy=unlist(lapply(val, function(x) if (!is.null(x)) mean(x, na.rm=TRUE) else
NA ))
NAs=unlist(lapply(val, function(x) length(which(is.na(x)==T)) ))
rang=unlist(lapply(val, function(x) if (!is.null(x)) range(x, na.rm=TRUE)
else NA ))
n.pix=unlist(lapply(val, length))
n.pix.ok=n.pix-NAs
pour.pix.vid=round((n.pix-n.pix.ok)/n.pix*100,0)

# Ecriture dans le fichier de sortie
cat(nomfich, "NORD", n.pix[1], n.pix.ok[1],NAs[1],pour.pix.vid[1],
moy[1],rang[1],rang[2],file=con,"\n")
cat(nomfich, "NORD-OUEST", n.pix[2],
n.pix.ok[2],NAs[2],pour.pix.vid[2],moy[2],rang[3],rang[4], file=con,"\n")
cat(nomfich, "SUD-OUEST", n.pix[3],
n.pix.ok[3],NAs[3],pour.pix.vid[3],moy[3], rang[5],rang[6],file=con,"\n")
cat(nomfich, "SUD", n.pix[4], n.pix.ok[4],NAs[4],pour.pix.vid[4],moy[4],
rang[7],rang[8],file=con,"\n")
} # fin boucle fichier
close(con) # fermeture fichier resultats

```

Nous arrivons au cœur du calcul, l'extraction et la création du fichier des résultats. La boucle « for » prendra chaque fichier et fera l'extraction des données. Il crée l'objet « exo » et les polygones des zones, puis stocke dans l'objet « val » toutes les valeurs de l'image raster [j] sur chaque polygone. Faites **str(val)**, puis **length(val[[1]])** pour voir la structure de cette liste et la quantité de valeurs dans la zone z1 par exemple.

Comme il se peut qu'il y ait des valeurs négatives ou égales à zéro qui ne nous intéressent pas, une deuxième boucle « for » est lancée pour convertir toutes ces valeurs en « NA ».

Pour chaque zone, sont calculés les moyennes (moy), la quantité des NA (NAs) qui représentent les pixels non valides ou inutilisables, le rang des valeurs (min et max), le nombre de pixels totaux du secteur (n.pix), le nombre de pixels utilisables (n.pix.ok), et le pourcentage de pixels inutilisables (pour.pix.vid). Il pourrait s'agir de toute autre valeur statistique également, en fonction de nos besoins.

Par la suite, le script procède à l'écriture de valeurs calculées dans le fichier de résultats.

La boucle est finie avec la fermeture de la connexion de la console.

En ouvrant le fichier de résultats **SSTmoyenne\_06.txt** nous observons la structure de données suivante :

```

SSTmoyenne_06.txt - Bloc-notes
Fichier Edition Format Affichage ?
nomfich zone n.pix n.pix.ok NAS pour pix.vid moy min max
sst20070601 NORD 38 38 0 0 13.61421 13.55 13.64
sst20070601 NORD-OUEST 24 24 0 0 12.81 12.8 12.82
sst20070601 SUD-OUEST 36 36 0 0 13.25722 13.23 13.32
sst20070601 SUD 28 28 0 0 14.07179 14.05 14.09
sst20070602 NORD 38 38 0 0 13.92816 13.87 13.97
sst20070602 NORD-OUEST 24 24 0 0 12.95583 12.95 12.97
sst20070602 SUD-OUEST 36 36 0 0 13.4825 13.45 13.54
sst20070602 SUD 28 28 0 0 14.81929 14.7 14.94
sst20070605 NORD 38 38 0 0 14.02158 13.97 14.05
sst20070605 NORD-OUEST 24 24 0 0 13.13333 13.12 13.14
sst20070605 SUD-OUEST 36 36 0 0 13.77528 13.74 13.83
sst20070605 SUD 28 28 0 0 15.54571 15.44 15.63
sst20070606 NORD 38 38 0 0 14.14421 14.09 14.18
sst20070606 NORD-OUEST 24 24 0 0 13.27 13.26 13.28
sst20070606 SUD-OUEST 36 36 0 0 14.095 14.05 14.16
sst20070606 SUD 28 28 0 0 16.45107 16.31 16.54
sst20070607 NORD 38 38 0 0 14.14 14.09 14.18
sst20070607 NORD-OUEST 24 24 0 0 13.265 13.25 13.29
    
```

Nous avons, pour chaque jour, quatre lignes avec les résultats demandés pour chaque zone :

- NORD = z1,
- NORD-OUEST = z2,
- SUD-OUEST = z3, et
- SUD = z4

### 3.5.- PREPARATION DES DONNEES ET FICHIERS PARTIE 2

A ce stade, la partie 1 qui produit les fichiers de base est finie. Nous passerons à la partie 2 de ce script, où est décrite la procédure pour faire la synthèse. Dans cet exemple nous voudrions obtenir les moyennes par secteur et par mois.

```

#*****
#                               P A R T I E  2
#   Compilation fichiers d'extraction bruts
#*****

# SYNTHESE A PARTIR DES FICHIERS INDIVIDUELS PAR ANNEE/MOIS/SAISON...

# Une fois les fichiers SSTmoyenne 19xx-19xx.txt crees, calcul des moyennes
par secteur et mise dans un fichier "SST_synt_19xx-19yy.txt"

#-----
# PREPARATION DES DONNEES ET FICHIERS
#-----
con=file("SSTsynt_06-08.txt", open="w") #ouverture fichiers extraction
fich.txt=list.files(path=".", pattern="SSTmoy") # [ ATTENTION au pattern]
length(fich.txt)
    
```

Il est donc nécessaire de prendre les fichiers "SSTmoyenne\_08.txt", "SSTmoyenne\_07.txt" et "SSTmoyenne\_06.txt", et calculer les moyennes de chaque mois par secteur. Puis remettre ces résultats dans un fichier de synthèse que nous nommerons "SSTsynt\_06-08.txt".

Il faut suivre la même procédure que pour la première partie pour créer le fichier de résultats et réaliser la liste de fichiers à utiliser avec le bon *pattern*. Vérifiez toujours dans le répertoire temporaire qu'il n'y ait pas des fichiers avec des noms qui pourraient présenter la même chaîne de caractères et perturber le calcul.

## 3.6.- BOUCLE DE CALCUL PARTIE 2

```

#-----
# BOUCLE DE CALCUL
#-----
# boucle fichier
for (j in 1:length(fich.txt)) {

# Charger fichier fichier extraction .txt
data0=read.table(fich.txt[j], header=F, skip=1)

N=apply(subset(data0, V2=="NORD", V7),2,mean)
NO=apply(subset(data0, V2=="NORD-OUEST", V7),2,mean)
SO=apply(subset(data0, V2=="SUD-OUEST", V7),2,mean)
S=apply(subset(data0, V2=="SUD", V7),2,mean)

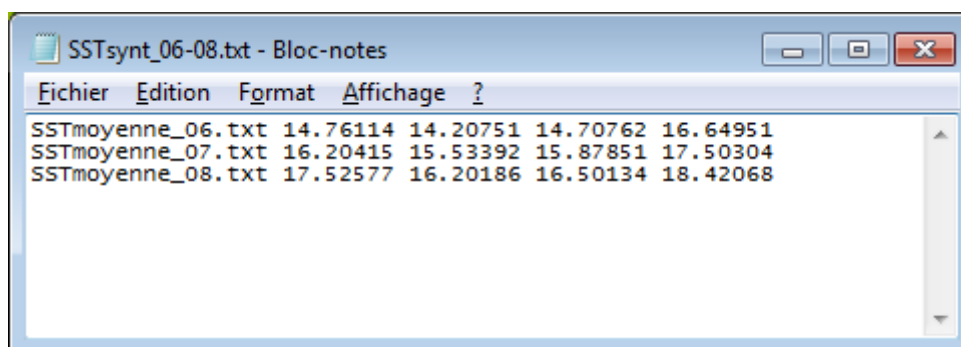
cat(fich.txt[j], N[1], NO[1],SO[1],S[1],file=con,"\n")

} # fin boucle fichier
close(con) # fermeture fichier resultats

#----- F I N P A R T I E 2 -----

```

De la même façon que précédemment, nous calculons les moyennes par zone et par fichier et nous reportons ceci dans le fichier "SSTsynt\_06-08.txt".



Fichier	14.76114	14.20751	14.70762	16.64951
SSTmoyenne_06.txt	14.76114	14.20751	14.70762	16.64951
SSTmoyenne_07.txt	16.20415	15.53392	15.87851	17.50304
SSTmoyenne_08.txt	17.52577	16.20186	16.50134	18.42068

Par la suite il est possible de récupérer ce fichier avec Excel ou tout autre logiciel de traitement ou de visualisation (R inclus). Il est possible de créer des colonnes avec les mois, les années ou les saisons si nécessaire, puis obtenir des statistiques sur ces périodes. Afin de changer de format (mode colonne à mode tableau) la commande R « reshape » peut s'avérer très utile.

Les calculs peuvent être très longs en fonction des images et de la taille des secteurs. S'il y a un bug ou un plantage du run, il arrêtera tout. Pour cette raison, il vaut mieux ne pas tout calculer d'un seul coup si il dépasse 3 heures de temps (en gros 2 années avec 45 zones dans la région Bretagne), mais cela dépend de la quantité et de la taille des zones. Faites vos essais.

Il est possible d'accélérer les calculs en lançant plusieurs fenêtres « R » au même temps. Mais ça dépendra de la quantité de cœurs de votre ordinateur. S'il possède 4 cœurs il est toujours possible de lancer deux fenêtres R et continuer à travailler sur une autre chose qui ne prenne

pas trop de ressources, sinon, il est préférable d'utiliser un ordinateur dédié pour cette tâche. L'utilisation des packages de « parallélisation » ne donne pas des meilleurs résultats du fait du taux de transfert entre les fichiers.

Dans le répertoire « Documentations » se trouvent les tutoriels de QGIS et R qui pourraient vous être utiles.

# ANNEXES

ANNEXE 1 : script Sat\_geotiff.r

ANNEXE 2 : script Sat\_AELB\_geotiff.r

## ANNEXE 1 : script Sat\_geotiff.r

Vous pouvez copier directement ce script sur R par coller-copier ou sur une feuille en mode texte et sur Tinn-R. Tout éditeur de texte suffira. Même à partir des fichiers PDF, il est possible de copier-coller les commandes sur la fenêtre R.

```

#*****
#   EXTRACTION DONNEE DES IMAGES SATELLITE (.tiff)
#
# Luis Lampert : 22/01/2018
# Modifications :
#*****

# Objectif : Extraction des donnees des images satellite, puis creation des
# fichiers de resultats (moyennes, P90, max, min, etc...).
#-----

#*****
#           P A R T I E  1
#   Fichier d'extraction brut
#*****
#-----
# NETTOYAGE ET OUVERTURE DES PACKAGES
#-----
rm(list=ls()) # Effacement de toutes les données en mémoire
graphics.off() # Effacement de tous les graphiques en mémoire

# Ouverture des libraeries necessaires
library(shapefiles)
library(raster)

#-----
# CREATIONS ZONES D'EXTRACTION DES VALEURS
#-----
# Nous aurons dans cet exemple quatre zones selon un decoupage fait avec
# QGIS.
shap=read.shapefile("X:\\TutoSAT\\Exemple\\data permanent\\exo")
#identification des 4 zones
z1=shap$shp$shp[[1]]$points
z2=shap$shp$shp[[2]]$points
z3=shap$shp$shp[[3]]$points
z4=shap$shp$shp[[4]]$points

#-----
# PREPARATION DES DONNEES ET FICHIERS
#-----
# Creation du fichier de resultats.

# ***** [ATTENTION changer le nom du fichier de sortie chq fois] *****
con=file("SSTmoyenne_08.txt", open="w") #creation fichier resultats
cat("nomfich", "zone", "n.pix", "n.pix.ok", "NAs", "pour.pix.vid", "moy",
"min", "max", file=con, "\n") # creation des labels des colonnes

# Identification des fichiers a traiter [ATTENTION au pattern a modifier]
fich.tif=list.files(path=".", pattern="sst200708")
length(fich.tif)

```

```

#-----
# BOUCLE DE CALCUL
#-----
# Boucle fichier
  for (j in 1:length(fich.tif)) {

# Charger fichier GEOTIFF
exo=raster(fich.tif[j])
nomfich=exo@data@names

#polygones des zones
polys= SpatialPolygons(list(Polygons(list(Polygon(z1)), 1),
                             Polygons(list(Polygon(z2)), 2),
                             Polygons(list(Polygon(z3)), 3),
                             Polygons(list(Polygon(z4)), 4)))

val=extract(exo, polys) # extraction des valeurs
for (i in 1:length(shap$shp$shp)) { val[[i]][val[[i]]<=0]=NA }
#BOUCLE ZONE
moy=unlist(lapply(val, function(x) if (!is.null(x)) mean(x, na.rm=TRUE)
else NA ))
NAs=unlist(lapply(val, function(x) length(which(is.na(x)==T)) ))
rang=unlist(lapply(val, function(x) if (!is.null(x)) range(x, na.rm=TRUE)
else NA ))
n.pix=unlist(lapply(val,length))
n.pix.ok=n.pix-NAs
pour.pix.vid=round((n.pix-n.pix.ok)/n.pix*100,0)

# Ecriture dans le fichier de sortie
cat(nomfich, "NORD", n.pix[1], n.pix.ok[1],NAs[1],pour.pix.vid[1],
moy[1],rang[1],rang[2],file=con,"\n")
cat(nomfich, "NORD-OUEST", n.pix[2],
n.pix.ok[2],NAs[2],pour.pix.vid[2],moy[2],rang[3],rang[4], file=con,"\n")
cat(nomfich, "SUD-OUEST", n.pix[3],
n.pix.ok[3],NAs[3],pour.pix.vid[3],moy[3], rang[5],rang[6],file=con,"\n")
cat(nomfich, "SUD", n.pix[4], n.pix.ok[4],NAs[4],pour.pix.vid[4],moy[4],
rang[7],rang[8],file=con,"\n")
      } # fin boucle fichier
close(con) # fermeture fichier resultats

#----- F I N           P A R T I E 1 -----

#*****
#           P A R T I E 2
#   Compilation fichiers d'extraction bruts
#*****

# SYNTHESE A PARTIR DES FICHIERS INDIVIDUELS PAR ANNEE/MOIS/SAISON...

# Une fois les fichiers SSTmoyenne 19xx-19xx.txt crees, calcul des
moyennes par secteur et mise dans un fichier "SST_synt_19xx-19yy.txt"

#-----
# PREPARATION DES DONNEES ET FICHIERS
#-----
con=file("SSTsynt_06-08.txt", open="w") #ouverture fichiers extraction
fich.txt=list.files(path=".", pattern="SSTmoy") # [ ATTENTION au pattern]
length(fich.txt)

```

```
#-----  
# BOUCLE DE CALCUL  
#-----  
# boucle fichier  
  for (j in 1:length(fich.txt)) {  
  
# Charger fichier fichier extraction .txt  
data0=read.table(fich.txt[j], header=F, skip=1)  
  
N=apply(subset(data0, V2=="NORD", V7),2,mean)  
NO=apply(subset(data0, V2=="NORD-OUEST", V7),2,mean)  
SO=apply(subset(data0, V2=="SUD-OUEST", V7),2,mean)  
S=apply(subset(data0, V2=="SUD", V7),2,mean)  
  
cat(fich.txt[j], N[1], NO[1],SO[1],S[1],file=con,"\n")  
  
} # fin boucle fichier  
  close(con) # fermeture fichier resultats  
  
#----- F I N P A R T I E 2 -----
```



## ANNEXE 2 : script Sat\_AELB\_geotiff.r

A titre d'autre exemple, ce script a été utilisé pour générer les fichiers des moyennes des paramètres satellite pour le projet DYNAFLIT, où plusieurs années sont considérées, ainsi que 45 masses d'eau côtières.

Vous pouvez copier directement ce script sur R par coller-copier ou sur une feuille en mode texte et sur Tinn-R. Tout éditeur de texte suffira. Même à partir des fichiers PDF, il est possible de copier-coller les commandes sur la fenêtre R.

```

#*****
#   LECTURE IMAGES SATELLITE (.tiff)
#
# Luis Lampert : 05/02/2016
# Modifications :
#*****

# Objectif : Lire les images de SST du Golfe de Gascogne (GdG) des
# produits divers.

rm(list=ls()) # Effacement de toutes les données en mémoire
graphics.off() # Effacement de tous les graphiques en mémoire

# Ouverture des librairies
library(shapefiles)
library(raster)

#-----
# PART 1 : CREATION DES FICHIERS ANNUELS (JOUR PAR JOUR)
#-----
# CREATIONS ZONES MEC
shap=read.shapefile("X:\\Touria AELB\\MEC+MET AELB")
z1=shap$shp$shp[[1]]$points
z2=shap$shp$shp[[2]]$points
z3=shap$shp$shp[[3]]$points
z4=shap$shp$shp[[4]]$points
z5=shap$shp$shp[[5]]$points
z6=shap$shp$shp[[6]]$points
z7=shap$shp$shp[[7]]$points
z8=shap$shp$shp[[8]]$points
z9=shap$shp$shp[[9]]$points
z10=shap$shp$shp[[10]]$points
z11=shap$shp$shp[[11]]$points
z12=shap$shp$shp[[12]]$points
z13=shap$shp$shp[[13]]$points
z14=shap$shp$shp[[14]]$points
z15=shap$shp$shp[[15]]$points
z16=shap$shp$shp[[16]]$points
z17=shap$shp$shp[[17]]$points
z18=shap$shp$shp[[18]]$points
z19=shap$shp$shp[[19]]$points
z20=shap$shp$shp[[20]]$points
z21=shap$shp$shp[[21]]$points
z22=shap$shp$shp[[22]]$points
z23=shap$shp$shp[[23]]$points
z24=shap$shp$shp[[24]]$points

```

```
z25=shap$shp$shp[[25]]$points
z26=shap$shp$shp[[26]]$points
z27=shap$shp$shp[[27]]$points
z28=shap$shp$shp[[28]]$points
z29=shap$shp$shp[[29]]$points
z30=shap$shp$shp[[30]]$points
z31=shap$shp$shp[[31]]$points
z32=shap$shp$shp[[32]]$points
z33=shap$shp$shp[[33]]$points
z34=shap$shp$shp[[34]]$points
z35=shap$shp$shp[[35]]$points
z36=shap$shp$shp[[36]]$points
z37=shap$shp$shp[[37]]$points
z38=shap$shp$shp[[38]]$points
z39=shap$shp$shp[[39]]$points
z40=shap$shp$shp[[40]]$points
z41=shap$shp$shp[[41]]$points
z42=shap$shp$shp[[42]]$points
z43=shap$shp$shp[[43]]$points
z44=shap$shp$shp[[44]]$points
z45=shap$shp$shp[[45]]$points

# FICHIER DE SORTIE
# ATTENTION changer le nom du fichier de sortie
con=file("HSMAX2016.txt", open="w") #ouverture fichier resultats
fich.tif=list.files(path=".", pattern="hsmax")

# BOUCLE CREATION FICHIER
T1=Sys.time()
for (j in 1:length(fich.tif)) {
# Charger fichier GEOTIFF
exo=raster(fich.tif[j])
nomfich=exo@data@names

#polygones des zones
polys= SpatialPolygons(list(Polygons(list(Polygon(z1)), 1),
Polygons(list(Polygon(z2)), 2),
Polygons(list(Polygon(z3)), 3),
Polygons(list(Polygon(z4)), 4),
Polygons(list(Polygon(z5)), 5),
Polygons(list(Polygon(z6)), 6),
Polygons(list(Polygon(z7)), 7),
Polygons(list(Polygon(z8)), 8),
Polygons(list(Polygon(z9)), 9),
Polygons(list(Polygon(z10)), 10),
Polygons(list(Polygon(z11)), 11),
Polygons(list(Polygon(z12)), 12),
Polygons(list(Polygon(z13)), 13),
Polygons(list(Polygon(z14)), 14),
Polygons(list(Polygon(z15)), 15),
Polygons(list(Polygon(z16)), 16),
Polygons(list(Polygon(z17)), 17),
Polygons(list(Polygon(z18)), 18),
Polygons(list(Polygon(z19)), 19),
Polygons(list(Polygon(z20)), 20),
Polygons(list(Polygon(z21)), 21),
Polygons(list(Polygon(z22)), 22),
Polygons(list(Polygon(z23)), 23),
Polygons(list(Polygon(z24)), 24),
Polygons(list(Polygon(z25)), 25),
Polygons(list(Polygon(z26)), 26),
```

```

Polygons(list(Polygon(z27)),27),
Polygons(list(Polygon(z28)),28),
Polygons(list(Polygon(z29)),29),
Polygons(list(Polygon(z30)),30),
Polygons(list(Polygon(z31)),31),
Polygons(list(Polygon(z32)),32),
Polygons(list(Polygon(z33)),33),
Polygons(list(Polygon(z34)),34),
Polygons(list(Polygon(z35)),35),
Polygons(list(Polygon(z36)),36),
Polygons(list(Polygon(z37)),37),
Polygons(list(Polygon(z38)),38),
Polygons(list(Polygon(z39)),39),
Polygons(list(Polygon(z40)),40),
Polygons(list(Polygon(z41)),41),
Polygons(list(Polygon(z42)),42),
Polygons(list(Polygon(z43)),43),
Polygons(list(Polygon(z44)),44),
Polygons(list(Polygon(z45)),45))

val=extract(exo, polys)
for (i in 1:45) { val[[i]][val[[i]]<=0]=NA }

#BOUCLE INTERNE ZONE
moy=unlist(lapply(val, function(x) if (!is.null(x)) mean(x, na.rm=TRUE)
else NA ) )
NAs=unlist(lapply(val, function(x) length(which(is.na(x)==T)) ))
n.pix=unlist(lapply(val,length))
n.pix.ok=n.pix-NAs

cat(nomfich,"z1",n.pix[1],n.pix.ok[1],NAs[1],moy[1],file=con,"\n")
cat(nomfich,"z2",n.pix[2],n.pix.ok[2],NAs[2],moy[2],file=con,"\n")
cat(nomfich,"z3",n.pix[3],n.pix.ok[3],NAs[3],moy[3],file=con,"\n")
cat(nomfich,"z4",n.pix[4],n.pix.ok[4],NAs[4],moy[4],file=con,"\n")
cat(nomfich,"z5",n.pix[5],n.pix.ok[5],NAs[5],moy[5],file=con,"\n")
cat(nomfich,"z6",n.pix[6],n.pix.ok[6],NAs[6],moy[6],file=con,"\n")
cat(nomfich,"z7",n.pix[7],n.pix.ok[7],NAs[7],moy[7],file=con,"\n")
cat(nomfich,"z8",n.pix[8],n.pix.ok[8],NAs[8],moy[8],file=con,"\n")
cat(nomfich,"z9",n.pix[9],n.pix.ok[9],NAs[9],moy[9],file=con,"\n")
cat(nomfich,"z10",n.pix[10],n.pix.ok[10],NAs[10],moy[10],file=con,"\n")
cat(nomfich,"z11",n.pix[11],n.pix.ok[11],NAs[11],moy[11],file=con,"\n")
cat(nomfich,"z12",n.pix[12],n.pix.ok[12],NAs[12],moy[12],file=con,"\n")
cat(nomfich,"z13",n.pix[13],n.pix.ok[13],NAs[13],moy[13],file=con,"\n")
cat(nomfich,"z14",n.pix[14],n.pix.ok[14],NAs[14],moy[14],file=con,"\n")
cat(nomfich,"z15",n.pix[15],n.pix.ok[15],NAs[15],moy[15],file=con,"\n")
cat(nomfich,"z16",n.pix[16],n.pix.ok[16],NAs[16],moy[16],file=con,"\n")
cat(nomfich,"z17",n.pix[17],n.pix.ok[17],NAs[17],moy[17],file=con,"\n")
cat(nomfich,"z18",n.pix[18],n.pix.ok[18],NAs[18],moy[18],file=con,"\n")
cat(nomfich,"z19",n.pix[19],n.pix.ok[19],NAs[19],moy[19],file=con,"\n")
cat(nomfich,"z20",n.pix[20],n.pix.ok[20],NAs[20],moy[20],file=con,"\n")
cat(nomfich,"z21",n.pix[21],n.pix.ok[21],NAs[21],moy[21],file=con,"\n")
cat(nomfich,"z22",n.pix[22],n.pix.ok[22],NAs[22],moy[22],file=con,"\n")
cat(nomfich,"z23",n.pix[23],n.pix.ok[23],NAs[23],moy[23],file=con,"\n")
cat(nomfich,"z24",n.pix[24],n.pix.ok[24],NAs[24],moy[24],file=con,"\n")
cat(nomfich,"z25",n.pix[25],n.pix.ok[25],NAs[25],moy[25],file=con,"\n")
cat(nomfich,"z26",n.pix[26],n.pix.ok[26],NAs[26],moy[26],file=con,"\n")
cat(nomfich,"z27",n.pix[27],n.pix.ok[27],NAs[27],moy[27],file=con,"\n")
cat(nomfich,"z28",n.pix[28],n.pix.ok[28],NAs[28],moy[28],file=con,"\n")
cat(nomfich,"z29",n.pix[29],n.pix.ok[29],NAs[29],moy[29],file=con,"\n")
cat(nomfich,"z30",n.pix[30],n.pix.ok[30],NAs[30],moy[30],file=con,"\n")

```

```

cat(nomfich,"z31",n.pix[31],n.pix.ok[31],NAs[31],moy[31],file=con,'\n')
cat(nomfich,"z32",n.pix[32],n.pix.ok[32],NAs[32],moy[32],file=con,'\n')
cat(nomfich,"z33",n.pix[33],n.pix.ok[33],NAs[33],moy[33],file=con,'\n')
cat(nomfich,"z34",n.pix[34],n.pix.ok[34],NAs[34],moy[34],file=con,'\n')
cat(nomfich,"z35",n.pix[35],n.pix.ok[35],NAs[35],moy[35],file=con,'\n')
cat(nomfich,"z36",n.pix[36],n.pix.ok[36],NAs[36],moy[36],file=con,'\n')
cat(nomfich,"z37",n.pix[37],n.pix.ok[37],NAs[37],moy[37],file=con,'\n')
cat(nomfich,"z38",n.pix[38],n.pix.ok[38],NAs[38],moy[38],file=con,'\n')
cat(nomfich,"z39",n.pix[39],n.pix.ok[39],NAs[39],moy[39],file=con,'\n')
cat(nomfich,"z40",n.pix[40],n.pix.ok[40],NAs[40],moy[40],file=con,'\n')
cat(nomfich,"z41",n.pix[41],n.pix.ok[41],NAs[41],moy[41],file=con,'\n')
cat(nomfich,"z42",n.pix[42],n.pix.ok[42],NAs[42],moy[42],file=con,'\n')
cat(nomfich,"z43",n.pix[43],n.pix.ok[43],NAs[43],moy[43],file=con,'\n')
cat(nomfich,"z44",n.pix[44],n.pix.ok[44],NAs[44],moy[44],file=con,'\n')
cat(nomfich,"z45",n.pix[45],n.pix.ok[45],NAs[45],moy[45],file=con,'\n')

    } # fin boucle fichier
close(con) # fermeture fichier resultats

T2=Sys.time()
T2-T1

#*****
# FIN PART 1
#*****

#-----
# PART 2 : CONCATENER FICHIERS INDIVIDUELS EN FICHER UNIQUE
#-----

lf=list.files(pattern="SST_AVHRR")
ltab <- lapply(lf , function(x) read.table(x , h = F))
data0 <- do.call("rbind" , ltab)
head(data0)
tail(data0)
dim(data0)
write.table(data0,"SST_AVHRR_1986-2009.txt")

#-----
# PART 3 : MISE EN FORME FICHER UNIQUE
#-----

data2=data0[,c(1:2,6)]
colnames(data2)=c("fichier","zone","SST")

data3=reshape(data2,v.names="SST",idvar="fichier",timevar="zone",direction=
"wide")

an=substr(data3[,1],4,7)
mois= as.numeric(substr(data3[,1],8,9))
jour= as.numeric(substr(data3[,1],10,11))
data4=data.frame(data3$fichier,an,mois,jour,data3[,2:28])

write.csv(data4,"SST2002-2009.csv")

```