Evaluation In Situ et spatialisation des contaminations en éléments traces métalliques par la méthode de flourescence X : Cas d'étude à Liège

Auteur : Duchesne, François
Promoteur(s) : Colinet, Gilles; Liénard, Amandine
Faculté : Gembloux Agro-Bio Tech (GxABT)
Diplôme : Master en bioingénieur : sciences et technologies de l'environnement, à finalité spécialisée
Année académique : 2017-2018
URI/URL : http://hdl.handle.net/2268.2/5121

ANNEXES : ÉVALUATION *IN SITU* ET SPATIALISATION DES CONTAMINATIONS EN ÉLÉMENTS TRACES MÉTALLIQUES PAR LA MÉTHODE DE FLUORESCENCE X : CAS D'ÉTUDE À LIÈGE

FRANÇOIS DUCHESNE

TRAVAIL DE FIN D'ÉTUDES PRÉSENTÉ EN VUE DE L'OBTENTION DU DIPLÔME DE

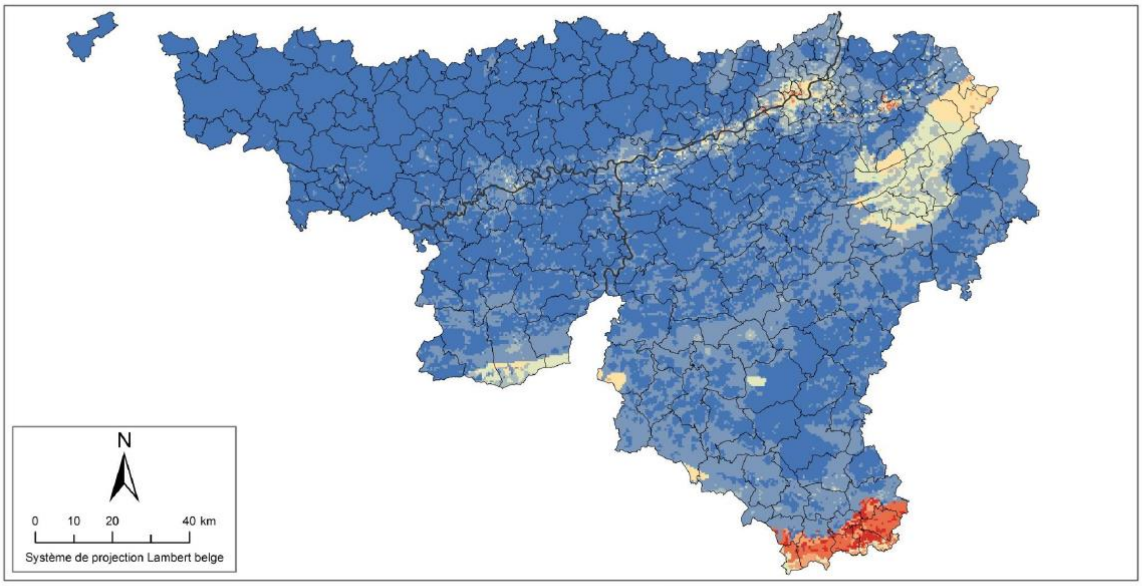
MASTER BIOINGÉNIEUR EN SCIENCES ET TECHNOLOGIES DE L'ENVIRONNEMENT

ANNÉE ACADÉMIQUE 2017-2018

(CO)-PROMOTEUR(S): GILLES COLINET, AMANDINE LIÉNARD

Table des matières

Annexes 1 : Cartes des teneurs attendues en ETMs dans les sols de Wallonie



Carte A3wal_As : Arsenic (As) : Teneurs attendues dans les sols de Wallonie

Projet POLLUSOL 2

Légende

Concentration en As (mg/kg matière sèche)

- < 10 (LQ)
- 11 - 13
- 14 - 16
- 17 - 19
- 20 - 22
- 23 - 25
- 26 - 40
- 41 - 121

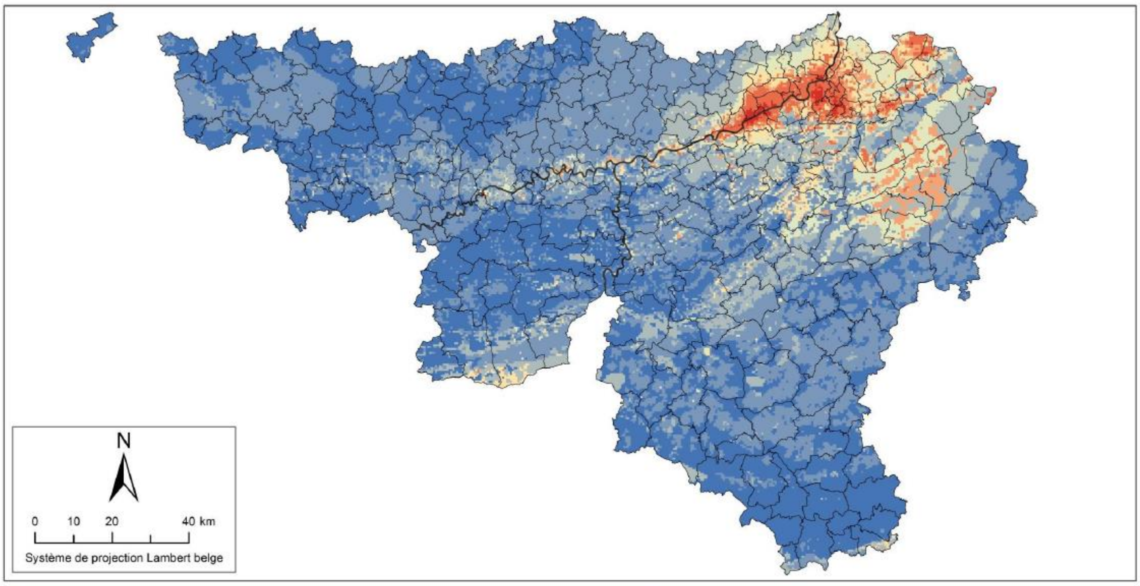
Escaut, Sambre ou Meuse

Sources :
- Concentration en polluant prédite par modélisation des résultats des projets POLLUSOL 1 et POLLUSOL 2 (UCL : Pereira B., Titeux H., Sonnet Ph. ; PGeostat LLC : Goovaerts P.)

Carte produite à l'aide de données © SPW-DGO3 :
- Carte Numérique des Sols de Wallonie : PCNSW, projet du Gouvernement wallon (GWVIII/2007/Doc.58.12/12.07/B.L & GWVII/2000/Doc.1331/07.12/JH.)
- Carte Numérique d'Occupation du Sol de Wallonie : PCNOSW, projet du Gouvernement wallon GWVIII/2005/Doc.1033/28.04/B.L.)

Carte réalisée dans le cadre du Projet POLLUSOL 2 (SPAQuE/UCL/ULg-GxABT/ULg-AQUAPÔLE/FPMs) financé par SPAQuE

Réalisation :
UCL, Earth and Life Institute
- Environmental Sciences (ELIE)



Carte A3wal_Cd : Cadmium (Cd) : Teneurs attendues dans les sols de Wallonie

Projet POLLUSOL 2

Légende

Concentration en Cd (mg/kg matière sèche)

- < 0.4 (LQ)
- 0.5 - 0.6
- 0.7 - 0.8
- 0.9 - 1.0
- 1.1 - 1.5
- 1.6 - 2.0
- 2.1 - 5.0
- 5.1 - 21.4

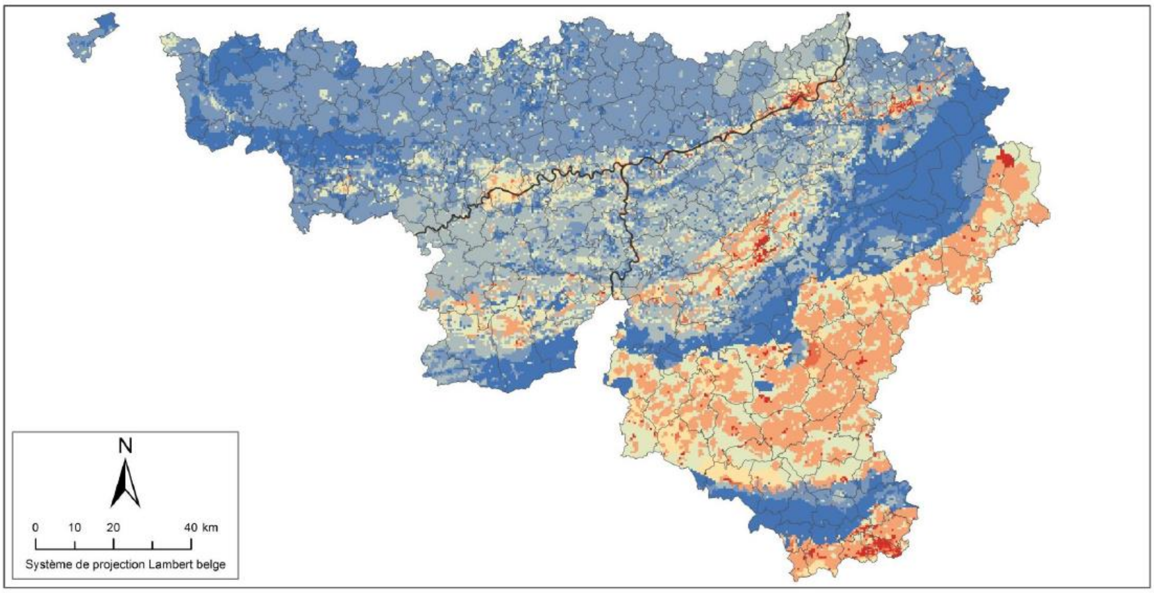
Escaut, Sambre ou Meuse

Sources :
- Concentration en polluant prédite par modélisation des résultats des projets POLLUSOL 1 et POLLUSOL 2 (UCL : Pereira B., Titeux H., Sonnet Ph. ; PGeostat LLC : Goovaerts P.)

Carte produite à l'aide de données © SPW-DGO3 :
- Carte Numérique des Sols de Wallonie : PCNSW, projet du Gouvernement wallon (GWVIII/2007/Doc.58.12/12.07/B.L & GWVII/2000/Doc.1331/07.12/JH.)
- Carte Numérique d'Occupation du Sol de Wallonie : PCNOSW, projet du Gouvernement wallon GWVIII/2005/Doc.1033/28.04/B.L.)

Carte réalisée dans le cadre du Projet POLLUSOL 2 (SPAQuE/UCL/ULg-GxABT/ULg-AQUAPÔLE/FPMs) financé par SPAQuE

Réalisation :
UCL, Earth and Life Institute
- Environmental Sciences (ELIE)



Carte A3wal_Ni : Nickel (Ni) : Teneurs attendues dans les sols de Wallonie

Projet POLLUSOL 2

Légende

Concentration en Ni (mg/kg matière sèche)

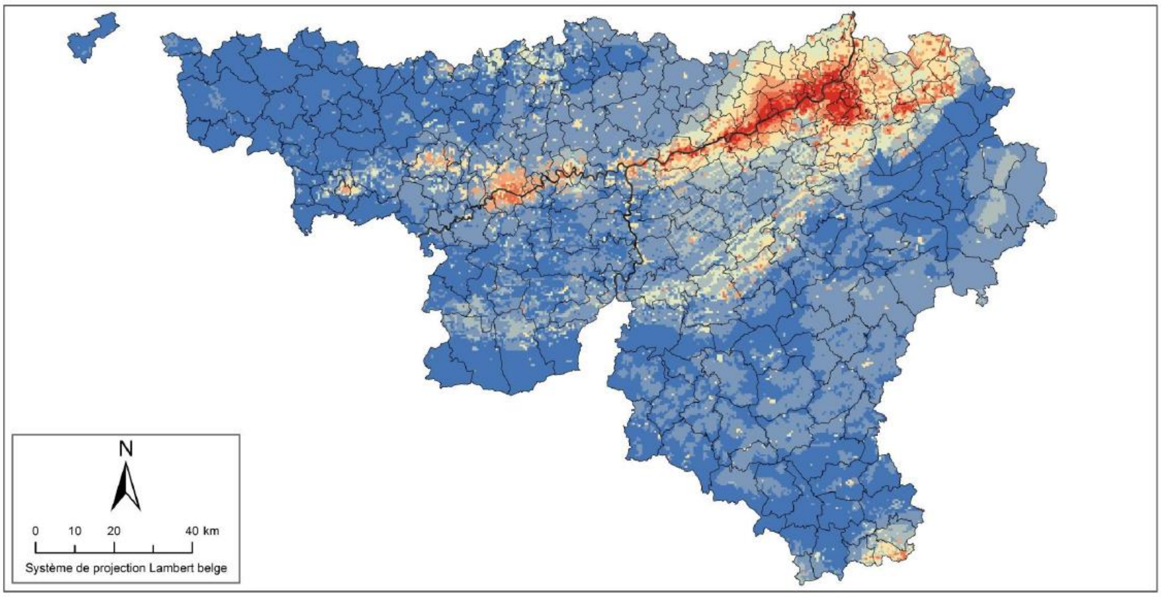
- 1 - 15
- 16 - 20
- 21 - 25
- 26 - 30
- 31 - 35
- 36 - 40
- 41 - 45
- 46 - 134

Escaut, Sambre ou Meuse

Sources :
- Concentration en polluant prédite par modélisation des résultats des projets POLLUSOL 1 et POLLUSOL 2 (UCL : Pereira B., Titeux H., Sonnet Ph. ; PGeostat LLC : Goovaerts P.)
Carte produite à l'aide de données © SPW-DGO3 :
- Carte Numérique des Sols de Wallonie : PCNSW, projet du Gouvernement wallon (GWVIII/2007/Doc.58.12/12.07/B.L & GWVII/2000/Doc.1331/07.12/JH.)
- Carte Numérique d'Occupation du Sol de Wallonie : PCNOSW, projet du Gouvernement wallon GWVIII/2005/Doc.1033/28.04/B.L.)

Carte réalisée dans le cadre du Projet POLLUSOL 2 (SPAQuE/UCL/ULg-GxABT/ULg-AQUAPÔLE/FPMs) financé par SPAQuE

Réalisation :
UCL, Earth and Life Institute
- Environmental Sciences (ELIE)



Carte A3wal_Zn : Zinc (Zn) : Teneurs attendues dans les sols de Wallonie

Projet POLLUSOL 2

Légende

Concentration en Zn (mg/kg matière sèche)

- 21 - 60
- 61 - 90
- 91 - 120
- 121 - 150
- 151 - 200
- 201 - 300
- 301 - 500
- 501 - 2235

Escaut, Sambre ou Meuse

Sources :
- Concentration en polluant prédite par modélisation des résultats des projets POLLUSOL 1 et POLLUSOL 2 (UCL : Pereira B., Titeux H., Sonnet Ph. ; PGeostat LLC : Goovaerts P.)
Carte produite à l'aide de données © SPW-DGO3 :
- Carte Numérique des Sols de Wallonie : PCNSW, projet du Gouvernement wallon (GWVIII/2007/Doc.58.12/12.07/B.L & GWVII/2000/Doc.1331/07.12/JH.)
- Carte Numérique d'Occupation du Sol de Wallonie : PCNOSW, projet du Gouvernement wallon GWVIII/2005/Doc.1033/28.04/B.L.)

Carte réalisée dans le cadre du Projet POLLUSOL 2 (SPAQuE/UCL/ULg-GxABT/ULg-AQUAPÔLE/FPMs) financé par SPAQuE

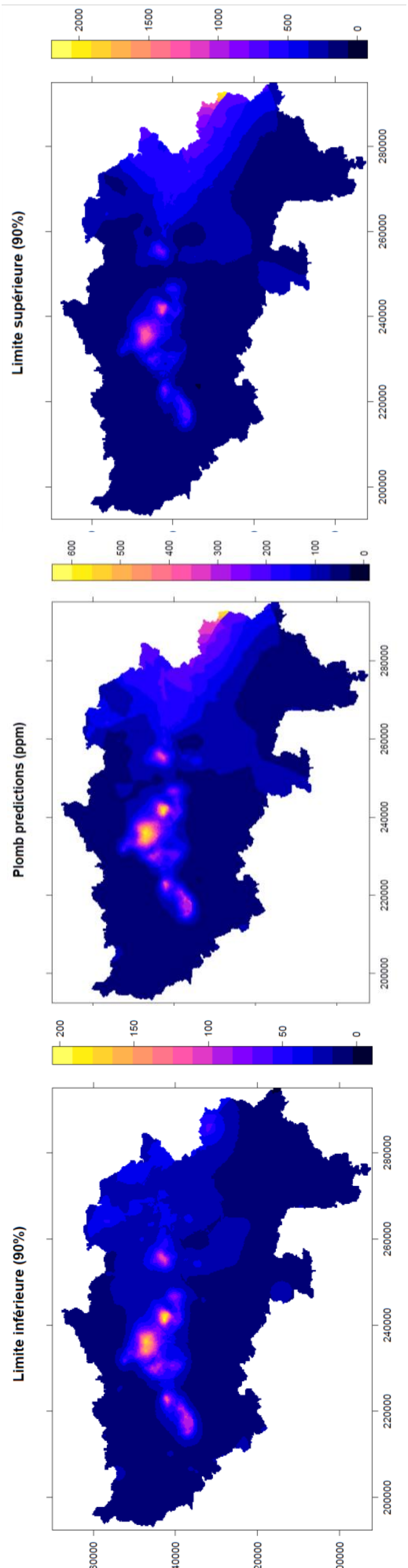
Réalisation :
UCL, Earth and Life Institute
- Environmental Sciences (ELIE)

Annexe 2 : Tableau des résultats d'analyse des ETMs

echant	Cr_er	Ni_er	Cu_er	Zn_er	As_er	Pb_er	Cr_ter	Ni_ter	Cu_ter	Zn_ter	As_ter	Pb_ter	Cr_lab	Ni_lab	Cu_lab	Zn_lab	As_lab	Pb_lab
gc1	31	27	65	280	17	130	/	27.3	47.0	206.3	12.0	113.7	58	32.3	65.3	295.0	18.3	180.0
gc10	27	23	35	210	13	76	/	25.7	31.7	202.7	8.0	76.0	62	25.7	40.3	213.7	11.7	91.7
gc2	29	27	85	340	18	180	/	17.3	43.7	219.7	10.0	139.0	77	28.0	74.0	349.3	20.7	188.0
gc3	30	32	96	390	20	230	/	25.3	59.0	261.3	13.7	170.7	87	37.7	582.0	401.7	19.7	244.7
gc4	29	30	86.5	390	20	210	/	28.0	58.7	254.0	9.0	170.3	88	33.7	96.0	366.3	21.7	211.3
gc5	30	32	100	410	24	320	/	34.0	72.3	303.7	14.3	205.0	82	39.0	112.7	439.3	23.3	276.3
gc6	28	27	74	360	18	150	/	18.5	43.3	231.3	8.7	109.3	58	30.0	90.0	368.7	22.7	147.3
gc7	24	21	46	230	14	92	/	16.3	32.0	157.7	10.0	/	50	22.0	44.7	222.3	17.0	89.0
gc8	28	25	64	330	17	160	48	19.0	42.0	232.7	7.7	138.7	65	27.7	67.0	331.3	19.7	161.3
gc9	24	24	96	280	16	150	/	20.0	48.0	216.7	8.7	134.7	53	22.0	76.7	302.0	20.7	160.7
pdt1	29	23	31	200	11	54	/	27.5	18.7	144.3	8.7	46.7	45	26.3	28.3	195.0	13.0	61.3
pdt2	29	23	22	170	12	48	/	21.7	13.7	98.3	8.3	28.0	51	27.7	26.0	173.3	14.7	48.0
pdt3	38	27	28	210	12	52	/	19.5	17.0	133.3	9.0	60.0	68	25.0	31.0	213.3	10.3	67.0
pdt4	33	26	33	250	13	73	/	22.7	19.3	177.7	9.7	57.7	61	20.0	33.0	236.7	16.7	59.7
pdt5	28	22	36	240	12	68	/	21.0	16.3	141.3	9.5	53.3	62	22.0	33.3	231.3	14.7	67.3
pdt6	32	26	36	240	12	83	/	28.3	26.7	160.0	9.3	50.3	57	22.0	179.3	224.0	16.7	63.0
pdt7	31	26	25	200	12	51	/	15.5	17.7	145.7	9.0	41.7	60	23.5	19.3	101.7	11.7	30.7
pdt8	29	24	27	240	11	57	/	17.3	18.7	143.3	7.7	45.7	64	26.5	28.3	201.3	14.7	56.0
pl1	31	26	29	200	13	72	/	24.0	19.3	148.7	8.3	59.0	85	27.0	31.0	181.7	15.0	67.0
pl10	30	25	36	250	14	84	/	19.5	22.0	168.7	9.0	61.0	79	22.3	33.3	253.7	17.7	79.0
pl11	32	28	26	220	13	66	/	23.7	14.7	137.7	6.5	102.0	39	24.7	13.7	89.0	11.0	35.3
pl2	29	28	40	300	14	95	/	24.0	22.0	195.0	7.5	79.3	74	32.3	42.0	289.7	15.3	116.3
pl3	27	24	49	290	17	100	/	19.0	32.5	231.5	14.0	81.5	54	25.3	44.7	295.3	16.3	104.0
pl4	29	27	35	270	16	90	/	20.0	24.3	163.7	10.0	65.7	70	19.5	37.3	254.7	18.0	88.7
pl5	32	28	42	270	15	85	/	23.5	25.3	157.0	8.0	58.0	62	27.3	45.0	268.3	15.3	91.7
pl6	28	25	41	270	15	89	/	24.5	28.0	189.0	8.3	72.0	69	21.7	40.3	274.0	17.3	85.3
pl7	31	28	38	300	15	140	/	28.3	23.7	197.7	8.5	84.7	59	19.0	27.3	181.0	17.0	61.0
pl8	26	24	35	290	16	130	/	16.0	23.7	209.7	12.0	98.7	72	23.0	35.3	290.0	18.0	116.0
pl9	27	25	27	250	12	77	/	18.0	16.7	166.3	6.7	61.3	64	22.0	26.7	246.3	14.7	75.3
gc1	32	28	37	170	14	74	/	19.7	29.3	159.7	8.7	82.3	73	25.0	40.3	157.3	17.7	66.0
gc10	31	24	15	64	10	19	/	37.3	13.0	46.7	8.0	24.7	59	29.0	18.3	56.3	15.0	22.0
gc2	23	18	25	110	9.7	72	/	17.5	19.3	87.7	8.0	50.7	71	17.0	28.3	117.0	11.0	66.3
gc3	29	29	61	250	17	300	/	26.3	43.7	206.3	13.0	133.7	87	30.0	65.3	234.3	24.7	184.0
gc4	29	30	77	350	20	210	/	24.7	49.0	233.7	9.0	169.3	78	35.0	81.7	338.3	20.3	212.0
gc5	29	26	39	170	13	83	/	25.7	39.7	153.3	10.3	91.0	68	26.7	40.3	165.7	18.0	77.7
gc6	24	23	50	280	15	100	/	19.0	38.7	214.7	9.7	81.0	69	27.3	55.7	299.3	17.0	106.0
gc7	21	16	18	120	7.7	33	/	17.0	20.3	93.7	6.0	28.5	50		20.3	120.0	9.5	37.3
gc8	30	25	41	220	14	94	/	50.0	23.3	128.3	8.0	60.7	58	24.7	40.0	199.7	13.0	96.0
gc9	24	18	19	84	9.2	31	/	16.0	15.0	63.7	5.0	36.0	46	17.0	23.0	85.3	11.0	38.0
pdt1	36	25	17	94	10	26	/	21.3	11.7	65.3	8.3	20.0	47	28.3	21.0	92.0	14.0	27.0
pdt2	40	29	19	100	12	29	/	43.7	13.3	83.0	6.7	25.0	74	26.0	19.3	102.0	15.0	27.3
pdt3	30	23	18	120	11	33	/	24.0	10.0	86.3	8.0	42.5	64	28.3	19.7	106.7	9.3	41.0
pdt4	35	26	18	100	11	28	/	28.0	13.7	108.3	9.3	29.0	65	28.3	16.7	95.7	11.7	34.0
pdt5	30	21	16	100	9.6	28	/	21.0	9.0	59.3	7.3	21.3	45	18.3	18.7	97.0	16.5	31.7
pdt6	31	24	61	290	13	110	/	23.3	37.3	168.0	7.5	81.3	52	25.0	55.0	365.3	14.0	103.7
pdt7	35	26	18	100	11	28	/	32.7	12.3	50.7	10.5	19.5	48	19.3	27.3	198.3	13.3	57.7
pdt8	39	27	26	120	11	37	/	55.0	33.7	158.0	7.0	59.3	56	19.7	28.3	109.3	18.0	28.0
pl1	38	28	21	110	12	38	/	31.0	10.7	49.7	8.7	28.0	56	19.3	22.0	109.3	10.0	48.0
pl10	29	21	15	110	11	53	/	18.7	14.7	93.7	9.7	31.7	63	18.3	17.0	106.0	15.7	29.0
pl11	31	24	16	93	11	28	/	23.0	14.3	67.7	6.0	27.7	62	30.5	28.3	212.7	15.3	70.3
pl2	31	27	32	250	15	98	/	27.0	17.3	218.7	9.7	55.3	70	25.0	32.3	206.3	15.3	68.0
pl3	30	25	38	270	14	90	/	16.3	23.0	165.0	9.3	60.3	72	30.7	44.3	255.7	18.3	98.3
pl4	30	23	20	120	12	41	/	19.0	11.0	59.7	6.0	29.5	48	26.3	24.3	115.3	15.3	40.0
pl5	30	24	25	130	11	43	/	23.0	15.3	84.7	6.0	39.0	58	17.5	24.3	121.7	11.7	44.3
pl6	29	22	48	140	11	41	/	21.0	12.0	73.3	8.7	27.0	44	19.7	26.7	133.0	15.3	41.3
pl7	25	22	30	200	13	68	/	26.0	24.0	156.7	10.0	67.3	64	21.5	53.7	291.7	17.3	101.3

pl8	30	22	22	160	11	62	/	15.5	15.3	94.3	6.5	41.3	69	20.7	26.0	149.0	16.3	49.0
pl9	27	23	18	140	10	43	/	20.7	13.0	118.3	9.3	37.0	48	18.5	19.3	138.0	10.5	51.0
34	55	51	256	1519	40	556	/	/	/	/	/	/	54	53	270	1611	41	606
35	49	46	215	1513	30	495	/	/	/	/	/	/	52	48	229	1623	31	520
38	68	48	252	1592	33	609	/	/	/	/	/	/	63	47	250	1622	33	620
46	119	40	202	1196	28	413	/	/	/	/	/	/	112	38	195	1138	28	388
68	77	44	215	1360	31	524	42	49	175	1089	19	403	78	44	210	1390	31	521
92	47	37	162	1154	28	356	/	/	/	/	/	/	45	34	161	1134	26	330
97	81	46	238	1753	30	557	/	/	/	/	/	/	76	45	234	1628	30	494
100	58	49	233	1447	35	522	/	/	/	/	/	/	55	47	222	1335	33	489
101	46	44	214	1400	33	505	/	/	/	/	/	/	59	54	272	1760	39	630
104	58	48	214	1351	34	505	/	/	/	/	/	/	56	47	211	1338	33	506
111	81	55	257	1726	37	596	/	/	/	/	/	/	79	54	249	1711	36	575
139	50	47	258	1635	33	628	/	/	/	/	/	/	51	47	256	1531	32	593
145	44	49	215	1473	27	464	/	/	/	/	/	/	43	48	212	1447	27	447
164	94	49	219	1513	32	524	/	/	/	/	/	/	92	49	216	1486	31	507
180	48	35	104	738	19	253	/	/	/	/	/	/	48	35	104	738	19	253
193	38	40	197	1586	44	514	/	/	/	/	/	/	38	41	208	1677	41	507
204	57	50	254	1569	33	574	/	/	/	/	/	/	64	52	257	1628	34	614
216	45	47	190	1300	32	449	/	/	/	/	/	/	45	46	182	1314	32	448
238	45	48	212	1568	31	557	/	/	/	/	/	/	42	45	202	1390	30	514
252	69	43	262	1271	28	491	52	41	155	992	14	447	52	46	242	1387	29	481
259	59	46	217	1395	33	473	/	/	/	/	/	/	58	46	215	1385	33	471
263	41	41	200	1316	27	430	/	/	/	/	/	/	47	45	213	1362	27	432
266	53	40	212	1249	28	511	/	/	/	/	/	/	53	40	189	1137	29	541
277	50	40	156	1038	25	390	/	/	/	/	/	/	49	41	148	1002	24	358
286	43	48	257	1536	32	614	/	/	/	/	/	/	39	45	260	1573	31	635
106-107-108	52	49	226	1505	34	520	/	/	/	/	/	/	58	52	218	1422	36	505
1-2-3-4	63	55	263	1819	38	681	68	42	163	1189	22	539	67	56	259	1733	37	657
143-144	57	43	208	1509	30	473	/	/	/	/	/	/	58	43	213	1554	30	478
180-181	49	35	119	810	19	312	/	/	/	/	/	/	47	35	138	853	19	341
188-189	56	44	232	1568	33	552	/	/	/	/	/	/	56	45	228	1623	33	576
220-221	41	39	178	1235	26	389	/	/	/	/	/	/	37	37	160	1174	25	365
270-271	38	44	239	1425	31	714	/	/	/	/	/	/	36	44	261	1662	34	896
304-305	47	46	177	1289	30	433	/	/	/	/	/	/	48	47	171	1275	31	434
86-87	56	39	151	1133	26	321		35	106	726	16	247	58	40	155	1129	27	318
Min	21	16	15	64	8	19	42	16	9	47	5	20	36	17	14	56	9	22
max	119	55	263	1819	44	714	68	55	175	1189	22	539	112	56	582	1760	41	896

Annexes 3 : Limites supérieures et inférieures de prédiction (90%) du plomb



Annexe 4 : Krigeage ordinairee et analyse des co-variables (Code R)

```
setwd("D:/aa TFE/3CARTO_province de Liège/2_R_krigeage/RW")
# empty memory and workspace
gc()
rm(list=ls())
# load libraries
require(sp)
require(raster)
require(rgdal)
require(randomForest)
require(leaflet)
require(plotKML)
require(htmlwidgets)
library(maptools)
library(gstat)
library(rgeos)
library(MASS)
library(readxl)
require(caret)
require(plyr)
require(gridExtra)
require(lattice)

etm.plomb = read_xlsx('ETM_total_cokri.xlsx')
etm.plomb <- etm.plomb[complete.cases(etm.plomb[,"Pb"]),]
etm <- subset(etm.plomb, select = c(4,5,6))
# Lecture des différents tif utilisés
mask <- readGDAL("prov_liege.tif")
mnt <- readGDAL("mnt_250_extract_ok.tif")
cosw <- readGDAL("cosw_extract_ok.tif")
ser <- readGDAL("ser_extract_ok.tif")
slope <- readGDAL("slope_extract_ok.tif")
province <- readOGR("D:/aa TFE/3CARTO_province de Liège/1_Carto_Liège/province.shp")
mask[mask = 3] <- 1
plot(mask)
summary(mask)

# Exploration des données à faire pour chaque élément
etm
dim(etm)
names(etm)
summary(etm)
class(etm)
#Pb
dev.off()

h1 <- histogram(etm$Pb, main = "Distribution plomb",
           col="LightBlue", xlab = "Pb (ppm)", ylab = "%")
h2 <- histogram(log10(etm$Pb), main = "",
           col="LightBlue", xlab = "log10(Pb)", ylab = "%")
grid.arrange(h1,h2, ncol=2)
#Rajout des colonnes des transformations logarithmiques
etm <- cbind(etm,
         lpb = log10(etm$Pb))
#SPATIALISATION
class(etm)
coordinates(etm) <- ~X+Y
class(etm)
projection(etm) <- "+proj=lcc +lat_1=49.8333339 +lat_2=51.16666723333333 +lat_0=90 +lon_0=4.367486666666666
+x_0=150000.01256 +y_0=5400088.4378 +ellps=intl +units=m +no_defs"
projection(province) <- "+proj=lcc +lat_1=49.8333339 +lat_2=51.16666723333333 +lat_0=90 +lon_0=4.367486666666666
+x_0=150000.01256 +y_0=5400088.4378 +ellps=intl +units=m +no_defs"
etm<-etm[-zerodist(etm)[,1],]
spplot(etm, zcol = "Pb", xlim = c(192500,300000),
     ylim = c(92000,170000), cex = 0.8, main = "Plomb en province liégeoise (mg/kg)",
     key.space = list(x = 0.1, y = 0.29, corner = c(0,1)),
     cuts = c(0,50,100,150,200,300,500,1000),
     sp.layout = list(list("sp.polygons", province)),
     scales=list(draw=T), col.regions = bpy.colors())
dev.copy(pdf,paper='a4', width = 800, height = 700,
```

```
    paste0("D:/aa TFE/3CARTO_province de Liège/2_R_krigeage/RW/out_liege/plomb_point.pdf"))
dev.off()

 #VARIOGRAMME
# # GSTAT et semi-variogramme expérimentale
pb <- gstat(formula = lpb~1, data = etm)
vpb <- variogram(pb)
plot(vpb, plot.nu=FALSE)
# Initialisation
vgmpb <- vgm(nugget = 0.05, psill = 0.15, range = 15000, model = "Sph")
plot(vpb, vgmpb)
# fit semivariogramme
vgmpb <- fit.variogram(vpb, vgmpb, fit.method=7)
plot(vpb, vgmpb, xlab = "distance (m)", main = "Semi-variogramme du plomb")
vgmpb
#Sum of Squared Errors between sample and fitted semivariogram to compare methods
attr(vgmpb, "SSErr")
# Krigeage ordinaire
pb.krig <- krige(formula = lpb~1, locations = etm, newdata = mask,
            model = vgmpb, nmax= 30, maxdist=17150, debug.level=-1)
names(pb.krig)

#Transformation inverse
pb.krig$var1.pred.norm <- 10^(pb.krig$var1.pred+0.5*pb.krig$var1.var)
names(pb.krig)
spplot(pb.krig, zcol = "var1.pred.norm", col.regions = bpy.colors(),
      xlim = c(192500,295000), ylim = c(92000,170000),
      main="Plomb predictions (ppm) ",
      scales = list(draw=TRUE))
dev.copy(pdf,paper='a4', width = 800, height = 700,
    paste0("D:/aa TFE/3CARTO_province de Liège/2_R_krigeage/RW/out_liege/plomb_ok.pdf"))
dev.off()
pb.krig$Upper.boundary <- 10^(pb.krig$var1.pred + 1.64*sqrt(pb.krig$var1.var))
pb.krig$Lower.boundary <- 10^(pb.krig$var1.pred - 1.64*sqrt(pb.krig$var1.var))
names(pb.krig)
pb.krig
pb.krig$sd <- 10^(sqrt(1.64*pb.krig$var1.var))
spplot(pb.krig, zcol = "sd", col.regions = bpy.colors(),
      main="écart-type", xlim = c(192500,295000), ylim = c(92000,170000),
      scales = list(draw=TRUE))
dev.copy(pdf,paper='a4', width = 800, height = 700,
    paste0("D:/aa TFE/3CARTO_province de Liège/2_R_krigeage/RW/out_liege/plomb_ok_sd.pdf"))
dev.off()

summary(pb.krig)
spplot(pb.krig, zcol = "Upper.boundary", col.regions = bpy.colors(),
      main="Limite supérieure (90%)", xlim = c(192500,295000), ylim = c(92000,170000),
      scales = list(draw=TRUE))
dev.copy(pdf,paper='a4', width = 800, height = 700,
    paste0("D:/aa TFE/3CARTO_province de Liège/2_R_krigeage/RW/out_liege/plomb_ok_ls.pdf"))
dev.off()
spplot(pb.krig, zcol = "Lower.boundary", col.regions = bpy.colors(),
      main="Limite inférieure (90%)", xlim = c(192500,295000), ylim = c(92000,170000),
      scales = list(draw=TRUE))
dev.copy(pdf,paper='a4', width = 800, height = 700,
    paste0("D:/aa TFE/3CARTO_province de Liège/2_R_krigeage/RW/out_liege/plomb_ok_li.pdf"))
dev.off()
pb.krig$var1.sd <- sqrt(pb.krig$var1.var)
max(pb.krig$sd, na.rm=T)
min(pb.krig$sd, na.rm=T)

# CROSS VALIDATION

cv.o <- krige.cv(lpb ~ 1, etm, model=vgmpb, nfold=nrow(etm), nmax= 30, maxdist=17150,
            debug.level=-1)
summary(cv.o)
res <- as.data.frame(cv.o)$residual
sqrt(mean(res^2))  #RMSE
mean(res)        #MEAN ERROR
mean(res^2/as.data.frame(cv.o)$var1.var)
rm(res)
diff <- 10^(cv.o$var1.pred+0.5*cv.o$var1.var) - 10^(etm$lpb)
```

```
summary(diff)
histogram(diff, col="lightblue2",
        main="Cross-Validation erreurs", xlab="ppm", breaks = 10)
##################        COKRIGEAGE      ###################
cokri <- subset(etm.plomb, select = c(4,5,6,14,15,16))
summary(cokri)
cokri <-subset(cokri, slope>=1)

#CO-VARIABLE SLOPE
h1 <- histogram(cokri$slope,
            col="LightBlue", xlab = "Pente (°)")
h2 <- histogram(log10(cokri$slope),
            col="LightBlue", xlab = "Pente (log)")
grid.arrange(h1,h2, ncol=2)

h1 <- histogram(cokri$RASTERVALU,
            col="LightBlue", xlab = "élévation (m)")
h2 <- histogram(log10(cokri$RASTERVALU),
            col="LightBlue", xlab = "élévation (log)")
grid.arrange(h1,h2, ncol=2)

cokri <-subset(cokri, slope>=1)
cokri <- cbind(cokri,
           lpb = log10(cokri$Pb),
           lslope = log10(cokri$slope),
           lele = log10(cokri$RASTERVALU))
str(cokri)
a <- xyplot(cokri$lpb ~ cokri$lslope, pch=20, cex=1.2, col="red4",
     ylab="log10(Pb)", xlab="pente (log)")
(r <- cor(cokri$lpb,log10(cokri$slope), use = "complete"))
(r2 <- r^2)# ---> 19% expliqué par le slope

#CO-VARIABLE MNT
 b<- xyplot(cokri$lpb ~ cokri$lele, pch=20, cex=1.2, col="red4",
     ylab="log10(Pb)", xlab="log10(élévation)")
(r <- cor(cokri$lpb,cokri$RASTERVALU, use = "complete"))
(r2 <- r^2)
r2  #---> 11.6 % expliqué par le MNT
grid.arrange(a,b,ncol = 2, nrow = 3)
dev.copy(pdf,paper='a4', width = 800, height = 700,
       paste0("D:/aa TFE/3CARTO_province de Liège/2_R_krigeage/RW/out_liege/corré_pente_ele.pdf"))
dev.off()
coordinates(cokri) <- ~X+Y
class(cokri)
projection(cokri) <- "+proj=lcc +lat_1=49.8333339 +lat_2=51.16666723333333 +lat_0=90 +lon_0=4.367486666666666
+x_0=150000.01256 +y_0=5400088.4378 +ellps=intl +units=m +no_defs"

cokri<-cokri[-zerodist(cokri)[,1],]
slope <- gstat(formula = lslope~1, data = cokri)
vslope <- variogram(slope)
plot(vslope, plot.nu=FALSE)
# define initial semivariogram model
vgmslope <- vgm(nugget = 0.035, psill = 0.05, range = 30000, model = "Exp")
plot(vslope, vgmslope)
# fit semivariogram model
vgmslope <- fit.variogram(vslope, vgmslope, fit.method=7)
plot(vslope, vgmslope)
vgmslope

(g <- gstat(NULL, id = "lpb", form = lpb ~ 1, data=cokri))
(g <- gstat(g, id = "lslope", form = lslope ~ 1, data=cokri))

v.cross <- variogram(g)
str(v.cross)
plot(v.cross, pl=F)

(g <- gstat(g, id = "lpb", model = vgmpb, fill.all=T))
(g <- fit.lmc(v.cross, g))
plot(variogram(g), model=g$model)

k.c <- predict(g, mask, nmax= 30,debug.level=-1)
names(k.c)
```

```
k.c$lpb.pred.norm <- 10^(k.c$lpb.pred+0.5*k.c$lpb.var)
names(k.c)
plot.kresults(k.c, "lpb.pred.norm", mask, cokri, "Plomb", "CK avec la pente ")
spplot(k.c, zcol = "lpb.pred.norm", col.regions = bpy.colors(),
    main="CK du plomb avec la pente ", xlim = c(192500,295000), ylim = c(92000,170000),
    scales = list(draw=TRUE))
dev.copy(pdf,paper='a4', width = 800, height = 700,
     paste0("D:/aa TFE/3CARTO_province de Liège/2_R_krigeage/RW/out_liege/ck_pb_slope.pdf"))
dev.off()
```

Annexe 5: Modèle Random Forest (Code R)

```
#RANDOM FOREST LIEGE
setwd("D:/aa TFE/3CARTO_province de Liège/2_R_krigeage/RW")
gc()
rm(list=ls())
# packages nécessaires
require(sp)
require(raster)
require(rgdal)
require(randomForest)
require(leaflet)
require(plotKML)
require(htmlwidgets)
library(htmltools)
library(maptools)
library(gstat)
library(rgeos)
library(MASS)
library(readxl)
require(caret)
require(plyr)
#création de la variable à prédire
# retrait des colonnes non nécessaires
etm = read_xlsx('ETM_total_liege.xlsx')
summary(etm)
etm <- etm[,-c(1,2)]
varpb = etm[,c(1,2,3)]
str(varpb)
head(varpb)
names(varpb) = c("X","Y","Pb")
varpb_ok = as.data.frame(varpb)
#Traitement des données
d1= subset(varpb_ok, varpb_ok$Pb != -9999)
dum = duplicated(d1)
summary(dum)
save(d1, file = "pointData.rda")
d = d1
str(d)
head(d)
save(d, file = "pointData_Plomb.rda")
summary(d$Pb)
dev.off()
hist(d$Pb, col = "lightblue", xlab="mg/kg", main="Plomb")
coordinates(d) <- ~ X+Y
plot(d)
str(d)
# légende
breaks <- c(seq(0,900,100),seq(1000,5000,1000))
# plot
spplot(d, c("Pb"),
    cuts = breaks,
    cex = 0.7,
    sp.layout = list(arrow),
    key.space = "right",
```

```
      main = "Concentrations en Plomb (mg/kg)",
      xlab = "easting (m)",
      ylab = "northing (m)",
      scales = list(draw=T))
d= as(varpb_ok, Class="data.frame")

#Création d'un stack de co-variables
mask <- readGDAL("prov_liege.tif")
mnt <- readGDAL("mnt_250_extract_ok.tif")
cosw <- readGDAL("cosw_extract_ok.tif")
ser <- readGDAL("ser_extract_ok.tif")
slope <- readGDAL("slope_extract_ok.tif")
cov.lst <- list.files(path="stack", pattern = ".tif")
r1 <- stack(paste0("stack/", cov.lst))
cov.lst
class(r1)
dim(r1)
plot(r1[["cosw_extract_ok"]], main = "Cosw Model", xlab = "Easting (m)", ylab ="Northing(m)")
m <- raster(paste0("prov_liege.tif"))
r2 <- mask(r1,m)
plot(r2[["cosw_extract_ok"]], main = "Cosw Model", xlab = "Easting (m)", ylab ="Northing(m)")
r4 <- as(r2, "SpatialGridDataFrame")
r5 <- as(r4, Class = "data.frame")
nzv <- nearZeroVar(r5, saveMetrics = TRUE)
nzv
summary(nzv$zeroVar)
summary(nzv$nzv)
r6 <- r5
summary(r6)
r <- r6[complete.cases(r6),]
names(r6)
r <- cbind(r6[,c("s1","s2")], r6[,!names(r6) %in% c("s1","s2")])
names(r6)
save(r, file='covariateStack.rda')

#Matrice de régression
coordinates(d) <- ~X+Y
proj4string(d)<- CRS("+proj=lcc +lat_1=49.8333339 +lat_2=51.16666723333333
                +lat_0=90 +lon_0=4.367486666666666 +x_0=150000.01256
                +y_0=5400088.4378 +ellps=intl +units=m +no_defs")
str(d)
gridded(r) <- ~s1+s2
proj4string(r)<- CRS("+proj=lcc +lat_1=49.8333339 +lat_2=51.16666723333333
                +lat_0=90 +lon_0=4.367486666666666 +x_0=150000.01256
                +y_0=5400088.4378 +ellps=intl +units=m +no_defs")
dum <- over(d,r)
rm <- cbind(d@data, d@coords, dum)
save(rm, file="regressionMatrix.rda")

#RANDOM FOREST
# set random seed
set.seed(20180523)
load("regressionMatrix.rda")
load("covariateStack.rda")
summary(complete.cases(rm))
rm[!complete.cases(rm),]
rm <- rm[complete.cases(rm),]
summary(rm$Pb)
rm <- rm[rm$Pb!=0,]
names(rm)
d <- rm$Pb
covar <- rm[,4:10]
names(covar)
rf <- randomForest(x = covar, y = d)
str(rf, max.level=2)
head(getTree(rf, k=1, labelVar = TRUE))
dev.off
```

```
varImpPlot(rf, main = "Random Forest")
plot(rf$predicted, rf$y, xlab="Plomb prédit (mg/kg)", ylab="Plomb mesuré (mg/kg)")
abline(0,1)
summary(rf$predicted)
summary(etm$Pb)
round(rf$mse[rf$ntree],1) #16543.2
round(rf$rsq[rf$ntree],2) #0.44
plot(rf$mse, xlab = "trees", ylab = "MSE")
plot(rf$rsq, xlab = "trees", ylab = "Rsquared")
r <- r[complete.cases(r),]
p.temp <- predict(rf, newdata = r)
str(p.temp)
summary(p.temp)
names(r)
# Compilation des prédictions
p <- data.frame(x = r$s1, y = r$s2, Pb = p.temp)
head(p)
gridded(p) <- ~x+y
proj4string(p) <- CRS("+proj=lcc +lat_1=49.8333339 +lat_2=51.16666723333333
            +lat_0=90 +lon_0=4.367486666666666 +x_0=150000.01256
            +y_0=5400088.4378 +ellps=intl +units=m +no_defs")
p.r <- raster(p)
spplot(p, zcol = "Pb")
plot(p.r)
spplot(p, zcol = "Pb", at = c(0,50,100,140,150,160,170,180,190,200,210,220,240,260,280,300,320,340,360,380,400,1000) ,
    scales = list(draw=T),
    main = "RF Plomb (mg/kg)")
dev.copy(pdf,paper='a4', width = 800, height = 700,
     paste0("D:/aa TFE/3CARTO_province de Liège/2_R_krigeage/RW/out_liege/rf_pb.pdf"))
dev.off()
p$safe <- factor(ifelse(p$Pb > 200,
             1, 0), labels=c("Non-contaminé","Contaminé"))
spplot(p, zcol = "safe", #at = c(0,50,100,140,150,160,170,180,190,200,210,220,240,260,280,300,320,340,360,380,400,1000) ,
    scales = list(draw=T),col.regions = c("darkgreen", "red"),
    main = "Contamination au plomb")
dev.copy(pdf,paper='a4', width = 800, height = 700,
     paste0("D:/aa TFE/3CARTO_province de Liège/2_R_krigeage/RW/out_liege/contami_pb.pdf"))
dev.off()
names(p)
# Créatino d'un fichier de sortie
dir.create(paste0(getwd(),"/out"))
# save as GeoTIFF
writeGDAL(p["Pb"], fname = "./out/Plomb_random_forest.tif", drivername = "GTiff", type = "Float32")
marai <- readOGR("points_maraichers.shp")
pop <- read.csv2("marai_pops.csv", sep=";")
str(pop)

########### #Cartes interactives
# Matrice de régression à un nouvel objet
d.ll <- rm
# Projection des coordonnées
coordinates(d.ll) <- ~X+Y
proj4string(d.ll) <- CRS("+proj=lcc
              +lat_1=49.8333339 +lat_2=51.16666723333333
              +lat_0=90 +lon_0=4.367486666666666 +x_0=150000.01256
              +y_0=5400088.4378 +ellps=intl +units=m +no_defs")
d.ll <- spTransform(d.ll, CRSobj = CRS("+init=epsg:4326"))
# Carte basique
leaflet() %>%
  addTiles() %>%
  addMarkers(data = d.ll)
# maraîchers de la ceinture alimentaire
leaflet() %>%
  addTiles() %>%
  addMarkers(data = marai)
```

```
# Popups
my_pops <- paste0(
  "<strong>Site: </strong>",
  d.ll$ProfID,
  '<br>
  <strong> Plomb (mg/kg): </strong>',
  round(d.ll$Pb,1))
####### Créatino de la carte interactive
leaflet() %>%
  addProviderTiles("Esri.WorldImagery") %>%
  addMarkers(data = d.ll, popup = my_pops)
leaflet() %>%
  addProviderTiles("Esri.WorldImagery") %>%
  addMarkers(data = marai)
# Jeu de couleurs
pal1 <- colorQuantile("YlOrBr", domain = rm$Pb)
pal2 <- colorNumeric(SAGA_pal[[1]], domain = d.ll$Pb, na.color = "transparent")
# Quantiles du plomb
(ll1 <- leaflet() %>%
    addProviderTiles("Esri.WorldImagery") %>%
    addCircleMarkers(data = d.ll, color = ~pal1(Pb), popup = my_pops) %>%
    addLegend("bottomright", pal = pal1, values = d.ll$Pb,
              title = "Plomb",
              opacity = 0.8))
# Concentrations du plomb
(ll2 <- leaflet() %>%
    addProviderTiles("Esri.WorldImagery") %>%
    addCircleMarkers(data = d.ll, color = ~pal2(Pb), popup = my_pops) %>%
    addLegend("bottomright", pal = pal2, values = d.ll$Pb,
              title = "Concentration en Plomb (mg/kg)",
              opacity = 0.8))
saveWidget(ll1, file = paste0(getwd(),"/out/Plomb_liege.html",sep=""))
# Légende
header <- "Concentrations en Plomb(mg/kg)"
pal <- colorBin(SAGA_pal[[1]], values(p.r),bins = c(0,50,100,150,200,250,300,350,400,450,500,600,700,800,900,1000,1500), na.color =
"transparent")
# Création du Leaflet
description <- paste(sep = "<br>", marai$PopupInfo)
(ll <- leaflet() %>%
    addProviderTiles("Esri.WorldImagery") %>%
    addRasterImage(p.r, colors=pal, opacity=0.5) %>%
    addLegend(pal=pal, values=values(p.r), title=header)%>%
    addMarkers(data = marai, popup = description))
saveWidget(ll, file = paste0(getwd(),"/out/plomb_province_liege.html",sep=""))
# Carte
save(p, p.r, file="rfplombmap.rda")
# Modèle
save(rm,rf,file="randomForestModel.rda")
# Environnement
save.image("randomForest.rda")

             #VALIDATION RANDOM FOREST
# set random seed
set.seed(20180523)
load("randomForestModel.rda")
# Erreurs de prédiction
pe <- rf$predicted - rf$y
# Statistiques
summary(pe)
hist(pe, breaks = 10)
# ME
(me <- round(mean(pe),3)) #round permet d'imposer le nombre de décimal
# MSE
(mse <- round(mean(pe**2),2))
# RMSE
(rmse <- round(sqrt(mean(pe**2)),2))
# RMedSE
```

```
(rmedse <- round(sqrt(median(pe**2)),2))
# AVE
(ave <- round(1-(sum(pe**2)/(sum((rf$y-mean(rf$y))**2))),3))
# create empty matrix
valstat <- matrix(nrow=6, ncol=1)
# add statistics
valstat <- rbind(me,mse,rmse,rmedse,ave,rsqCor) #remplir les lignes avec rbind et les colonnes avec cbind
# give row and column names
row.names(valstat) <- c('ME', 'MSE', 'RMSE', 'RMedSE', 'AVE', 'RSQ-COR')
colnames(valstat) <- c('value')
# export
write.csv(valstat, "rf_validation.csv")
## ----correlation_plot, message=F-----------------------------------------
# scatterplot
plot(rf$predicted, rf$y, xlab="Plomb prédit (ppm)", ylab="Plomb mesuré (ppm)")
abline(0,1)
## ----trainControl, results = 'hide', message=F--------------------------
# create an object with the training parameters
cvPar <- trainControl(
  method = "cv",
  number = 5,
  verboseIter = TRUE,
  savePredictions = TRUE)
# inspect
str(cvPar)
# copy soil property values to a new vectors
d <- rm$Pb
# copy the covariates to new data.frames
covar <- rm[,4:10]
# define the mtry parameter (1/3 of the number of covariates by default)
mtry <- data.frame(mtry = floor(ncol(covar)/3)) # or: rf$mtry
# cross-validation with caret package
rf.cv <- train(x = covar, y = d, method = "rf", trControl = cvPar, tuneGrid = mtry, do.trace=25)
## ----train_inspect, results = 'hide', message=F-------------------------
# inspect
str(rf.cv, max.level = 1)
# object class
class(rf.cv)
# cross validation predictions
str(rf.cv$pred)
head(rf.cv$pred)
# results per fold
rf.cv$resample
# aggregated results
rf.cv$results
## ----finalModel, message=F----------------------------------------------
# MSE
round(rf.cv$finalModel$mse[500], digits = 3)
# RMSE
round(sqrt(rf.cv$finalModel$mse[500]), digits = 3)
# R2 (= AVE for OOB)
round(rf.cv$finalModel$rsq[500], digits = 3)
## ----originalModel, message=F-------------------------------------------
# MSE
round(rf$mse[500], digits = 3)
# RMSE
round(sqrt(rf$mse[500]), digits = 3)
# R2 (= AVE for OOB)
round(rf$rsq[500], digits = 3)
# MSE
round(mean((rf.cv$pred$pred-rf.cv$pred$obs)**2), digits = 3)
# RMSE
round(sqrt(mean((rf.cv$pred$pred-rf.cv$pred$obs)**2)), digits = 3)
# AVE
round(1-(sum((rf.cv$pred$pred-rf.cv$pred$obs)**2)/(sum((rf.cv$pred$obs-mean(rf.cv$pred$obs))**2))), digits = 3)
# copy cross-validation predictions to new data.frame
cv.pred <- rf.cv$pred
```

```
# order
cv.pred <- cv.pred[order(cv.pred$rowIndex, decreasing = FALSE),]
# append to dataset
rm$cv.pred <- cv.pred$pred
# calculate prediction error
rm$pe <- rm$cv.pred-rm$Pb
# save various outputs
save.image("validation.rda")
```