



**UNIVERSITÉ
DE GENÈVE**



GRID
Geneva

Expérimentation d'automatisation du suivi de la couverture du sol

Natacha Klein

Supervisé par: Bruno Chatenoux & Karin Allenbach

Rapport de stage au GRID-Genève
Certificat complémentaire en géomatique
Université de Genève

Juillet 2016

Table des matières

Résumé.....	1
1. Introduction	2
1.1. GRID Genève et LiMES	2
1.2. Objectif du stage	2
2. Détection du changement de la couverture du sol.....	2
3. Données	3
4. Méthodes de détection du changement testées	4
4.1. Comparaison Post-Classification	4
4.1.1. Principe	4
4.1.2. Méthodologie.....	5
4.1.3. Résultats	7
4.2. Analyse en composantes principales (ACP).....	11
4.2.1. Principe	11
4.2.2. Méthodologie.....	11
4.2.3. Résultats	14
4.3. Différences entre bandes spectrales.....	18
4.3.1. Principe	18
4.3.2. Méthodologie.....	18
4.3.3. Résultats	20
4.4. Échantillons aléatoires	23
4.4.1. Principe	23
4.4.2. Méthodologie.....	23
4.4.3. Résultats	26
4.5. Zone d'entraînement totale	29
4.5.1. Principe	29
4.5.2. Méthodologie.....	29
4.5.3. Résultats	31
5. Discussion des méthodes et conclusion.....	34
6. Conclusions personnelles.....	35
7. Bibliographie	36
8. Annexes.....	38

Table des figures

Table 1: Information sur les sites traités	4
Figure 1: Workflow de la méthode 1.....	6
Figure 2: Compositions colorées/Classification scène 1 et scène 2	7
Figure 3: Image de surfaces changées vs. surfaces inchangées.	8
Figure 4 : Compositions colorées/Classification scène 1 et scène 2	10
Figure 5: Workflow de la méthode 2.....	13
Figure 6: Boxplots des ACP des 5 sites	15
Figure 7: Classification scène 1 et scène 2	17
Image de changement et classification finale scène 2 selon méthode ACP	17
Figure 8: Workflow de la méthode 3.....	19
Table 2: Pourcentage de surfaces selon les classes de différences	20
Figure 9: Classification scène 1 et scène 2	22
Classification des différences	22
Classification scène 2 selon la méthode des différences	22
Figure 10: Workflow de la méthode 4.....	25
Figure 11: Les classifications de la scène 2 avec les échantillons aléatoires.....	28
Figure 12: Workflow de la méthode 5.....	30
Figure 13: Compositions colorées en fausses couleurs.....	32
Classification scène 1 et scène 2 (méthode 1)	32
Classification scène 1 et scène 2 (méthode 5)	32
Figure 14: Image de changement selon méthode 1.....	33
Image de changement selon méthode 5	33
Figure 15: Classification scène 1 et scène 2 selon méthode 5	34

Résumé

L'analyse du changement et de la composition de la couverture du sol par images satellites est un sujet qui suscite beaucoup d'intérêt dans le domaine de la recherche scientifique. De multiples méthodes et processus existent suivant le type de changement étudié et l'environnement physique considéré. Les satellites Landsat sont des sources précieuses fournissant continuellement des images de qualité et sans frais à la communauté scientifique (Rawat & Kumar, 2015). Dans une perspective d'automatisation, ce travail se propose de chercher et de tester les processus les plus fiables et stables permettant de 1) détecter automatiquement le changement de la couverture du sol sur une série d'images multi-temporelles de Landsat-8 et de 2) classier automatiquement la couverture du sol d'une série d'images en tenant compte des classifications précédentes. Avec le logiciel GRASS, cinq méthodes ont été testées sur 5 sites comprenant chacun 2 scènes : 1) la comparaison post-classification, 2) l'analyse en composantes principales, 3) la méthode des différences spectrales, 4) la méthode des échantillons aléatoires et 5) la méthode de la zone d'entraînement totale. Toutes les méthodes présentent des inconvénients et limitations empêchant une automatisation totale des processus de traitement et de réaliser des classifications multi-temporelles fiables et stables. La limitation principale semble être que les classifications multi-temporelles tenant compte de classifications précédentes exige que les scènes/signatures spectrales puissent être classifiées de manière identique par rapport à l'ordre et aux catégories des classes. Une exigence qui n'est pas systématiquement atteinte pour chaque site. Ainsi pour des recherches ultérieures, l'alternative pourrait être de combiner plusieurs méthodes de détection du changement afin d'améliorer la validité et la fiabilité des résultats.

1. Introduction

1.1. GRID Genève et LiMES

Le GRID-Genève est une organisation qui fait partie des centres d'information environnementale appartenant au PNUÉ (Programme des Nations-Unies pour l'Environnement). Lancée en 1985, le GRID-Genève a pour objectif de faciliter l'accès à l'information environnementale et de produire des données pour la prise de décisions et la mise en place de politiques (UNEP/DEWA/GRID-Geneva, 2016). Dans la continuité du projet "UNEP atlas of our changing environment", le GRID-Genève est en train de mettre en place un processus de suivi automatique de sites à l'aide d'images satellites. Le projet est nommé LiMES - Live Monitoring of Earth Surface (<http://limes.grid.unep.ch/>) et ne se base que sur des logiciels de SIG et de télédétection open source. L'un des objectifs du projet est de pouvoir suivre la classification de la couverture du sol de sites dans une perspective multi-temporelle. Cependant, bien que les processus de classification automatique d'images satellites existent, ils ne se sont pas adaptés à un contexte d'images multi-temporelles.

1.2. Objectif du stage

Dans le contexte du projet LiMES, le but de ce stage est de chercher et tester les processus les plus fiables et stables permettant de détecter le changement de la couverture du sol dans une série d'images multi-temporelles provenant de Landsat-8. De plus, les méthodes les plus stables recherchées doivent permettre de classer automatiquement la couverture du sol d'une série d'images en tenant compte des classifications précédentes. Ceci, afin de garantir une bonne comparabilité de deux images quelque soit l'intervalle de temps entre leurs prises de vues.

2. Détection du changement de la couverture du sol

La littérature scientifique sur l'étude du changement de la couverture du sol est très vaste. Plusieurs auteurs regroupent les méthodes de détection du changement basées sur l'analyse de pixels en 2 grandes catégories: 1) les méthodes post-classification et 2) les méthodes pré-classification (Lu et al., 2004; Hussain et al., 2013; Yu et al., 2016). La première catégorie signifie que le changement est détecté entre 2 images déjà classifiées de manière indépendante (Fichera et al. 2012). La deuxième catégorie rassemble des méthodes qui identifient le changement en traitant directement les images brutes donc sans être classifiées (Yu et al., 2016). Les méthodes peuvent aussi se différencier par le niveau d'information généré. Les méthodes post-classification fournissent des informations sur le type de changement survenu tandis que les méthodes de traitement pré-classification ne fournissent que de l'information binaire sur le changement ("changement" ou "non-changement") (Lu et al., 2004; Munyati, 2004; Yuan et al., 2005; Fichera et al., 2012; Hussain et al., 2013; Rawat & Kumar, 2015). Les méthodes de détection les plus couramment utilisées dans l'analyse de la couverture du sol sont notamment la comparaison post-classification, la méthode des différences spectrales (intitulée "image

differencing” en anglais) et l’analyse en composantes principales (Lu et al., 2004). Il existe également des techniques de détection du changement basées sur l’analyse d’objets et non plus sur l’analyse de pixels qui n’ont pas été expérimentées dans cette étude pour des raisons de contrainte de temps. Ces techniques plus récentes, dont les méthodes de segmentation, ont pour objectif d’améliorer les classifications qui deviendraient des représentations plus précises du monde réel et qui fourniraient plus d’informations contextuelles (Hussain et al., 2013).

Les caractéristiques et spécificités de la zone étudiée sont des éléments cruciaux à prendre en compte dans le choix de la méthode à utiliser dans la détection du changement. En effet, la qualité et la précision des résultats d’une méthode varient en fonction de l’environnement étudié et de l’objectif de l’analyse (quel type de changement recherché) (Lu et al., 2004; Berberoglu & Akin, 2009). De plus, la littérature mentionne que la qualité des résultats d’une méthode est aussi influencée par la sélection du seuil qui différencie le changement du non-changement (Théau, 2012; Hussain et al., 2013). La définition du seuil de différenciation n’est généralement pas clairement expliquée dans la littérature (Hussain et al., 2013).

Plusieurs études et revues de la littérature se sont intéressées à la démarche des techniques de détection du changement. Le cheminement des techniques est généralement décrit en 3 parties: 1) le pré-traitement, 2) la sélection d’une méthode de détection/classification, 3) l’évaluation de la précision (Lu et al., 2004; Hussain et al., 2013). Ce travail se concentre sur la partie de sélection de différentes méthodes de détection du changement.

3. Données

Plusieurs sites d’études ont été sélectionnés dans le cadre de ce travail. Pour chaque site, deux scènes ont été téléchargées et clippées afin de pouvoir tester le processus de classification et les différentes méthodes de détection du changement de la couverture du sol. Les images brutes ont été corrigées et les nuages masqués lors d’une étape de pré-traitement (Timoner, 2016).

Par conséquent, le téléchargement, le découpage et le masquage des nuages sont effectués aux moyens de 2 scripts (LiMES_batcher_v003.py; LiMES_grasser_v003.py) créés dans le cadre du projet LiMES. Ce projet s’applique uniquement aux images récoltées par Landsat-8, ce qui ne permet de sélectionner que des images entre 2013 et 2016. Toutes les images multi-spectrales sont donc à moyenne résolution (30 mètres). Les tests et analyses de ce travail ont été réalisés principalement via des scripts bash dans GRASS GIS, et dans une moindre mesure sur les logiciels R et Qgis. Les sites, les dates des 2 scènes traitées ainsi que la couverture nuageuse lors de la prise de vue sont présentées dans le tableau ci-dessous (pour plus d’information, voir annexe 1):

Site	Scene 1 Date	Cloud Cover	Scene 2 Date	Cloud Cover
Papua New Guinea	25.1.2015	4%	22.9.2015	0%

Irak	20.9.2015	0%	30.3.2016	0%
Iceland	4.5.2015	40%	25.9.2015	11%
Dubti, Ethiopia	13.12.2014	0%	20.4.2015	0%
Geneva, Switzerland	15.8.2013	1%	5.8.2015	0%

Table 1: Information sur les sites traités

4. Méthodes de détection du changement testées

Dans le cadre de ce stage, 5 méthodes de détection du changement et de classification de la couverture du sol ont été testées:

1. Comparaison post-classification
2. Analyse en composantes principales (ACP)
3. Différences entre bandes spectrales
4. Échantillons aléatoires
5. Zone d'entraînement totale

Certaines de ces méthodes sont inspirées de la littérature scientifique et d'autres ont été imaginées et élaborées selon une logique autonome en vue des résultats souhaités. Dans la partie qui suit, chaque méthode est présentée par un chapitre subdivisé lui-même en trois sous-parties à chaque fois: 1) le principe de la méthode à tester, 2) le processus effectué pour arriver aux résultats, 3) la présentation et discussion des résultats.

4.1. Comparaison Post-Classification

4.1.1. Principe

Cette méthode consiste à examiner/suivre le changement de la couverture du sol après avoir effectué 2 classifications non-supervisées des 2 scènes multi-temporelles indépendamment l'une de l'autre. Il s'agit de la seule méthode permettant de connaître la nature de changement (Fichera, Modica & Pollino, 2012). Le processus d'une classification non-supervisée requiert un minimum de connaissance de terrain de la part de l'utilisateur étant donné qu'il faut identifier le nombre de classes approprié au site ainsi que les catégories à attribuer à ces différentes classes. Par conséquent, la précision des résultats de détection du changement va dépendre de la qualité des 2 classifications distinctes (Lu et al., 2004; Yuan et al., 2005; Deng et al., 2009; Hussain et al., 2013).

4.1.2. Méthodologie

Afin d'arriver à une classification non-supervisée dans GRASS, il faut dans un premier temps créer 2 groupes contenant toutes les bandes spectrales de chaque scène (i.group) Dans le cas de Landsat-8, les bandes spectrales à intégrer dans le processus de classification vont de la bande 2 à la bande 7. Ces 2 groupes servent ensuite à créer 2 fichiers de signatures spectrales (i.cluster) Chaque fichier va servir ainsi à classer chaque pixel selon leur ressemblance avec la signature spectrale du fichier (i.maxlik). L'utilisateur doit également préciser le nombre de classes selon lesquelles doivent se répartir les signatures spectrales. Finalement, étant donné que l'utilisateur a connaissance du nombre de classes, il peut aussi y préciser la catégorie/l'étiquette de chacune d'elles en ajoutant un fichier de catégories (r.category).

Étant donné que l'objectif du travail est de détecter le changement entre les 2 scènes, une image identifiant les zones de changement de couverture du sol et les zones de non-changement a été créée (r.mapcalc).

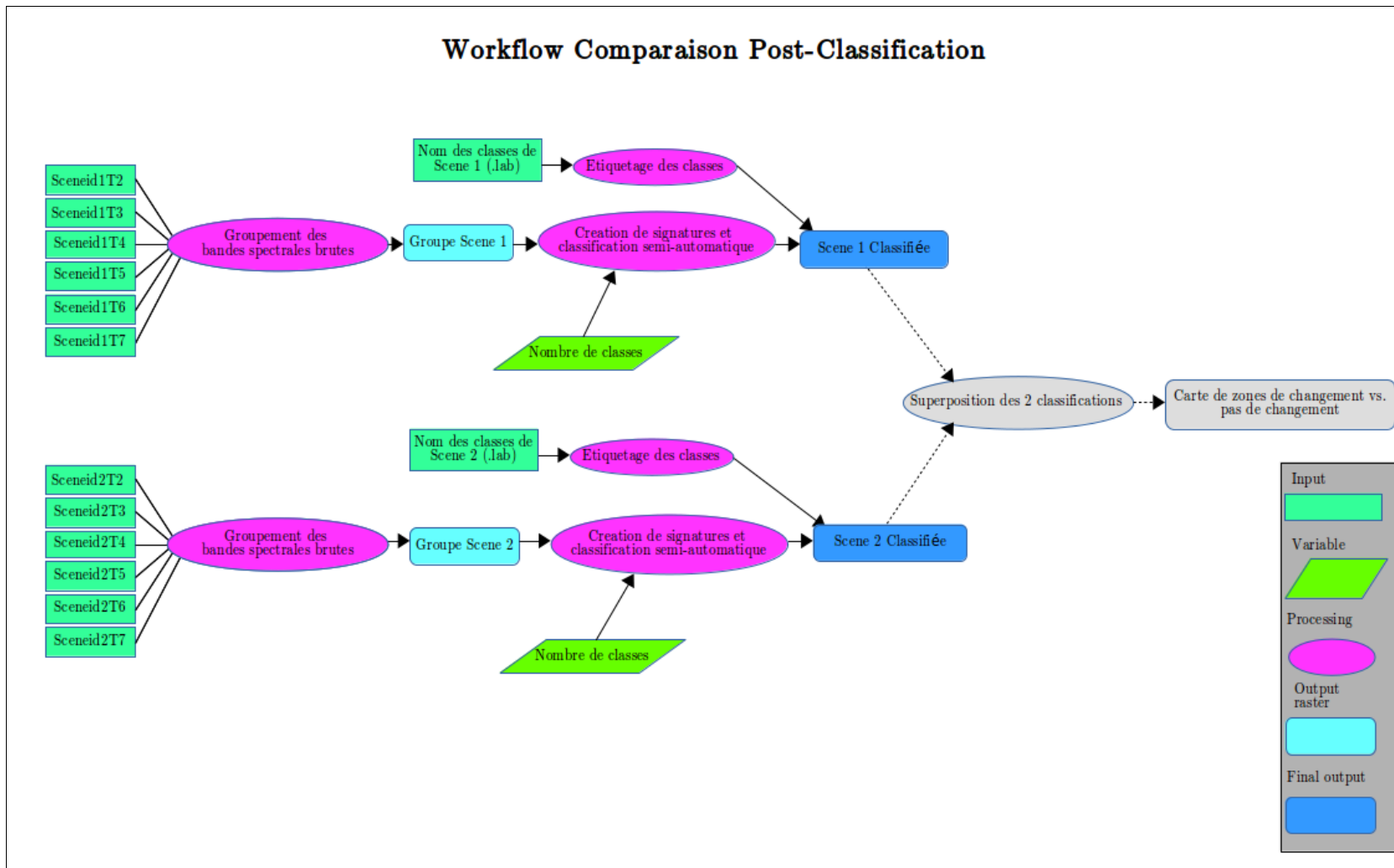


Figure 1: Workflow de la méthode 1

4.1.3. Résultats

Afin de pouvoir comparer et juger de la fiabilité de la méthode, une composition en fausse couleur de chaque scène a été créée pour l'exemple du site en Papouasie Nouvelle-Guinée à la figure 2. Par conséquent, en examinant les 4 images ci-dessous, le résultat de cette méthode semble être fiable. Les changements majeurs de couverture du sol semblent se retrouver dans les classifications et dans la figure 3, l'image présentant les surfaces qui ont changées et pas changées.

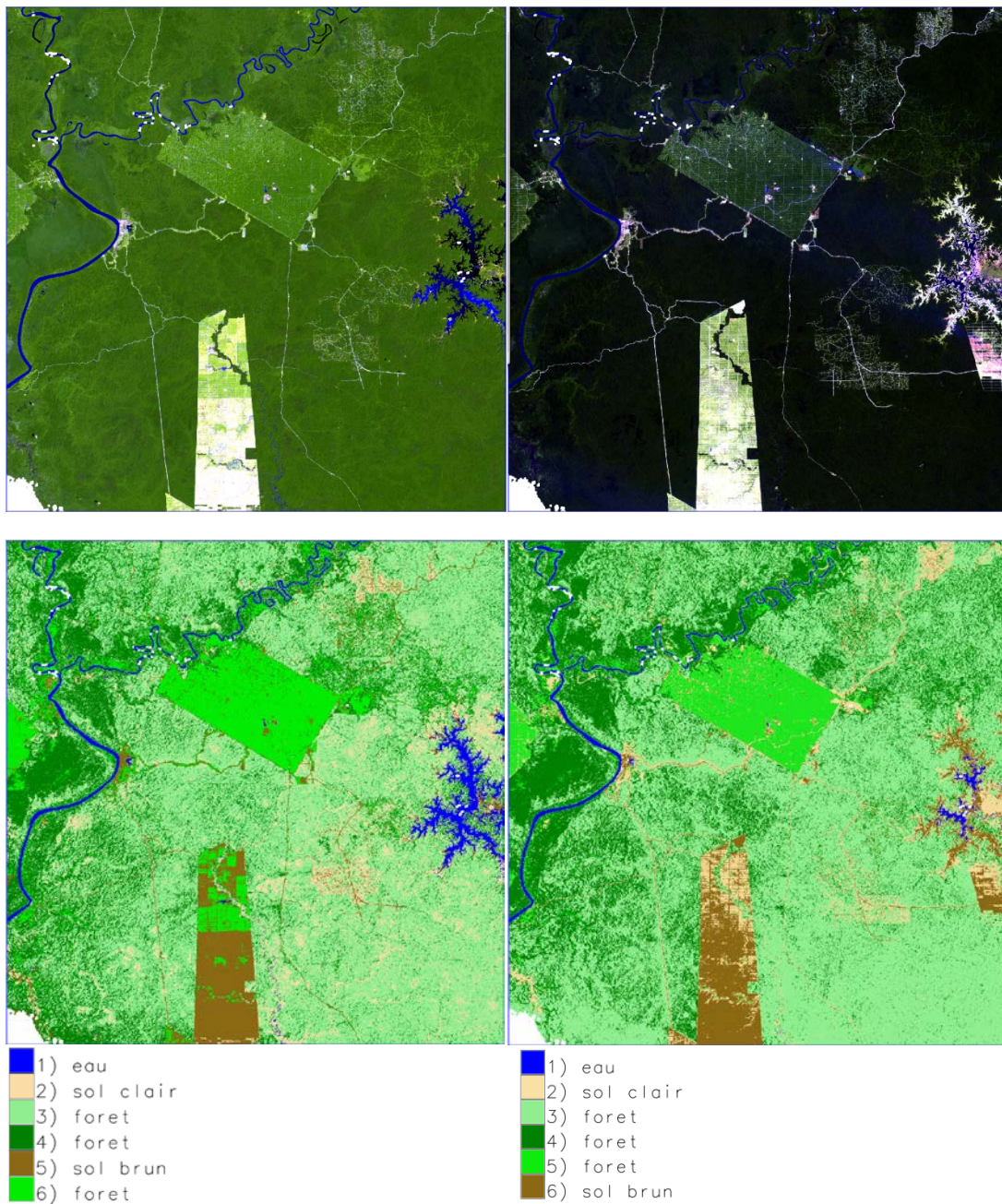


Figure 2: Compositions colorées en fausses couleurs (764) (+ i.colors.enhance)

Classification scène 1 et classification scène 2

Site: Papua (25.1.2015 et 22.9.2015)

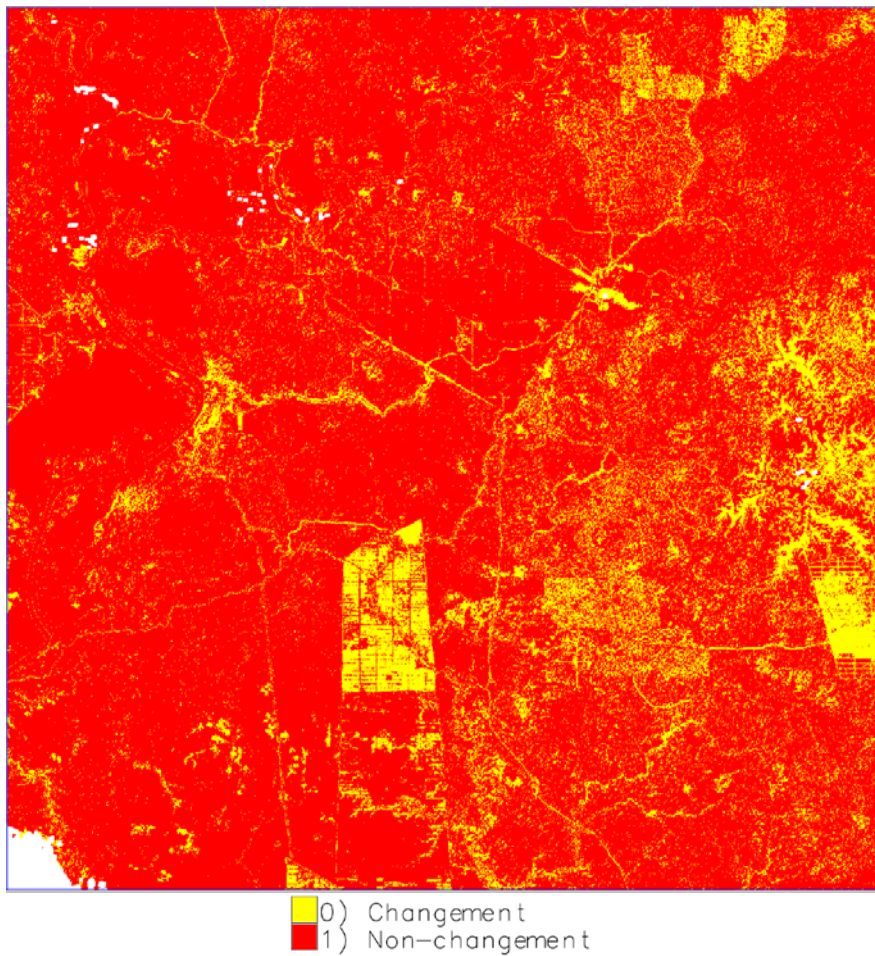


Figure 3: Image de surfaces changées vs. surfaces inchangées.

Site: Papua (25.1.2015 et 22.9.2015)

L'avantage de cette méthode est qu'elle est composée de peu d'étapes de traitement et s'effectue ainsi en très peu de temps. La comparaison post-classification a également l'avantage de fonctionner tant sur des sites qui contiennent une minorité de changement dans le temps, tant sur des sites caractérisés par une majorité de changement entre 2 scènes.

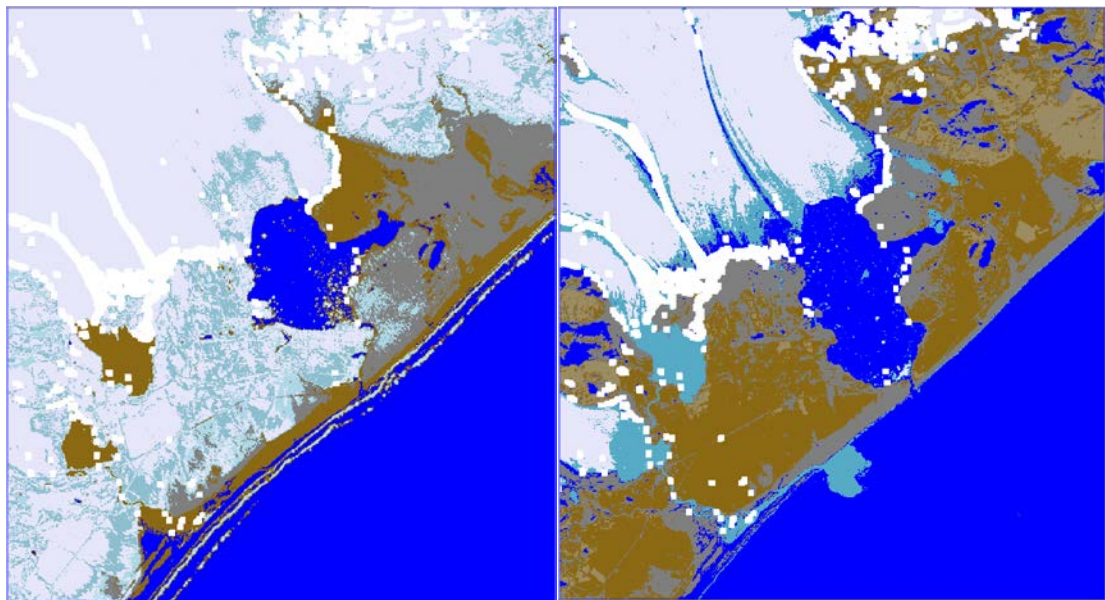
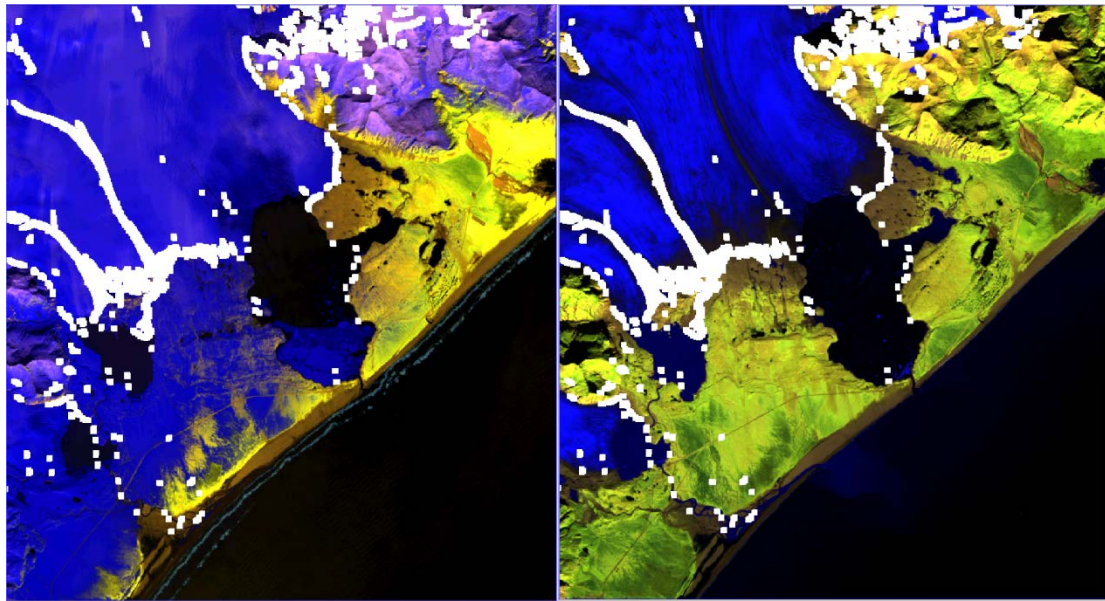
Cependant, cette méthode requiert que l'utilisateur sache combien de classes il souhaite et lesquelles. Cela ne permet donc pas une automatisation totale du processus du suivi de la couverture du sol. En effet, il faut examiner manuellement quelle classe correspond à quelle catégorie et comme on peut l'observer, les classifications ne sont pas exactement les mêmes entre les scènes et requiert ainsi l'intervention de l'utilisateur.

L'inconvénient également de cette méthode est que la détection du changement entre les 2

scènes s'effectue après avoir classé la deuxième scène. Ainsi, la deuxième scène classifiée ne tient pas compte de la première scène.

Pour le début des processus de détection de changement, la classification non-supervisée est le traitement qui va systématiquement être utilisé pour classer la première scène étant donné qu'il n'existe pas de référence au départ. Par conséquent, la classification de la première scène servira de référence dans l'identification du changement dans les prochaines méthodes testées.

Dans le cas de classification non-supervisée de la couverture du sol, il est important de souligner le fait que les résultats (images classifiées) doivent être évalués avec beaucoup de précaution. Non seulement, l'objectif est d'arriver à une classification la plus détaillée possible de façon à ce qu'elle soit la plus semblable à la réalité mais, dans le contexte de ce travail de recherche, la classification doit également être simplifiée au maximum pour permettre le bon fonctionnement de chaque méthode à tester. Par conséquent, un site comme Island qui révèle avoir 2 classifications très différentes entre ses 2 scènes à la figure 4, ne donne pas de résultat précis en correspondance à la réalité mais la méthode montre tout de même la composition du sol et son changement de manière grossière et schématique.



- 1) eau
- 2) sol
- 3) sol
- 4) glace fonce
- 5) glace fonce
- 6) glace fonce
- 7) neige/glace
- 8) neige/glace

- 1) eau
- 2) eau
- 3) glace fonce
- 4) sol
- 5) sol
- 6) sol
- 7) neige/glace
- 8) neige/glace

Figure 4 : Compositions colorées en fausses couleurs (764) (+ i.colors.enhance)

Classification scène 1 et classification scène 2 (de gauche à droite)

Site: Iceland (4.5.2015 et 25.9.2015)

Il peut en effet être observé quelques erreurs dans la classification de la deuxième scène notamment dans le fait que l'ombre des montagnes (en haut à droite et à gauche de l'image) soit classifié en bleu comme étant des surfaces d'eau. Par conséquent, tout au long de ce travail, bien que ces classifications servent de base pour l'évaluation des autres résultats, il est important de reconnaître le manque de précision potentiel dans certaines images de couverture du sol.

4.2. Analyse en composantes principales (ACP)

4.2.1. Principe

L'analyse en composantes principales est une méthode très utilisée dans le domaine de la détection du changement de la couverture du sol. Cependant, la littérature est unanime sur le fait que les résultats d'une ACP sont difficile à interpréter (Gong, 1993; Lu et al., 2004; Hussain et al., 2013). Dans le cadre de l'étude du changement de la couverture du sol, l'hypothèse du résultat d'une analyse en composantes principales est que les pixels inchangés sont affichés dans les premières composantes tandis que les pixels qui ont changés sont affichés dans les dernières composantes (Fuchs, Hoffmann, & Schwonke, 2008; Chisanga, 2012; Théau, 2012).

De plus, en suivant ce principe de fonctionnement de l'ACP pour le suivi de la couverture du sol, l'hypothèse suivante peut être émise: plus il y a du changement entre les scènes, plus cette information (les valeurs des composantes) sera clairement distinguée dans les premières composantes.

4.2.2. Méthodologie

La première étape est de faire une analyse en composantes principales incluant les 6 bandes spectrales des 2 scènes (12 bandes au total) (i.pca). En sortie de ce traitement, les différentes composantes principales contenant un maximum d'information des 12 bandes sont alors créées. Étant donné qu'il est très difficile d'interpréter quelles valeurs des composantes principales signifient du changement et quelles valeurs signifient de la surface inchangée, des boxplots ont été créés en séparant les valeurs en 2 groupes selon les catégories de changement/pas de changement (0 et 1) de la première méthode (figure 6). Pour ce faire, les rasters de l'ACP et l'image des changements ont directement été traités dans le logiciel R pour produire les boxplots. Il a donc été nécessaire d'installer le logiciel R ainsi que les libraires *sp*, *raster*, *rgdal*, *rgrass7* (voir Annexe 2).

En outre, pour identifier manuellement la valeur des pixels changés et inchangés, les couches rasters ont été exportées (*r.out.dgal*) sur Qgis et visualisées avec l'outil Raster Transparency.

Après avoir examiné les valeurs de changement et les boxplots des trois premières composantes pour chaque site, la méthode requiert l'extraction du fichier statistique des composantes où se trouvent les valeurs des pixels correspondant à du non-changement (*r.univar*). Avec ces valeurs extraites dans une nouvelle couche, il s'agit ensuite d'utiliser les surfaces de changement comme

masque sur la(es) composante(s) principale(s) qui contient les valeurs de non-changement (r.mapcalc/r.mask). Finalement, la dernière étape est ainsi d'utiliser la première scène classifiée et partiellement masquée par la(es) composante(s) principale(s), comme zone d'entraînement pour la classification de la deuxième scène (i.gensig/i.maxlik). Il est important de noter qu'étant donné que les signatures spectrales de la classification finale sont créées sur la base de zones d'entraînement de la classification de la première scène, les catégories et symbologie correspondante de cette première classification sont à appliquer également sur la classification finale produite par la méthode ACP (r.category).

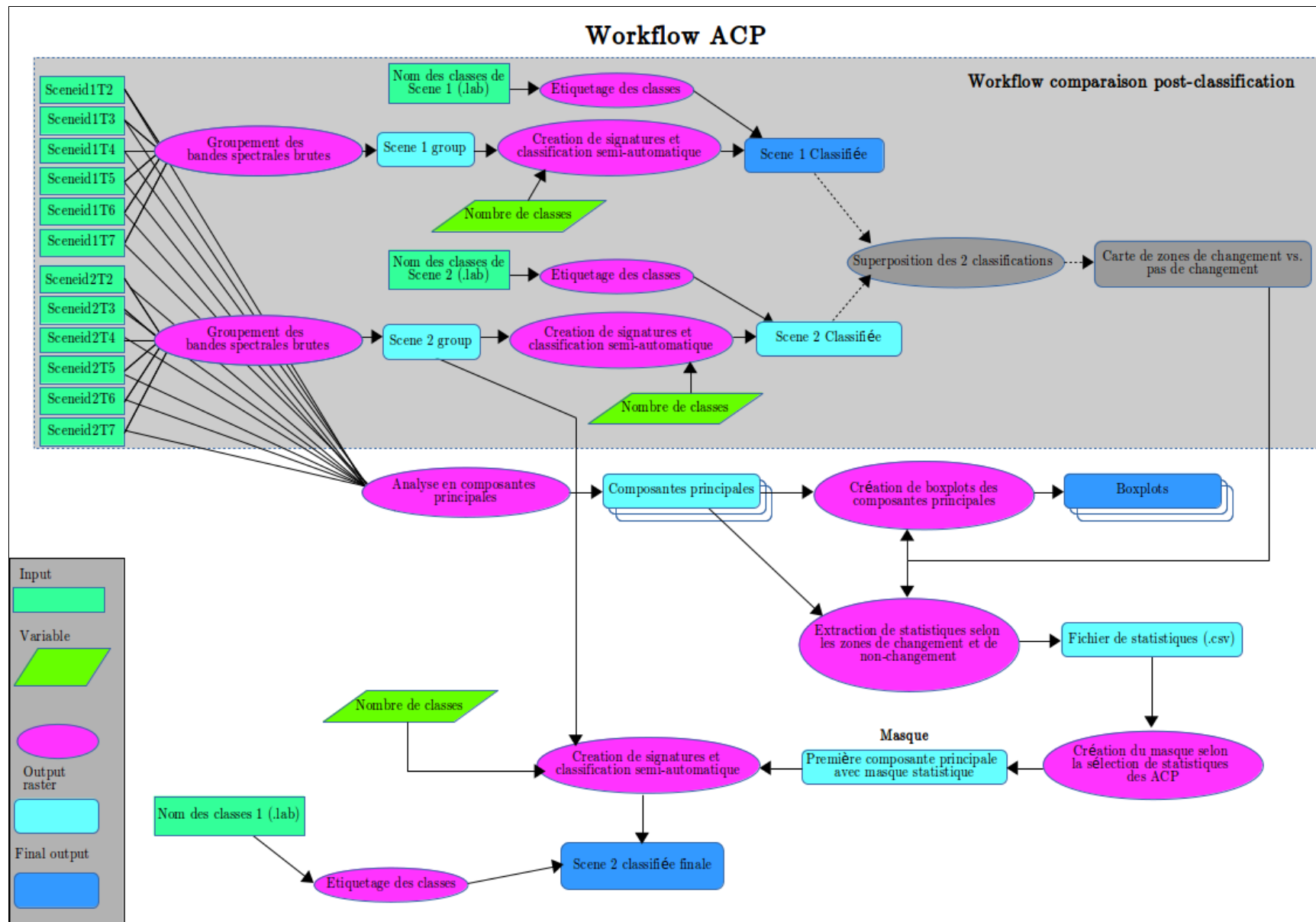


Figure 5: Workflow de la méthode 2

4.2.3. Résultats

Dans les boxplots (figure 6), la distribution des valeurs des trois premières composantes dans les zones inchangées se révèle être très différente d'un site à l'autre et semble ne pas se distinguer de la distribution des valeurs dans les surfaces de changement. De plus, si les valeurs réparties dans les 2 catégories ne sont pas clairement distinguées, alors il n'y a aucun moyen de pouvoir différencier les zones de changement des zones de non-changement dans les composantes principales, rendant ainsi la méthode non-applicable.

Théoriquement, selon l'hypothèse émise plus haut, la méthode ACP ne serait cohérente que lorsque la majorité du site change entre les 2 scènes et ainsi les valeurs de non-changement seraient identifiables dans les premières composantes. Or pour les sites contenant une majorité de changement entre les scènes (Iceland, 62% et Irak, 59%), les boxplots montrent des valeurs qui se chevauchent entre les 2 catégories et donc celles-ci ne se distinguent pas clairement, ce qui conduit à rejeter l'hypothèse émise précédemment. Et s'il s'agit d'un site où il y a peu de changements alors la première composante peut contenir d'autres informations, il est alors impossible d'établir un seuil de différenciation entre les pixels de changement et de non-changement.

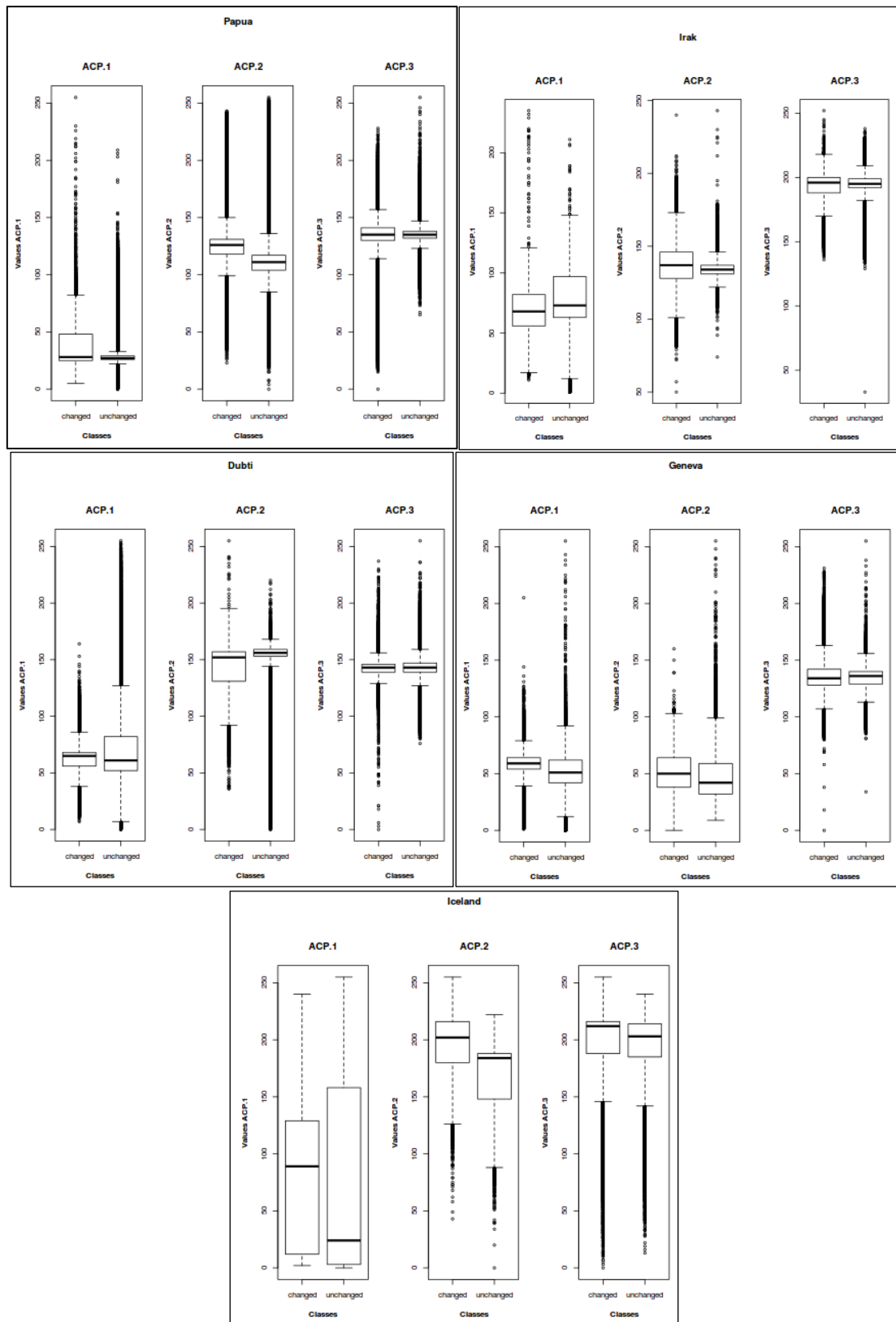


Figure 6: Boxplots des ACP des 5 sites

Le deuxième défaut majeur de cette méthode est que, si la distribution des valeurs d'ACP change

d'un site à un autre, il est alors impossible de déterminer de manière statistique, et donc automatique, quelles valeurs sélectionner ou quel seuil établir pour créer le masque nécessaire pour la deuxième classification pour plusieurs sites.

Cependant, pour le site Papua, un test a été effectué pour identifier manuellement quelles valeurs des composantes principales correspondraient aux zones de non-changement et voir à quelle statistique ces intervalles pourraient correspondre. Les valeurs de non-changement ont ainsi été identifiées comme allant de 0 jusqu'au troisième quartile de la première composante principale pour le site Papua. La figure 7 présente le résultat de la méthode ACP pour ce site-là (image en bas à gauche). Précisément, cet exemple montre que le problème de la méthode ACP est que les valeurs de non-changement sont relatives au site et qu'il n'est pas pertinent d'utiliser cet intervalle de sélection pour le traitement d'autres sites. L'image en bas à droite dans la figure 7 montre le résultat lorsque ces valeurs sont utilisées comme masque pour la classification de la deuxième scène. En comparant avec une classification non-supervisée (image en haut à droite), il peut être affirmé qu'en effet la méthode ACP n'est pas une méthode qui donne un résultat en correspondance avec la réalité notamment avec les surfaces déforestées classifiées en eau.

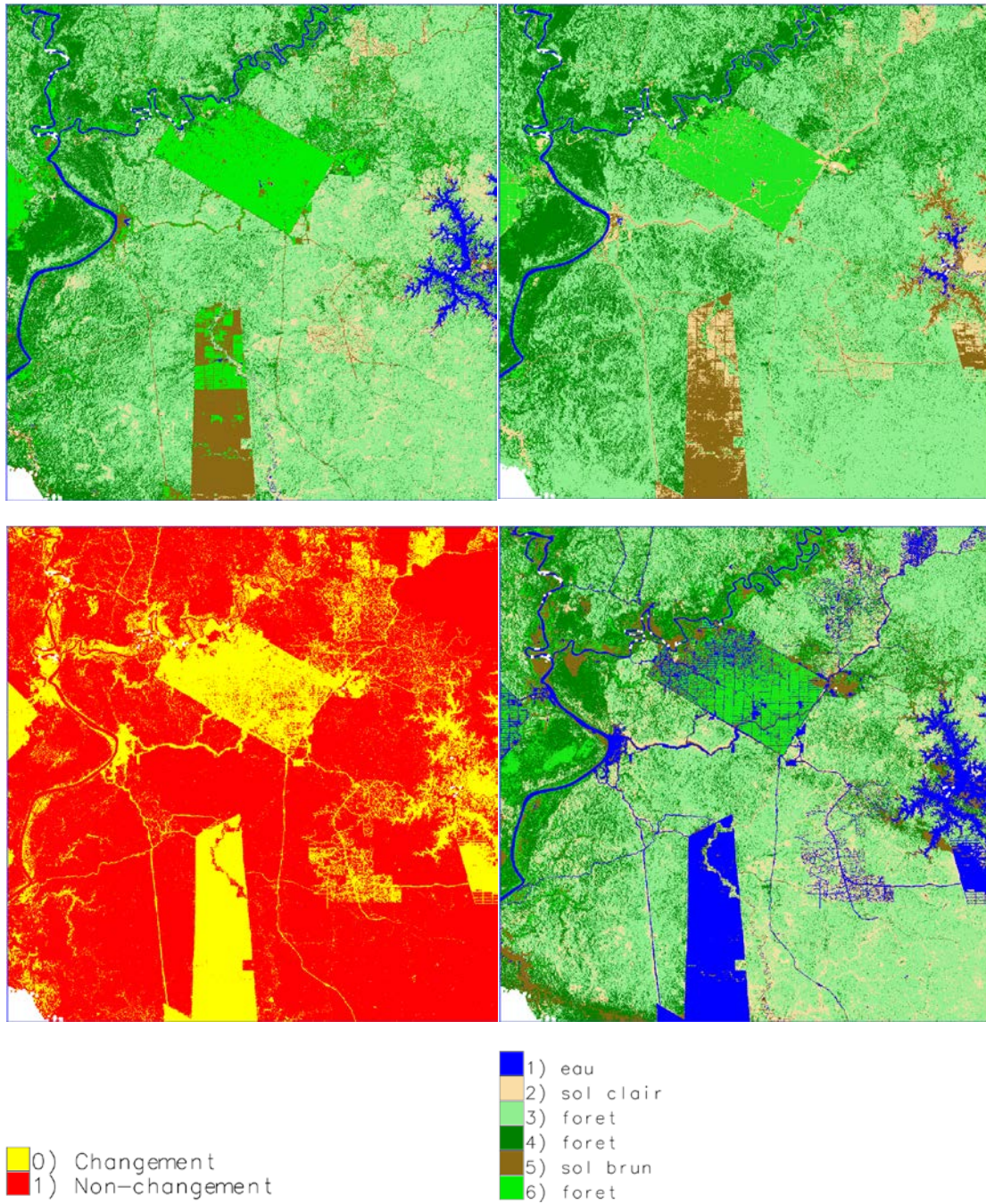


Figure 7: Classification scène 1 et classification scène 2

Image de changement vs. non-changement et classification finale scène 2 selon méthode ACP

Site: Papua (25.1.2015 et 22.9.2015)

Par conséquent, avec les observations ci-dessus, la méthode utilisant une ACP pour identifier le changement de la couverture du sol n'est pas une méthode fiable et stable dans le cas du projet LIMES.

4.3. Différences entre bandes spectrales

4.3.1. Principe

Cette troisième méthode a été choisie car elle est l'une des méthodes les plus couramment utilisées dans la littérature (Lu et al., 2004; Berberoglu & Akin, 2009). Dans ce travail, le processus de traitement se base sur l'idée d'une classification des différences entre les bandes spectrales respectives de chaque scène pour identifier le changement de la couverture du sol nécessaire à la réalisation de la classification de la deuxième scène.

4.3.2. Méthodologie

Comme à chaque début de traitement, la classification non-supervisée de la première scène a été effectuée comme référence et produit de sortie final pour la première scène.

Dans un deuxième temps, une boucle a été créée pour soustraire chaque bande spectrale de la deuxième scène à la bande spectrale correspondante de la première scène (`r.mapcalc`). Puis ces 6 couches contenant les valeurs de différences ont été groupées (`i.group`) pour créer un fichier de signatures spectrales (`i.cluster`) et une classification non-supervisée basée sur ces signatures (`i.maxlik`).

Afin de pouvoir estimer à combien de classes la classification des différences allait se faire, plusieurs tests ont été effectués sur plusieurs sites. Ainsi des classifications à 4,5,6,7 et parfois 8 classes ont été réalisées et exportées (`r.out.gdal`) pour ensuite identifier, grâce à l'outil Raster Transparency sur Qgis, si les classes établies correspondent clairement à des changements ou à des non-changements ou si les classes contiennent des pixels des 2 catégories mélangées.

Le résultat de la classification des différences spectrales est ainsi une image qui identifie, en plusieurs classes, les surfaces de changement et les surfaces qui n'ont pas changé. Cette classification a été utilisée dans la suite du processus de traitement comme masque sur les zones changées (`r.mask`).

Finalement, la dernière étape a été d'utiliser la première scène classifiée et partiellement masquée par la classification des différences, comme zone d'entraînement pour la classification de la deuxième scène (`i.gensig/i.maxlik`). Les étiquettes et symbologie de chaque catégorie de la classification ont été également ajoutées en correspondance avec la première classification (`r.category`).

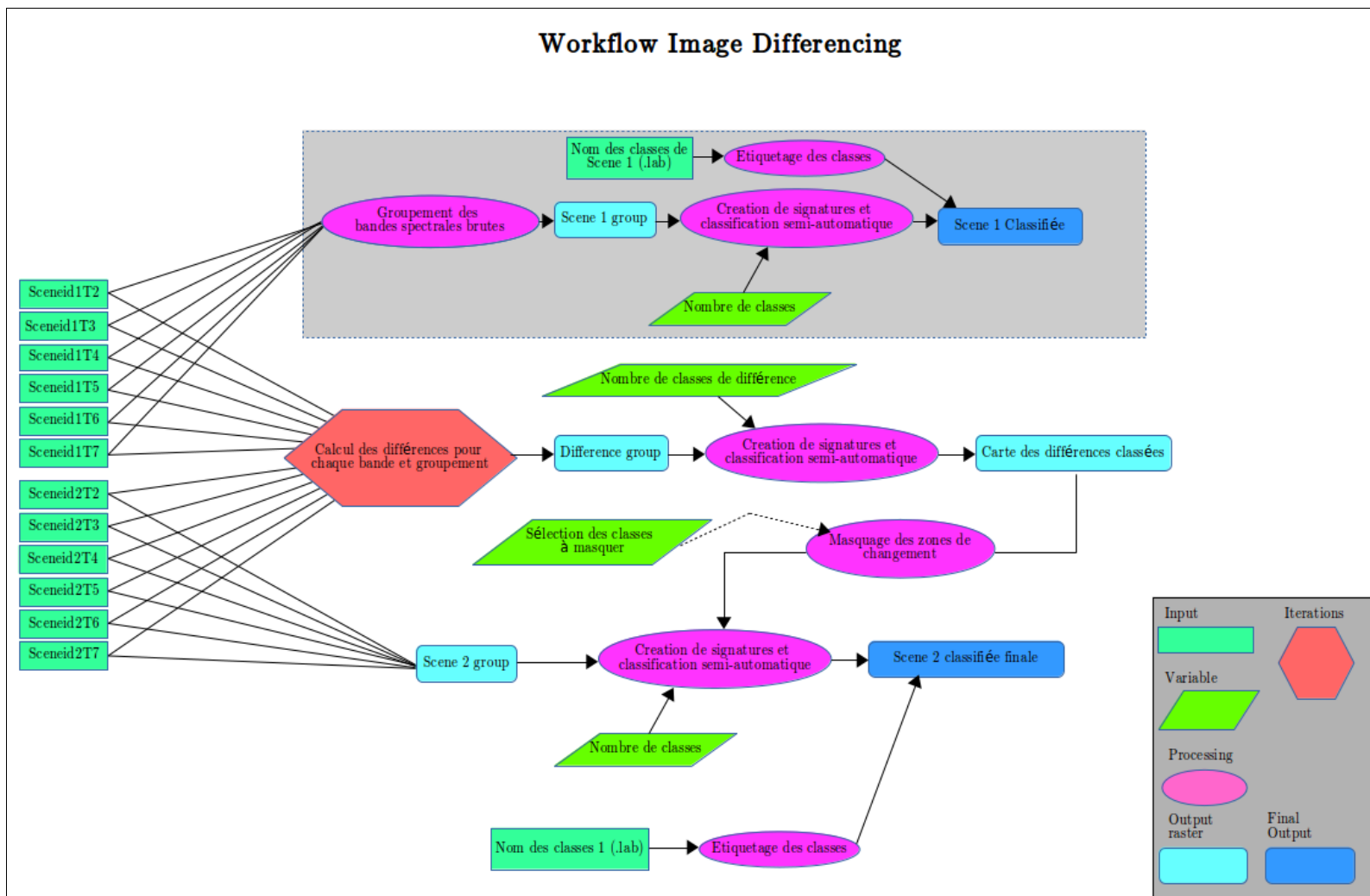


Figure 8: Workflow de la méthode 3

4.3.3. Résultats

Lors des essais de classification des différences entre les bandes spectrales des 2 scènes sur les différents sites, il semble que les classifications les plus adéquates sont des classifications à 4 ou 5 classes. Le choix du nombre de classes semble se faire en fonction de la quantité de changement entre les 2 scènes. En effet, l'enjeu du choix du nombre de classes est qu'il ne faut:

- Ni sous-estimer la proportion de zones changées à masquer car le risque est d'avoir des surfaces de changement identifiées comme étant inchangées et ainsi risquer d'intégrer ces surfaces dans la zone d'entraînement, ce qui influence la création des signatures spectrales finales. Donc si une image contient une minorité de changement, il faut éviter un trop grand nombre de classes.
- Ni surestimer le masque sur les zones changées (éviter de surestimer la proportion de changement) pour intégrer tous les types de couverture du sol dans le fichier de signatures. Donc si une image contient une majorité de changement, il faut éviter un trop petit nombre de classes,

Le deuxième constat est qu'il semble que la distribution des classes de changement dépend du degré et de la direction (négative ou positive) des différences. Ainsi, pour tous les sites analysés, les classes de changement ne sont pas les mêmes. Le tableau ci-dessous présente un résumé des classifications des différences des sites traitées et la répartition des pourcentages des surfaces jugées comme étant soit du changement, soit du non-changement (surligné en jaune):

Site	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	No data
Papua	3.49%	37.55%	48.68%	7.64%	2.07%	0.56%
Dubti	8.67%	51.63%	31.96%	7.68%	-	0.07%
Geneva	13.88%	26.65%	50.58%	7.08%	-	1.81%
Geneva	6.79%	19.52%	37.55%	28.21%	6.12%	1.81%
Iceland	6.34%	19.19%	18.46%	9.48%	39.69%	6.84%
Irak	6.93%	18.39%	27.76%	26.89%	4.68%	15.35%

Table 2: Pourcentage de surfaces (changées ou inchangées) selon les classes de différences (= % de surface inchangée)

Par conséquent, en observant la table 2, une possible hypothèse de règle pour la création du

masque serait de sélectionner les deux classes contenant le plus grand pourcentage de surface et de les catégoriser comme surfaces inchangées. Cependant, cette hypothèse n'est pas valable pour les sites Iceland et Geneva (4 classes). Par conséquent, l'hypothèse émise n'est pas valable. De plus, l'inconvénient dans l'utilisation de cette méthode dans le projet LiMES est qu'il faut identifier manuellement quelles classes correspondent à du changement et celle qui correspondent à du non-changement, empêchant ainsi une automatisation totale du processus de traitement/classification multi-temporelle.

Cependant, avec une classification des différences étudiée et une identification minutieuse des surfaces de changement et de non-changement, le résultat final de cette troisième méthode testée dans ce travail peut sembler cohérent. Par exemple, pour le site Papua, la figure 9 présente une comparaison entre les résultats de classification des scènes indépendamment l'une de l'autre et le résultat de classification de la deuxième scène basées sur les différences spectrales ainsi que l'image des différences classifiée.

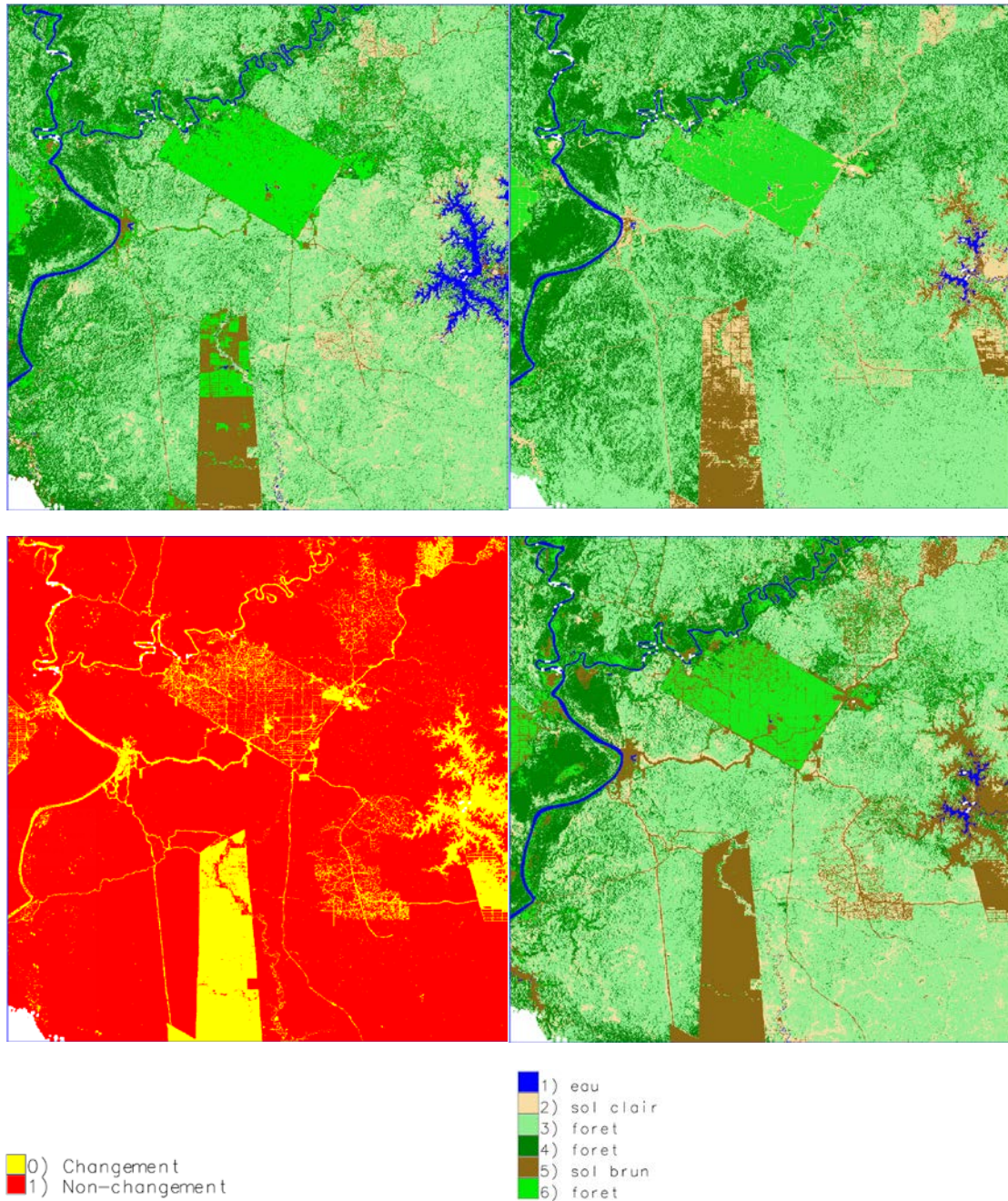


Figure 9: Classification scène 1 et classification scène 2

Classification des différences et classification finale scène 2 selon la méthode des différences

Site: Papua (25.1.2015 et 22.9.2015)

Le site Papua présente une minorité de surfaces changées ainsi, dans cette méthode, le masque (surface jaune dans l'image en bas à gauche) tend à réduire la proportion de surfaces inchangées calculée lors de la méthode de comparaison post-classification (voir figure 3).

La deuxième scène classifiée avec le masquage établi selon les différences spectrales (image en bas à droite) montre une couverture du sol en cohérence avec le résultat de la classification

“basique” (image en haut à droite). Néanmoins, il y a des différences non-négligeables notamment la proportion de sol brun devenue plus grande avec la méthode des différences. Ainsi, il peut être conclu que construire le masque des zones de changement sur la base d’une classification des différences spectrales a des fortes chances de produire une classification de la deuxième scène qui soit simplifiée, faussée et par conséquent ne correspondant pas de manière satisfaisante à la réalité. (Voir annexe 3 pour les résultats de chaque site)

En somme, il est difficile d’automatiser le suivi du changement de la couverture du sol par cette méthode des différences essentiellement car la catégorisation de la classification va dépendre du degré et de la direction des différences et par conséquent, la distribution des catégories change d’un site à un autre. De plus, une automatisation complète de la méthode est aussi impossible car le processus doit être interrompu pour que l’utilisateur puisse identifier les classes correspondant à du changement ou à du non-changement et ainsi entrer les nombres de classes et les classes à masquer en variables.

4.4. Échantillons aléatoires

4.4.1. Principe

Cette quatrième méthode est basée sur l’idée de créer des échantillons de la classification de la première scène récoltés de manière aléatoire pour les utiliser comme zones d’entraînement résultant en plusieurs classifications de la deuxième scène. Puis, en superposant toutes les classifications de la deuxième scène, il serait possible de créer une image montrant les surfaces de changement et de non-changement. Finalement, l’image créée servira à masquer les zones changées et faire une classification finale de la deuxième scène en prenant les surfaces inchangées comme zones d’entraînement. Cette méthode du suivi de la couverture du sol présente ainsi un processus de traitement relativement lourd avec plusieurs étapes intermédiaires dont un passage en mode vecteur.

4.4.2. Méthodologie

La première étape de cette méthode des échantillons aléatoires a été, comme les méthodes précédentes, de faire une classification non-supervisée de la première scène (i.group/i.cluster/i.maxlik). Étant donné que cette première classification va servir de zone d’entraînement pour la classification finale, la classification a été reclassée selon la majorité (r.mode) en exigeant une taille minimum de surface (3 pixels) pour les pixels d’une même classe afin d’éliminer les surfaces trop petites (r.reclass.area).

Le deuxième objectif était alors de créer des échantillons sur la base de cette première classification. Cependant le traitement qui crée des échantillons de manière aléatoire n’est disponible qu’en mode vecteur. Par conséquent, il a fallu procéder à une segmentation de la

première scène classifiée (i.segment). Pour ce faire, il a fallu ajuster des paramètres tels que la taille minimum des objets et le seuil de différenciation pour la création d'objets. Après de multiples tests de segmentation sur plusieurs sites, le choix d'une segmentation relativement fine a été adopté pour prendre en compte un maximum de la structure de la couverture du sol.

Les éléments créés dans la couche segmentée en sortie ne contenant pas les valeurs de la classification, il a ensuite fallu les récupérer de la classification de la première scène (r.mode). Puis, il a fallu vectoriser la couche segmentée (r.to.vect) pour établir une boucle qui crée plusieurs couches d'échantillons aléatoires (v.extract/v.select).

Ces multiples couches d'échantillons ont été utilisées comme zones d'entraînement pour faire des classifications de la deuxième scène par une boucle de traitement (i.gensig/i.maxlik). C'est donc à cette étape qu'il a fallu repasser en mode raster (v.to.rast). Pour avoir des classifications comparables à la première scène classifiée, il a fallu reclasser et grouper les pixels de manière similaire (r.reclass/r.mode). Les classifications de la deuxième scène ont été superposées les unes aux autres pour créer l'image qui va détecter les surfaces de changement et de non-changement (figure 11). Pour ce faire, il a fallu faire appel à la fonction *r.series* qui utilise dans ce cas-ci la méthode *range* (max-min) pour identifier les zones de non-changement (égal à 0) et de changement (pas égal à 0). Donc ensuite, il a fallu recalculer les valeurs des pixels de façon à ce qu'elles ne soient comprises qu'entre 0-1, correspondant ainsi aux 2 catégories de changement et non-changement (r.mapcalc). Dans la suite du processus, la couche de sortie a été utilisée comme masque sur les zones changées (r.mask).

Finalement, la dernière étape a été d'utiliser la première scène classifiée masquée des zones changées comme zone d'entraînement pour la classification de la deuxième scène (i.gensig/i.maxlik). Les étiquettes et symbologie de chaque catégorie de la classification ont été également ajoutées en correspondance avec la première classification (r.category).

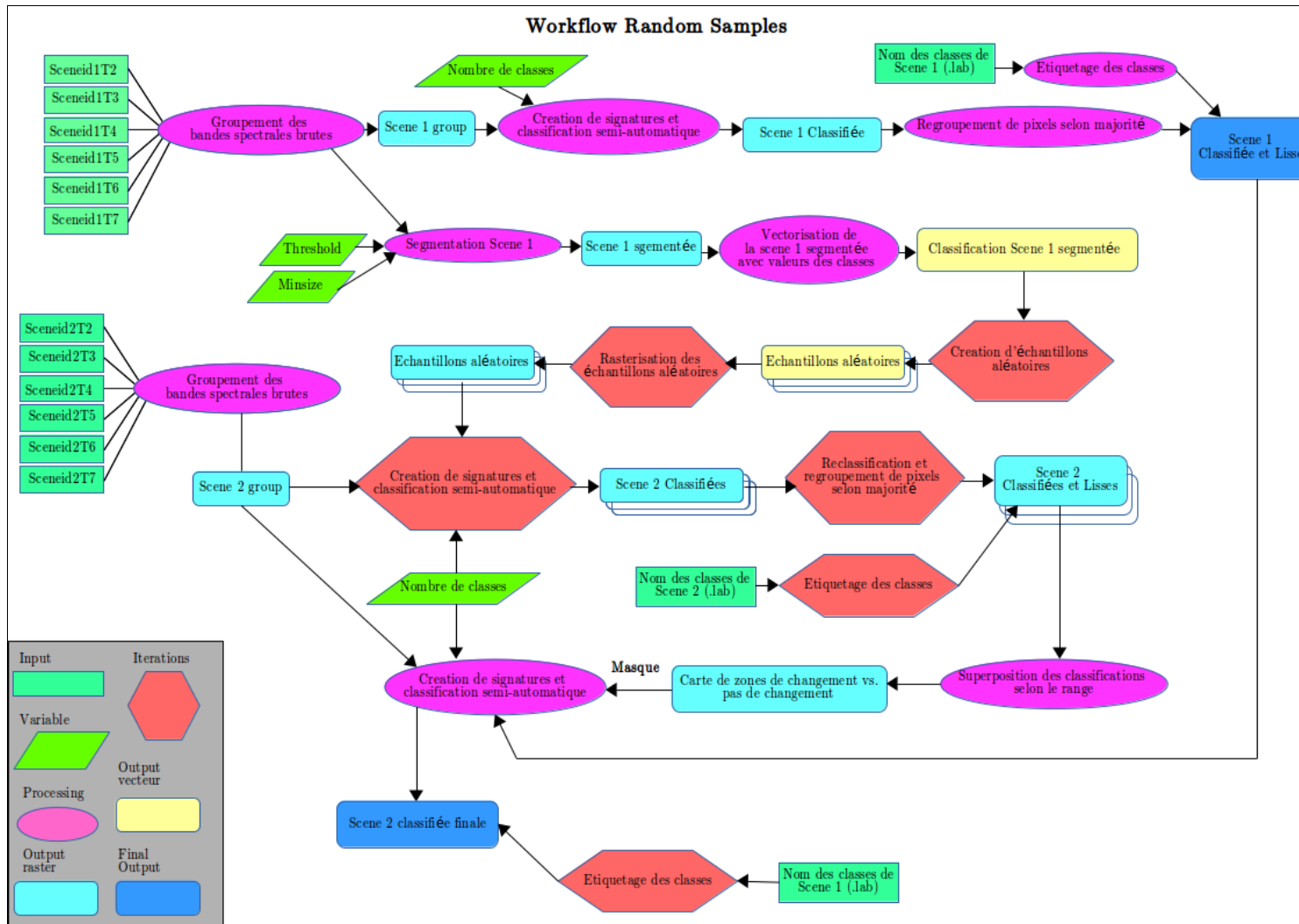
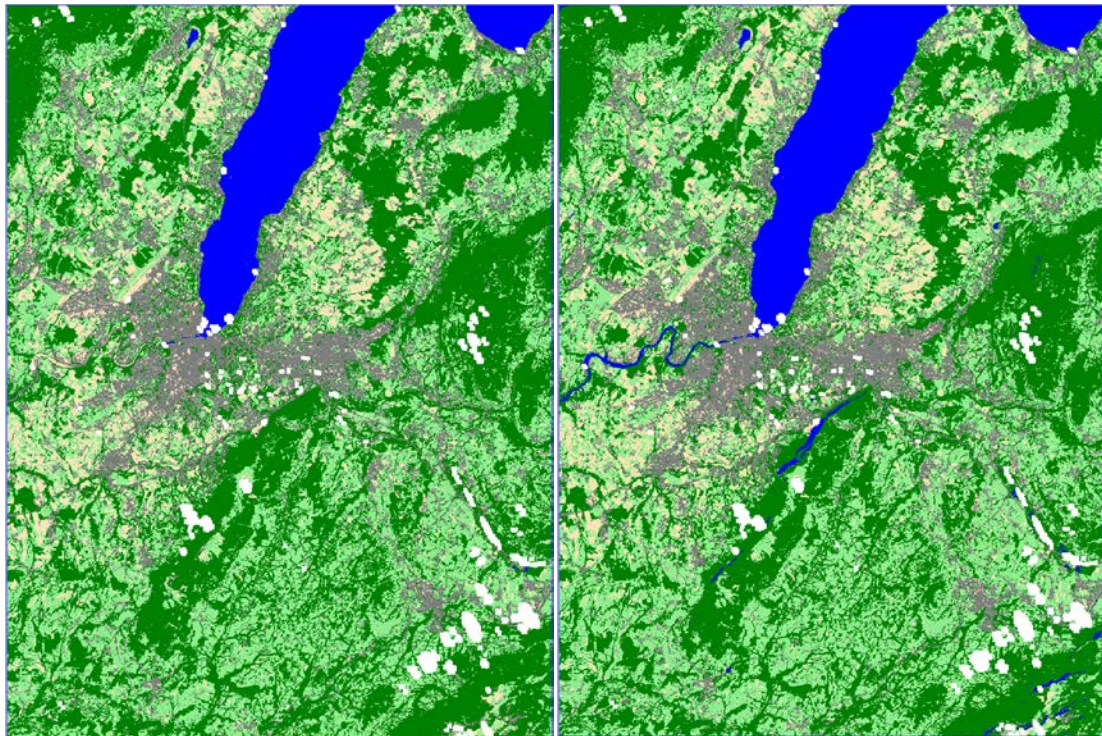


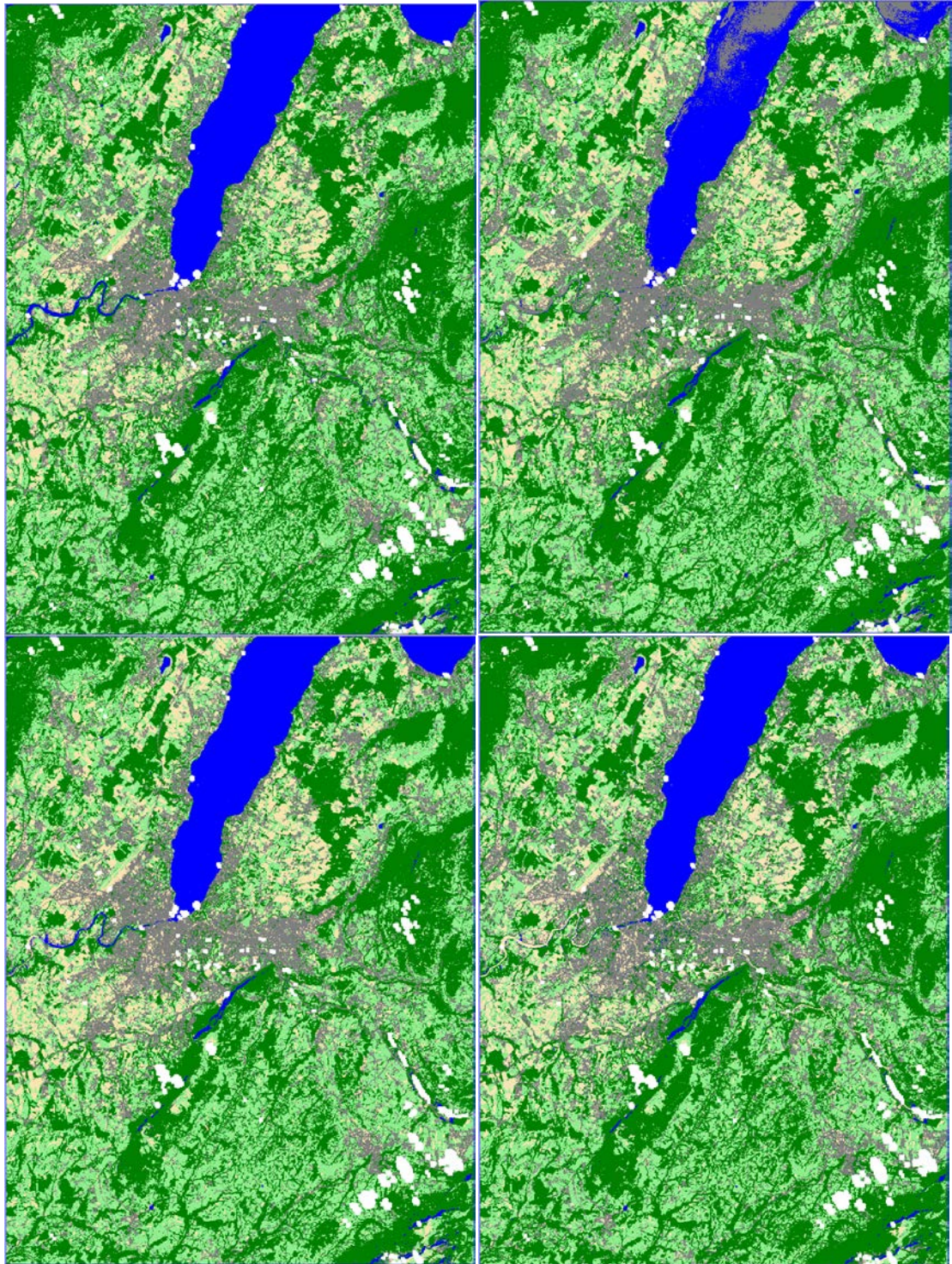
Figure 10: Workflow de la méthode 4

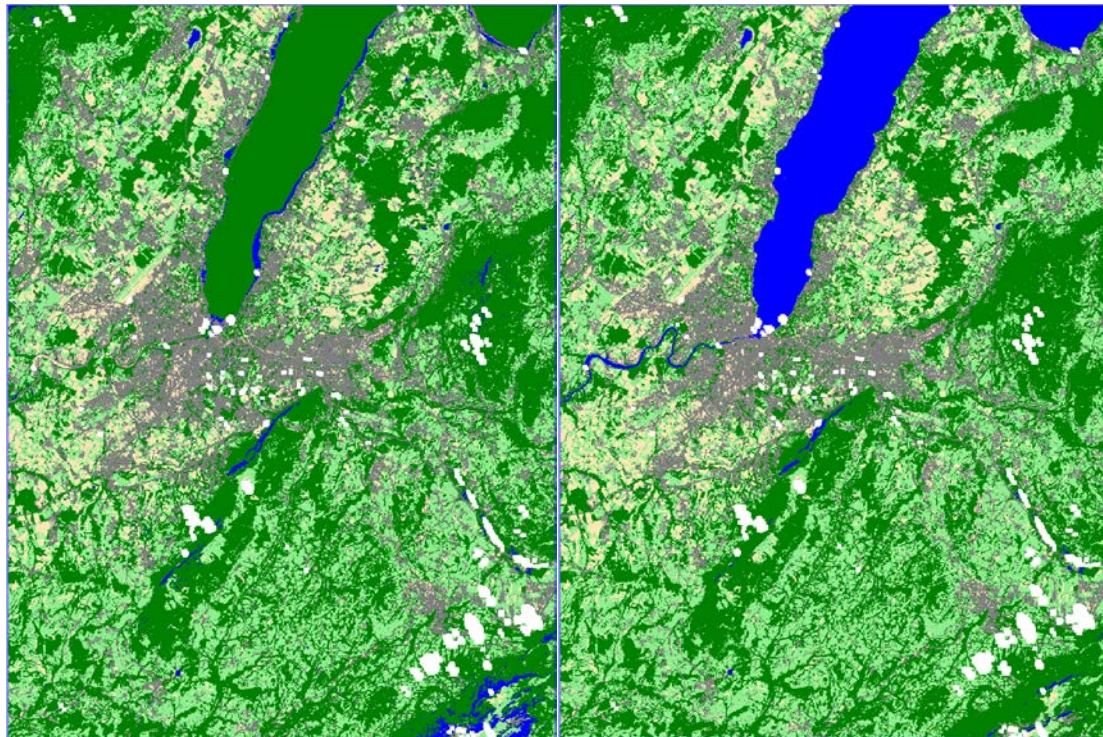
4.4.3. Résultats

Les classifications de la deuxième scène effectuées en boucle sur la base des échantillons aléatoires donnent des résultats de composition du sol qui ne semblent pas correspondre à la couverture terrestre réelle.

La méthode des échantillons aléatoires comporte en effet deux faiblesses qu'il est important de mentionner. Premièrement, la segmentation est un traitement où les paramètres doivent être adaptés au site auxquels ils s'appliquent car les objets n'auront pas les mêmes caractéristiques d'un site à l'autre. Par conséquent, en établissant des paramètres standards appliqué à plusieurs sites différents, le risque est que la qualité de la segmentation ne soit pas toujours la même et ainsi crée des objets de taille très différentes. Pour permettre une segmentation adaptée à chaque site, le processus de traitement devrait être interrompu pour que l'utilisateur puisse juger du résultat de la segmentation et entrer les paramètres optimaux, empêchant ainsi une automatisation complète du processus. Deuxièmement, lors de l'étape de la création des échantillons, le risque de la sélection aléatoire est que les objets sélectionnés peuvent être de taille différente d'une fois à l'autre et donc risquent de couvrir des surfaces différentes d'une exécution à l'autre, influençant directement la création des signatures spectrales et les classifications en boucle à chaque exécution du processus d'analyse. La figure 11 présente un exemple de cette limitation de la méthode avec le résultat des classifications basées sur les échantillons aléatoires pour le site Geneva.







- 1) eau
- 2) forêt
- 3) urbain
- 4) sol/herbe
- 5) champs

Figure 11: Les classifications de la deuxième scène basées sur les échantillons aléatoires

Site: Geneva (5.8.2015)

En observant le résultat des différentes classifications représentées ci-dessus, il peut être noté que, dans les quatrième et septième classifications, le lac Léman a été classifié en partie comme de la forêt ou comme de la surface urbaine. Le Rhône n'est également pas à chaque fois classifié comme de la surface aquatique. Par conséquent, la superposition de ces classifications va qualifier la surface du lac Léman et le Rhône comme une surface de changement de couverture du sol, or, avec la connaissance du terrain, il est certain que le lac Léman est resté une surface aquatique dans les deux scènes sélectionnées.

Une autre faiblesse se trouve également dans l'utilisation de la fonction *r.series*. En effet, pour que la fonction *r.series* soit correctement appliquée, les classifications en boucle de la deuxième scène devraient être simplifiées au maximum, d'où la nécessité de reclasser la deuxième scène en passant de 8 à 5 classes. Cependant, les résultats ne seront donc jamais précis et ne correspondront pas en précision à la réalité. (Voir annexe 3 pour résultats des sites Papua, Dubti et Geneva)

En somme, de par les constats émis ci-dessus, il peut être conclu que la méthode des échantillons aléatoires n'est pas une méthode fiable et stable pour être considérée dans le projet

d'automatisation du suivi de la couverture du sol.

4.5. Zone d'entraînement totale

4.5.1. Principe

La dernière méthode testée dans ce travail est basée sur l'idée que, contrairement à la méthode des échantillons aléatoires où des échantillons partiels de la première classification utilisés comme zones d'entraînement pour la classification finale, l'ensemble de la première classification peut être utilisé comme zone d'entraînement et donner des résultats fiables. Cette méthode peut être potentiellement stable si les changements entre les scènes sont minimales car cela laisserait moins d'erreur dans la détection et création des signatures de la dernière classification. Ainsi l'avantage de cette méthode serait de l'utiliser précisément pour créer des classifications de multiples scènes qui sont relativement proches les unes des autres d'un point de vue temporel.

4.5.2. Méthodologie

La classification non-supervisée de la première scène a été faite (*i.group/i.cluster/i.maxlik*) dans un premier temps de façon à utiliser la couche en sortie comme zone d'entraînement dans le processus de classification de la deuxième scène (*i.group/i.gensig/i.maxlik*). Étant donné que la classification finale est basée sur la classification de la première scène, les catégories des classes à ajouter seront les mêmes que celles identifiées pour la première scène (*r.category*). Finalement, semblable à la méthode de comparaison post-classification, l'image de changement a aussi été calculée après avoir réalisé les scènes classifiées (*r.mapcalc*).

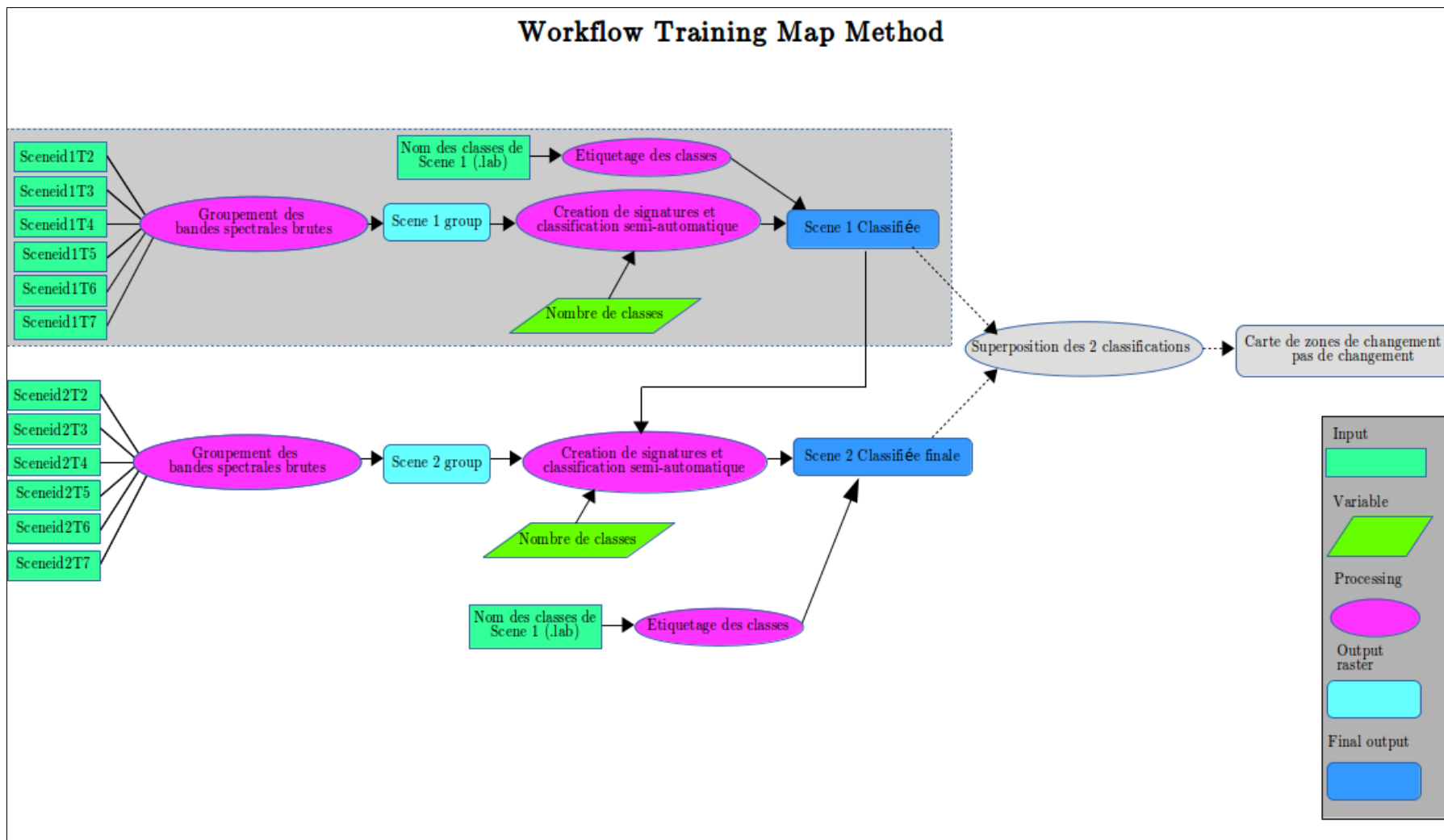
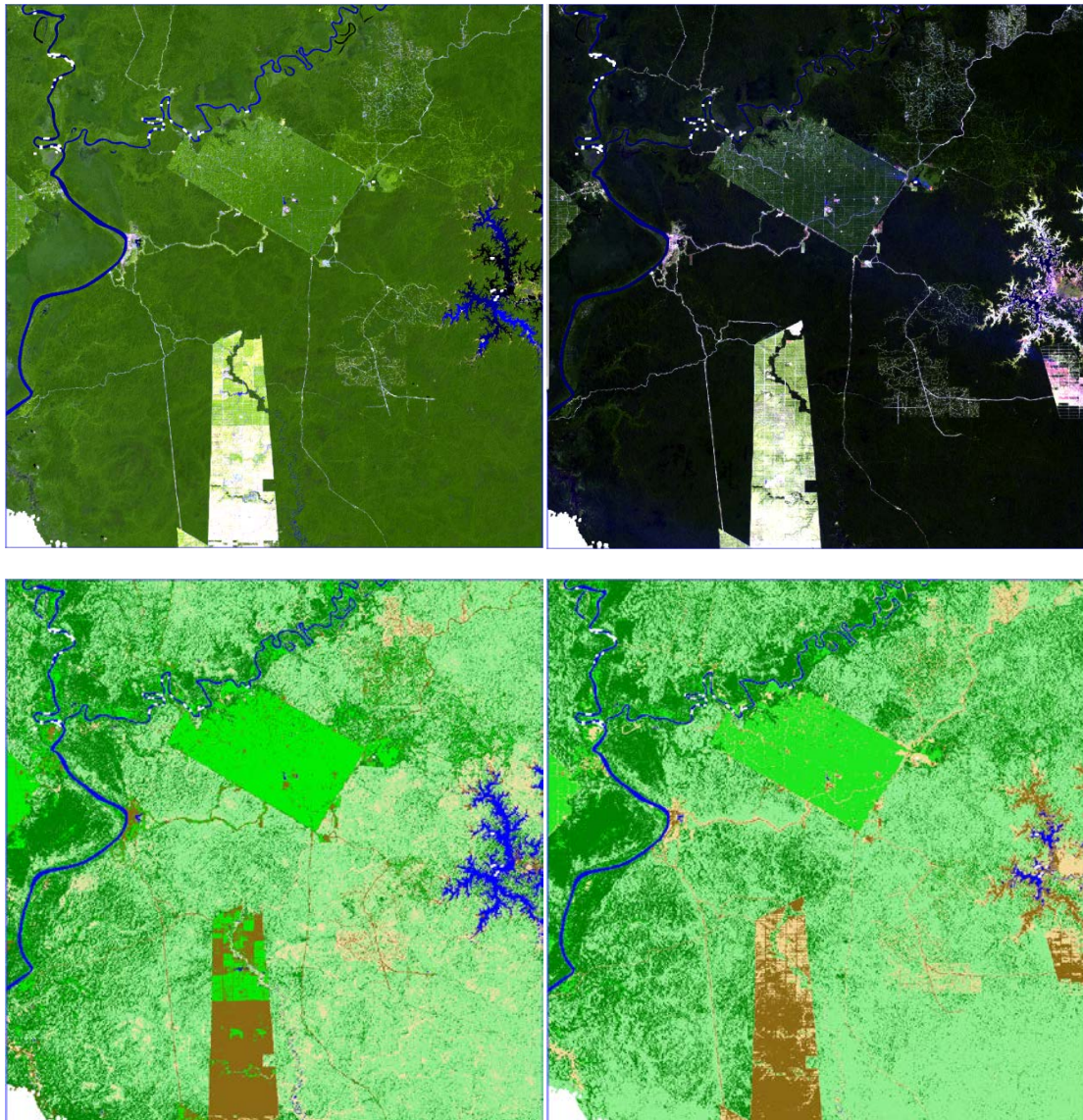


Figure 12: Workflow de la méthode 5

4.5.3. Résultats

Parmi les sites qui, lors de la méthode de comparaison post-classification, ont été identifiés comme contenant une minorité de changement, cette dernière méthode ne semble pas donner les résultats pertinents et fiables.

En prenant l'exemple du site Papua qui présente seulement 14% de surface changée, on observe en effet, à la figure 13, différents résultats notamment au niveau du lac/marais. Ces éléments ne réduisent pas en surface avec cette dernière méthode à la différence des résultats d'autres méthodes où il y a clairement un assèchement de ces surfaces aquatiques entre les 2 scènes. De plus, au niveau des champs déforestés en bas des images, il y a des zones identifiées comme du sol dans la composition d'images brutes qui sont en partie classifiées comme étant de l'eau, ce qui ne correspond pas aux autres résultats plus fiables testés dans ce travail.



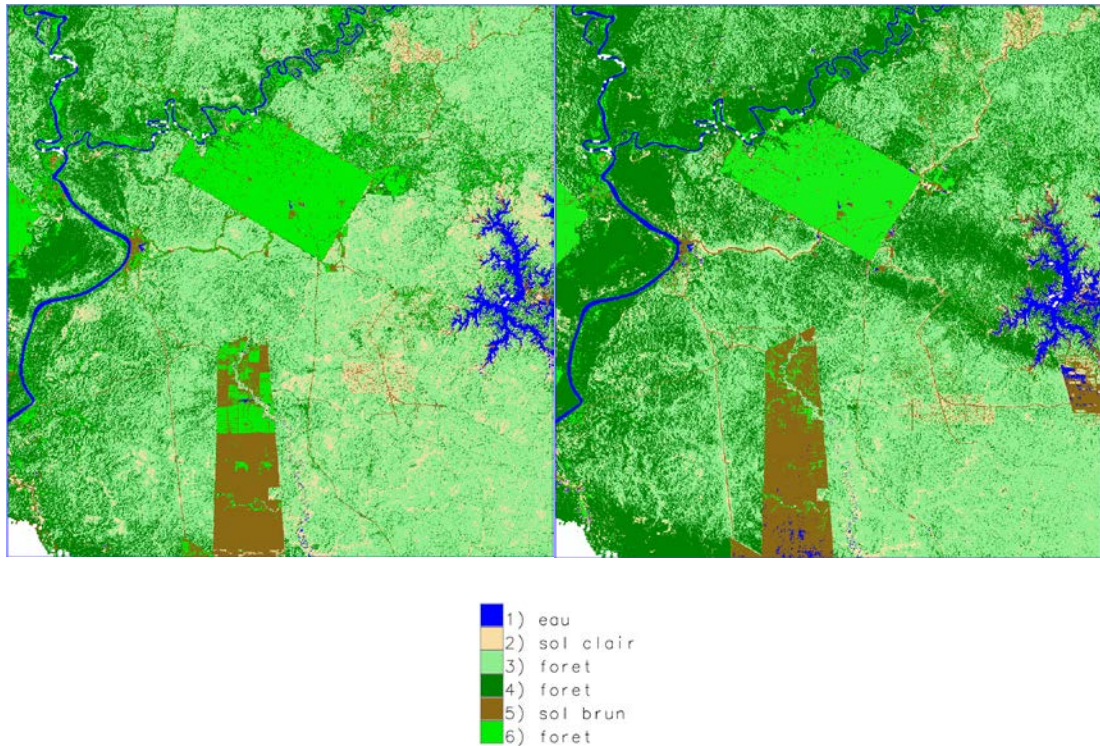


Figure 13: Compositions colorées en fausses couleurs (764) (+ i.colors.enhance)

Classification scène 1 et classification scène 2 (méthode 1) (haut)

Classification scène 1 et classification scène 2 (méthode 5) (bas)

Site: Papua (25.1.2015 et 22.9.2015)

Ces différences de résultats peuvent aussi être observées dans l'image montrant le changement détecté selon cette méthode en comparaison avec l'image de changement selon la méthode de comparaison post-classification (figure 14). En effet, selon les statistiques des couches, cette dernière méthode détecte moins de changement global et le lac/marais n'est pas reconnu comme étant une surface qui a changé de couverture du sol.

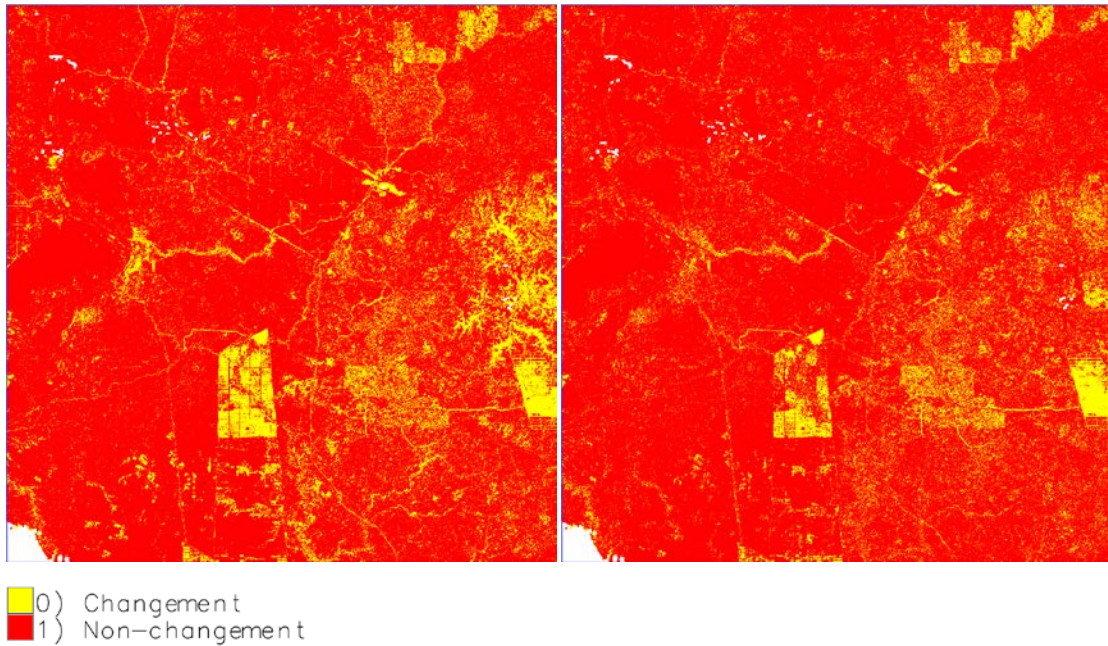


Figure 14: Gauche: image de changement (en jaune) vs. non-changement (en rouge) selon méthode 1

Droite: image de changement (en jaune) vs. non-changement (en rouge) selon méthode 5

Site: Papua (25.1.2015 et 22.9.2015)

Concernant les tests effectués sur les autres sites, la méthode semble donner une classification de la deuxième scène relativement cohérente pour les scènes ayant une classification identique entre elles (voir résultats de la méthode 1), comme par exemple le site Dubti (8 % de changement) ou encore Geneva (40% de changement) (voir annexe). Cependant, pour le site Iceland qui a 2 scènes très différentes dans leur classification, cette dernière méthode ne donne pas de résultats pertinents. À la figure 15, dans la moitié gauche de l'image, il y a des zones d'eau/de glace selon la première méthode qui sont maintenant classées comme étant du sol. Ainsi pour donner des résultats plausibles, il est nécessaire que les classes et catégories des 2 scènes se correspondent exactement car la classification de la deuxième scène utilise la première scène comme carte d'entraînement. Il peut être conclu que la proportion de changement n'a pas d'influence sur le résultat avec cette méthode-ci tandis que l'ordre des classes et leurs catégories influencent la qualité des résultats.

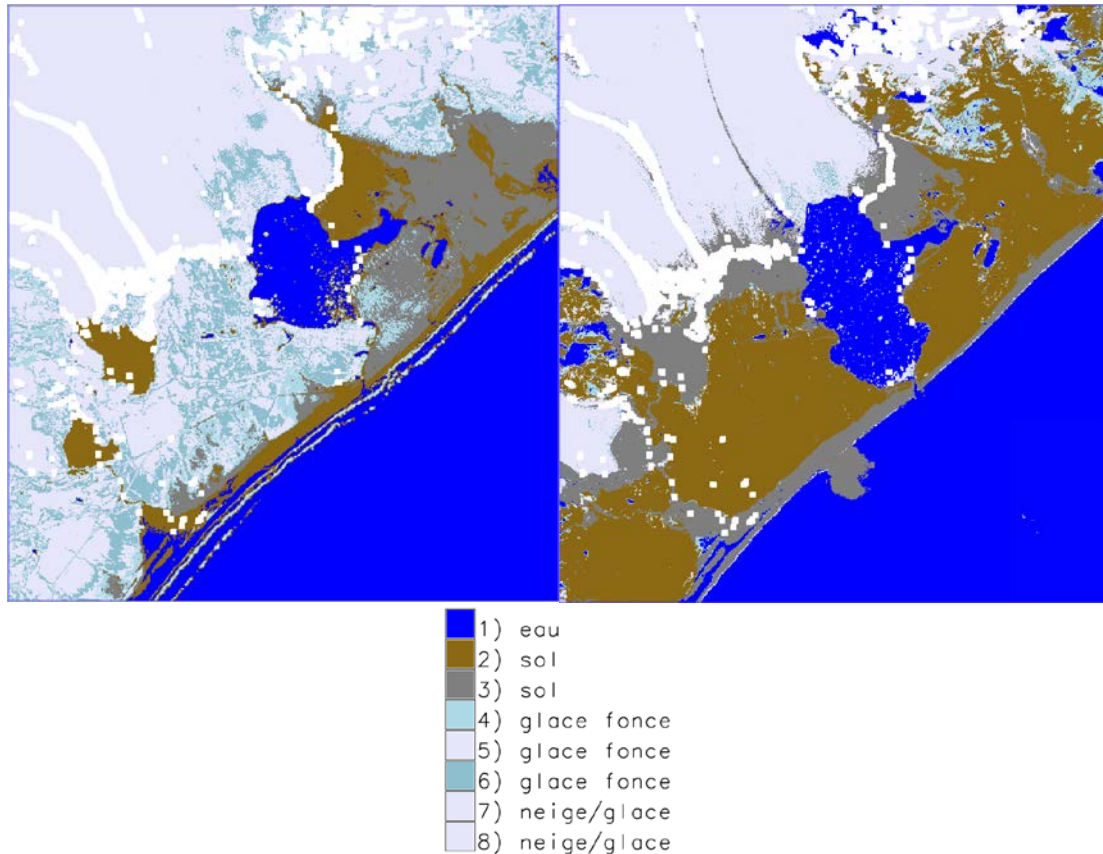


Figure 15: Classification scène 1 et classification scène 2 selon méthode 5

Site: Iceland (4.5.2015 et 25.9.2015)

5. Discussion des méthodes et conclusion

Le test des 5 méthodes de détection du changement et de classification montre qu'il y a des limitations dans chaque méthode pour parvenir à des résultats fiables et processus stables et pour atteindre une automatisation complète des processus de traitement.

Ce travail de recherche a montré que les 5 méthodes requièrent une examination préalable afin de déterminer le nombre de classes et les catégories correspondantes pour la classification de la première scène.

Il a été démontré que les résultats des méthodes sont souvent dépendants du site traité. Soit les changements dans les composantes principales sont impossible à déterminer/détecter de manière statistique et automatique pour créer le masque nécessaire à la classification de la deuxième scène. Soit les résultats semblent dépendre du degré et de la direction des différences spectrales et donc la distribution des catégories de différences change d'un site à un autre rendant la création du masque non-automatisable. Soit les paramètres de segmentation nécessaire pour la création d'échantillons aléatoires sont dépendants des caractéristiques du site, ce qui a une influence sur la précision des classifications et ainsi sur la création automatique du masque. Soit les résultats de classification varient et donc sont instables suivant si le site a des

scènes classifiées de manière identiques ou non.

Une des difficultés majeures dans la recherche d'une méthode stable et fiable donnant des résultats cohérents et précis pour le suivi automatique de la couverture du sol semble être le fait qu'il faille réaliser des classifications non-supervisées entre scènes qui soient simples et comparables (donc identiques) dans le nombre de classes et surtout dans l'ordre des catégories de classes. Cette observation se révèle être un problème conséquent étant donné que, dans une perspective multi-temporelle et multi-site, il est difficile d'avoir ces 2 prérequis. Pour permettre un traitement d'images de manière totalement automatique, il faut essayer de trouver la meilleure classification possible qui serait, d'un côté, simple mais, dans ce cas, trop grossière par rapport à la réalité et qui serait, de l'autre côté, bien détaillée mais donc difficile à appliquer sur diverses scènes si le changement est trop grand.

En fin de compte, il semblerait que dans le but de classifier automatiquement la couverture du sol en tenant compte des classifications précédentes, il est nécessaire que les scènes prises en compte puissent être classifiées de manière identique par rapport à l'ordre et aux catégories des classes. En effet, dans les méthodes où la classification de la première scène est utilisée partiellement ou entièrement comme zone d'entraînement, les résultats se révèlent en majorité ne pas être fiables. En revanche, la méthode de comparaison post-classification peut obtenir des résultats pertinents avec des classifications différentes entre les 2 scènes car celle-ci effectue les classifications indépendamment l'une de l'autre.

C'est pourquoi, la méthode qui pourrait prétendre présenter les meilleurs résultats est celle de la zone d'entraînement totale dans le cas où les classifications sont identiques au niveau des classes/signatures spectrales. Cependant, le seul moyen de savoir si les scènes entraîneront des classifications identiques est d'effectuer la méthode de comparaison post-classification qui a été jugée impossible à automatiser.

Par conséquent, pour la suite de la recherche d'une méthode stable et fiable de classifications multi-temporelles sur la base de détection du changement, la littérature affirme qu'une combinaison de méthodes peut améliorer les résultats et faciliter la détection des changements. En effet, Lu et al. (2004) et Gong (1993) suggèrent de combiner 2 méthodes en utilisant, par exemple, les images de différences ou/et des indices de végétation (NDVI) dans une analyse en composantes principales pour de potentiels meilleurs résultats.

6. Conclusions personnelles

Ce stage de 3 mois au sein du GRID-Genève a permis de développer et d'accroître mes compétences et connaissances en télédétection et analyse d'images satellitaires. Cette expérience enrichissante a été une opportunité d'acquérir une certaine aisance dans l'utilisation du logiciel GRASS et d'entraîner les techniques de classification apprises durant le certificat en géomatique. J'ai pu également apprendre comment créer et développer des scripts bash

automatisants des processus de traitement GRASS. De plus, le fait d'avoir un autre stagiaire travaillant aussi dans le cadre du projet LiMES a créé l'occasion d'échanger des informations et expériences permettant ainsi une bonne entraide.

Finalement, je tiens à remercier Bruno Chantenoux, Karin Allenbach et le GRID-Genève de m'avoir donné l'opportunité de faire ce stage et d'avoir pu participer à un projet aussi intéressant que LiMES. J'espère que ce début de travail de recherche aidera à développer et découvrir des méthodes de classification qui permettront un bon suivi automatique de la couverture du sol.

7. Bibliographie

Berberoglu, S. & Akin, A., 2009. International Journal of Applied Earth Observation and Geoinformation Assessing different remote sensing techniques to detect land use / cover changes in the eastern Mediterranean. *International Journal of Applied Earth Observation and Geoinformation*, 11, pp.46–53.

Chisanga, C., 2012. Free Open Source Software (FOSS) Geographic Resource Analysis Support System (GRASS GIS 6 . 4) for Mapping Land use , land cover dynamics , Luswishi Farm Block , Lufwanyama District.

Deng, J.S. et al., 2009. Spatio-temporal dynamics and evolution of land use change and landscape pattern in response to rapid urbanization. *Landscape and Urban Planning*, 92, pp.187–198.

Fichera, C.R., Modica, G. & Pollino, M., 2012. Land Cover classification and change-detection analysis using multi-temporal remote sensed imagery and landscape metrics. *European Journal of Remote Sensing*, (45), pp.1–18.

Fuchs, M., Hoffmann, R. & Schwonke, F., 2008. Change Detection with GRASS GIS – Comparison of images taken by different sensors. *Geoinformatics FCE CTU*, pp.25–38.

Gong, P., 1993. Change Detection Using Principal Component Analysis and Fuzzy Set Theory. *Canadian Journal of Remote Sensing*, 19(1), pp.22–29.

Hussain, M. et al., 2013. ISPRS Journal of Photogrammetry and Remote Sensing Change detection from remotely sensed images : From pixel-based to object-based approaches. *ISPRS Journal of Photogrammetry and Remote Sensing*, 80, pp.91–106. Available at:

<http://dx.doi.org/10.1016/j.isprsjprs.2013.03.006>.

Lu, D. et al., 2004. Change detection techniques. *International Journal of Remote Sensing*, 25(12), pp.2365–2407.

Munyati, C., 2004. Use of Principal Component Analysis (PCA) of Remote Sensing Images in Wetland Change Detection on the Kafue Flats , Zambia. *Geocarto International*, 19(3), pp.11–22.

Rawat, J.S. & Kumar, M., 2015. Monitoring land use / cover change using remote sensing and GIS techniques : A case study of Hawalbagh block , district Almora, Uttarakhand, India. *The Egyptian Journal of Remote Sensing and Space Sciences*, 18(1), pp.77–84. Available at: <http://dx.doi.org/10.1016/j.ejrs.2015.02.002>.

Théau, J., 2012. Change Detection. In W. Kresse & M. D. Danko, eds. *Springer Handbook of Geographic Information*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 75–94. Available at: http://dx.doi.org/10.1007/978-3-540-72680-7_7.

Timoner, P., 2016. Détection de nuages et analyses de NDVI sur des images multi-temporelles Landsat-8, Projet LIMES, UNEP/DEWA/GRID-Genève (Rapport de stage).

UNEP/DEWA/GRID-Geneva, 2016. *United Nations Environment Programme DEWA/GRID-Geneva, "Taking the global pulse for a greener world ": About us*. Available from: http://www.grid.unep.ch/index.php?option=com_content&view=article&id=1&Itemid=511&lang=fr (Accessed 13.07.2016)

Yu, W. et al., 2016. A new approach for land cover classification and change analysis : Integrating backdating and an object-based method. *Remote Sensing of Environment*, 177, pp.37–47. Available at: <http://dx.doi.org/10.1016/j.rse.2016.02.030>.

Yuan, F. et al., 2005. Land cover classification and change analysis of the Twin Cities (Minnesota) Metropolitan Area by multitemporal Landsat remote sensing. *Remote Sensing of Environment*, 98, pp.317–328.

8. Annexes

Annexe 1: Liste des sites	39
Annexe 2: Scripts.....	39
LiMES_batcher_v003.py.....	39
LiMES_grasser_v003.py.....	45
Change_detection1.sh - Méthode Comparaison post-classification	53
Change_detection2.0.sh - Méthode ACP	54
Change_detection2.1.sh - Méthode des différences spectrales.....	58
Change_detection2.2.sh - Méthode des échantillons aléatoires.....	60
Change_detection2.3.sh - Méthode de la zone d'entraînement totale	65
Annexe 3: Résultats des tests par sites	67
Comparaison post-classification	67
Papua.....	67
Irak.....	70
Iceland.....	73
Dubti.....	76
Geneva	79
Analyse en composantes principales (ACP)	82
Papua.....	82
Irak.....	82
Iceland.....	83
Dubti.....	83
Geneva	83
Différences des images	84
Papua.....	84
Irak.....	85
Iceland.....	86
Dubti.....	87
Geneva	89
Échantillons aléatoires	92
Papua.....	92
Dubti.....	94
Geneva	96
Zone d'entraînement totale	99
Papua.....	99
Irak.....	100
Iceland.....	101
Dubti.....	101
Geneva	102

Annexe 1: Liste des sites

Métadonnées et information sur classifications non-supervisées des scènes 1 et 2

Site	Location	Scene ID 1	Date	Cloud Cover	Scene ID 2	Date	Cloud Cover	Classes T1, T2
Papua_123	Papua	LC81000652015025 LGN00	25.1.2015	4%	LC81000652015265LG N00	22.9.2015	0%	6
mesopotamia_m arshlands	Irak	LC81660382015263 LGN00	20.9.2015	0%	LC81660382016090LG N00	30.3.2016	0%	8
breidamerkurjok ul_118	Iceland	LC82170152015124 LGN00	4.5.2015	40%	LC82170152015268LG N00	25.9.2015	11%	8
Dubti	Dubti	LC81670522014347 LGN00	13.12.2014	0%	LC81670522015110LG N00	20.4.2015	0%	7
Geneva_2	Geneva	LC81960282013227 LGN00	15.8.2013	1%	LC81960282015217LG N00	5.8.2015	0%	8

Annexe 2: Scripts

Liste des scripts

LiMES_batcher_v003.py

LiMES_grasser_v003.py

Change_detection1.sh - Méthode Comparaison post-classification

Change_detection2.0.sh - Méthode ACP

Change_detection2.1.sh - Méthode des différences spectrales

Change_detection2.2.sh - Méthode des échantillons aléatoires

Change_detection2.3.sh - Méthode de la zone d'entraînement totale

LiMES_batcher_v003.py

```
#!/usr/bin/env python
```

```
#
```

```
#####
```

```

#
# MODULE:      LiMES_batcher
#
# VERSION:    0.0.3
#
# AUTHOR(S):  Bruno Chatenoux
#
# PURPOSE:    Main script for LiMES process
#
# REQUIREMENT: - Python 2.7 (tested)
#              - gsutil (http://www.bchate.name/shaarli/?tQadYw)
#              - pandas (sudo pip install pandas)
#
# DATE:       10.5.2016
#
# CHANGELOG:  0.0.3  Check gsutil availability
#             0.0.2  Multi source switcher
#             LS8 benchmarking scripts integrated
#
# KNOWN BUGS: - Do not switch is several scene ar in one line and
#             one scene cannot be downloaded > list all scene to be
#             downloaded and remove from list when done, switching
#             data provider til the list is empty
#
#####

import os
import sys
import atexit
import argparse
import time
import logging
import ConfigParser
import csv
import shutil
import ogr
import subprocess
from osgeo import osr
import pandas as pd # sudo apt-get install python-pandas, To be removed when > DB

# import gis_tools_v001 as gt
import get_images_v002 as geti
import general_tools_v002 as gent

```

```

def main(argv):
    # Parse arguments and create help
    parser = argparse.ArgumentParser(description='Main LiMES batch module')
    parser.add_argument('batch_file_name', help='Batch file Name')
    args = parser.parse_args()

    initt = time.time()
    config = ConfigParser.RawConfigParser() # To be removed when > DB
    config.read('configFile.cfg')          # To be removed when > DB

    logging.basicConfig(filename='liMES_batcher.log',
                        level=logging.DEBUG,
                        format='%(asctime)s |%(levelname)s| %(message)s', datefmt='%m/%d/%Y %H:%M:%S')
    msg = '%s STARTED' % ' '.join(sys.argv[:])
    print '* %s' % msg
    logging.info(msg)

    # Check gsutil availability
    try:
        command = 'gsutil,version'
        comlist = command.split(',')
        subprocess.call(comlist, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    except:
        msg = 'Cannot find gsutil: SCRIPT INTERRUPTED'
        print '* %s' % msg
        logging.error(msg)
        sys.exit()

    # Open tables DENISA: > DB
    pb_db = pd.read_csv('./db/process_bands.txt',sep='\t')
    pp_db = pd.read_csv('./db/providers_priority.txt',sep='\t')

    # TBD: Create a unique batchID to be transfered in the db and used in the log

    if not os.path.isfile(args.batch_file_name):
        msg = 'Cannot find file %s: SCRIPT INTERRUPTED' % args.batch_file_name
        print '* %s' % msg
        logging.error(msg)
        sys.exit()
    else:
        # Create an empty scenes folder
        os.umask(0000) # To create 777 directories
        if os.path.exists('scenes'):
            shutil.rmtree('scenes')

```

```

# Download all files
with open(args.batch_file_name, 'rb') as f:
    next(f) # Skip header
    rows = csv.reader(f, delimiter='\t')
    # process sites one by one
    for row in rows:
        print '* Downloading %s' % row[0]
        # Check if shapefile is readable
        try:
            ds = ogr.Open('./sites/%s/%s.shp' % (row[0], row[0]))
            lyr = ds.GetLayer(0)
        except AttributeError:
            msg = 'Cannot read %s shapefile' % (row[0])
            print '* %s' % msg
            logging.error(msg)
            continue
        # Check if shapefile is a polygon
        if lyr.GetGeomType() != 3:
            msg = '%s shapefile is not a polygon' % row[0]
            print '* %s' % msg
            logging.error(msg)
            continue
        # Check if shapefile epsg code is 4326
        srs = osr.SpatialReference(str(lyr.GetSpatialRef()))
        try:
            res = srs.AutoIdentifyEPSG()
            if int(srs.GetAuthorityCode(None)) != 4326:
                msg = '%s shapefile projection is not in WGS84' % row[0]
                print '! %s' % msg
                logging.warning(msg) # Or error and continue ?
        except:
            msg = '%s shapefile miss a .prj file' % row[0]
            print '! %s' % msg
            logging.warning(msg) # Or error and continue ?

# Get image sensor
unik = []
scene_list = []
for sceneid in row[2].split(','):
    if sceneid[:3] not in unik:
        unik.append(sceneid[:3])
    # Check for duplicate sceneids
    if sceneid not in scene_list:
        scene_list.append(sceneid)

```

```

if len(unik) > 1:
    msg = 'You have a mix of sensors %s will not be processed' % (row[0])
    print '! %s' % msg
    logging.warning(msg)
    continue
if len(row[2].split(',')) != len(scene_list):
    msg = 'Some scenids were duplicated'
    print '* %s' % msg
    logging.info(msg)
# Get necessary bands DENISA: > DB
stopit = False
bands = []
try:
    mycol = pb_db[unik[0]]
except:
    mycol = None
    msg = '%s is not an implemented sensor' % unik[0]
    print '* %s' % msg
    logging.error(msg)
for p in row[3].split(','):
    try:
        myrow = pb_db[(pb_db.Process == p)].iloc[0]
    except:
        myrow = None
        msg = '%s is not an implemented process' % p
        print '* %s' % msg
        logging.error(msg)
    if myrow is None or mycol is None:
        stopit = True
    else:
        for b in pb_db[(pb_db.Process == p)].iloc[0][unik[0]].split(','):
            if b not in bands:
                bands.append(b)
bands.sort(key=int)
if stopit:
    continue

# Get bands by providers priority
pp_db_asc = pp_db.sort_values([unik[0]])
for p in pp_db_asc['Provider'].iloc[:]:
    try:
        func = getattr(geti, p)

```

```

        func(scene_list, ',').join(bands))
        break
    except:
        e = sys.exc_info() # DEV
        continue

f.close()
deltat = time.time() - initt
dsize = gent.dir_size('./scenes')
mbs = dsize / deltat
msg = '%i Mb downloaded in %i seconds (%0.1f Mb/s)' % (dsize, deltat, mbs)
print '* %s' % msg
logging.info(msg)

## Process all files
# with open(args.batch_file_name, 'rb') as f:
#     next(f) # Skip header
#     rows = csv.reader(f, delimiter='\t')
#     # process sites one by one
#     for row in rows:
#         print '* Processing %s' % row[0]

# f.close()

## Get shapefile extent
# w_extent, e_extent, s_extent, n_extent = lyr.GetExtent()
## Write in DB

msg = '%s COMPLETED in %i seconds\n' % (' '.join(sys.argv[:]), time.time() - initt)
print '* %s' % msg
logging.info(msg)

return 0

def cleanup():
    pass

if __name__ == '__main__':
    atexit.register(cleanup)
    sys.exit(main(sys.argv))

```

LiMES_grasser_v003.py

```
#!/usr/bin/env python
#
#####
#
# MODULE:      LiMES_grasser.py
#
# VERSION:     0.0.3
#
# AUTHOR(S):   Bruno Chatenoux & Pierric De Laborie
#
# PURPOSE:     Main LiMES script for Grass processing
#
# REQUIREMENT: - Python 2.7 (tested)
#              - python-gdal
#              - gdal-bin (sudo apt-get install gdal-bin)
#
# DATE:        10.05.2016
#
# CHANGELOG:   0.0.3  automatically import site shapefile and name it 'aoi'
#              1 line in the batch file > 1 GRASS location
#
# KNOWN BUGS:  None
#
#####

import os, sys, atexit
import argparse
import glob
from subprocess import Popen, PIPE
from datetime import date, timedelta
import time
import logging
import ConfigParser
import csv
import pandas as pd # sudo apt-get install python-pandas, To be removed when > DB

import subprocess
import ogr
```



```

import shutil
import grass_tools_v003 as grat

import cloud_mask as cloud

def main(argv):
    # Parse arguments and create help
    parser = argparse.ArgumentParser(description='Main LiMES script fro Grass processing')
    parser.add_argument('batch_file_name', help='Batch file Name')
    parser.add_argument('-c', action='store_true', default=False, help='Convert int16 output to
float64 (default: False)')
    parser.add_argument('-m', action='store_true', default=False, help='Merge all different
scene masks in order to work only with pixels that are clear in all the scenes (default: False)')
    args = parser.parse_args()

    initt = time.time()
    config = ConfigParser.RawConfigParser() # To be removed when > DB
    config.read('configFile.cfg')          # To be removed when > DB

    logging.basicConfig(filename='liMES_grasser.log', level=logging.DEBUG, format='%(asctime)s
|%(levelname)s| %(message)s', datefmt='%m/%d/%Y %H:%M:%S')
    msg = '%s STARTED' % ' '.join(sys.argv[:])
    print '* %s' % msg
    logging.info(msg)

    # Open tables DENISA: > DB
    pb_db = pd.read_csv('./db/process_bands.txt', sep='\t')

    # Process all files
    with open(args.batch_file_name, 'rb') as f:
        next(f) # Skip header
        rows = csv.reader(f, delimiter='\t')
        # process sites one by one
        for row in rows:
            # Define projection
            command = 'gdalsrsinfo,./scenes/%s/%s_BQA.TIF,-o,wkt' % (row[2].split(',')[0],
row[2].split(',')[0])
            comlist = command.split(',')
            info     = subprocess.check_output(comlist)#,      stdout=subprocess.PIPE,
stderr=subprocess.PIPE) # remove the 2 last arguments to become verbose
            k = info.rfind('EPSG')

```

```

# Reproject shapefile if needed
if not os.path.exists('./sites/%s/%s_%s.shp' % (row[0], row[0], info[k+7:k+12]]):
    print '** Creating %s_%s.shp' % (row[0], info[k+7:k+12])
    command = 'ogr2ogr,-t_srs,EPSG:%s,./sites/%s/%s_%s.shp,./sites/%s/%s.shp' %
(info[k+7:k+12], row[0], row[0], info[k+7:k+12], row[0], row[0])
    comlist = command.split(',')
    subprocess.call(comlist, stdout=subprocess.PIPE, stderr=subprocess.PIPE)

# Initiate grassDB
# grat.create_grassdb(row[0], row[1], row[2].split(',')[0])
grat.create_grassdb(row[0], row[1], info[k+7:k+12])
grat.connect_grassdb(row[1])
grat.import_aoi_grassdb(row[0], info[k+7:k+12])

# Looking for the MTL.txt files needed for the Fmask function
for scene in row[2].split(','):
    mydir = './scenes/%s/' % scene
    for file in os.listdir(mydir):
        if file.endswith("_MTL.txt"):
            MTLfile = file
    try:
        MTLfile
    except NameError:
        sys.exit('MTL.txt file from %s scene is missing. Script aborted' % scene)
    else:
        pass

# Each scene treatment
for scene in row[2].split(','):

    ## Set up yyyyymmdd format for outputs
    # yyyy = scene[9:13]
    # jday = scene[13:16]
    # dt = date(int(yyyy), 1, 1) + timedelta(days= (int(jday)-1))
    # yyyyymmdd = dt.strftime('%Y%m%d')
    # print yyyyymmdd

    # Fmask execution
    print '** Starting Fmask function on %s scene' % scene
    cloud.cloudmask(scene)

```

```

# Remove cropped folder if needed
if os.path.exists('./sites/%s/cropped/%s' % (row[0], scene)):
    shutil.rmtree('./sites/%s/cropped/%s' % (row[0], scene))
os.makedirs('./sites/%s/cropped/%s' % (row[0], scene))

# CLIP SCENE
print '** Clipping %s scene' % (scene)
ds = ogr.Open('./sites/%s/%s_%s.shp' % (row[0], row[0], info[k+7:k+12]))
lyr = ds.GetLayer(0)
LLLong,URLong,LLLat,URLat = lyr.GetExtent()
for tifname in glob.glob('./scenes/%s/*.tT][il][fF]' % scene):
    command = 'gdal_translate,-projwin,%s,%s,%s,%s,-eco,%s,%s' %
(str(float(LLLong)), str(float(URLat)), str(float(URLong)), str(float(LLLat)), tifname,
tifname.replace('/scenes/', './sites/%s/cropped/' % row[0]))
    comlist = command.split(',')
    subprocess.call(comlist, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)

# CLIP MASK
fmaskOutMask = '%s_MTLFmask' % scene
mask = './scenes/%s/%s' % (scene,fmaskOutMask)
command = 'gdal_translate,-projwin,%s,%s,%s,%s,-eco,%s,%s' %
(str(float(LLLong)), str(float(URLat)), str(float(URLong)), str(float(LLLat)), mask,
mask.replace('/scenes/', './sites/%s/cropped/' % row[0]))
    comlist = command.split(',')
    subprocess.call(comlist, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
os.rename('./sites/%s/cropped/%s/%s' % (row[0],scene,fmaskOutMask),
 './sites/%s/cropped/%s/%s' % (row[0],scene,fmaskOutMask) + '.tif')

# CLIP REFLECTANCES
fmaskOutRef = 'TOA3.2'
reflect = './scenes/%s/%s' % (scene,fmaskOutRef)
command = 'gdal_translate,-projwin,%s,%s,%s,%s,-eco,%s,%s' %
(str(float(LLLong)), str(float(URLat)), str(float(URLong)), str(float(LLLat)), reflect,
reflect.replace('/scenes/', './sites/%s/cropped/' % row[0]))
    comlist = command.split(',')
    subprocess.call(comlist, stdout=subprocess.PIPE, stderr=subprocess.PIPE)

# Multiband reflectance image to single band images
for band in range(1,8):
    if band == 7:

```

```

        outBand = 9
    else:
        outBand = band+1

    multiBand = './sites/%s/cropped/%s/%s' % (row[0],scene,fmaskOutRef)
    singleBand = './sites/%s/cropped/%s/%s_T%s.tif' %
(row[0],scene,scene,outBand)

    if args.c == True:
        command =
'gdal_translate,-ot,Float64,-scale,0,10000,0,1,%s,%s,-b,%s' % (multiBand,singleBand,band)
    else:
        command = 'gdal_translate,%s,%s,-b,%s' %
(multiBand,singleBand,band)

    comlist = command.split(',')
    subprocess.call(comlist, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)

    # Copy metadata file
    shutil.copyfile('./scenes/%s/%s_MTL.txt' % (scene, scene),
'./sites/%s/cropped/%s/%s_MTL.txt' % (row[0], scene, scene))

    # Import raster into the GRASS database
    grat.import_rast_grassdb(row[0], scene)

    # PROCESSING
    # grat.convert_2_reflectance(row[0], scene) # Not necessary anymore, since
Fmask calculate them
    # grat.NDVI (row[0], scene)

    listScenes = row[2].split(',')
    numScenes = (len(row[2].split(',')))
    if (numScenes == 1) | (args.m == True):
        grat.mask(numScenes,listScenes)
    else:
        pass

f.close()

msg = '%s COMPLETED in %i seconds\n' % (' '.join(sys.argv[:]), time.time()-initt)

```

```

print '* %s' % msg
logging.info(msg)

#    ## 8bit composite
#    print('Creating 8bit Composite ...')
#    for b in rgb :
#        lhm = 'B.int%s = round(B.toar%s * 1000000)' % (b,b)
#        grass.mapcalc(lhm, quiet= True, overwrite=True)
#        grass.run_command('r.rescale',input='B.int%s' % (b),output='B%s.8bit' %
(b),to='0,254')
#        i.group(group='composite8bitRGB',input='B%s.8bit,B%s.8bit,B%s.8bit' %
(rgb[0],rgb[1],rgb[2]),quiet=QUIET)
#    outputfilename = '%s/%s_%s_%s_composite8bit%s.TIF' % (resultpath, site, yyyyymmdd,
prfx, rgb)
#    deleteFileIfExists(outputfilename)
#    print('Exporting 8bit composite ...')
#    r.out_gdal(input="composite8bitRGB", output=outputfilename,format="GTiff",flags='c')

#    ### enhanced composite
#    # print('Creating composite ...')
#    # for b in rgb :
#    #     lhm = 'B.int%s = round(B.toar%s * 1000000)' % (b,b)
#    #     grass.mapcalc(lhm, quiet= True, overwrite=True)
#    #     grass.run_command('r.rescale',input='B.int%s' % (b),output='B%s.8bit' %
(b),to='0,254')
#    # grass.run_command('i.colors.enhance','f',red='B%s.8bit' % rgb[0],green='B%s.8bit' %
rgb[1],blue='B%s.8bit' % rgb[2])
#    # grass.run_command('r.composite',red='B%s.8bit' % (rgb[0]),green='B%s.8bit' %
(rgb[1]),blue='B%s.8bit' % (rgb[2]),output='tmpRGB')
#    # tmpfilename = '%s/%s_%s_%s_composite%s_tmp.TIF' % (resultpath, site, yyyyymmdd,
prfx, rgb)
#    # outputfilename = '%s/%s_%s_%s_composite%s.TIF' % (resultpath, site, yyyyymmdd, prfx,
rgb)
#    # deleteFileIfExists(outputfilename)
#    # print('Exporting composite ...')
#    # r.out_gdal(input="tmpRGB", output=tmpfilename,format="GTiff",type="UInt16")
#    # cmd = "gdal_translate -expand rgb %s %s" % (tmpfilename,outputfilename)
#    # startProcess(cmd)
#    # os.remove(tmpfilename)

```

```

# # os.remove('%s.aux.xml' % (tmpfilename))

# # ## urban composite (LC8)
# # print('Creating composite ...')
# # lhm = 'B.int7 = round(B.toar7 * 1000000)'
# # grass.mapcalc(lhm, quiet= True, overwrite=True)
# # grass.run_command('r.rescale',input='B.int7',output='B7.8bit',to='0,254')
# # grass.run_command('i.colors.enhance','f',red='B7.8bit',green='B5.8bit',blue='B7.8bit')
# # #
# # grass.run_command('r.composite',red='B7.8bit',green='B5.8bit',blue='B7.8bit',output='urbRGB')
# # # tmpfilename = '%s/%s_%s_%s_composite757_tmp.TIF' % (resultpath, site, yyyyymmdd,
# # prfx)
# # # outputfilename = '%s/%s_%s_%s_composite757.TIF' % (resultpath, site, yyyyymmdd,
# # prfx)
# # # deleteFileIfExists(outputfilename)
# # # print('Exporting composite ...')
# # # r.out_gdal(input="urbRGB", output=tmpfilename,format="GTiff",type="UInt16")
# # # cmd = "gdal_translate -expand rgb %s %s" % (tmpfilename,outputfilename)
# # # startProcess(cmd)
# # # os.remove(tmpfilename)
# # # os.remove('%s.aux.xml' % (tmpfilename))

# # # Nikos' Pansharpening -- requires installation of module i.fusion.hpf
# # # # - Pls see https://github.com/NikosAlexandris/i.fusion.hpf , steps are :
# # # # download files
# # # # Go to main directory and start the following command : make
# # # # MODULE_TOPDIR=$GISBASE
# # # # if sceneid[:3] == 'LC8' or sceneid[:3] == 'LE7' :
# # # #     grass.run_command('g.region', raster="B8")
# # # #     # grass.run_command('i.fusion.hpf',"l","c",pan='B8',msx='B%s,B%s,B%s' %
# # # # (rgb[0],rgb[1],rgb[2]),center="high",modulation="max")
# # # # #
# # # # grass.run_command('g.remove','f',type='group',name='compositeEqualisedRGB__HPF')
# # # # for b in rgb :
# # # #     grass.run_command('r.rescale.eq',input='B%s.hpf' % (b),output='B%s.hpf.eq' %
# # # # (b),to='0,254')
# # # # #
# # # # i.group(group='compositeRGB_Pansharp',input='B%s.hpf.eq,B%s.hpf.eq,B%s.hpf.eq' %
# # # # (rgb[0],rgb[1],rgb[2]),quiet=QUIET)
# # # # outputfilename = '%s/%s_%s_%s_composite%s_Pansharp.TIF' % (resultpath, site,
# # # # yyyyymmdd, prfx, rgb)

```

```

# # deleteFileIfExists(outputfilename)
# # print('Exporting Pansharpened composite ...')
# # r.out_gdal(input="compositeRGB_Pansharp",
output=outputfilename,format="GTiff",flags='c')

# ##### TO DO :

# #include Nikos' atmospheric correction ? His script-> Probably

# #publish layer on Geoserver

# ##### END #####

# ##### NOT USED :
# # ## Composite
# # print('Creating Composite ...')
# # i.group(group='compositeRGB',input='B%s,B%s,B%s' %
(rgb[0],rgb[1],rgb[2]),quiet=QUIET)
# # outputfilename = '%s/%s_%s_%s_composite%s.TIF' % (resultpath, site, yyyyymmdd, prfx,
rgb)
# # deleteFileIfExists(outputfilename)
# # print('Exporting composite ...')
# # r.out_gdal(input="compositeRGB", output=outputfilename,format="GTiff",flags='c')

# ##### END #####

# # delete location
# location_path = os.path.join(gisdb, location)
# shutil.rmtree(location_path)

# # sys.exit(0)
# print '\n* DONE !'
# return (0)

def cleanup():
    pass

if __name__ == '__main__':
    atexit.register(cleanup)
    sys.exit(main(sys.argv))

```

Change_detection1.sh - Méthode Comparaison post-classification

```
#!/bin/bash
```

```
#####
```

```
#
```

```
#
```

```
# VERSION:      0.0.3
```

```
#
```

```
# AUTHOR(S):    Natacha Klein
```

```
#
```

```
# PURPOSE:      Grass 7.0 functions for change detection and post-classification comparison
```

```
#
```

```
# REQUIREMENT:  - Python 2.7 (tested)
```

```
#               - Grass 7.0 (sudo apt-get install grass7 grass7-dev gdal-bin)
```

```
#               > grass installed twice !
```

```
#               - label files for the classification images (.lab)
```

```
#
```

```
# DATE:         28.06.2016
```

```
#
```

```
#
```

```
#####
```

```
export sceneid1=LC82170152015124LGN00
```

```
export sceneid2=LC82170152015268LGN00
```

```
export class_num=8
```

```
export site=Iceland
```

```
# Set region
```

```
g.region raster=${sceneid1}_MTLFmask
```

```
# Masquer les nuages
```

```
r.mapcalc "Mask_1 = (( ${sceneid1}_MTLFmask == 0) + ( ${sceneid1}_MTLFmask == 1) +  
( ${sceneid1}_MTLFmask == 3 ))" --overwrite
```

```
r.mapcalc "Mask_2 = (( ${sceneid2}_MTLFmask == 0) + ( ${sceneid2}_MTLFmask == 1) +  
( ${sceneid2}_MTLFmask == 3 ))" --overwrite
```

```
# 0 = no data
```

```
r.mapcalc "Mask_1_nodata = if ( Mask_1 == 0, null(), 1 )" --overwrite
```

```
r.mapcalc "Mask_2_nodata = if ( Mask_2 == 0, null(), 1 )" --overwrite
```

```
r.mask --overwrite raster=Mask_1_nodata
```



```

# Classification t1
g.remove -f type=group name=Time_1,Time_2
i.group                                group=Time_1                                subgroup=Time_1
input=${sceneid1}_T2,$sceneid1_T3,$sceneid1_T4,$sceneid1_T5,$sceneid1_T6,$sceneid1
}_T7
i.cluster --overwrite group=Time_1 subgroup=Time_1 signaturefile=Time_1_sig
classes=${class_num} reportfile=Time_1_Report
i.maxlik --overwrite group=Time_1 subgroup=Time_1 signaturefile=Time_1_sig output=Time_1
r.category map=Time_1 separator=: rules=Time_1_${site}.lab # il faut que les labels commencent
par un numéro/valeur

```

```

# Cloud mask of both scenes
r.mapcalc "Mask_tot = Mask_1_nodata + Mask_2_nodata" --overwrite
r.mask --overwrite raster=Mask_tot

```

```

# Classification t2
i.group                                group=Time_2                                subgroup=Time_2
input=${sceneid2}_T2,$sceneid2_T3,$sceneid2_T4,$sceneid2_T5,$sceneid2_T6,$sceneid2
}_T7
i.cluster --overwrite group=Time_2 subgroup=Time_2 signaturefile=Time_2_sig
classes=${class_num} reportfile=Time_2_Report
i.maxlik --overwrite group=Time_2 subgroup=Time_2 signaturefile=Time_2_sig output=Time_2
r.category map=Time_2 separator=: rules=Time_2_${site}.lab # il faut que les labels commencent
par un numéro/valeur

```

```

# Map unchanged areas
r.mapcalc "unchange_map = if(@Time_1 == @Time_2)" --overwrite #cela crée une carte de
change vs. no-change --> 0 = zones changées (en jaune) et 1 = zones inchangées (en rouge)

```

Change_detection2.0.sh - Méthode ACP

```
#!/bin/bash
```

```

#####
#
#
# VERSION:      0.0.3
#
# AUTHOR(S):    Natacha Klein
#
# PURPOSE:      Grass 7.0 functions for change detection and classification, second method:

```

ACP and building boxplots with R

```
#
# REQUIREMENT:  - Python 2.7 (tested)
#               - Grass 7.0 (sudo apt-get install grass7 grass7-dev gdal-bin)
#               > grass installed twice !
#               - R 3.3.1 (install libraries :sp, raster, rgdal, rgrass7)
#               - label files for the classification images (.lab)
#
# DATE:         28.06.2016
#
#
#####

export sceneid1=LC81960282013227LGN00
export sceneid2=LC81960282015217LGN00
export site=Geneva
export class_num=8

# Set region
g.region raster=${sceneid1}_MTLFmask

# Masquer les nuages
r.mapcalc "Mask_1 = (( ${sceneid1}_MTLFmask == 0) + ( ${sceneid1}_MTLFmask == 1) +
( ${sceneid1}_MTLFmask == 3 ))" --overwrite
r.mapcalc "Mask_2 = (( ${sceneid2}_MTLFmask == 0) + ( ${sceneid2}_MTLFmask == 1) +
( ${sceneid2}_MTLFmask == 3 ))" --overwrite
# 0 = no data
r.mapcalc "Mask_1_nodata = if ( Mask_1 == 0, null(), 1) " --overwrite
r.mapcalc "Mask_2_nodata = if ( Mask_2 == 0, null(), 1) " --overwrite
r.mask --overwrite raster=Mask_1_nodata

# Classification t1
g.remove -f type=group name=Time_1,Time_2
i.group                                group=Time_1                                subgroup=Time_1
input=${sceneid1}_T2,$sceneid1_T3,$sceneid1_T4,$sceneid1_T5,$sceneid1_T6,$sceneid1
}_T7
i.cluster --overwrite group=Time_1 subgroup=Time_1 signaturefile=Time_1_sig
classes=${class_num} reportfile=Time_1_Report
i.maxlik --overwrite group=Time_1 subgroup=Time_1 signaturefile=Time_1_sig output=Time_1
r.category map=Time_1 separator=: rules=Time_1_${site}.lab # il faut que les labels commencent
par un numéro/valeur
```

```

# Cloud mask of both scenes
r.mapcalc "Mask_tot = Mask_1_nodata + Mask_2_nodata" --overwrite
r.mask --overwrite raster=Mask_tot

# Classification t2
i.group                                group=Time_2                                subgroup=Time_2
input=${sceneid2}_T2,${sceneid2}_T3,${sceneid2}_T4,${sceneid2}_T5,${sceneid2}_T6,${sceneid2}_T7
i.cluster --overwrite group=Time_2 subgroup=Time_2 signaturefile=Time_2_sig
classes=${class_num} reportfile=Time_2_Report
i.maxlik --overwrite group=Time_2 subgroup=Time_2 signaturefile=Time_2_sig output=Time_2
r.category map=Time_2 separator=: rules=Time_2_${site}.lab # il faut que les labels commencent
par un numéro/valeur

# Map unchanged areas
r.mapcalc "unchange_map = if(@Time_1 == @Time_2)" --overwrite #cela crée une carte de
change vs. no-change --> 0 = zones changées (en jaune) et 1 = zones inchangées (en rouge)

#####

# ACP time 1 + time 2 (12 bands)
i.pca --overwrite
input=${sceneid1}_T2,${sceneid1}_T3,${sceneid1}_T4,${sceneid1}_T5,${sceneid1}_T6,${sceneid1}_T7,${sceneid2}_T2,${sceneid2}_T3,${sceneid2}_T4,${sceneid2}_T5,${sceneid2}_T6,${sceneid2}_T7 output=ACP #rescale=1,256
# r.out.gdal input=ACP.1
output=/media/natacha/Bigdata2/Natacha/Exported_Files/${site}/ACP.1.tif format=GTiff
nodata=0 # visualisation dans Qgis

Statistique de distribution des valeurs de ACP.1
r.univar -e -t --overwrite map=ACP.1 zones=unchange_map separator=comma
output=univar_ACP.csv
tail -n 1 univar_ACP.csv > zone_1.csv
perc=`cat zone_1.csv | cut -d , -f 17`
echo ${perc}

# Mask ACP.1 avec intervalle
r.mapcalc "ACP.1_mask = if(ACP.1 >=${perc}) & if(ACP.1 <= 0)" --overwrite # 0 = zones changées
(en jaune) et 1 = zones inchangées (en rouge)
r.mask --overwrite raster=ACP.1_mask maskcats=0 # pour masquer les zones de surface changée

```

pour avoir que la surface de non-changement comme zone d'entraînement

```
# Classification t2 final
g.remove -f type=group name=Time_2
i.group --overwrite group=Time_2 subgroup=Time_2
input=${sceneid2}_T2,${sceneid2}_T3,${sceneid2}_T4,${sceneid2}_T5,${sceneid2}_T6,${sceneid2}_T7
i.gensig trainingmap=Time_1 group=Time_2 subgroup=Time_2 signaturefile=Time_2_ACP_sig
r.mask -r
r.mask --overwrite raster=Mask_tot
i.maxlik --overwrite group=Time_2 subgroup=Time_2 signaturefile=Time_2_ACP_sig
output=Time_2_ACP
r.category map=Time_2_ACP separator=: rules=Time_1_${site}.lab # mettre les catégories de
Time_1 car cette classification se base dessus ET utiliser la symbologie de Time_1 si les classes ne
sont pas les mêmes!!
```

```
# R Script to create boxplots
```

```
R --vanilla ${site} << EOF # on crée un argument qu'on fait ensuite passer comme une variable
# R script starts here
library(sp)
library(raster)
library(rgdal)
library(rgrass7)
args <- commandArgs()
print(args)
boxtitle <- args[length(args)] # on crée la variable
print(boxtitle)
execGRASS("g.region", raster = "ACP.1")
x <- readRAST("ACP.1")
x <- raster(x)
y2 <- readRAST("unchange_map")
y2 <- raster(y2)
png('boxplots_ACP_${site}.png')
par(mfrow=c(1,3), oma=c(4,2,5,2))
boxplot(x, y2, maxpixels=3148778, names=c("changed", "unchanged"), main="ACP.1", ylab =
expression(bold("Values ACP.1")), xlab = expression(bold("Classes")))
x2 <- readRAST("ACP.2")
x2 <- raster(x2)
boxplot(x2, y2, maxpixels=3148778, names=c("changed", "unchanged"), main="ACP.2", ylab =
expression(bold("Values ACP.2")), xlab = expression(bold("Classes")))
```

```

x3 <- readRAST("ACP.3")
x3 <- raster(x3)
boxplot(x3, y2, maxpixels=3148778, names=c("changed", "unchanged"), main="ACP.3", ylab =
expression(bold("Values ACP.3")), xlab = expression(bold("Classes")))
title(main = boxtitle, outer= TRUE)
dev.off()
# R script finish here
q()
EOF

```

Change_detection2.1.sh - Méthode des différences spectrales

```

#!/bin/bash

#####
#
#
# VERSION:      0.0.3
#
# AUTHOR(S):    Natacha Klein
#
# PURPOSE:      Grass 7.0 functions for change detection and classification, third method:
Image Differencing
#
# REQUIREMENT:  - Python 2.7 (tested)
#               - Grass 7.0 (sudo apt-get install grass7 grass7-dev gdal-bin)
#               > grass installed twice !
#               - label files for the classification images (.lab)
#
# DATE:         29.06.2016
#
#
#####

export sceneid1=LC81660382015263LGN00
export sceneid2=LC81660382016090LGN00
export site=Irak
export class_num=8
export class_num_diff=5 # classes de différences
export class_to_mask1='3 4' # entrer les classes inchangées pour le masque

```

```

# Set region
g.region raster=${sceneid1}_MTLmask

# Masquer les nuages
r.mapcalc "Mask_1 = (( ${sceneid1}_MTLmask == 0) + ( ${sceneid1}_MTLmask == 1) +
( ${sceneid1}_MTLmask == 3 ))" --overwrite
r.mapcalc "Mask_2 = (( ${sceneid2}_MTLmask == 0) + ( ${sceneid2}_MTLmask == 1) +
( ${sceneid2}_MTLmask == 3 ))" --overwrite
# 0 = no data
r.mapcalc "Mask_1_nodata = if ( Mask_1 == 0, null(), 1) " --overwrite
r.mapcalc "Mask_2_nodata = if ( Mask_2 == 0, null(), 1) " --overwrite
r.mask --overwrite raster=Mask_1_nodata

# Classification t1
g.remove -f type=group name=Time_1,Time_2
i.group                group=Time_1                subgroup=Time_1
input=${sceneid1}_T2,${sceneid1}_T3,${sceneid1}_T4,${sceneid1}_T5,${sceneid1}_T6,${sceneid1}_T7
i.cluster --overwrite group=Time_1 subgroup=Time_1 signaturefile=Time_1_sig
classes=${class_num} reportfile=Time_1_Report
i.maxlik --overwrite group=Time_1 subgroup=Time_1 signaturefile=Time_1_sig output=Time_1
r.category map=Time_1 separator=: rules=Time_1_${site}.lab # il faut que les labels commencent
par un numéro/valeur

# Cloud mask of both scenes
r.mapcalc "Mask_tot = Mask_1_nodata + Mask_2_nodata" --overwrite
r.mask --overwrite raster=Mask_tot

# Classification t2
i.group                group=Time_2                subgroup=Time_2
input=${sceneid2}_T2,${sceneid2}_T3,${sceneid2}_T4,${sceneid2}_T5,${sceneid2}_T6,${sceneid2}_T7
i.cluster --overwrite group=Time_2 subgroup=Time_2 signaturefile=Time_2_sig
classes=${class_num} reportfile=Time_2_Report
i.maxlik --overwrite group=Time_2 subgroup=Time_2 signaturefile=Time_2_sig output=Time_2
r.category map=Time_2 separator=: rules=Time_2_${site}.lab # il faut que les labels commencent
par un numéro/valeur

# Map unchanged areas
r.mapcalc "unchange_map = if(@Time_1 == @Time_2)" --overwrite #cela crée une carte de
change vs. no-change --> 0 = zones changées (en jaune) et 1 = zones inchangées (en rouge)

```

```
#####

# Differences images all 6 bands
for b in $(seq 2 7)
do
r.mapcalc "diff_T${b} = ${sceneid2}_T${b} - ${sceneid1}_T${b}" --overwrite
done

# Classification of the differences (mask for unchanged areas)
i.group                group=differences                subgroup=differences
input=diff_T2,diff_T3,diff_T4,diff_T5,diff_T6,diff_T7
i.cluster --overwrite group=differences subgroup=differences signaturefile=diff_sig
classes=${class_num_diff} reportfile=diff_report_${site}
i.maxlik --overwrite group=differences subgroup=differences signaturefile=diff_sig
output=diff_map_${class_num_diff}
#          r.out.gdal          --overwrite          input=diff_map_${class_num_diff}
output=/media/natacha/Bigdata2/Natacha/Exported_Files/${site}/diff_map_${class_num_diff}.ti
f format=GTiff # pour visualisation dans Qgis
r.mask --overwrite raster=diff_map_${class_num_diff} maskcats="${class_to_mask1}" # pour
masquer les zones de surface changée

# Classification t2 based on map t1
g.remove -f type=group name=Time_2
i.group --overwrite          group=Time_2          subgroup=Time_2
input=${sceneid2}_T2,${sceneid2}_T3,${sceneid2}_T4,${sceneid2}_T5,${sceneid2}_T6,${sceneid2}
}_T7
i.gensig trainingmap=Time_1 group=Time_2 subgroup=Time_2 signaturefile=Time_2_diff_sig
r.mask -r
r.mask --overwrite raster=Mask_tot
i.maxlik --overwrite group=Time_2 subgroup=Time_2 signaturefile=Time_2_diff_sig
output=Time_2_diff
r.category map=Time_2_diff separator=: rules=Time_1_${site}.lab # mettre les catégories de
Time_1 car cette classification se base dessus ET utiliser la symbologie de Time_1 si les classes ne
sont pas les mêmes!!
```

Change_detection2.2.sh - Méthode des échantillons aléatoires

```
#!/bin/bash
```

```
#####
```

```

#
#
# VERSION:      0.0.3
#
# AUTHOR(S):    Natacha Klein
#
# PURPOSE:      Grass 7.0 functions for change detection and classification, fourth method:
échantillons aléatoires (Bourino)
#
# REQUIREMENT:  - Python 2.7 (tested)
#                - Grass 7.0 (sudo apt-get install grass7 grass7-dev gdal-bin)
#                > grass installed twice !
#                - label files for the classification images (.lab)
#
# DATE:         01.07.2016
#
#
#####

export sceneid1=LC81960282013227LGN00
export sceneid2=LC81960282015217LGN00
export class_num=8
export site=Geneva
export threshold=0.05
export minsize=20
# PARAMETERS to be used in the stratified sampling part
export split_nb=8 # nb of parts the vector will be splitted
export once=true # If false 1 feature can be used more than once
export prfx='Time_2.'
export sufx='_reclass'

# Set region
g.region raster=${sceneid1}_MTLFmask

# Masquer les nuages
r.mapcalc "Mask_1 = (( ${sceneid1}_MTLFmask == 0) + ( ${sceneid1}_MTLFmask == 1) +
( ${sceneid1}_MTLFmask == 3 ))" --overwrite
r.mapcalc "Mask_2 = (( ${sceneid2}_MTLFmask == 0) + ( ${sceneid2}_MTLFmask == 1) +
( ${sceneid2}_MTLFmask == 3 ))" --overwrite

# 0 = no data

```



```

r.mapcalc "Mask_1_nodata = if ( Mask_1 == 0, null(), 1) " --overwrite
r.mapcalc "Mask_2_nodata = if ( Mask_2 == 0, null(), 1) " --overwrite

# Classification t1 + labels
r.mask raster=Mask_1_nodata
i.group --overwrite group=Time_1 subgroup=Time_1
input=${sceneid1}_T2,${sceneid1}_T3,${sceneid1}_T4,${sceneid1}_T5,${sceneid1}_T6,${sceneid1}_T7
i.cluster --overwrite group=Time_1 subgroup=Time_1 signaturefile=Time_1_sig
classes=${class_num} reportfile=Time_1_Report
i.maxlik --overwrite group=Time_1 subgroup=Time_1 signaturefile=Time_1_sig output=Time_1
r.reclass.area --overwrite input=Time_1 output=Time_1_resize value=0.18 mode=lesser
method=rmarea # pour éliminer les éléments trop petits, ceci prend la valeur qui est en majorité
r.mode --overwrite base=Time_1_resize cover=Time_1 output=Time_1_resize_class
r.category map=Time_1_resize_class separator=: rules=Time_1_${site}.lab # il faut que les labels
commencent par un numéro/valeur

# Tests for training areas map

# Segmentation
i.segment --overwrite group=Time_1 output=Time_1_seg threshold=${threshold}
minsize=${minsize} memory=4096
r.mode --overwrite base=Time_1_seg cover=Time_1_resize output=Time_1_seg_class # pour
récupérer les valeurs de la classification dans les éléments de la segmentation
r.to.vect --overwrite input=Time_1_seg_class output=Time_1_seg_vect type=area
# v.out.ogr --overwrite input=Time_1_seg_vect output=Time_1_seg_vect format=ESRI_Shapefile

# Echantillons aléatoires
startt=$(date +%s)

# Stratified sampling start from here
# Loop through categories
eval `v.db.univar -g map=Time_1_seg_vect column=value`
export min_val=${min}
export max_val=${max}
for c in `seq ${min_val} ${max_val}`
do
    v.extract input=Time_1_seg_vect where=value=${c} output=tmp --overwrite --quiet
    eval `v.info -t map=tmp`
    export feat_cnt=${centroids}
    feat_split=$((feat_cnt/split_nb))

```

```

# Interrupt script if there are too few features
if [ ${feat_cnt} -lt ${split_nb} ]
then
    g.message message="! SCRIPT INTERRUPTED: there are less features than split number
for category ${c}!"
    exit
fi
# Loop through splits
for s in `seq 1 ${split_nb}`
do
    g.message message="* Split ${c}_${s}: ${feat_cnt} features in total splitted into
${feat_split}"
    if ! ${once}
    then
        v.extract input=tmp output=tmp_${c}_${s} random=${feat_split} --overwrite
--quiet
    else
        v.extract input=tmp output=tmp_${c}_${s} random=${feat_split} --overwrite
--quiet
        v.select -r ainput=tmp binput=tmp_${c}_${s} output=tmp2 operator=within
--overwrite --quiet
        if [ ${s} -eq $((split_nb - 1)) ]
        then
            eval `v.info -t map=tmp2`
            export feat_remain=${centroids}
            g.message message="* Split ${c}_${s}((s + 1)): ${feat_cnt} features in total
splitted into ${feat_remain}"
            g.rename vector=tmp2,tmp_${c}_${s}((s + 1)) --overwrite --quiet
            break
        fi
        g.rename vector=tmp2,tmp --overwrite --quiet

    fi
done
done

# Merge the categories
for s in `seq 1 ${split_nb}`
do
    g.message message="* Merging sample ${s}"
    v.patch -e input=`g.list type=vector sep=, pattern=tmp_*_${s}` out=tmp --overwrite --quiet

```

```

v.clean input=tmp output=sample_${s}on${split_nb} tool='snap,break,rmdupl,rms'
--overwrite --quiet
done

g.message message="* $(basename -- "$0") script ran in $((`date +%s`-${startt})) seconds" #
Ext4Part: script complet exécuté en 693 secondes (environ 12 minutes)

# Cloud mask of both scenes
r.mapcalc "Mask_tot = Mask_1_nodata + Mask_2_nodata" --overwrite
r.mask --overwrite raster=Mask_tot

# Loop classifications scene 2 based om random samples
i.group --overwrite group=Time_2 subgroup=Time_2
input=${sceneid2}_T2,${sceneid2}_T3,${sceneid2}_T4,${sceneid2}_T5,${sceneid2}_T6,${sceneid2}
}_T7
for a in $(seq 1 ${class_num})
do
v.to.rast --overwrite input=sample_${a}on${class_num} type=area
output=sample_${a}on${class_num} use=attr attribute_column=value label_column=label
memory=4096
i.gensig trainingmap=sample_${a}on${class_num} group=Time_2 subgroup=Time_2
signaturefile=Time_2_sample${a}
i.maxlik --overwrite group=Time_2 subgroup=Time_2 signaturefile=Time_2_sample${a}
output=Time_2.${a}
r.reclass --overwrite input=Time_2.${a} output=Time_2.${a}_reclass
rules=Time_2_${site}_reclass.lab
r.category map=Time_2.${a}_reclass separator=: rules=Time_1_${site}_final.lab
done

g.list type=raster pattern=Time_2.*_reclass separator=comma

# Change vs. no-change map (no change = 0)
startt=$(date +%s)

r.series -n input=`g.list type=raster pattern=Time_2.*_reclass separator=comma`
output=bourino_unchange_map method=range --overwrite # unité = range, le nombre de valeurs
qui varient

g.message message="* Done in "$((`date +%s`-${startt}))" seconds"

r.mapcalc "bourino_unchange_map2 = if( bourino_unchange_map != 0 )" --overwrite # 0 = no

```

change , pour convertir de DCELL à integer

Classification of scene 2 based on unchanged maps training areas

```
g.remove -f type=group name=Time_2
```

```
i.group --overwrite group=Time_2 subgroup=Time_2  
input=${sceneid2}_T2,${sceneid2}_T3,${sceneid2}_T4,${sceneid2}_T5,${sceneid2}_T6,${sceneid2}_T7
```

```
r.mask --overwrite raster=bourino_unchange_map2 maskcats=0 # pour masquer les zones de surface changée
```

```
i.gensig trainingmap=Time_1 group=Time_2 subgroup=Time_2 signaturefile=Time_2_final_sig
```

```
r.mask -r
```

```
r.mask --overwrite raster=Mask_tot
```

```
i.maxlik --overwrite group=Time_2 subgroup=Time_2 signaturefile=Time_2_final_sig  
output=Time_2_final
```

```
r.category map=Time_2_final separator=: rules=Time_1_${site}.lab # mettre les catégories de Time_1 car cette classification se base dessus ET utiliser la symbologie de Time_1 si les classes ne sont pas les mêmes!!
```

script exécuté en 33 minutes et 6 secondes pour Papua

script exécuté en 138 minutes et 20 secondes pour Dubti

script exécuté en 5 minutes et 32 secondes pour Geneva

Change_detection2.3.sh - Méthode de la zone d'entraînement totale

```
#!/bin/bash
```

```
#####
```

```
#
```

```
#
```

```
# VERSION: 0.0.3
```

```
#
```

```
# AUTHOR(S): Natacha Klein
```

```
#
```

```
# PURPOSE: Grass 7.0 functions for change detection and classification, fifth method: la zone d'entraînement totale (classification 1 on 2)
```

```
#
```

```

# REQUIREMENT: - Python 2.7 (tested)
#               - Grass 7.0 (sudo apt-get install grass7 grass7-dev gdal-bin)
#               > grass installed twice !
#               - label files for the classification images (.lab)
#
# DATE:         01.07.2016
#
#
#####

export sceneid1=LC81960282013227LGN00
export sceneid2=LC81960282015217LGN00
export site=Geneva
export class_num=8

# Set region
g.region raster=${sceneid1}_MTLFmask

# Masquer les nuages
r.mapcalc "Mask_1 = (( ${sceneid1}_MTLFmask == 0) + ( ${sceneid1}_MTLFmask == 1) +
( ${sceneid1}_MTLFmask == 3 ))" --overwrite
r.mapcalc "Mask_2 = (( ${sceneid2}_MTLFmask == 0) + ( ${sceneid2}_MTLFmask == 1) +
( ${sceneid2}_MTLFmask == 3 ))" --overwrite

# 0 = no data
r.mapcalc "Mask_1_nodata = if ( Mask_1 == 0, null(), 1) " --overwrite
r.mapcalc "Mask_2_nodata = if ( Mask_2 == 0, null(), 1) " --overwrite

# Classification t1 + labels
g.remove -f type=group name=Time_1,Time_2
r.mask --overwrite raster=Mask_1_nodata
i.group --overwrite group=Time_1 subgroup=Time_1
input=${sceneid1}_T2,${sceneid1}_T3,${sceneid1}_T4,${sceneid1}_T5,${sceneid1}_T6,${sceneid1}
}_T7
i.cluster --overwrite group=Time_1 subgroup=Time_1 signaturefile=Time_1_sig
classes=${class_num} reportfile=Time_1_Report
i.maxlik --overwrite group=Time_1 subgroup=Time_1 signaturefile=Time_1_sig output=Time_1
r.category map=Time_1 separator=: rules=Time_1_${site}.lab # il faut que les labels commencent
par un numéro/valeur

# Cloud mask of both scenes

```

```

r.mapcalc "Mask_tot = Mask_1_nodata + Mask_2_nodata" --overwrite
r.mask --overwrite raster=Mask_tot

# Classification t2 based on map t1
i.group --overwrite group=Time_2 subgroup=Time_2
input=${sceneid2}_T2,${sceneid2}_T3,${sceneid2}_T4,${sceneid2}_T5,${sceneid2}_T6,${sceneid2}_T7
i.gensig trainingmap=Time_1 group=Time_2 subgroup=Time_2 signaturefile=Time_2_gensig
i.maxlik --overwrite group=Time_2 subgroup=Time_2 signaturefile=Time_2_gensig
output=Time_2_gensig
r.category map=Time_2_gensig separator=: rules=Time_1_${site}.lab # il faut que les labels
commencent par un numéro/valeur

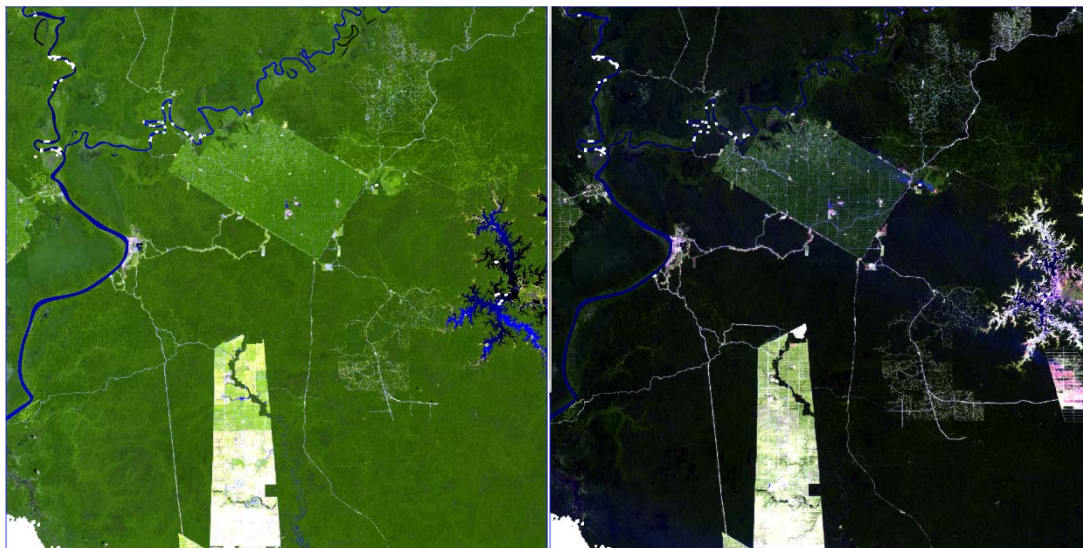
# Map unchanged areas
r.mapcalc "unchange_map_gensig = if(@Time_1 == @Time_2_gensig)" --overwrite # @ = par
rapprt aux labels; 0 = zones changées (en jaune) et 1 = zones inchangées (en rouge)

```

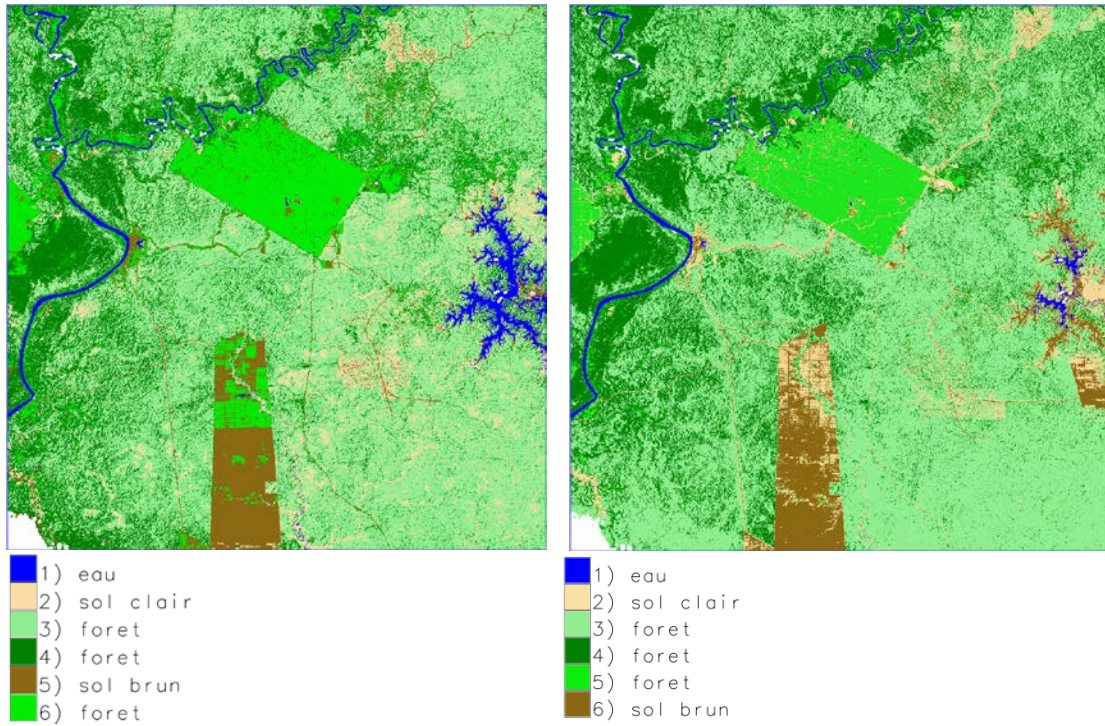
Annexe 3: Résultats des tests par sites

Comparaison post-classification

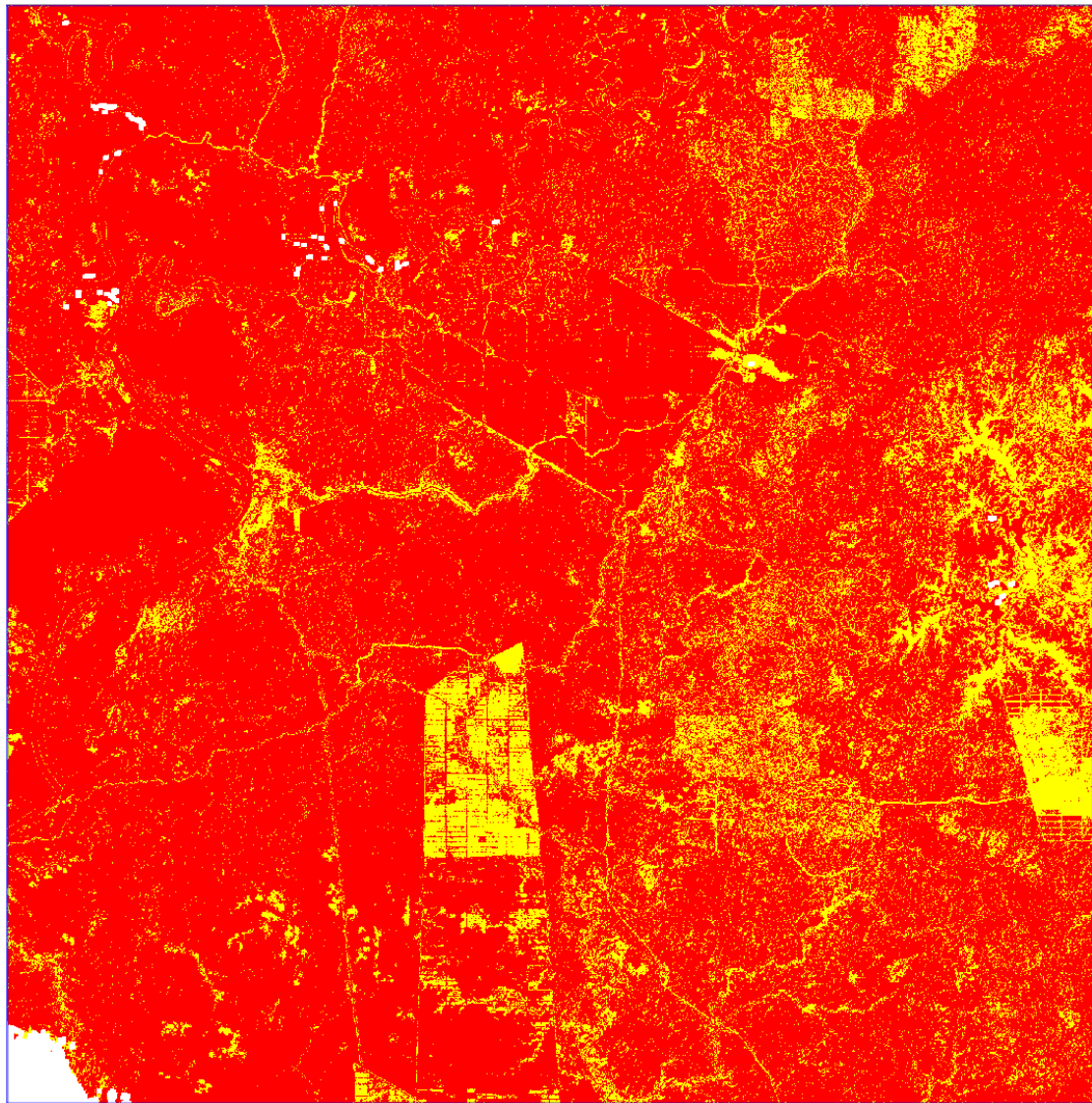
Papua



Compositions colorées en fausses couleurs (764) (+ i.colors.enhance)



Résultats Papua: Classification scène 1 et classification scène 2 (de gauche à droite)



0) Changement
1) Non-changement

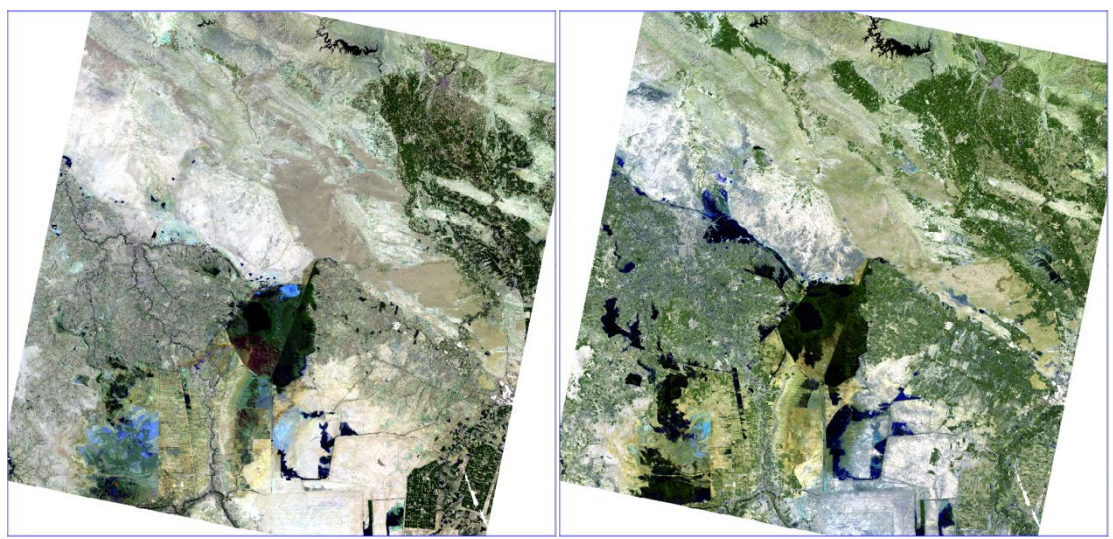
Résultat Papua: Image de surfaces changées vs. surfaces inchangées selon méthode post-classification


```
r.report map=unchange_map@PERMANENT units=h,c,p
+-----+
|                                RASTER MAP CATEGORY REPORT                                |
|LOCATION: papua                                                            Fri Jul  8 10:39:25 2016|
+-----+
|          north: -711375      east: 476415                                |
|REGION    south: -764865     west: 423435                                |
|          res:      30       res:      30                                |
+-----+
|MASK: Mask_tot in PERMANENT, categories 2                                |
+-----+
|MAP: (untitled) (unchange_map@PERMANENT in PERMANENT)                    |
+-----+
|          Category Information                                           |
| #|description                    | hectares| cell| %  |
+-----+-----+-----+-----+-----+
| 0| . . . . .                      | 41,328.81| 459209| 14.58|
| 1| . . . . .                      |240,466.14|2671846| 84.85|
|*|no data. . . . .                | 1595.07| 17723| 0.56|
+-----+-----+-----+-----+-----+
|TOTAL                                     |283,390.02|3148778|100.00|
+-----+

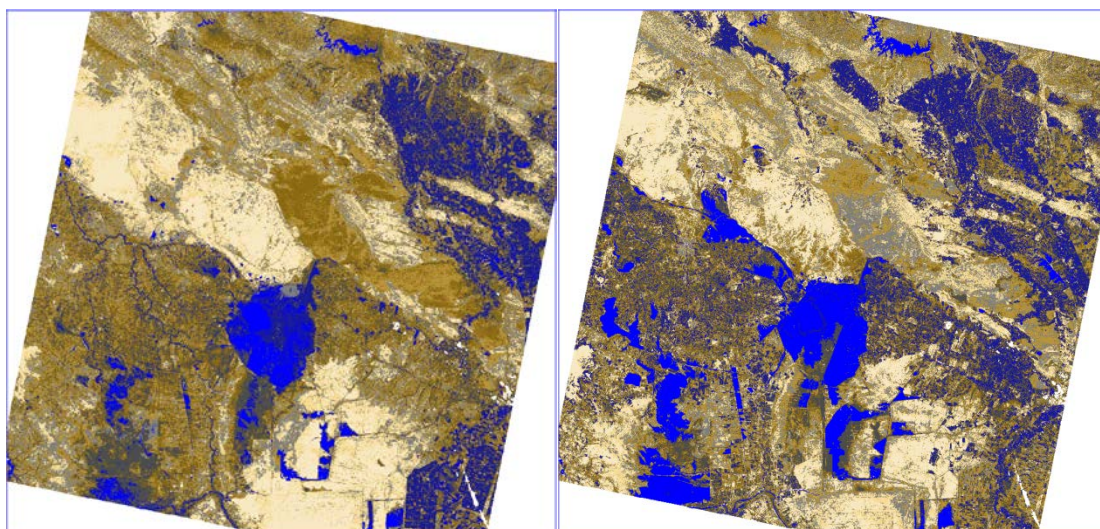
```

Résultat Papua: Statistiques de l'image de changement

Irak

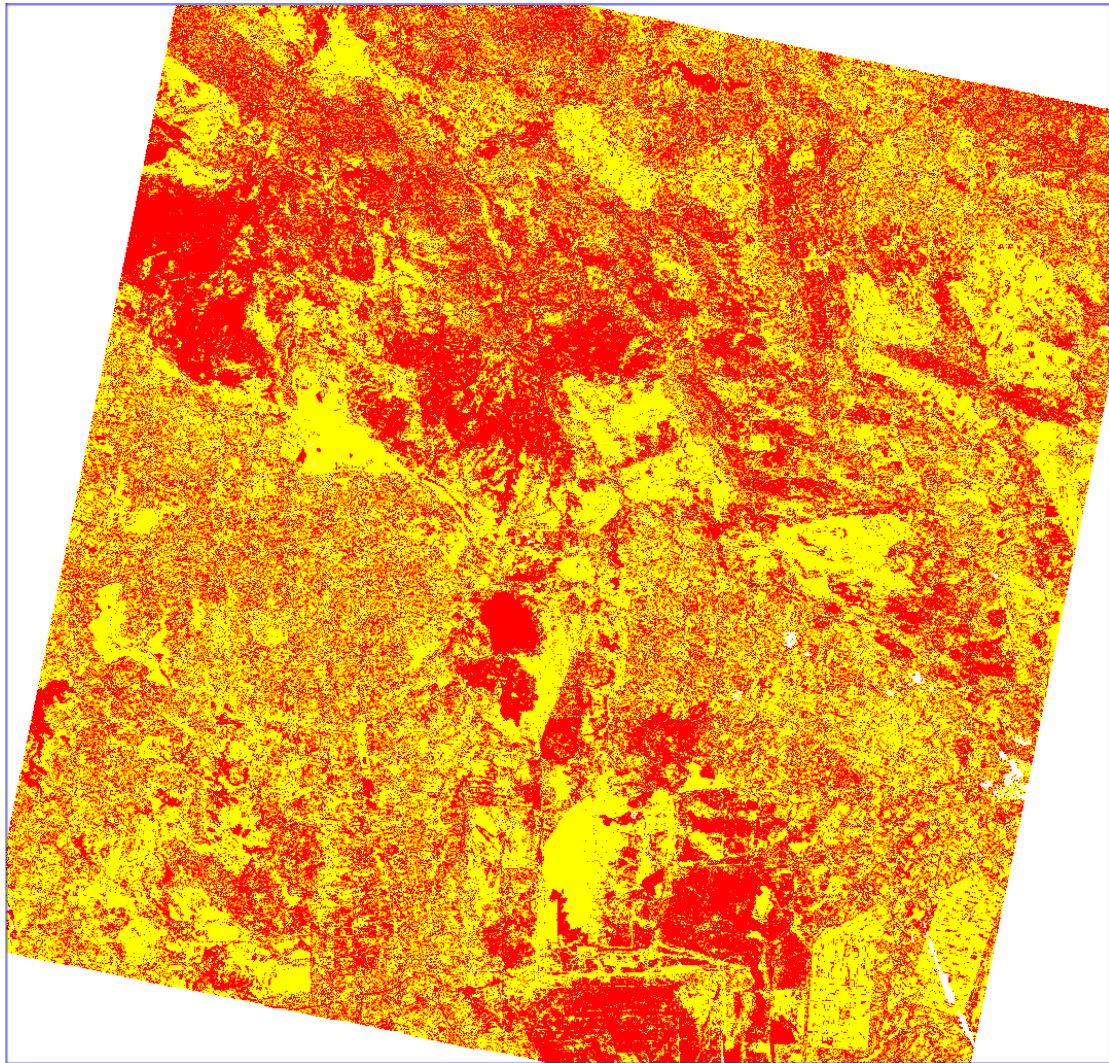


Compositions colorées en fausses couleurs (764) (+ i.colors.enhance)



- 1) eau
- 2) marais
- 3) sol humide
- 4) sol brun
- 5) sol brun
- 6) sol gris
- 7) sol clair
- 8) sol clair

Résultats Irak: Classification scène 1 et classification scène 2 (de gauche à droite)



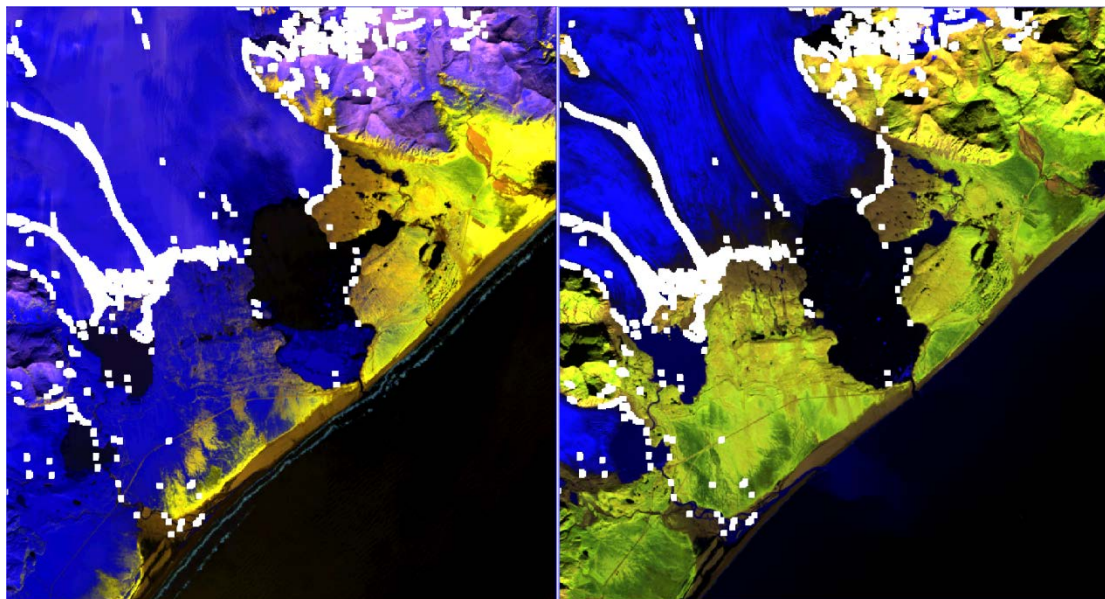
0) Changement
1) Non-changement

Résultat Irak: Image de surfaces changées vs. surfaces inchangées selon méthode post-classification

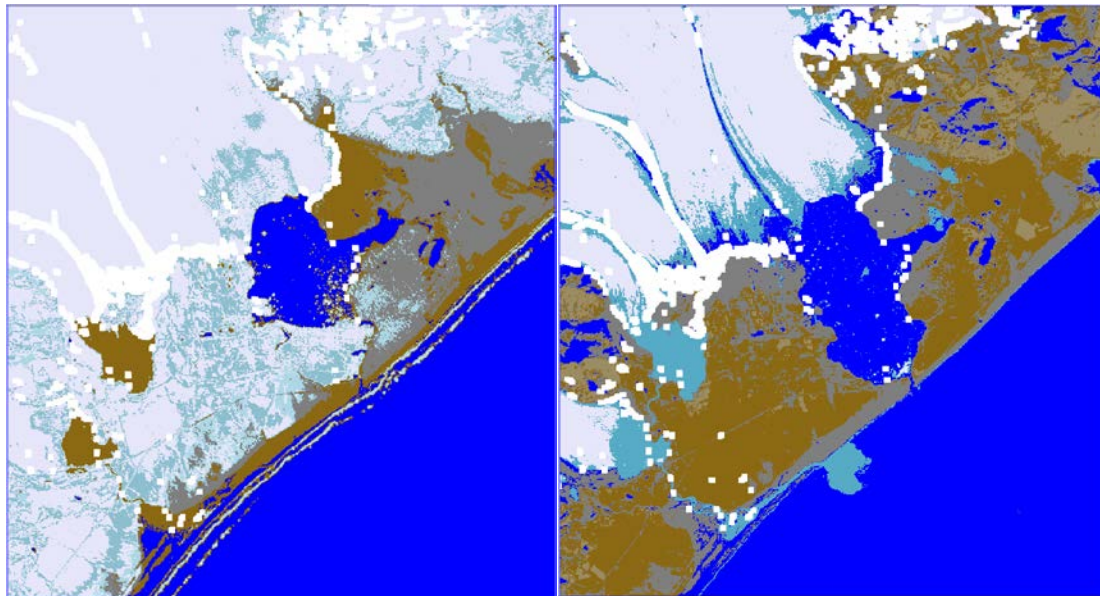
```
r.report map=unchange_map@PERMANENT units=h,c,p
+-----+
|                                RASTER MAP CATEGORY REPORT                                |
|LOCATION: irak_water                                                    Fri Jul  8 10:45:15 2016|
+-----+
|           north: 3614835      east: 862695                               |
|REGION      south: 3415665     west: 654795                               |
|           res:      30        res:      30                               |
+-----+
|MASK: Mask_tot in PERMANENT, categories 2                               |
+-----+
|MAP: (untitled) (unchange_map@PERMANENT in PERMANENT)                  |
+-----+
|                                Category Information                               |
| #|description                               | hectares| cell| %  |
|-----|-----|-----|-----|-----|
| 0| . . . . .                               | 1,943,591|21595460| 46.94|
| 1| . . . . .                               | 1,561,618|17351315| 37.71|
|*|no data. . . . .                         |   635,535| 7061495| 15.35|
+-----+
|TOTAL                               | 4,140,744|46008270|100.00|
+-----+
```

Résultat Irak: Statistiques de l'image de changement

Iceland



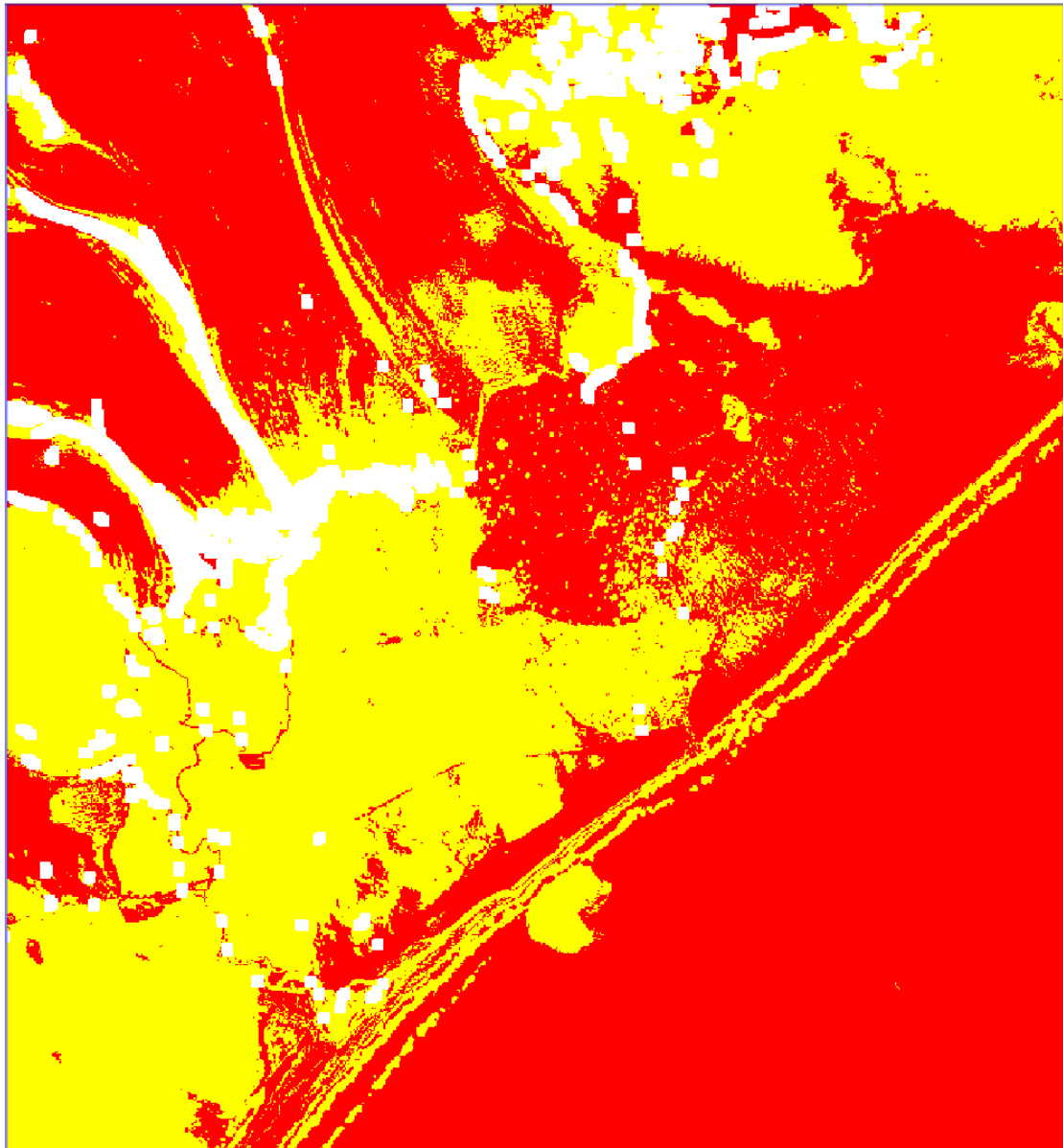
Compositions colorées en fausses couleurs (764) (+ i.colors.enhance)



- 1) eau
- 2) sol
- 3) sol
- 4) glace fonce
- 5) glace fonce
- 6) glace fonce
- 7) neige/glace
- 8) neige/glace

- 1) eau
- 2) eau
- 3) glace fonce
- 4) sol
- 5) sol
- 6) sol
- 7) neige/glace
- 8) neige/glace

Résultats Iceland: Classification scène 1 et classification scène 2 (de gauche à droite)



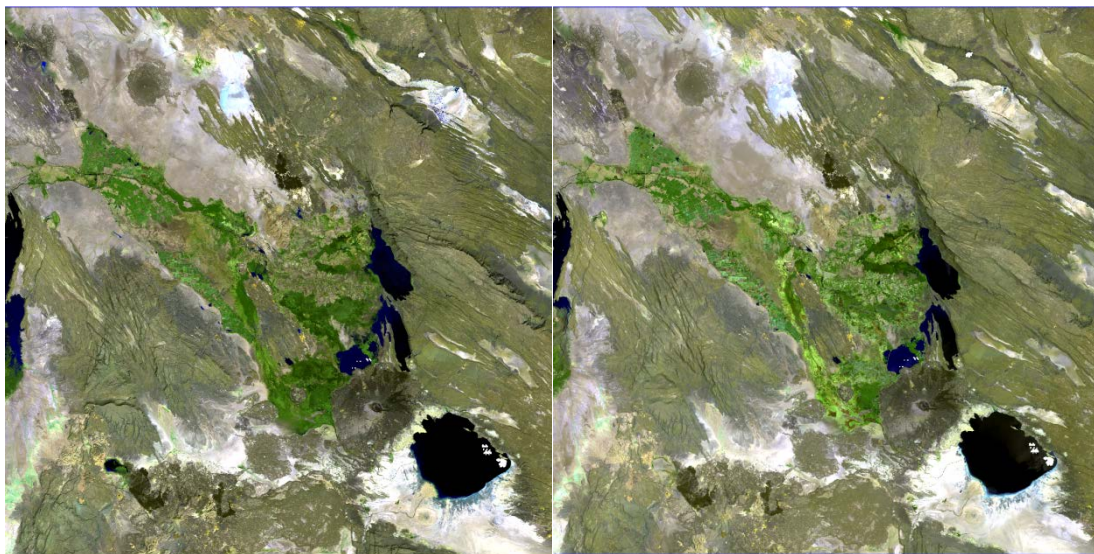
0) Changement
1) Non-changement

Résultat Iceland: Image de surfaces changées vs. surfaces inchangées selon méthode post-classification

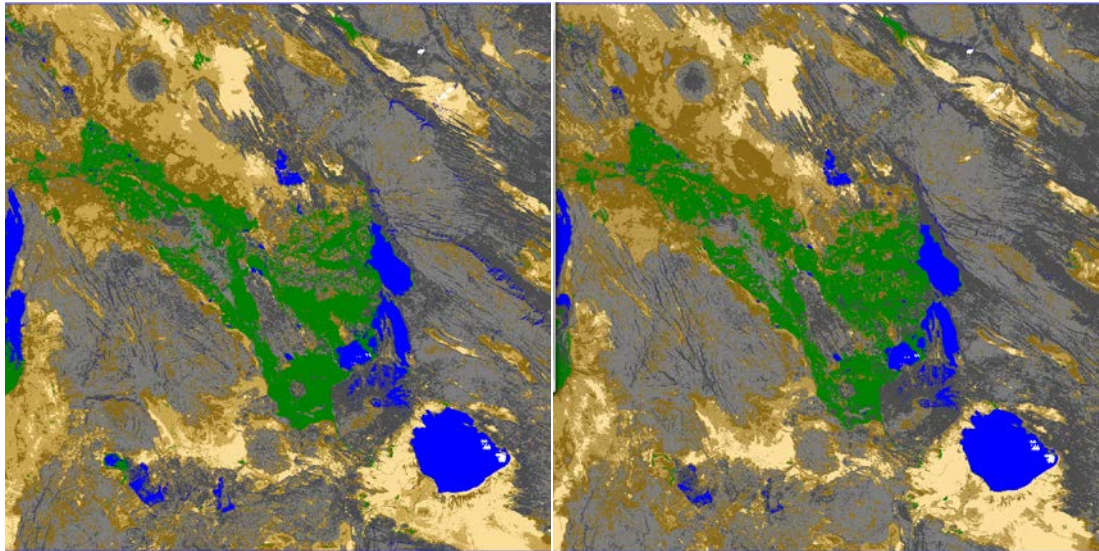
```
r.report map=unchange_map@PERMANENT units=h,c,p
+-----+
|                                     RASTER MAP CATEGORY REPORT                                     |
|LOCATION: iceland                                                                Fri Jul  8 10:47:21 2016|
+-----+
|          north: 7117095    east: 449865                                     |
|REGION    south: 7094175    west: 428715                                     |
|          res:      30      res:      30                                     |
+-----+
|MASK: Mask_tot in PERMANENT, categories 2                                     |
+-----+
|MAP: (untitled) (unchange_map@PERMANENT in PERMANENT)                         |
+-----+
|          Category Information                                     | cell| % |
| #|description                                     | hectares| count| cover|
+-----+
|0| . . . . . |17,577.180|195302| 36.26|
|1| . . . . . |27,585.000|306500| 56.90|
|*|no data. . . . . | 3313.620| 36818|  6.84|
+-----+
|TOTAL                                     |48,475.800|538620|100.00|
+-----+
```

Résultat Iceland: Statistiques de l'image de changement

Dubti

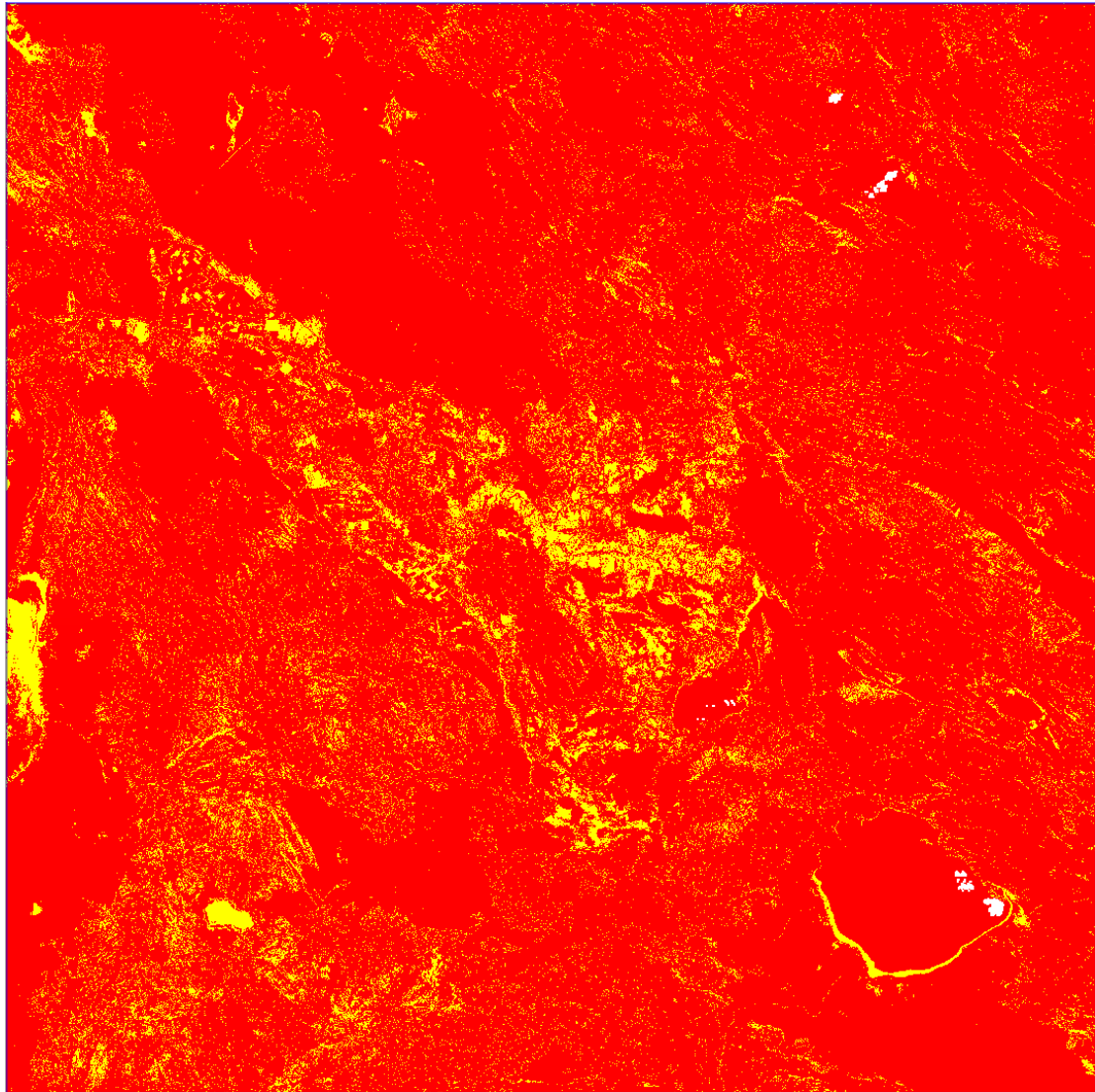


Compositions colorées en fausses couleurs (764) (+ i.colors.enhance)



- 1) eau
- 2) sol fonce
- 3) sol fonce
- 4) vegetation
- 5) sol clair
- 6) sol clair
- 7) sol clair

Résultat Dubti: Classification scène 1 et classification scène 2 (de gauche à droite)



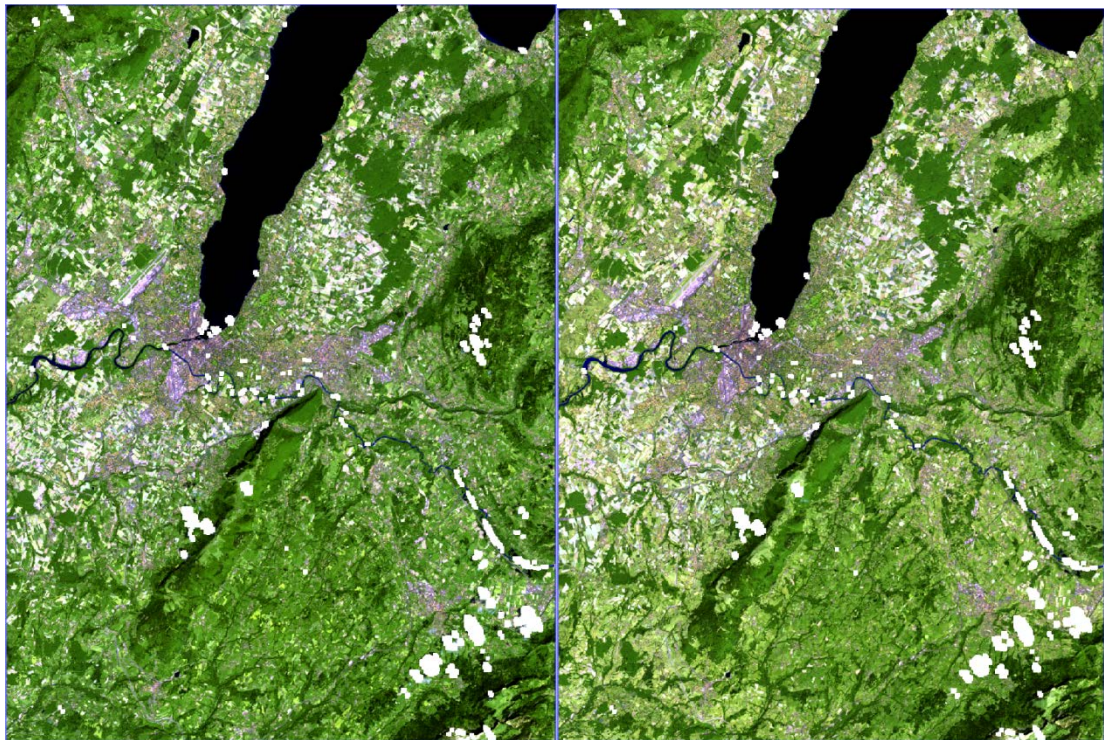
0) Changement
1) Non-changement

Résultat Dubti: Image de surfaces changées vs. surfaces inchangées selon méthode post-classification

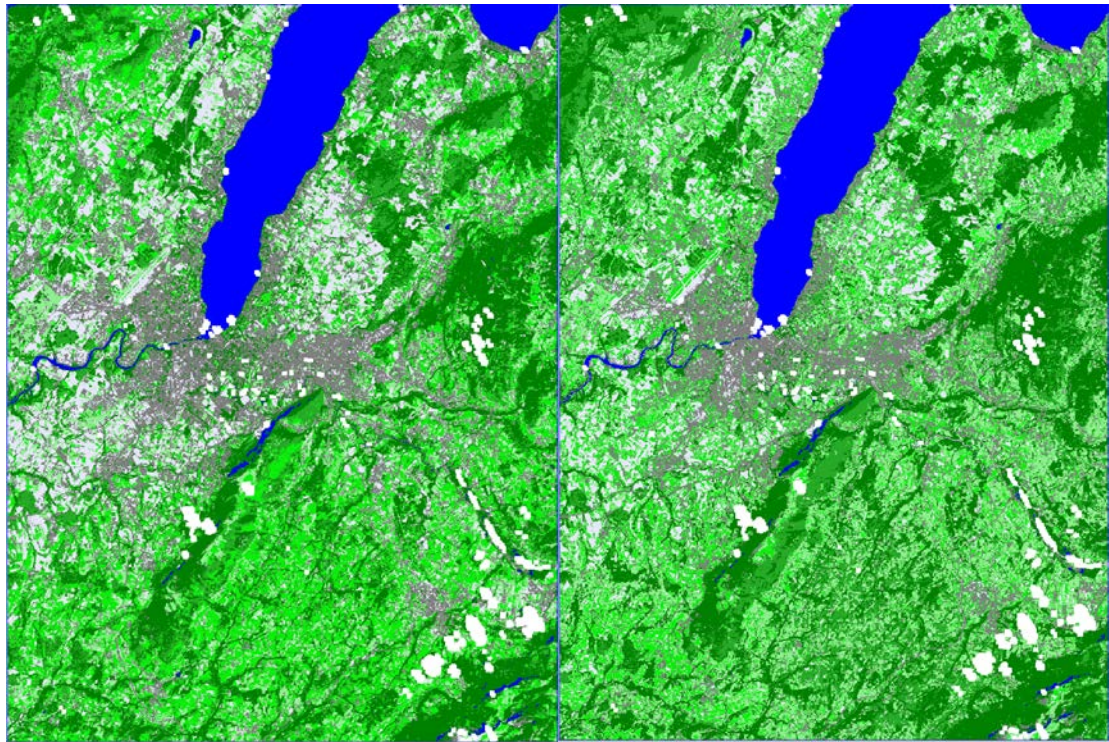
```
r.report map=unchange_map@PERMANENT units=h,c,p
+-----+
|                                     RASTER MAP CATEGORY REPORT                                     |
|LOCATION: dubti                                                                 Fri Jul  8 10:50:32 2016|
+-----+
|          north: 1330905      east: 824145                                     |
|REGION    south: 1218435     west: 711675                                     |
|          res:      30       res:      30                                     |
+-----+
|MASK: Mask_1_nodata in PERMANENT, categories 1                               |
+-----+
|MAP: (untitled) (unchange_map@PERMANENT in PERMANENT)                       |
+-----+
|          Category Information                                               | cell|  % |
|#|description                       | hectares| count| cover|
+-----+
|0| . . . . .                          | 107,689| 1196540| 8.51|
|1| . . . . .                          | 1,156,351|12848346| 91.41|
|*|no data. . . . .                   | 910| 10115| 0.07|
+-----+
|TOTAL                               | 1,264,950|14055001|100.00|
+-----+
```

Résultat Dubti: Statistiques de l'image de changement

Geneva

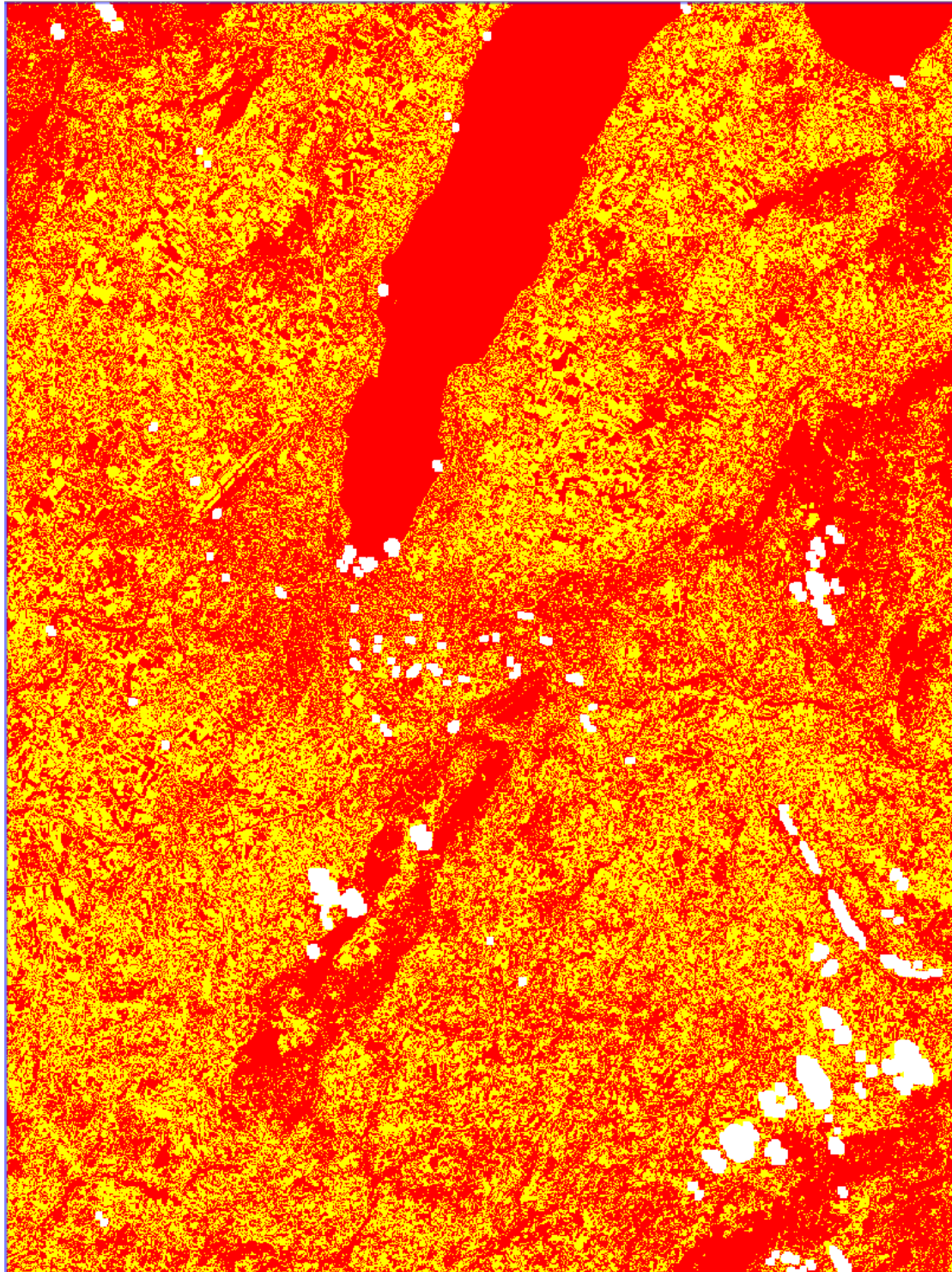


Compositions colorées en fausses couleurs (764) (+ i.colors.enhance)



- 1) eau
- 2) foret
- 3) foret
- 4) foret/herbe
- 5) urbain
- 6) sol/herbe
- 7) foret/herbe
- 8) champs

Résultats Geneva: Classification scène 1 et classification scène 2 (de gauche à droite)



0) Changement
1) Non-changement

Résultat Geneva: Image de surfaces changées vs. surfaces inchangées selon méthode post-classification

```
r.report map=unchange_map@PERMANENT units=h,c,p
+-----+
|                                     RASTER MAP CATEGORY REPORT                                     |
| LOCATION: geneva                                                             Fri Jul  8 10:54:30 2016 |
|-----|
|           north: 5140095           east: 762525                               |
| REGION    south: 5099415           west: 732195                               |
|           res:           30           res:           30                               |
|-----|
| MASK: Mask_tot in PERMANENT, categories 2                                   |
|-----|
| MAP: (untitled) (unchange_map@PERMANENT in PERMANENT)                       |
|-----|
|                                     Category Information                                     |
| #|description                                                                | hectares| cell| % |
|-----|-----|-----|-----|
| 0| . . . . . | 49,684.77| 552053| 40.27|
| 1| . . . . . | 71,462.07| 794023| 57.92|
| *|no data. . . . . | 2235.60| 24840| 1.81|
|-----|-----|-----|-----|
| TOTAL                                                                |123,382.44|1370916|100.00|
+-----+
```

Résultat Geneva: Statistiques de l'image de changement

Analyse en composantes principales (ACP)

Papua

```
Computing covariance matrix...
Calculating principal components...
Rescaling to range 0,255...
Eigen values, (vectors), and [percent importance]:
PC1 590731.73 ( 0.0515, 0.0754, 0.1495,-0.1037, 0.3062, 0.1981, 0.0796, 0.1271, 0.2559, 0.0147, 0.6884, 0.5134) [47.22%]
PC2 535173.15 ( 0.0155, 0.0044, 0.0523,-0.7931,-0.2494,-0.0697, 0.0346, 0.0401, 0.0832,-0.5340,-0.0648, 0.0433) [42.78%]
PC3 62691.78 ( 0.0540, 0.0863, 0.1194, 0.2271, 0.5780, 0.3147,-0.0735,-0.0969,-0.0899,-0.6457,-0.1957,-0.1120) [ 5.01%]
PC4 34130.23 ( 0.0765, 0.0568, 0.2010,-0.5249, 0.4167, 0.3072,-0.0899,-0.0787,-0.1769, 0.5214,-0.1267,-0.2712) [ 2.73%]
PC5 18374.56 (-0.1966,-0.3478,-0.5000,-0.1195, 0.0858, 0.0095,-0.2855,-0.3826,-0.4835,-0.0695, 0.2943, 0.1248) [ 1.47%]
PC6 4100.70 (-0.1354,-0.3843,-0.4965,-0.0788, 0.2536, 0.1408, 0.5656, 0.3250, 0.2036, 0.0311,-0.1573,-0.0664) [ 0.33%]
PC7 2879.53 (-0.0130, 0.1133,-0.0361, 0.0170,-0.0167,-0.1125, 0.1080, 0.0989,-0.0124,-0.1269, 0.5878,-0.7676) [ 0.23%]
PC8 1876.59 ( 0.2533, 0.4322,-0.1259, 0.0081,-0.1147, 0.0948, 0.4020, 0.2457,-0.6774,-0.0216, 0.0187, 0.1713) [ 0.15%]
PC9 547.33 ( 0.0427, 0.4926,-0.5608,-0.0678, 0.1294,-0.0464,-0.4990, 0.3311, 0.2266, 0.0345,-0.0801, 0.0025) [ 0.04%]
PC10 359.59 (-0.3154, 0.1111, 0.1429,-0.0771, 0.4551,-0.7766, 0.1124, 0.0683,-0.1457, 0.0135,-0.0559, 0.0991) [ 0.03%]
PC11 163.63 (-0.8640, 0.1869, 0.1465, 0.0196,-0.1656, 0.3386,-0.0407, 0.2056,-0.1033,-0.0069,-0.0084, 0.0069) [ 0.01%]
PC12 89.96 (-0.1388, 0.4747,-0.2242,-0.0367,-0.0004, 0.0332, 0.3755,-0.6985, 0.2660, 0.0455,-0.0275, 0.0072) [ 0.01%]
```

Résultats de l'ACP pour Papua

Irak

```
Computing covariance matrix...
Calculating principal components...
Rescaling to range 0,255...
Eigen values, (vectors), and [percent importance]:
PC1 5107744.08 ( 0.1349, 0.2016, 0.3020, 0.2233, 0.3906, 0.4052, 0.1322, 0.1894, 0.2823, 0.2250, 0.3936, 0.3805) [74.73%]
PC2 971782.64 ( 0.1463, 0.2114, 0.3197, 0.1932, 0.3831, 0.3521,-0.0945,-0.1361,-0.2176,-0.1937,-0.4536,-0.4533) [14.22%]
PC3 401609.36 ( 0.0803, 0.0802, 0.1080,-0.3510,-0.0285, 0.0815, 0.1251, 0.1253, 0.2253,-0.8499, 0.0203, 0.2002) [ 5.88%]
PC4 180171.06 (-0.0617,-0.0964,-0.0267,-0.8019, 0.1627, 0.4039,-0.1320,-0.1500,-0.2074, 0.2569, 0.0633, 0.0248) [ 2.64%]
PC5 106493.22 (-0.2283,-0.2770,-0.2988, 0.3133, 0.3053, 0.1268,-0.2662,-0.3493,-0.3562,-0.3245, 0.2596, 0.2865) [ 1.56%]
PC6 32746.43 ( 0.2475, 0.3321, 0.4995,-0.1001,-0.0956,-0.4292,-0.2149,-0.2826,-0.3783,-0.0074, 0.2840, 0.1604) [ 0.48%]
PC7 13319.93 ( 0.5329, 0.1741,-0.4449, 0.0009, 0.0421, 0.0435, 0.5170, 0.0927,-0.4514,-0.0070, 0.0552, 0.0381) [ 0.19%]
PC8 11411.55 ( 0.0664, 0.1407, 0.0243, 0.1650,-0.6359, 0.4761,-0.1587,-0.1054,-0.1113, 0.0433,-0.3076, 0.4144) [ 0.17%]
PC9 6447.79 ( 0.0411, 0.0613,-0.0398, 0.0851,-0.3770, 0.3257,-0.0426,-0.1153, 0.0642,-0.1127, 0.6160,-0.5711) [ 0.09%]
PC10 2153.39 ( 0.5287, 0.1686,-0.3992,-0.0422, 0.1320,-0.0746,-0.5400,-0.1740, 0.4267, 0.0291,-0.0551, 0.0061) [ 0.03%]
PC11 896.39 ( 0.2740,-0.4502, 0.2009, 0.0174,-0.0355, 0.0044, 0.4055,-0.6587, 0.2718, 0.0445,-0.0700, 0.0349) [ 0.01%]
PC12 351.67 (-0.4383, 0.6534,-0.2359,-0.0439, 0.0636,-0.0398, 0.2705,-0.4565, 0.1817, 0.0280,-0.0424, 0.0250) [ 0.01%]
```

Résultats de l'ACP pour Irak

Iceland

```

Calculating principal components...
Rescaling to range 0,255...
PC1 51523071.21 ( 0.4829, 0.4751, 0.5142, 0.4692,-0.0061, 0.0017, 0.1040, 0.1090, 0.1254, 0.1294, 0.0293, 0.0211) [90.33%]
PC2 3405291.67 ( 0.0314, 0.0593, 0.1014, 0.1843, 0.0596, 0.0445,-0.5249,-0.5107,-0.5236,-0.1390, 0.2879, 0.1818) [ 5.97%]
PC3 1693173.88 ( 0.1981, 0.1482, 0.0974,-0.1881,-0.3770,-0.2396,-0.0322,-0.0841,-0.1285,-0.6404,-0.4465,-0.2456) [ 2.97%]
PC4 202273.52 (-0.0856,-0.0736,-0.0345, 0.3162, 0.5998, 0.4420, 0.0947, 0.0370, 0.0649,-0.5108,-0.2345,-0.0224) [ 0.35%]
PC5 142454.43 (-0.0479,-0.0847,-0.0469, 0.2439, 0.1149,-0.0283,-0.1520,-0.1788,-0.2064, 0.4798,-0.5174,-0.5650) [ 0.25%]
PC6 38174.62 ( 0.3116, 0.2965, 0.1050,-0.7177, 0.3675, 0.3079,-0.1424,-0.0698,-0.0048, 0.1529,-0.0648,-0.0788) [ 0.07%]
PC7 13595.45 ( 0.5711,-0.1198,-0.4745, 0.0610, 0.0483, 0.0027, 0.4275, 0.0584,-0.4660, 0.0079, 0.1501,-0.0491) [ 0.02%]
PC8 10119.79 ( 0.5035,-0.4079,-0.1891, 0.0863,-0.1368, 0.1140,-0.3089,-0.2833, 0.5321, 0.0246,-0.1379, 0.1634) [ 0.02%]
PC9 4202.34 ( 0.0262,-0.5698, 0.6274,-0.1406,-0.1081, 0.2123, 0.3569,-0.1651,-0.2029, 0.0534,-0.0489, 0.0583) [ 0.01%]
PC10 3472.35 ( 0.1779,-0.3525, 0.2098,-0.0401, 0.3304,-0.4113,-0.2901, 0.3387, 0.0528,-0.1530, 0.3542,-0.4119) [ 0.01%]
PC11 2287.31 ( 0.0587,-0.1060, 0.0024, 0.0414,-0.3121, 0.4440,-0.4072, 0.6536,-0.2675, 0.0392,-0.1229, 0.0892) [ 0.00%]
PC12 1595.17 ( 0.0667,-0.0860, 0.0540,-0.0401, 0.3270,-0.4746,-0.0694, 0.1712,-0.1758, 0.1047,-0.4532, 0.6083) [ 0.00%]
    
```

Résultats de l'ACP pour Iceland

Dubti

```

Computing covariance matrix...
Calculating principal components...
Rescaling to range 0,255...
Eigen values, (vectors), and [percent importance]:
change_detection2.0.sh: 64: change_detection2.0.sh:
Statistitque: not found
PC1 3413291.96 ( 0.1699, 0.2266, 0.3001, 0.3590, 0.3671, 0.3208, 0.1531, 0.2080, 0.2773, 0.3312, 0.3416, 0.3052) [85.28%]
PC2 325517.33 ( 0.1640, 0.1769, 0.2988,-0.6235, 0.0498, 0.2509, 0.1386, 0.1459, 0.2159,-0.5305,-0.0535, 0.1640) [ 8.13%]
PC3 120022.80 ( 0.2661, 0.3339, 0.2757, 0.1410,-0.3670,-0.3360, 0.2406, 0.2987, 0.2310, 0.1608,-0.3648,-0.3388) [ 3.00%]
PC4 96916.71 ( 0.1198, 0.1600, 0.2126, 0.3933, 0.3599, 0.3134,-0.1470,-0.2031,-0.2957,-0.3356,-0.4220,-0.3023) [ 2.42%]
PC5 23929.18 (-0.0460,-0.0395,-0.1561, 0.5130,-0.0939,-0.3097, 0.1072, 0.0969, 0.2325,-0.6634, 0.1631, 0.2498) [ 0.60%]
PC6 7792.26 (-0.3211,-0.3641,-0.3095, 0.0485, 0.1461, 0.3471, 0.2665, 0.3257, 0.3869, 0.0458,-0.4234,-0.1272) [ 0.19%]
PC7 6913.56 ( 0.4194, 0.1233,-0.4373, 0.0617,-0.2767, 0.2942, 0.3582, 0.0793,-0.4212, 0.0293,-0.1492, 0.3401) [ 0.17%]
PC8 5291.44 (-0.2430,-0.0440, 0.3184, 0.1528,-0.5203, 0.2631,-0.3220,-0.1944, 0.1383, 0.0774,-0.3123, 0.4584) [ 0.13%]
PC9 1086.09 (-0.0864, 0.2403,-0.1807, 0.0793,-0.4173, 0.4843,-0.1787, 0.0857, 0.1099,-0.1222, 0.4376,-0.4790) [ 0.03%]
PC10 846.65 ( 0.5322,-0.4895, 0.0686, 0.0425,-0.1107, 0.1000, 0.1631,-0.5194, 0.3438, 0.0076, 0.0831,-0.1638) [ 0.02%]
PC11 560.62 (-0.3765,-0.2167, 0.4406, 0.0645,-0.1674, 0.0718, 0.6210,-0.0535,-0.3708,-0.0554, 0.1964,-0.1107) [ 0.01%]
PC12 169.01 ( 0.2942,-0.5336, 0.2349, 0.0530,-0.0815, 0.0327,-0.3453, 0.6045,-0.2600,-0.0565, 0.0818,-0.0300) [ 0.00%]
    
```

Résultats de l'ACP pour Dubti

Geneva

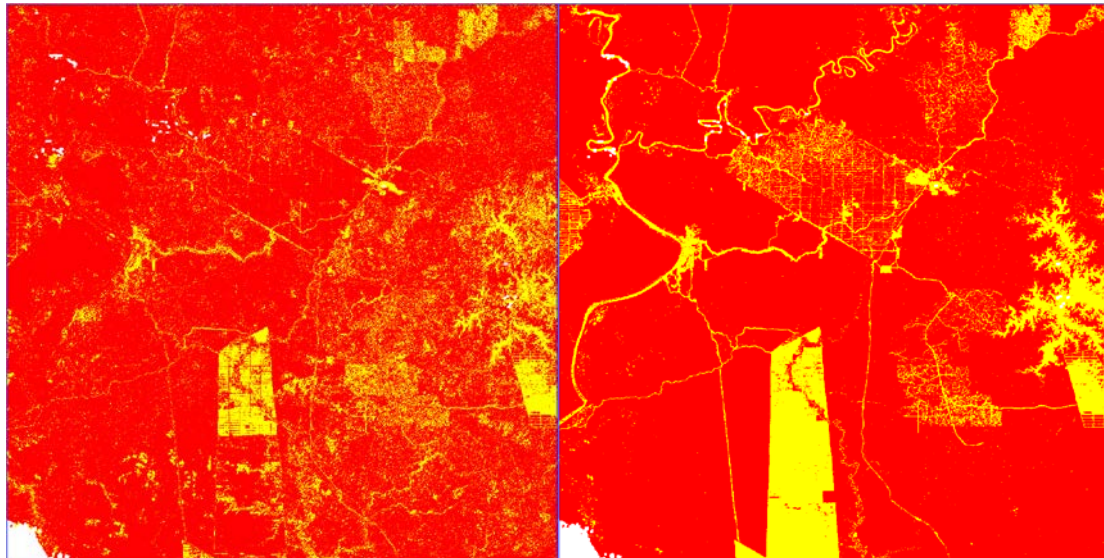
```

Computing covariance matrix...
Calculating principal components...
Rescaling to range 0,255...
PC1 2851234.61 ( 0.0309, 0.0794, 0.0945, 0.5713, 0.3720, 0.2107, 0.0447, 0.0906, 0.1323, 0.4383, 0.4281, 0.2675) [62.86%]
PC2 1244884.43 ( 0.1418, 0.1653, 0.2584,-0.4695, 0.2571, 0.3329, 0.1505, 0.1764, 0.2754,-0.4357, 0.2413, 0.3328) [27.44%]
PC3 262128.17 ( 0.0875, 0.0968, 0.2304,-0.4159, 0.3279, 0.3460,-0.0636,-0.0751,-0.1588, 0.5649,-0.2978,-0.2948) [ 5.78%]
PC4 93403.30 ( 0.0627, 0.1322, 0.1155, 0.4379, 0.4028, 0.2513,-0.1172,-0.1798,-0.2911,-0.5330,-0.2827,-0.2306) [ 2.06%]
PC5 57074.22 (-0.3070,-0.3287,-0.3281,-0.2644, 0.2977, 0.0541,-0.3153,-0.3490,-0.3452,-0.0554, 0.4080, 0.1367) [ 1.26%]
PC6 10858.87 (-0.2970,-0.3318,-0.3673, 0.0014, 0.4693,-0.0663, 0.1507, 0.2920, 0.4603,-0.0291,-0.3259,-0.1233) [ 0.24%]
PC7 8788.20 (-0.1370,-0.2323,-0.0707, 0.1062,-0.3151, 0.5429,-0.0780,-0.1308, 0.0344, 0.0553,-0.4385, 0.5451) [ 0.19%]
PC8 3535.59 ( 0.3783, 0.1126,-0.5259,-0.0505, 0.1017, 0.0182, 0.4938, 0.1890,-0.4747, 0.0422,-0.0779, 0.2043) [ 0.08%]
PC9 2079.80 (-0.0957,-0.0874,-0.2288, 0.0498,-0.3258, 0.5952, 0.1528, 0.1767, 0.0392,-0.0676, 0.3389,-0.5407) [ 0.05%]
PC10 993.20 (-0.3977, 0.5721,-0.1936,-0.0355,-0.0320, 0.0276,-0.4263, 0.5037,-0.1577, 0.0123,-0.0610, 0.0991) [ 0.02%]
PC11 608.65 ( 0.6519, 0.0027,-0.3730,-0.0059, 0.0059, 0.0584,-0.5761,-0.0159, 0.3134, 0.0045,-0.0013,-0.0445) [ 0.01%]
PC12 386.30 ( 0.1753,-0.5654, 0.3236, 0.0298,-0.0029,-0.0309,-0.2205, 0.6145,-0.3384,-0.0359, 0.0183, 0.0239) [ 0.01%]
    
```

Résultats de l'ACP pour Geneva

Différences des images

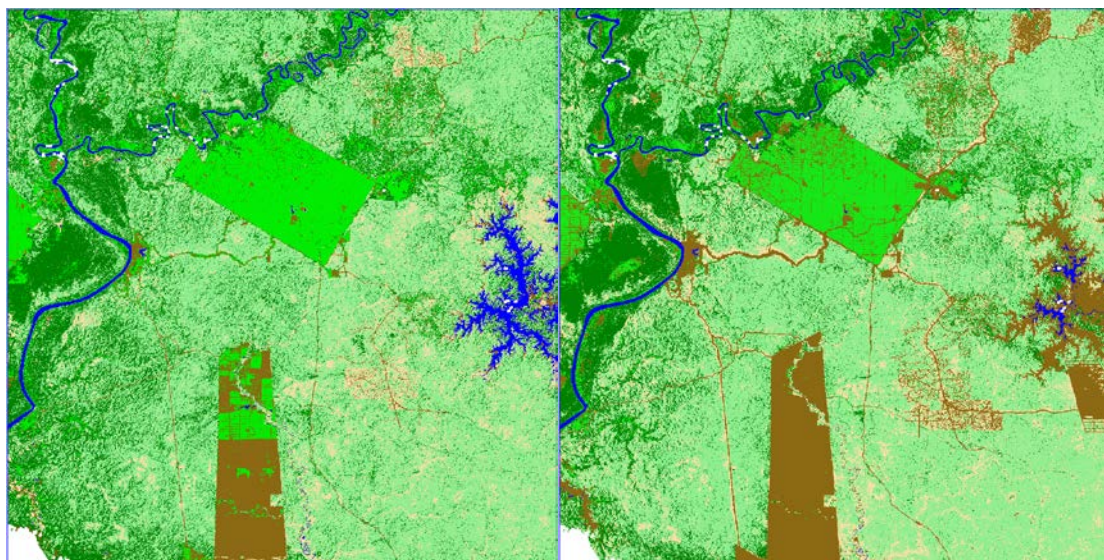
Papua



- 0) Changement
- 1) Non-changement

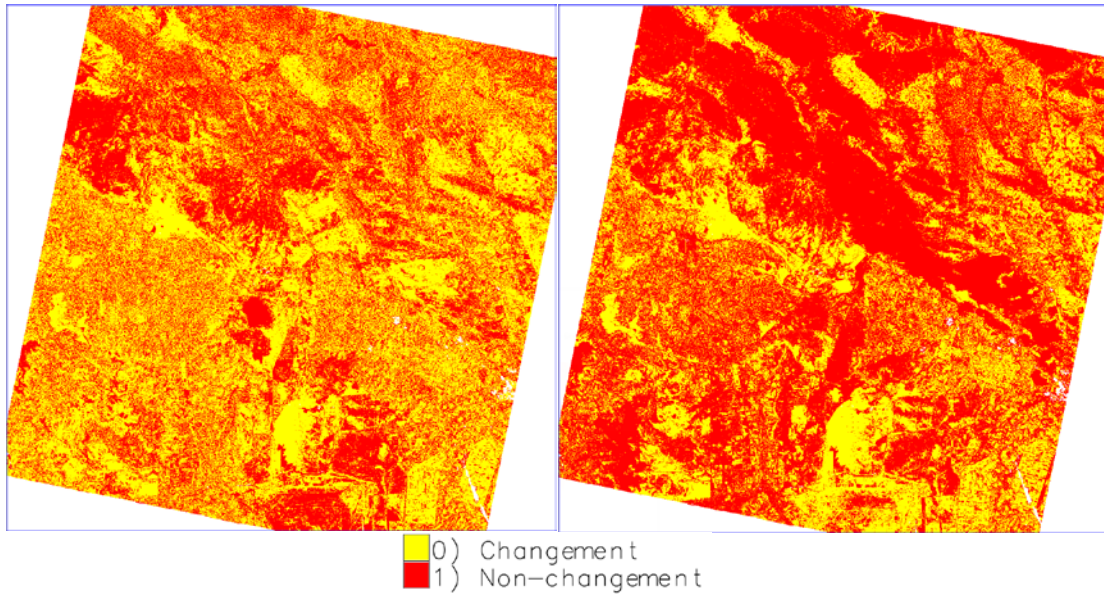
Résultat Papua: Image de surfaces changées vs. surfaces inchangées selon méthode post-classification (à gauche)

Image de surfaces changées (classes 1, 4 et 5) vs. surfaces inchangées (classes 2 et 3) selon classification des différences (à droite)



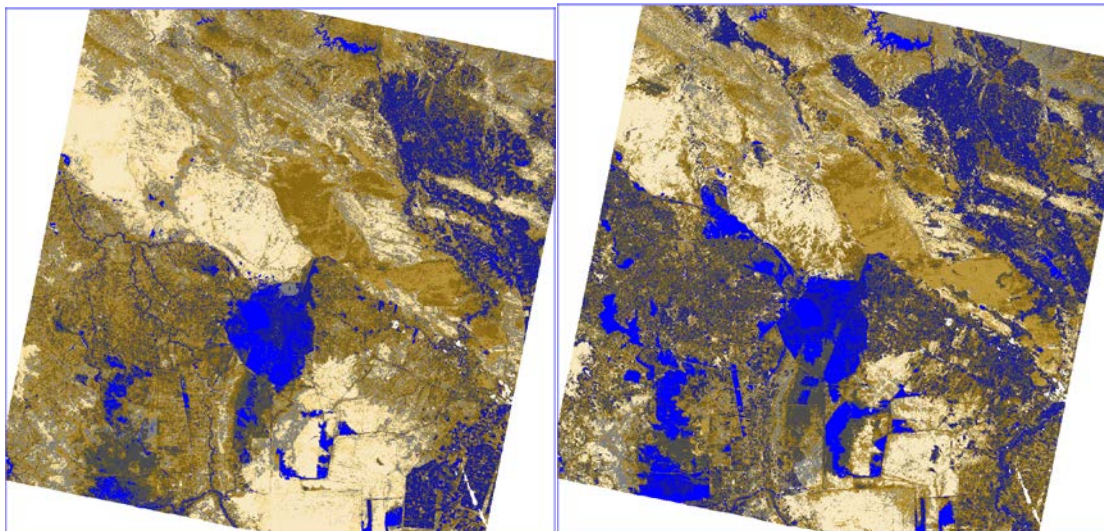
- 1) eau
- 2) sol clair
- 3) foret
- 4) foret
- 5) sol brun
- 6) foret

Irak



Résultat Irak: Image de surfaces changées vs. surfaces inchangées selon méthode post-classification (à gauche)

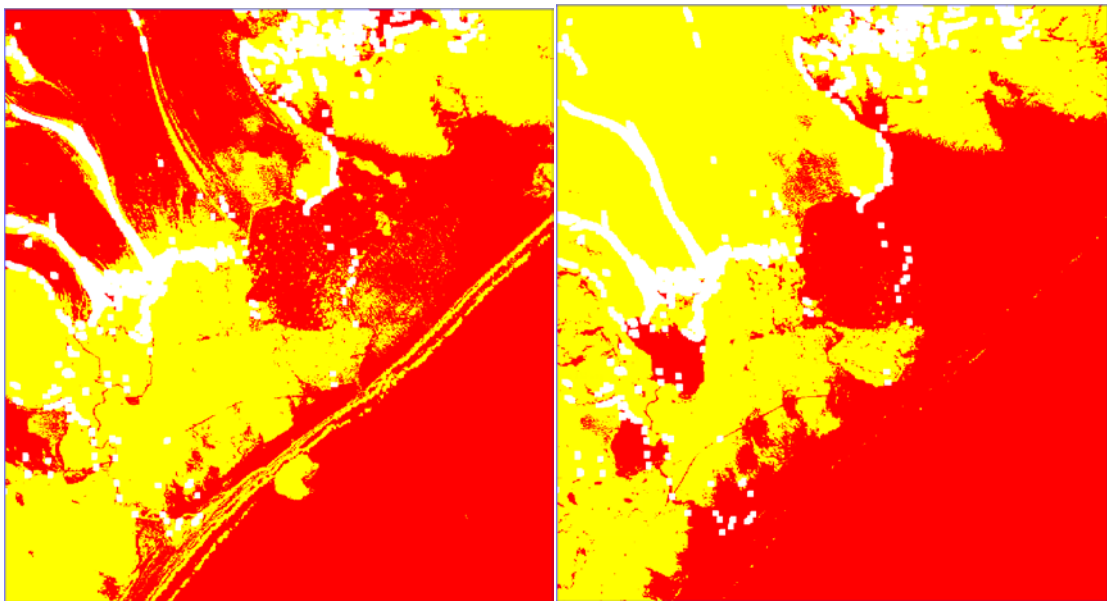
Image de surfaces changées (classes 1, 2 et 5) vs. surfaces inchangées (classes 3 et 4) selon classification des différences (à droite)



- 1) eau
- 2) marais
- 3) sol humide
- 4) sol brun
- 5) sol brun
- 6) sol gris
- 7) sol clair
- 8) sol clair

Résultat Irak: Classification scène 1 selon méthode 1 et classification finale scène 2 selon la méthode des différences

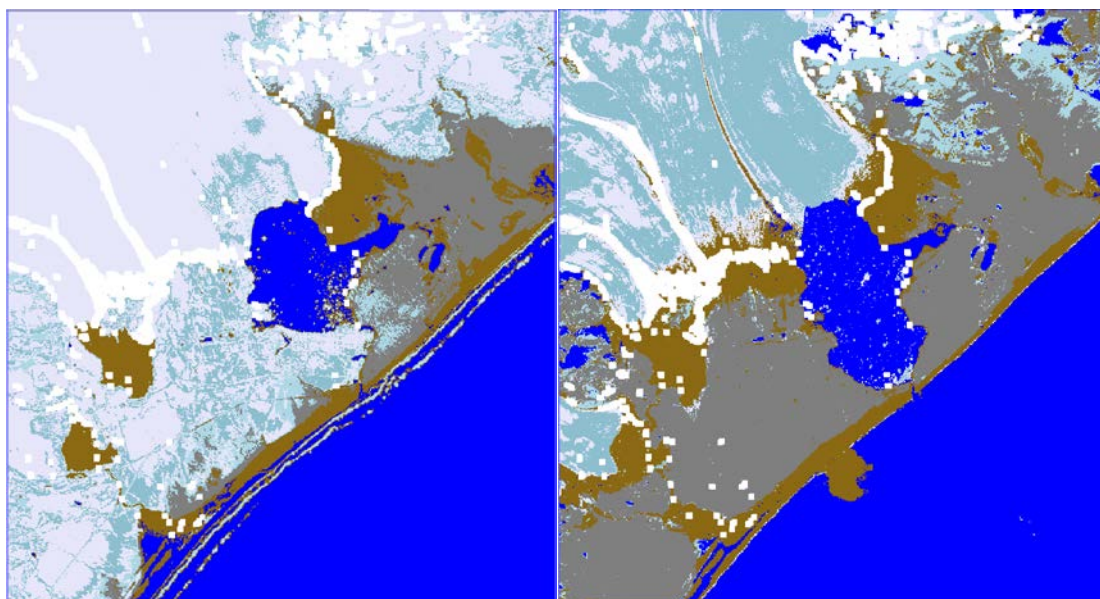
Iceland



- 0) Changement
- 1) Non-changement

Résultat Iceland: Image de surfaces changées vs. surfaces inchangées selon méthode post-classification (à gauche)

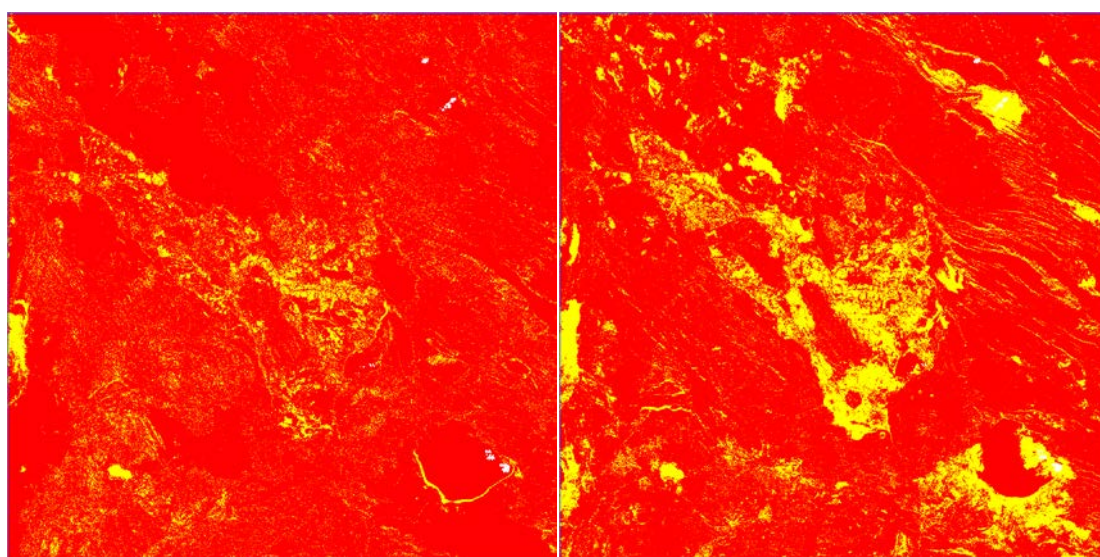
Image de surfaces changées (classes 1, 2 et 3) vs. surfaces inchangées (classes 4 et 5) selon classification des différences (à droite)



- 1) eau
- 2) sol
- 3) sol
- 4) glace fonce
- 5) glace fonce
- 6) glace fonce
- 7) neige/glace
- 8) neige/glace

Résultat Iceland: Classification scène 1 selon méthode 1 et classification finale scène 2 selon la méthode des différences

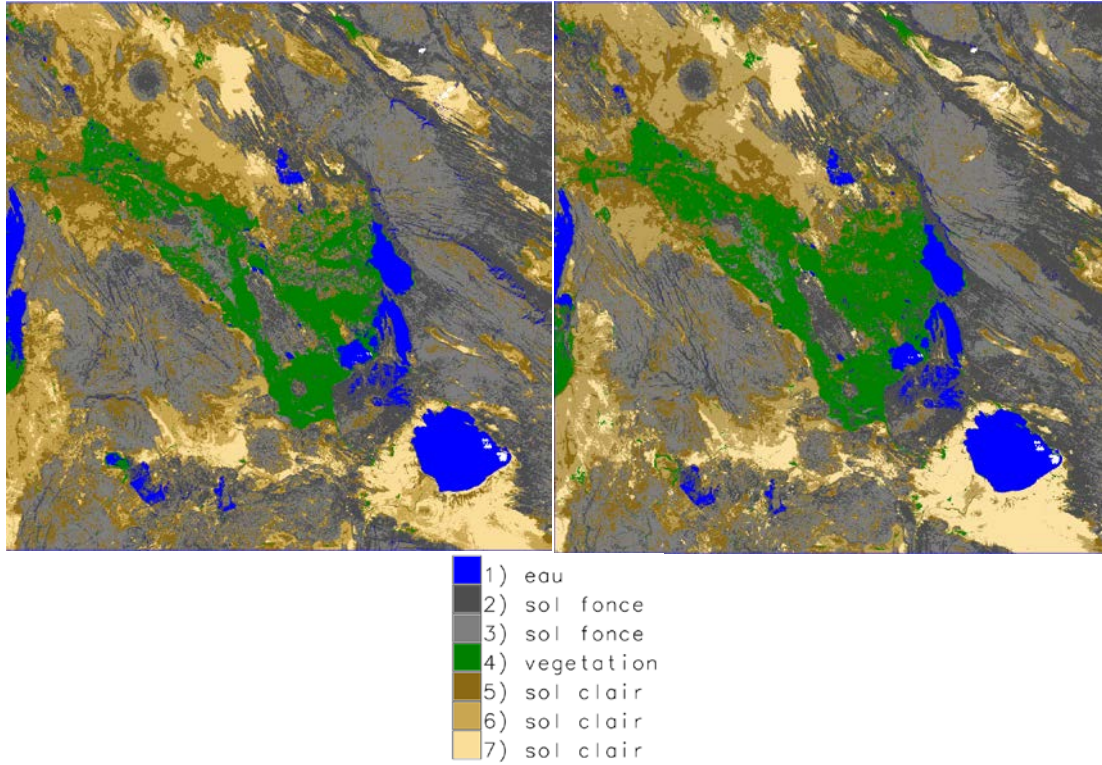
Dubti



- 0) Changement
- 1) Non-changement

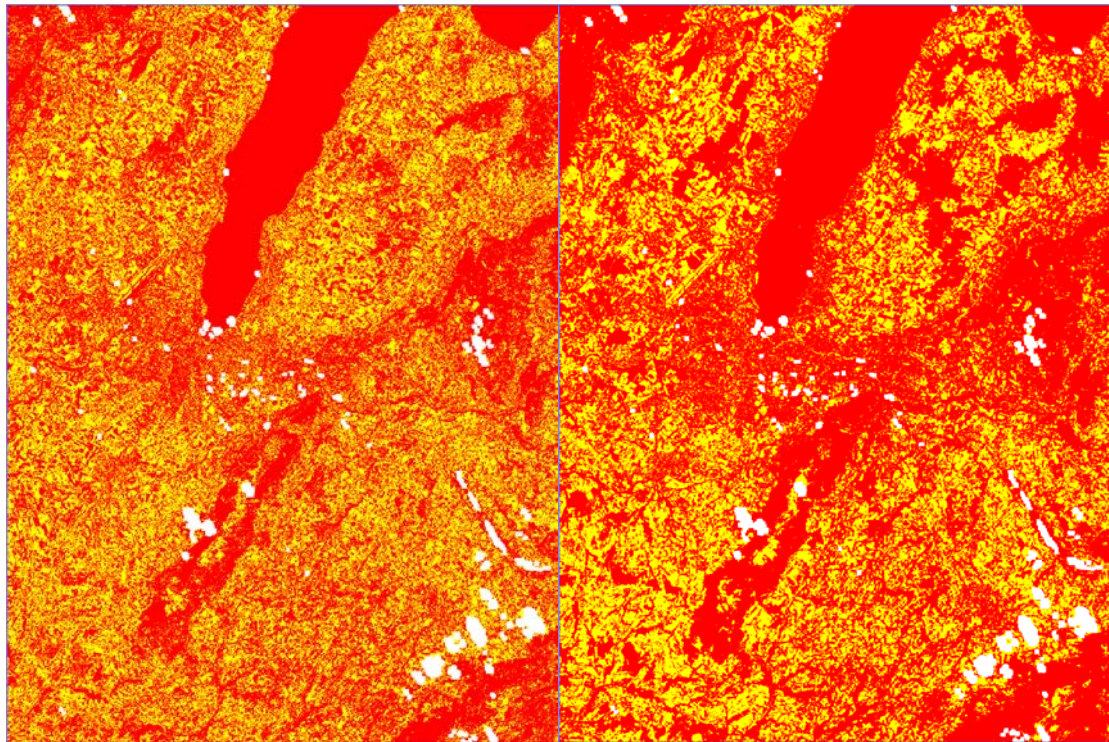
Résultat Dubti: Image de surfaces changées vs. surfaces inchangées selon méthode post-classification (à gauche)

Image de surfaces changées (classes 1 et 4) vs. surfaces inchangées (classes 2 et 3) selon classification des différences (à droite)



Résultat Dubti: Classification scène 1 selon méthode 1 et classification finale scène 2 selon la méthode des différences

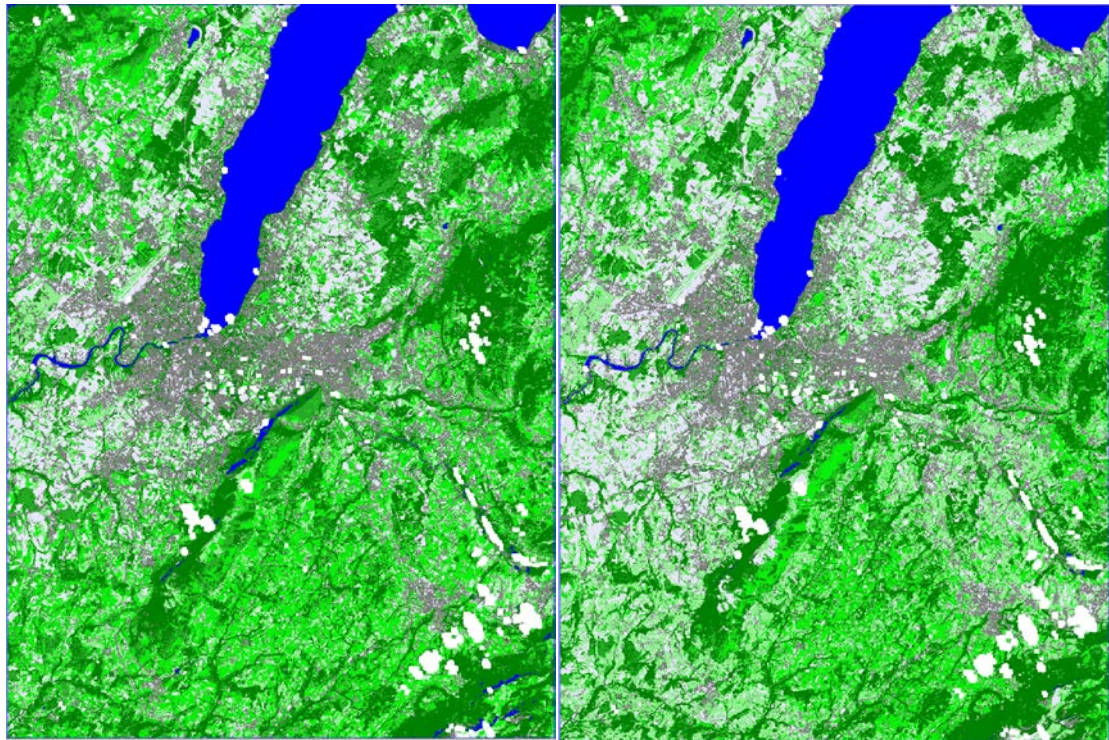
Geneva



0) Changement
1) Non-changement

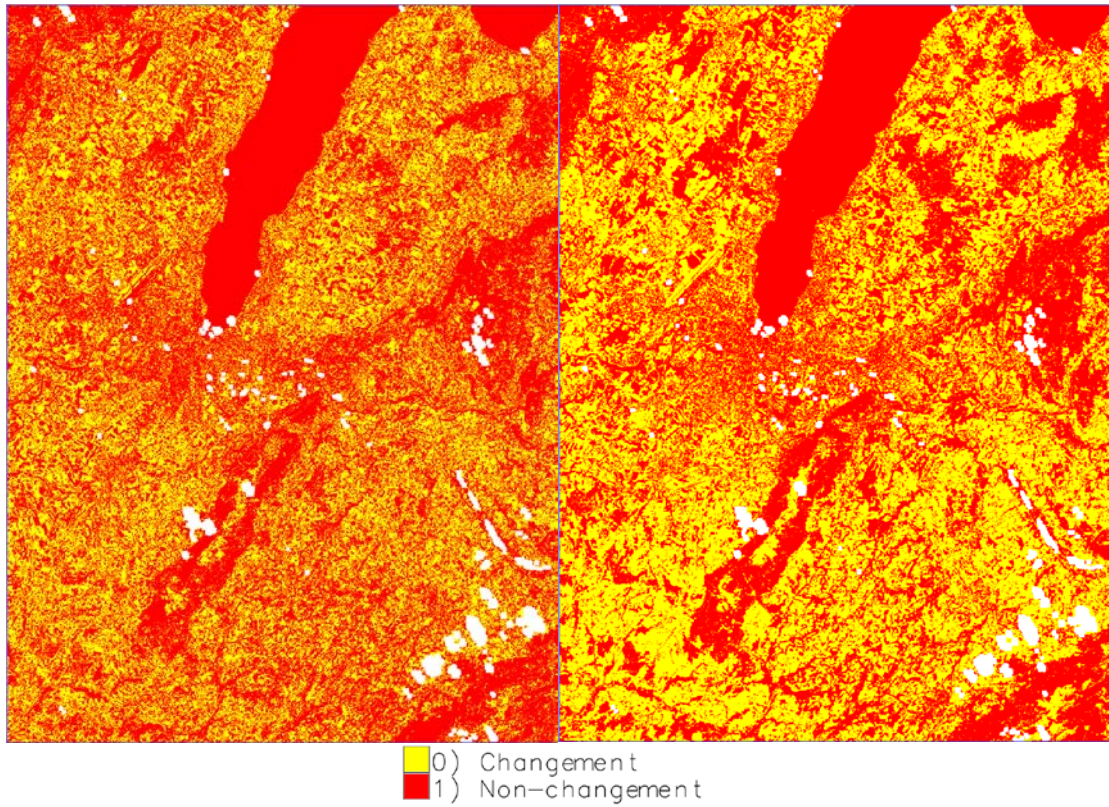
Résultat Geneva: Image de surfaces changées vs. surfaces inchangées selon méthode post-classification (à gauche)

Image de surfaces changées (classes 1,2 et 5) vs. surfaces inchangées (classes 3 et 4) selon classification des différences à 5 classes (à droite)



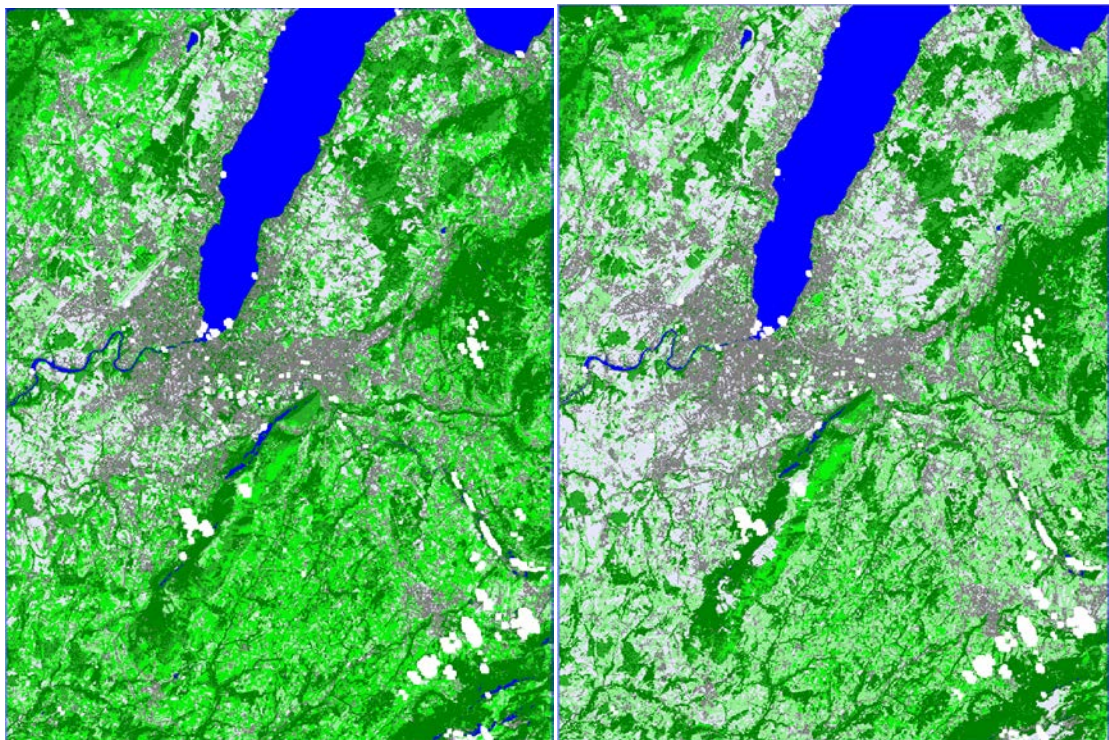
- 1) eau
- 2) foret
- 3) foret
- 4) foret/herbe
- 5) urbain
- 6) sol/herbe
- 7) foret/herbe
- 8) champs

Résultat Geneva: Classification scène 1 selon méthode 1 et classification finale scène 2 selon la méthode des différences (5 classes)



Résultat Geneva: Image de surfaces changées vs. surfaces inchangées selon méthode post-classification (à gauche)

Image de surfaces changées (classes 1,2 et 4) vs. surfaces inchangées (classes 3) selon classification des différences à 4 classes (à droite)



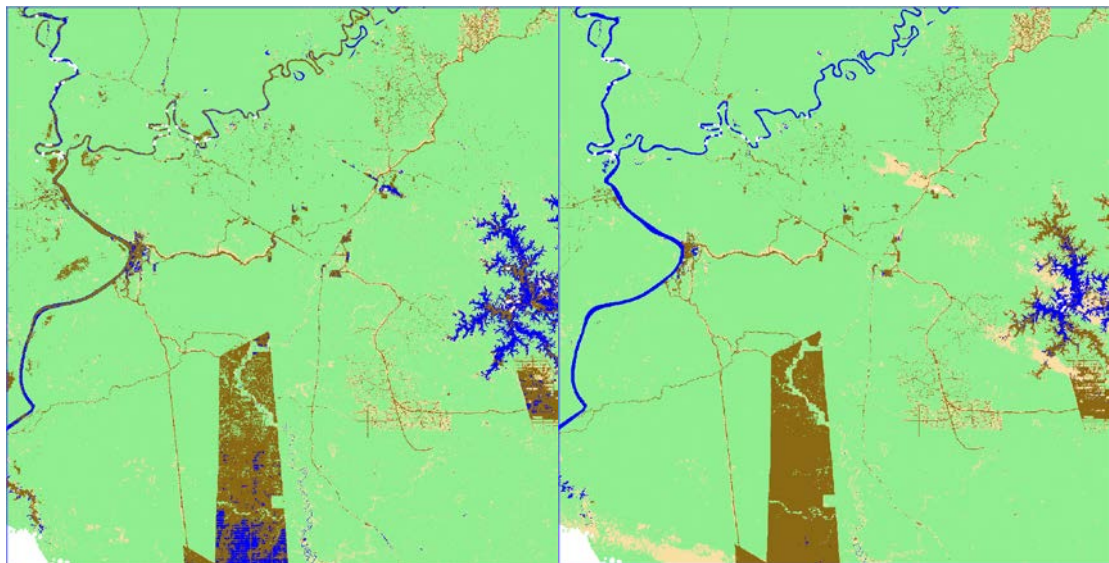
- 1) eau
- 2) foret
- 3) foret
- 4) foret/herbe
- 5) urbain
- 6) sol/herbe
- 7) foret/herbe
- 8) champs

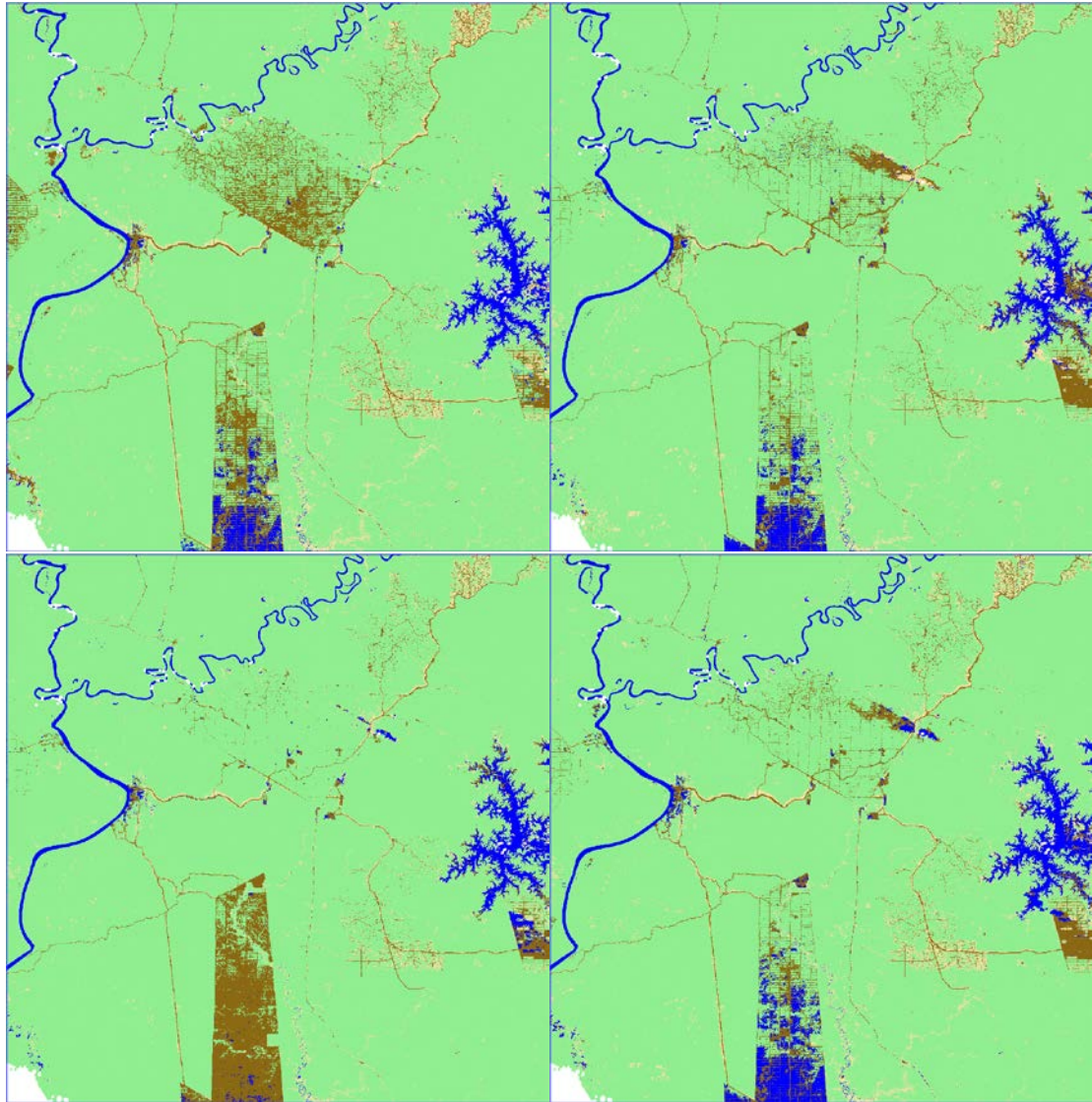
Résultat Geneva: Classification scène 1 selon méthode 1 et classification finale scène 2 selon la méthode des différences (4 classes)

Échantillons aléatoires

Cette méthode a été testée sur 3 sites: Papua, Dubti et Geneva

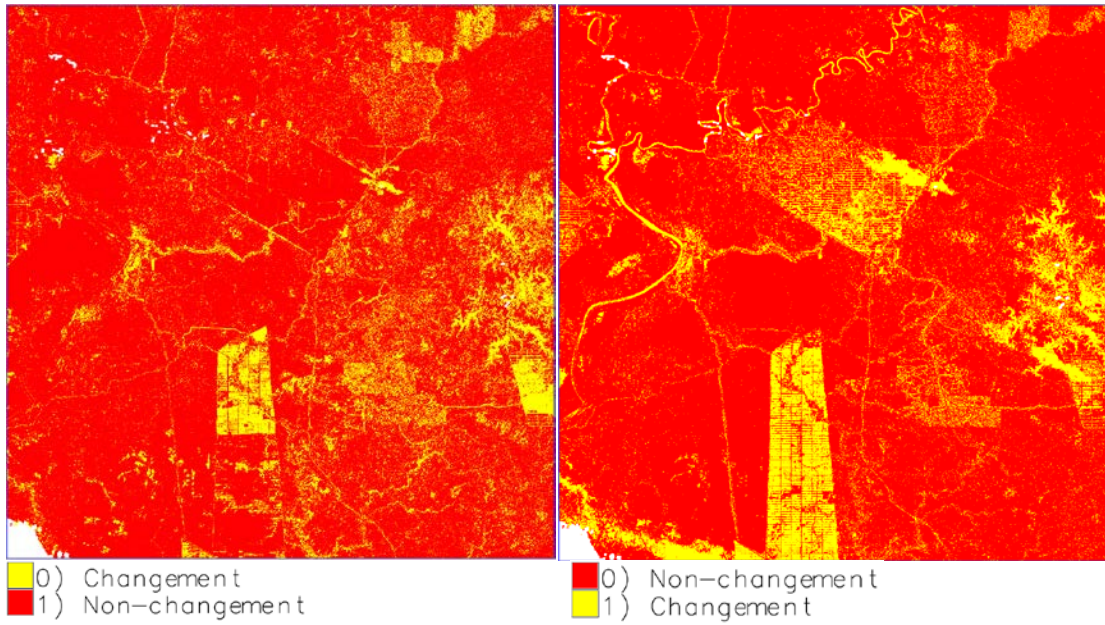
Papua





- 1) eau
- 2) sol clair
- 3) foret
- 4) sol brun

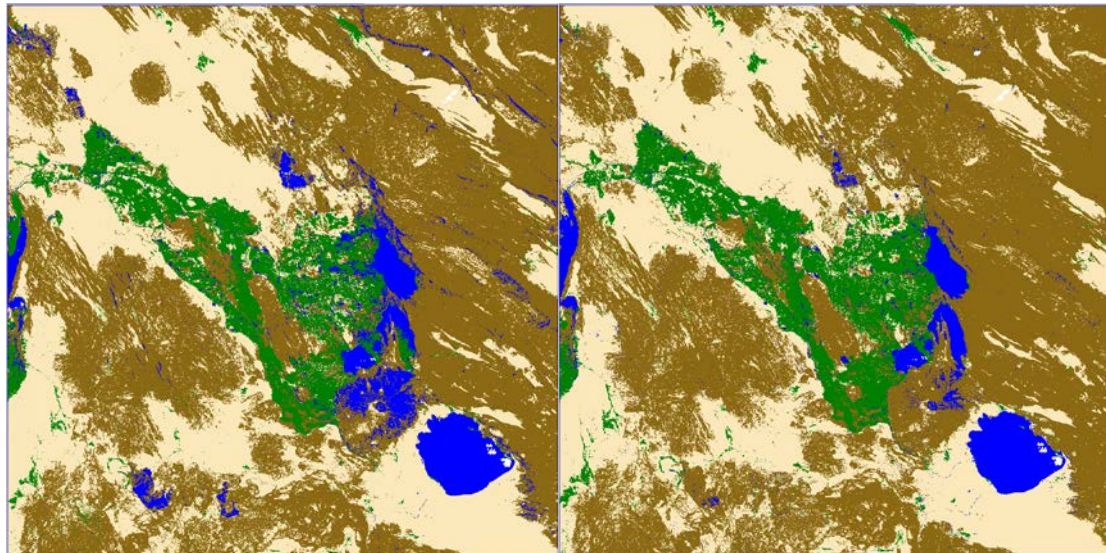
Résultat Dubti: Classifications de la deuxième scène basées sur les échantillons aléatoires

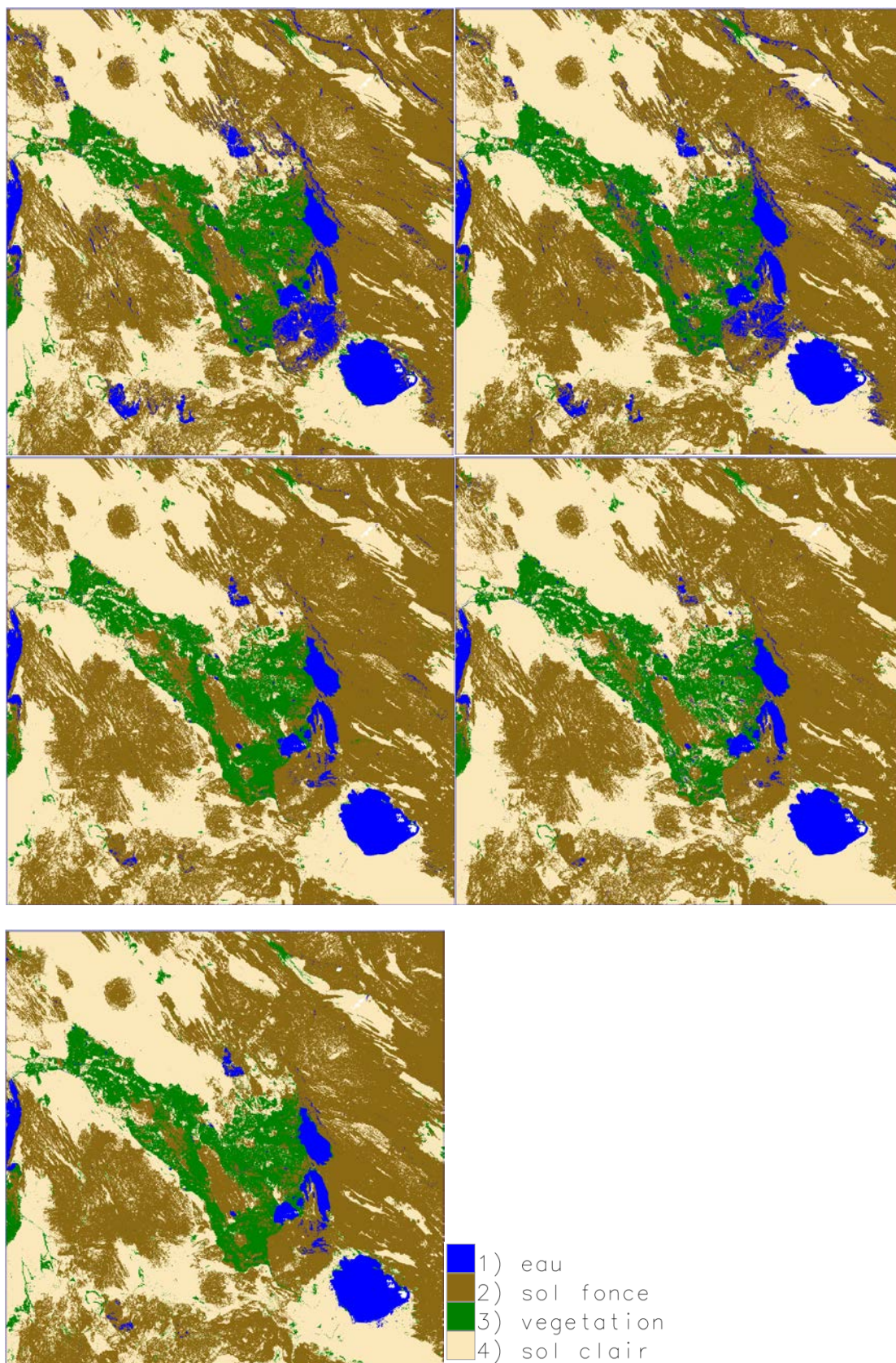


Résultat Papua: Image de surfaces changées vs. surfaces inchangées selon méthode post-classification (à gauche)

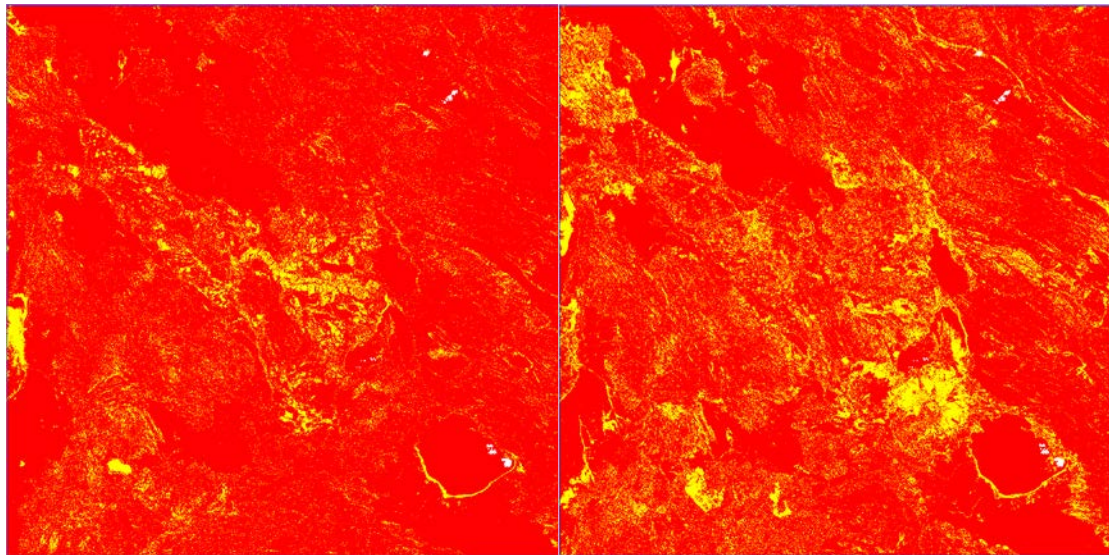
Image de surfaces changées vs. surfaces inchangées selon classification des échantillons aléatoires (à droite)

Dubti





Résultat Dubti: Classifications de la deuxième scène basées sur les échantillons aléatoires



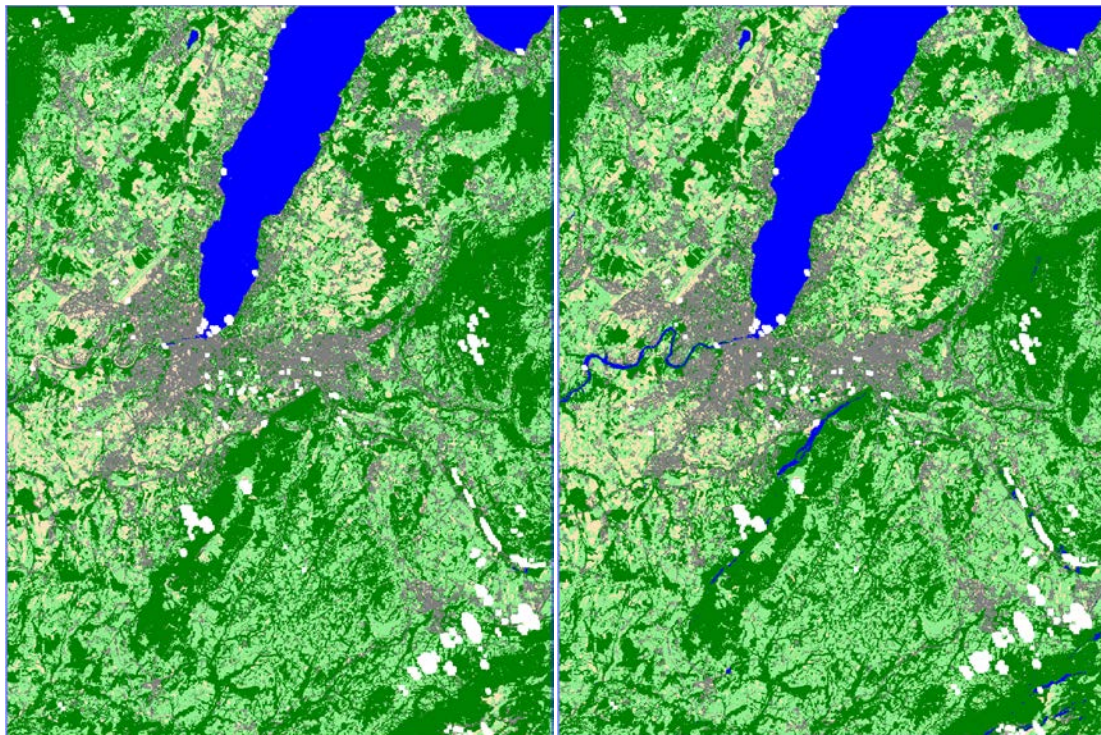
■ 0) Changement
■ 1) Non-changement

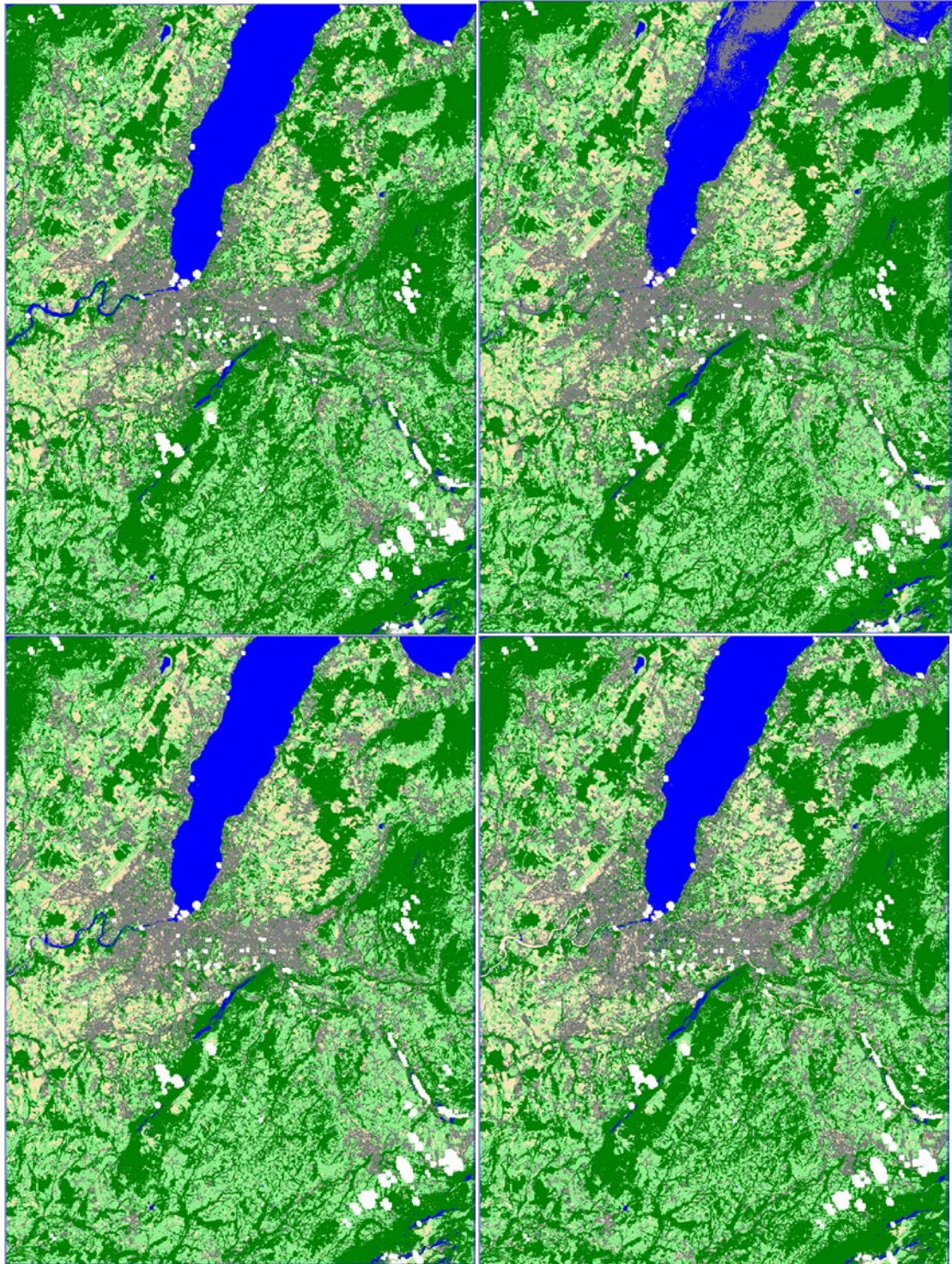
■ 0) Non-changement
■ 1) Changement

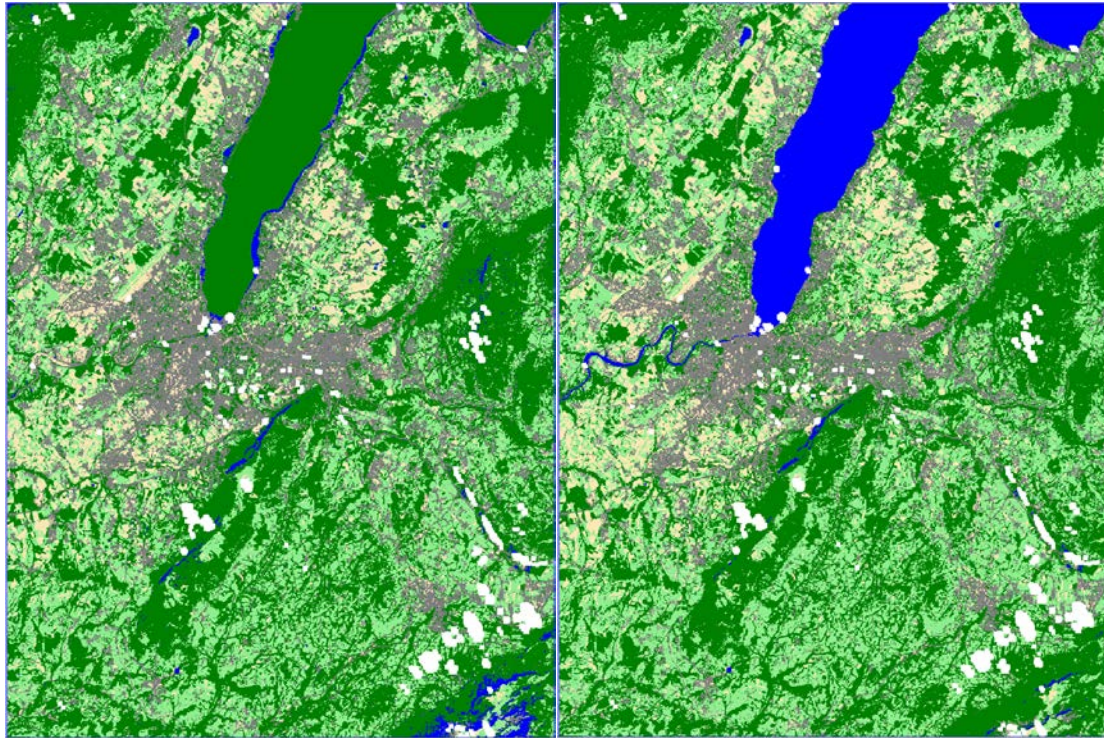
Résultat Dubti: Image de surfaces changées vs. surfaces inchangées selon méthode post-classification (à gauche)

Image de surfaces changées vs. surfaces inchangées selon classification des échantillons aléatoires (à droite)

Geneva

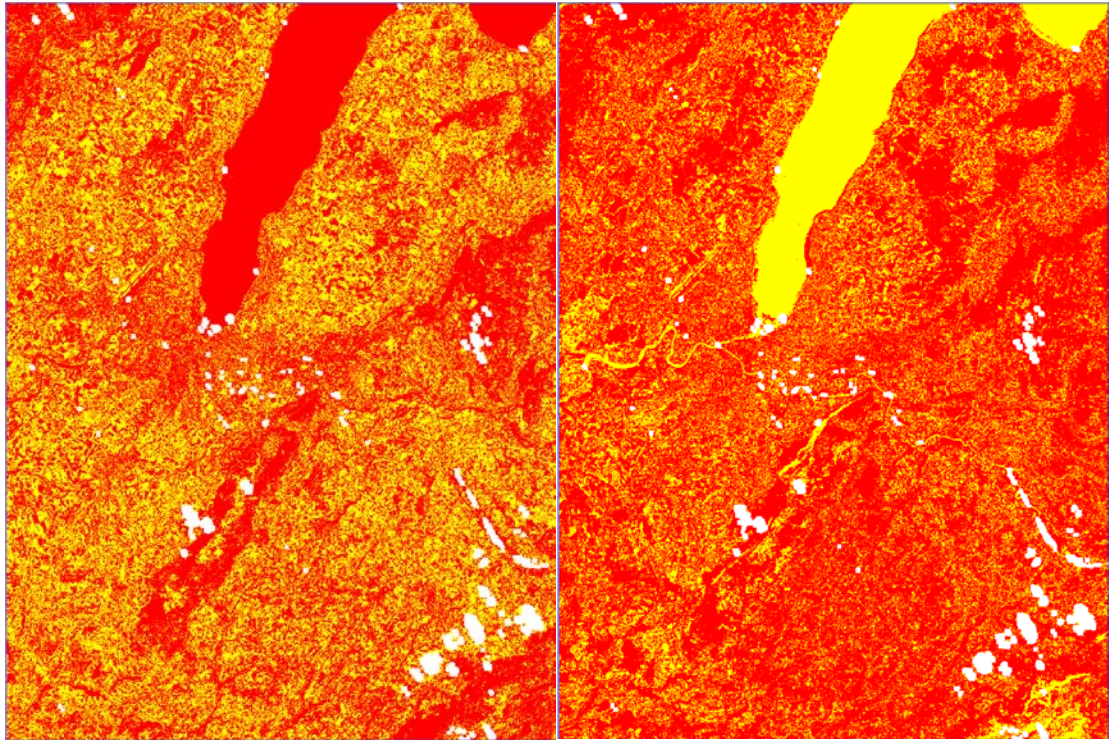






- 1) eau
- 2) foret
- 3) urbain
- 4) sol/herbe
- 5) champs

Résultat Geneva: Classifications de la deuxième scène basées sur les échantillons aléatoires



0) Changement
 1) Non-changement

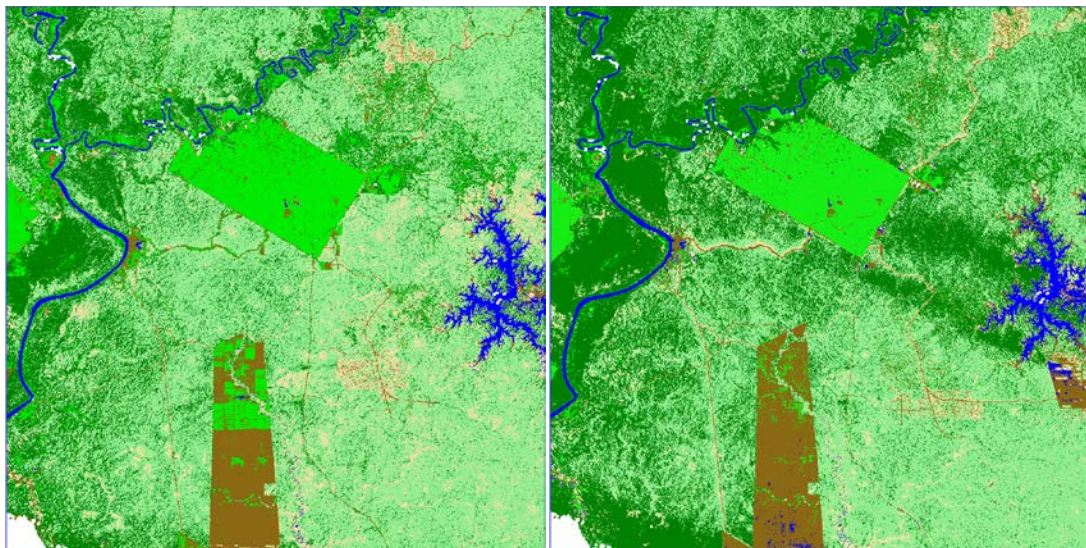
0) Non-changement
 1) Changement

Résultat Geneva: Image de surfaces changées vs. surfaces inchangées selon méthode post-classification (à gauche)

Image de surfaces changées vs. surfaces inchangées selon classification des échantillons aléatoires (à droite)

Zone d'entraînement totale

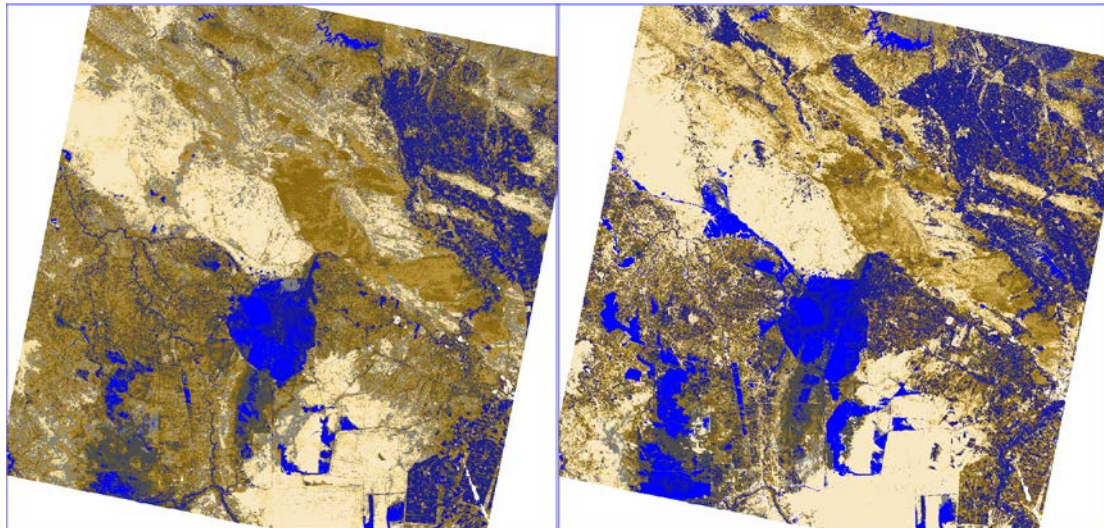
Papua



- 1) eau
- 2) sol clair
- 3) foret
- 4) foret
- 5) sol brun
- 6) foret

Résultats Papua: Classification scène 1 et classification scène 2 selon méthode 5

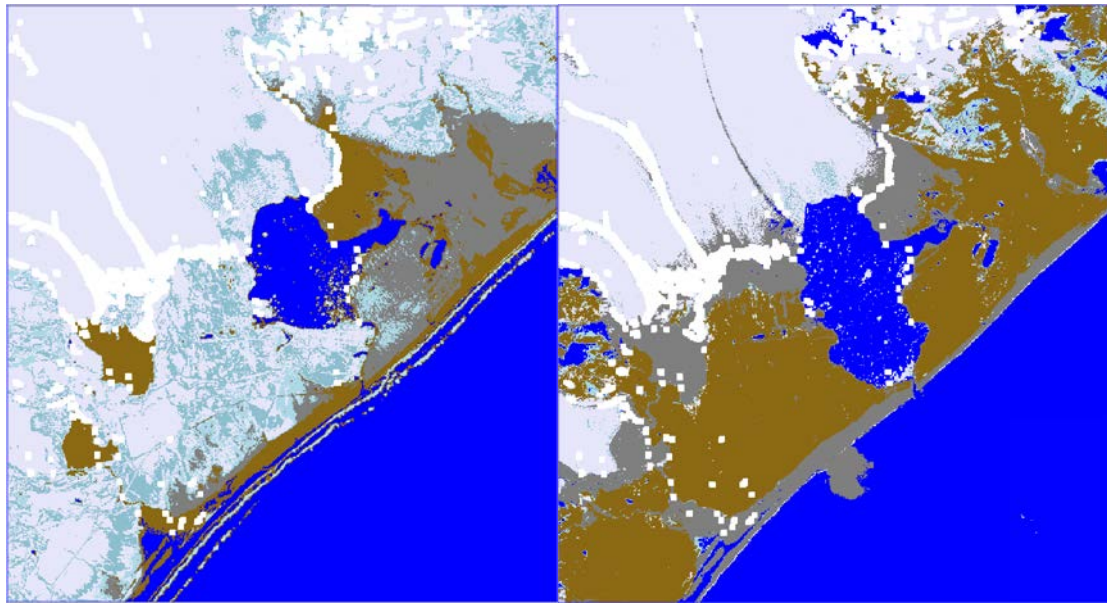
Irak



- 1) eau
- 2) marais
- 3) sol humide
- 4) sol brun
- 5) sol brun
- 6) sol gris
- 7) sol clair
- 8) sol clair

Résultats Irak: Classification scène 1 et classification scène 2 selon méthode 5

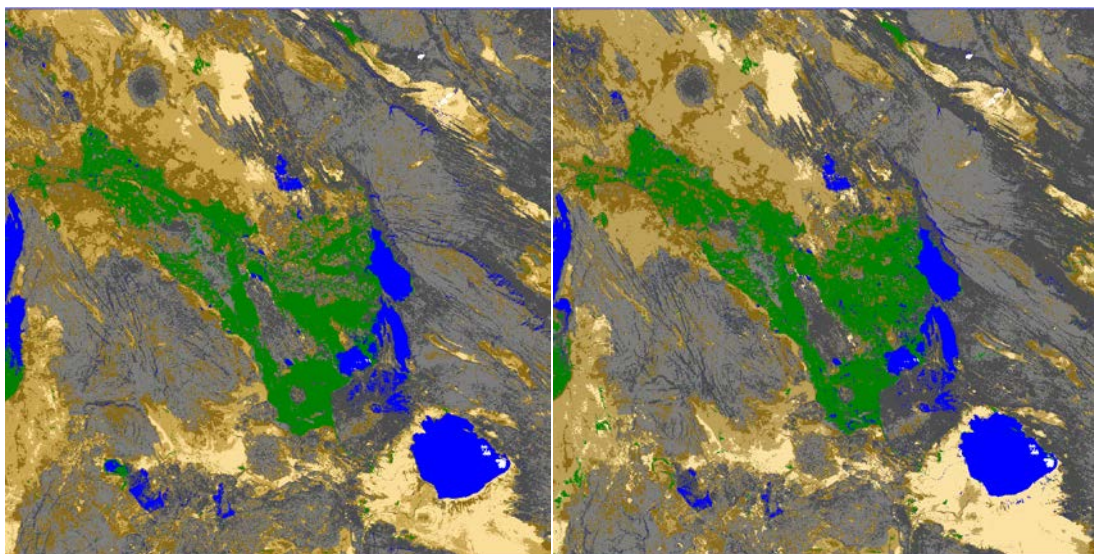
Iceland



- 1) eau
- 2) sol
- 3) sol
- 4) glace fonce
- 5) glace fonce
- 6) glace fonce
- 7) neige/glace
- 8) neige/glace

Résultats Iceland: Classification scène 1 et classification scène 2 selon méthode 5

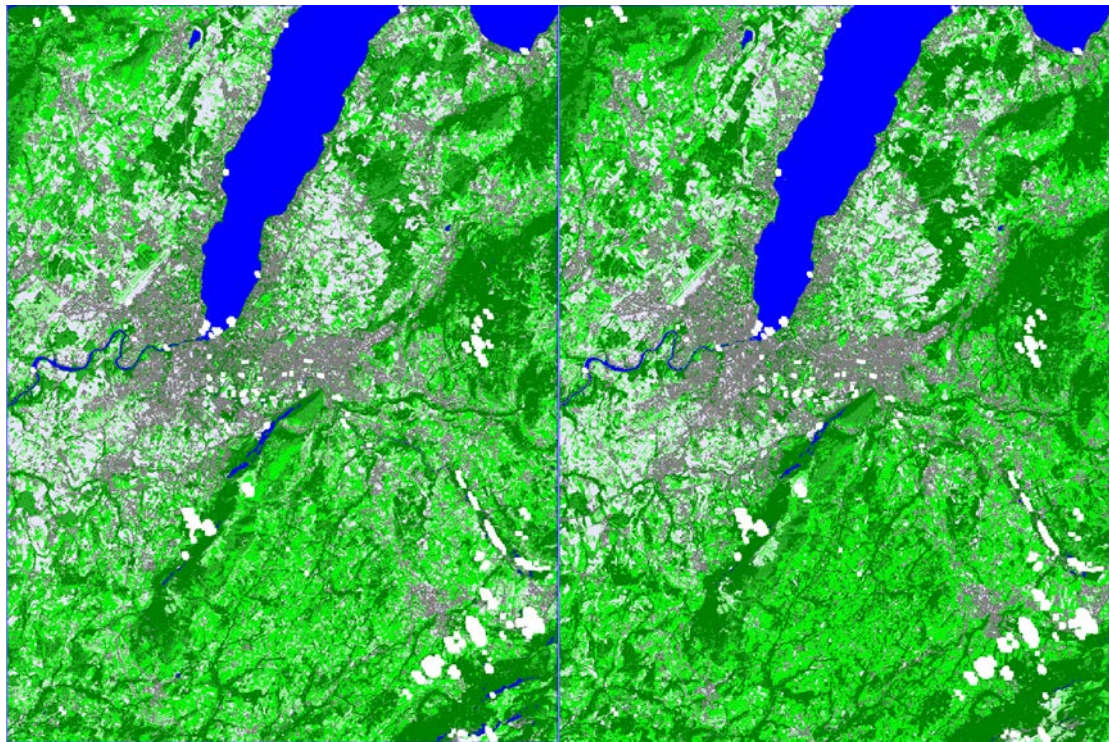
Dubti



- 1) eau
- 2) sol foncé
- 3) sol foncé
- 4) végétation
- 5) sol clair
- 6) sol clair
- 7) sol clair

Résultats Dubti: Classification scène 1 et classification scène 2 selon méthode 5

Geneva



- 1) eau
- 2) forêt
- 3) forêt
- 4) forêt/herbe
- 5) urbain
- 6) sol/herbe
- 7) forêt/herbe
- 8) champs

Résultats Geneva: Classification scène 1 et classification scène 2 selon méthode 5