

L'analyse de l'utilisation de l'espace par la faune sauvage

Clément CALENGE

Office National de la Chasse et de la Faune Sauvage

L'analyse de l'utilisation de l'espace par la faune sauvage.

Utilisation du package adehabitat et des autres packages d'analyse spatiale pour le logiciel R

Clément Calenge

Office national de la chasse et de la faune sauvage
Saint Benoist - Le Perray en Yvelines

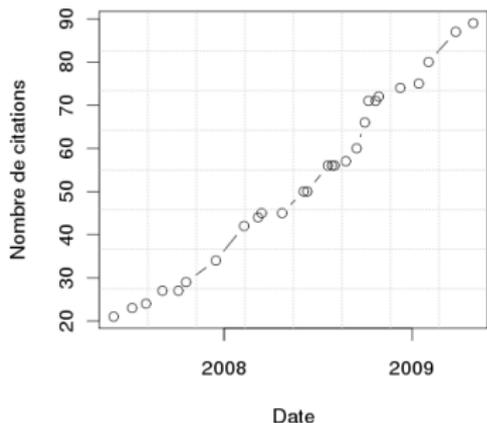
13 mai 2009

Plan

- 1 Introduction
- 2 Découverte des classes d'objet d'adehabitat
- 3 Vers une meilleure gestion du spatial : le package sp

Le package adehabitat

Nombre de citations de adehabitat
(Google scholar)



Objectif de adehabitat : implémenter des méthodes statistiques pour l'**analyse exploratoire de la sélection de l'habitat par la faune sauvage**.

Développement commencé en Décembre 2002

Première version déposée sur CRAN en Septembre 2004

Calenge (2006) Ecological Modelling, 197 :516-519.

La structure de adehabitat : quatre parties

Grande diversité de données et d'approche pour l'étude de la sélection de l'habitat.

Les fonctions du package sont regroupées en quatre parties :

- Gestion de cartes raster ;
- Estimation du domaine vital ;
- Analyse de la sélection de l'habitat ;
- Analyse de trajets.

Analyse de la sélection de l'habitat

Le cœur du package (adehabitat) :

- Echelle de la population : Distances de Mahalanobis, fonctions de sélection des ressources, rapports de sélection, algorithmes DOMAIN, ENFA, MADIFA, GNESFA, etc. ;
- Echelle individuelle (radio-pistage) : rapports de sélection, analyse compositionnelle, Analyse factorielle des rapports de sélection, analyse OMI, analyse K-select, etc. ;

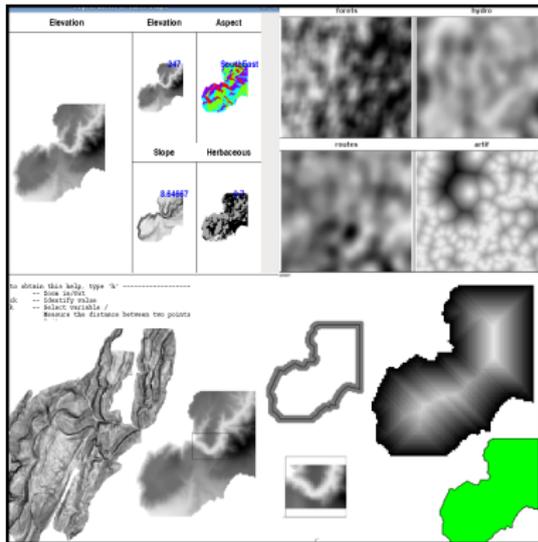
Nombreuses analyses basées sur le concept "formel" de niche écologique (*sensu Grinnell, 1917*).

Calenge (2007) Journal of Statistical Software, 22.

Calenge (2005) Des outils statistiques pour l'analyse des semis de points dans l'espace écologique Université Claude

Bernard Lyon 1.

La gestion des cartes raster



Nombreuses fonctions permettant la gestion des cartes raster :

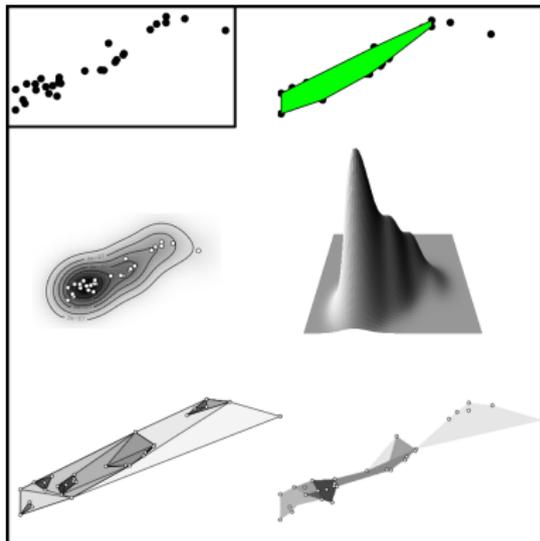
- Importation/exportation de et vers les SIG (e.g. [import.asc](#));
- Affichage statique et dynamique des cartes (simples ou multicouches ; e.g. [explore.kasc](#), [image](#));
- Opération SIG de base (jointure spatiale, compter des points dans les pixels de la carte, zones tampon, distances aux caractéristiques du milieu, changement de résolution, morphologie mathématique, etc.) ;

Utilisation limitée ! (pas trop plus d'un million de pixels gérables)

→ Système d'information géographique (notamment SIG libre, e.g. GRASS, QGIS).

Estimation du domaine vital

Fonctions permettant l'estimation du domaine vital à l'aide de données de radio-pistage :

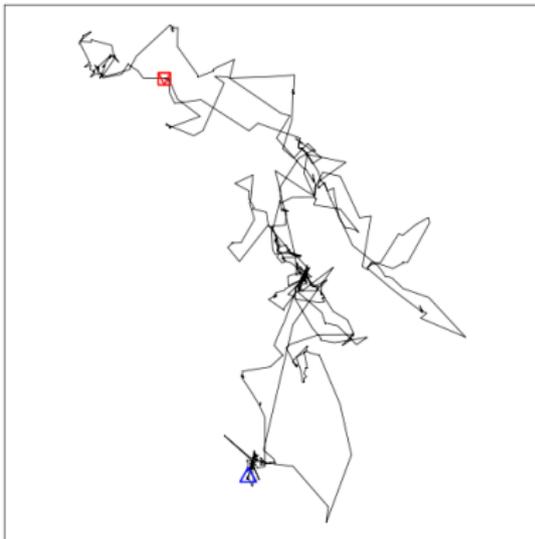


- Polygone convexe minimum (`mcp()`);
- Estimateur du domaine vital et de la distribution d'utilisation par la méthode du noyau (paramètres de lissage *ad hoc* et LSCV, `kernelUD()`);
- Cluster home range (`clusthr()`);
- Nearest neighbour convex hull ou LoCoH (`NNCH()`, developed by Scott Fortmann Roe, Univ. Berkeley);

Voir aussi Animove pour de nombreuses discussions sur ces méthodes

(<http://www.faunalia.it/animov/>).

Analyse de trajets



Suivis GPS d'animaux sauvage =
technologie récente ;
mode de suivi intensif → nouvelle classe de
donnée.

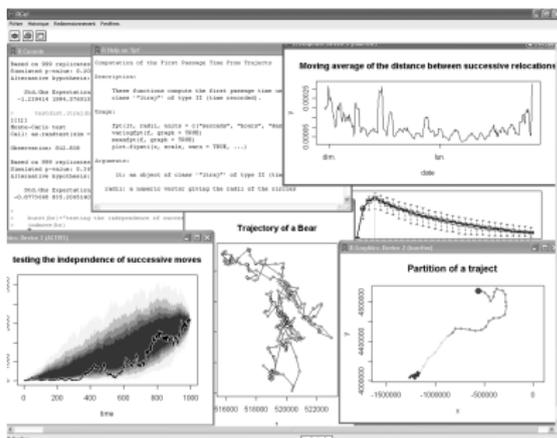
Autocorrélation temporelle de la position
des localisations est structurelle : elle doit
être intégrée à l'aide de modèles décrivant
les déplacements de l'animal.

Martin, Calenge, Quenette & Allainé (2008) Ecological Modelling, 213 : 257-262.

Dray, Royer & Calenge (soumis).

Analyse de trajets

Implémentation d'une classe de données permettant la manipulation des trajets d'animaux :



- Calcul de paramètres caractérisant chaque localisation d'un trajet (distance entre localisations, "turning angles", etc.) : [as.ltraj](#), etc.
- Exploration graphique dynamique ou non des données de trajet : [trajdyn](#), [plotltr](#), etc.
- test de l'autocorrélation des paramètres du trajet, rediscrétisation de trajets ([redisltraj](#)), partition d'un trajet en segments homogènes (e.g. [partmod.ltraj](#)), *first passage time* ([fpt](#)), etc.

Calenge, Royer & Dray (2009). Ecological informatics. 4 :34-41.

Apprendre l'utilisation du package

- A l'aide des demos commentées :

Type:

`demo(rastermaps)` for demonstration of raster map analysis

`demo(homerange)` for demonstration of home-range estimation

`demo(managltraj)` for demonstration of animals trajectory management

`demo(analysisltraj)` for demonstration of animals trajectory analysis

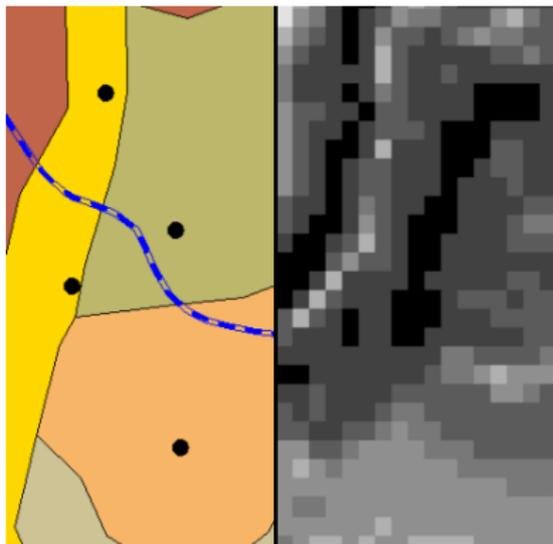
`demo(nichehs)` for demonstration of niche/habitat selection analysis

- A l'aide du didacticiel (`help.start()` puis "packages", puis "adehabitat" puis "browse directory");
- Lisez les fiches d'aide et essayez les exemples !
- A l'aide des articles scientifiques liés au package ;
- sur internet : didacticiel et forum de discussion :
<http://www.faunalia.it/animov/>

Plan

- 1 Introduction
- 2 Découverte des classes d'objet d'adehabitat
- 3 Vers une meilleure gestion du spatial : le package sp

Cartes vecteur vs. cartes raster

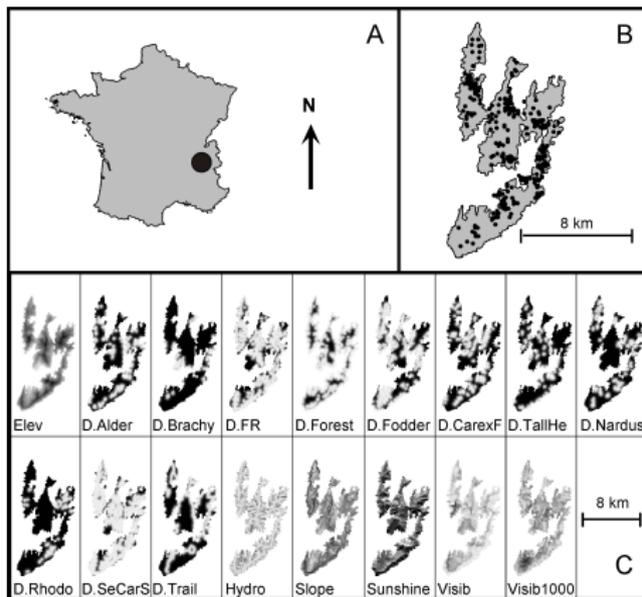


- Carte **vecteur** : enregistre l'information spatiale sous la forme d'objets (*object-oriented mode*) – lignes, points, polygones – possédant des attributs ;
- Carte **raster** : uniquement pour représenter des informations de type surfacique (*image-oriented mode*). La zone est représentée par un assemblage de pixels.

Avantage du raster en termes d'analyse statistique : toutes les unités de bases de la carte ont la même surface.

adehabitat ne gère pour ainsi dire que les cartes raster.

Le chamois dans le massif des Bauges (Haute-Savoie)



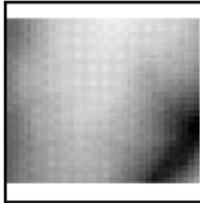
Parcours de la totalité de la zone d'étude une fois par an au printemps (juin) : tous les chamois en zone ouverte sont localisés : 400 chamois localisés de 1994 à 2000.

Cartes de 17 variables d'habitat, relative au relief (e.g. altitude, pente, exposition), à la végétation (e.g. distance aux prairies à Seslerie), au dérangement par l'homme (e.g. distance aux chemins).

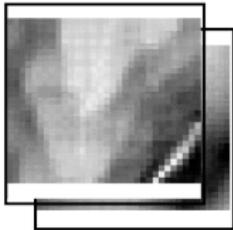
Analyse exploratoire de la sélection de l'habitat

Calenge, Darmon, Basille, Loison & Jullien (2008). *Ecology*, 89 :555-566.

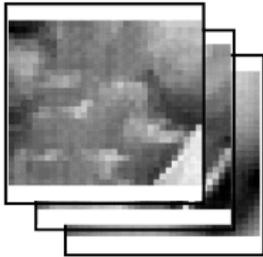
Utilisation du concept de niche écologique



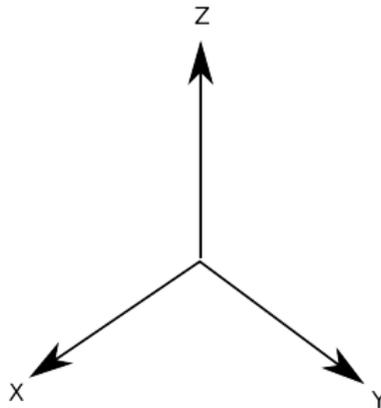
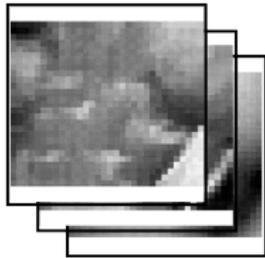
Utilisation du concept de niche écologique



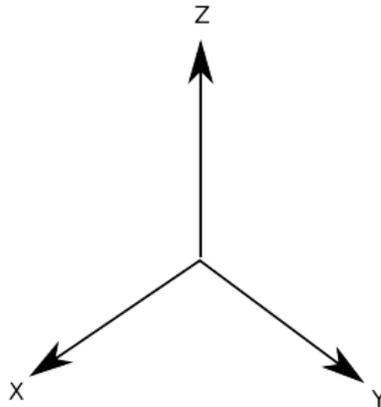
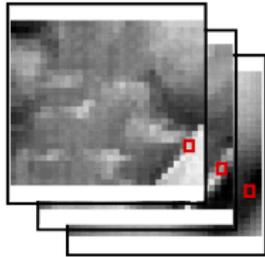
Utilisation du concept de niche écologique



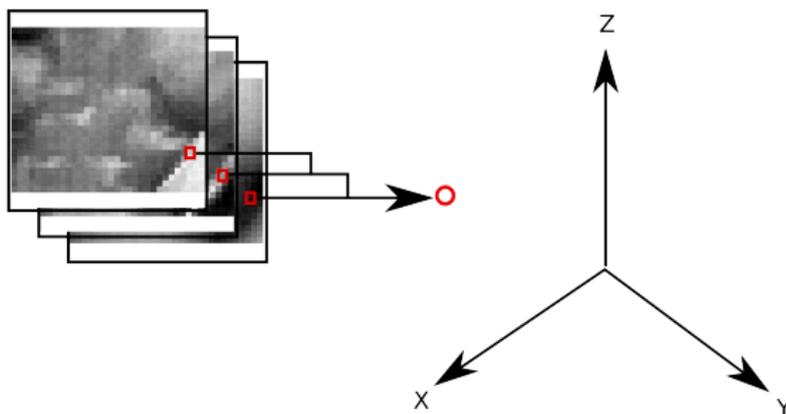
Utilisation du concept de niche écologique



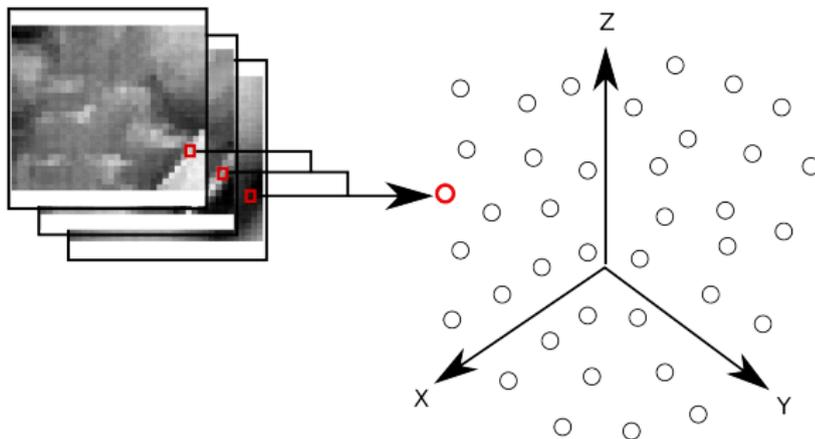
Utilisation du concept de niche écologique



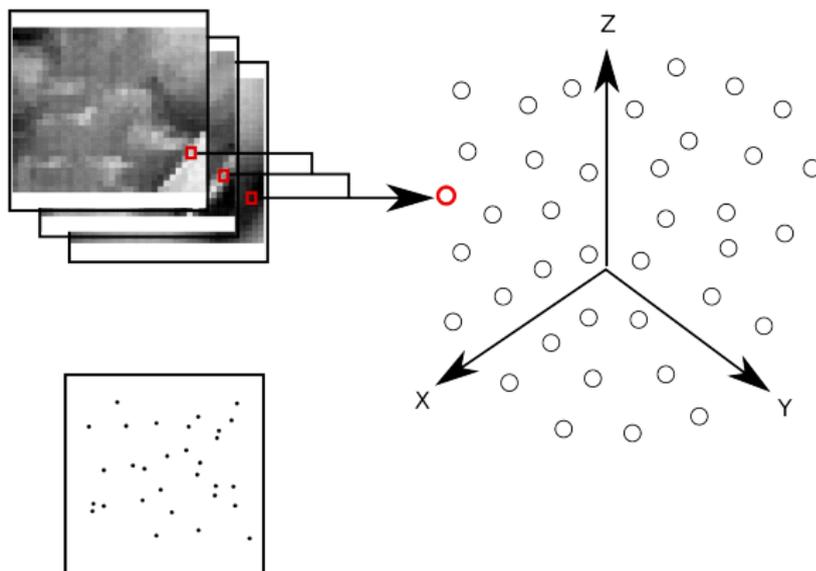
Utilisation du concept de niche écologique



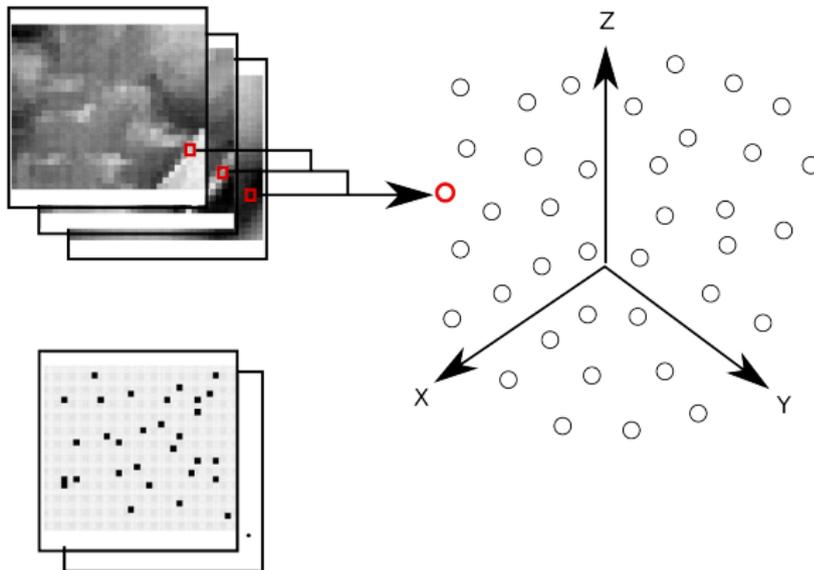
Utilisation du concept de niche écologique



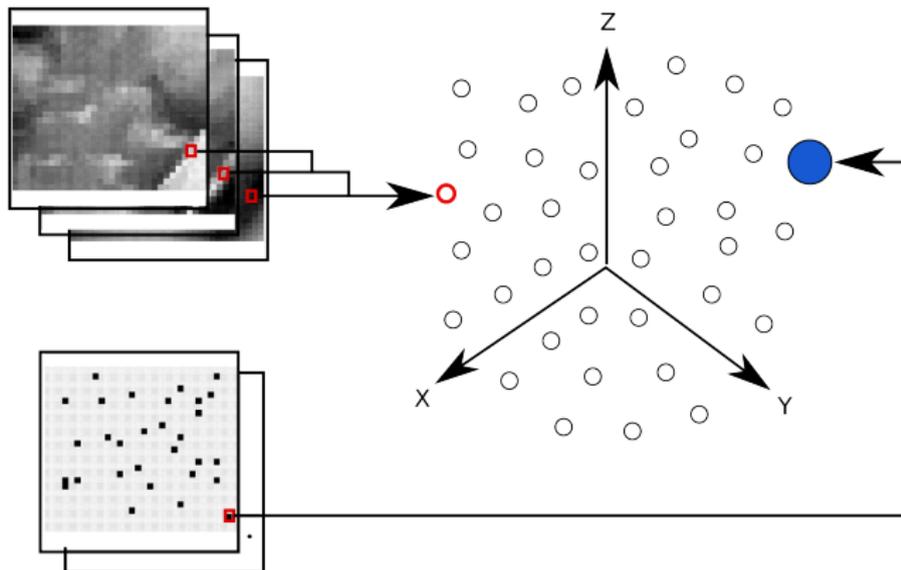
Utilisation du concept de niche écologique



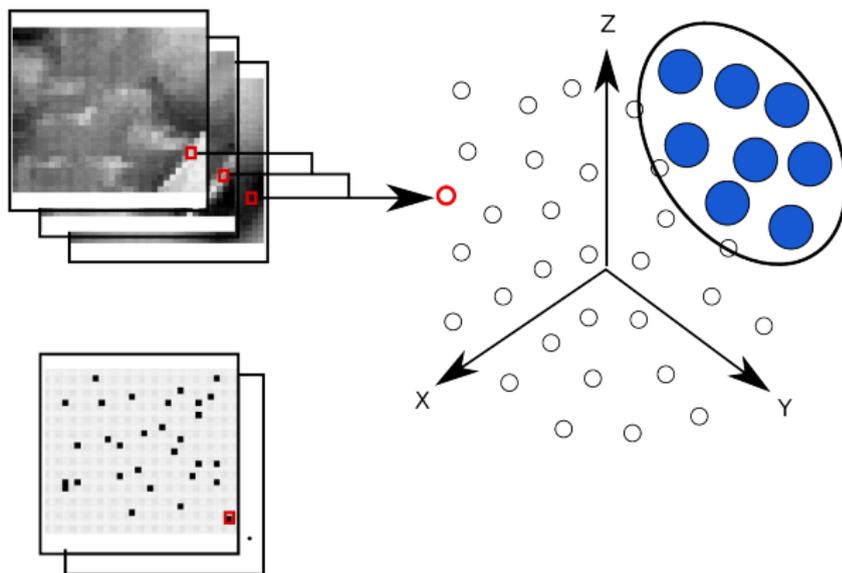
Utilisation du concept de niche écologique



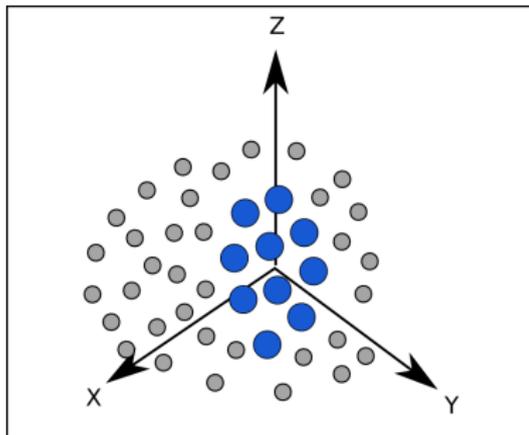
Utilisation du concept de niche écologique



Utilisation du concept de niche écologique



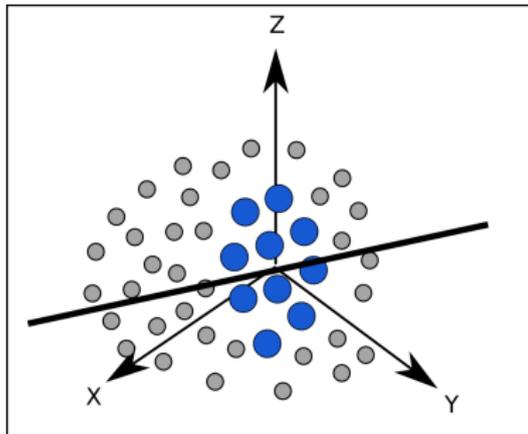
La Mahalanobis distances factor analysis (MADIFA)



→ centrage de l'espace sur la niche

Calenge, Darmon, Basille, Loison & Jullien (2008). Ecology, 89 :555-566.

La Mahalanobis distances factor analysis (MADIFA)

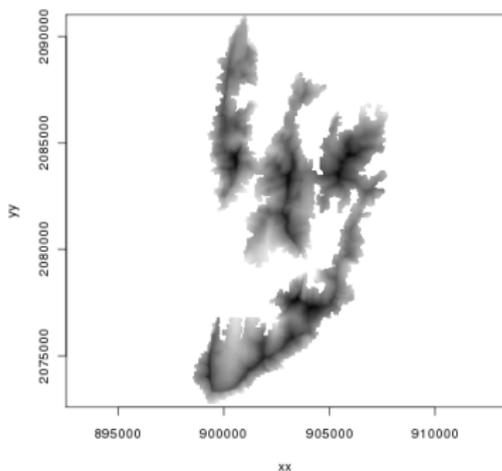


→ centrage de l'espace sur la niche

→ recherche des directions dans lesquelles les points disponibles sont les plus éloignés en moyenne du barycentre de la niche (au sens de la métrique de Mahalanobis).

Calenge, Darmon, Basille, Loison & Jullien (2008). Ecology, 89 :555-566.

La classe "asc" pour stocker des cartes raster



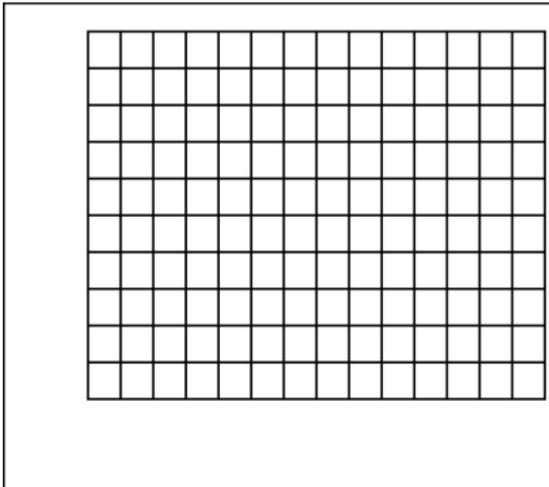
Fichiers raster ASCII générés par un grand nombre de logiciels

```
> carte <- import.asc("altitude.asc")
> carte
Raster map of class "asc":
Cell size: 50
Number of rows: 369
Number of columns: 224
Type: numeric
>
> image(carte)
```

La classe "asc" pour stocker des cartes

La classe asc, c'est :

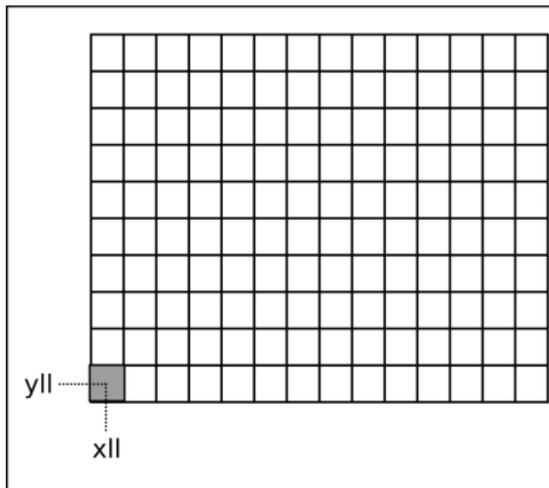
- Une matrice à ℓ lignes et c colonnes ;



La classe "asc" pour stocker des cartes

La classe asc, c'est :

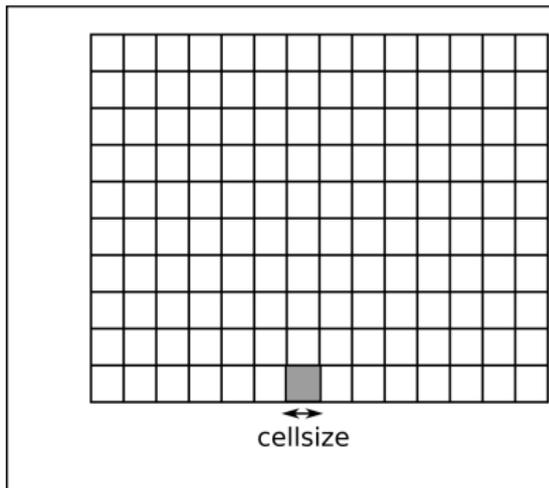
- Une matrice à ℓ lignes et c colonnes ;
- Des attributs x_{11} et y_{11} ;



La classe "asc" pour stocker des cartes

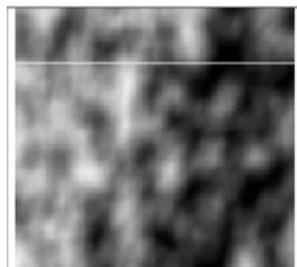
La classe asc, c'est :

- Une matrice à ℓ lignes et c colonnes ;
- Des attributs $x11$ et $y11$;
- Un attribut `cellsize` contenant la taille d'un pixel ;

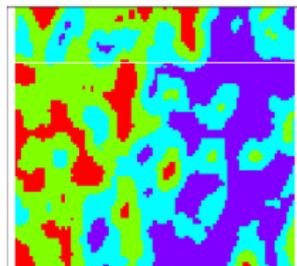


La classe "asc" pour stocker des cartes

numeric



factor

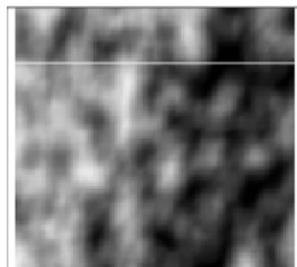


La classe asc, c'est :

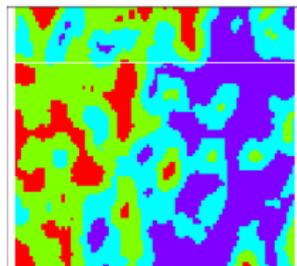
- Une matrice à ℓ lignes et c colonnes ;
- Des attributs $x11$ et $y11$;
- Un attribut `cellsize` contenant la taille d'un pixel ;
- Un attribut `type` définissant le type de la carte ;

La classe "asc" pour stocker des cartes

numeric



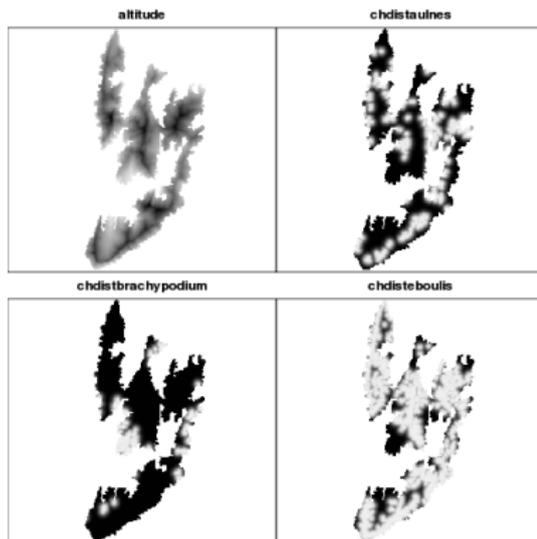
factor



La classe asc, c'est :

- Une matrice à ℓ lignes et c colonnes ;
- Des attributs x_{11} et y_{11} ;
- Un attribut `cellsize` contenant la taille d'un pixel ;
- Un attribut `type` définissant le type de la carte ;
- éventuellement, si `type == "factor"`, un attribut `levels` définissant les niveaux de la carte.

La classe "kasc" pour stocker des cartes multicouches



Plusieurs variables sont souvent cartographiées sur la même zone (même nombre de lignes et de colonnes, mêmes attributs xll et yll, même résolution) → cartes multicouches "kasc" ;

La classe "kasc" pour stocker des cartes multicouches

```
> carte <- import.asc("altitude.asc")
> carte2 <- import.asc("chdistaulnes.asc")
> carte3 <- import.asc("chdistbrachipodium.asc")
> carte4 <- import.asc("chdisteboulis.asc")
>
> cartetot <- as.kasc(list(altitude = carte,
+                          chdistaulnes = carte2,
+                          chdistbrachipodium = carte3,
+                          chdisteboulis = carte4))
> cartetot
```

Raster map of class "kasc":

Cell size: 50

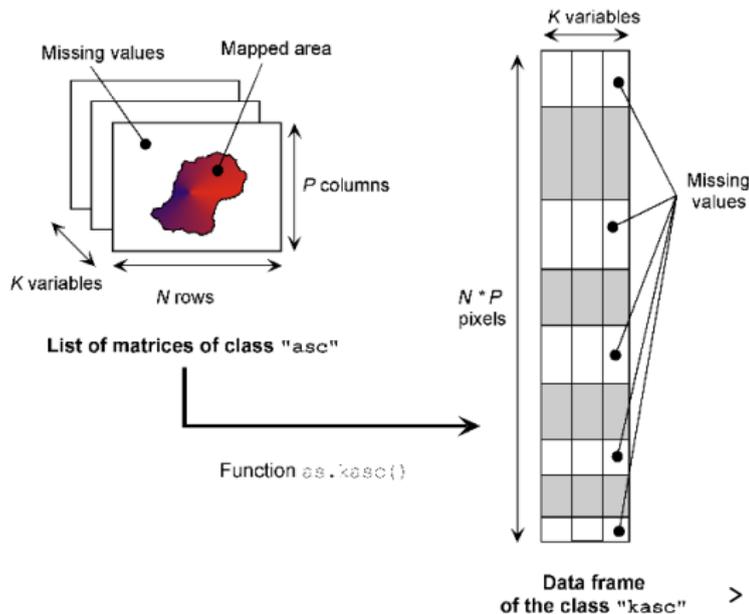
Number of rows: 369

Number of columns: 224

Variables measured:

1. altitude: numeric
2. chdistaulnes: numeric
3. chdistbrachypodium: numeric
4. chdisteboulis: numeric

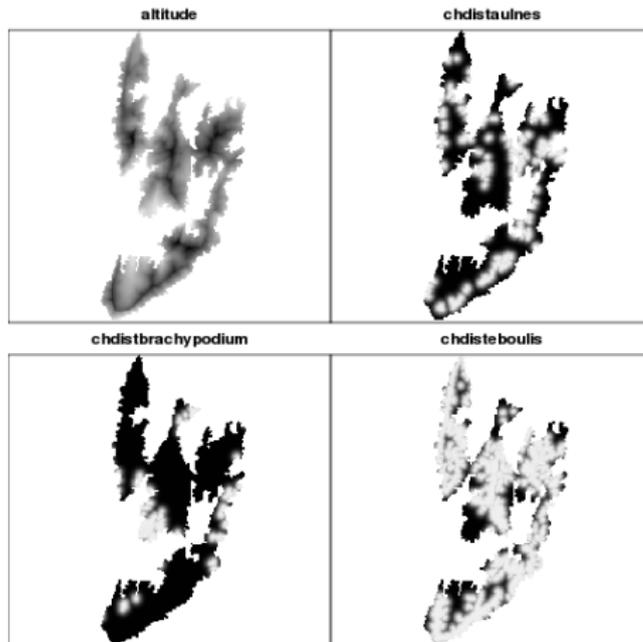
La classe "kasc" pour stocker des cartes multicouches



- Chaque matrice "asc" est transformée en vecteur ;
- Ces vecteurs sont concaténés en un `data.frame` ;
- attributs : `xll`, `yll`, `cellsize`, `nrow` et `ncol` ;

```
> image(cartetot)
> explore.kasc(cartetot)
```

La classe "kasc" pour stocker des cartes multicouches



- Chaque matrice "asc" est transformée en vecteur ;
- Ces vecteurs sont concaténés en un data.frame ;
- attributs : xll, yll, cellsize, **nrow** et **ncol** ;

```
> image(cartetot)
> explore.kasc(cartetot)
```

Nombreuses fonctions de gestion des cartes

En pratique : mettre toutes les cartes dans un dossier, puis :

```
> dir <- list.files("cartes")  
> dir2 <- paste("cartes", dir, sep="/")  
> liasc <- lapply(dir2, import.asc)  
> names(liasc) <- sapply(strsplit(dir, "\\."), function(x) x[[1]])  
> kasc <- as.kasc(liasc)
```

Fonction de gestion des cartes :

- **subsetmap** : Récupérer une partie d'une carte ;
- **setmask** : Positionner un masque sur une carte ;
- **lowres** : diminuer la résolution d'une carte ;
- **labcon** : compter le nombre de composantes connexes
- etc.

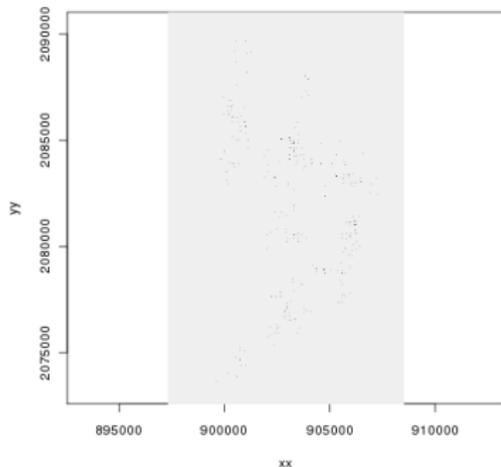
Mise en relation avec localisations de chamois



Les localisations peuvent être stockées dans un data frame classique.

```
> head(df)
      x      y
1 903600 2084300
2 903800 2083650
3 904100 2084100
4 904600 2083900
5 905300 2084300
6 905750 2078750
>
> image(kasc, 1)
> points(df)
```

Dénombrement des chamois dans les pixels de la carte

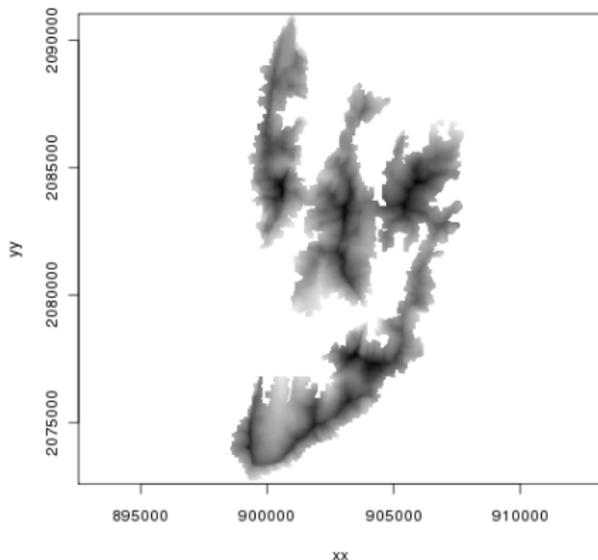


Plusieurs opérations possibles dans adehabitat, impliquant des points et des cartes raster (`join.kasc`, `buffer`, etc.).

Nous allons dénombrer les localisations de chamois dans chaque pixel de la carte.

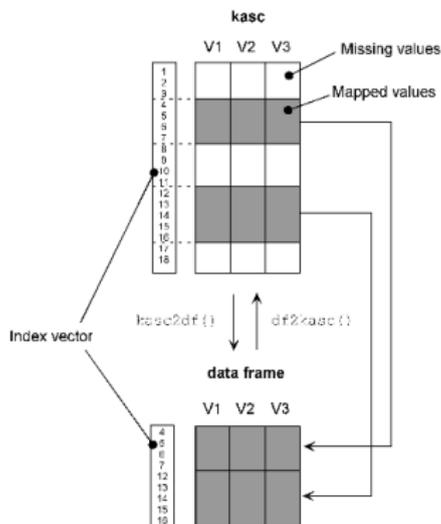
```
> cp <- count.points(df, kasc)
> cp
Raster map of class "asc":
Cell size: 50
Number of rows: 369
Number of columns: 224
Type: numeric
> image(cp)
```

Transformer un kasc en tableau



Une carte est codée sous la forme d'une matrice rectangulaire de pixels. La zone étudiée n'est pas nécessairement rectangulaire. Les zones non cartographiées sont remplies avec des NA.

La fonction `kasc2df`



La fonction `kasc2df` permet de transformer un objet de classe `kasc` en `data frame`.

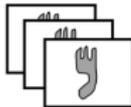
... En éliminant les “zones de données manquantes” du `kasc`.

La composante `index` permettra de reconstruire une carte à partir du tableau (fonction `df2kasc`).

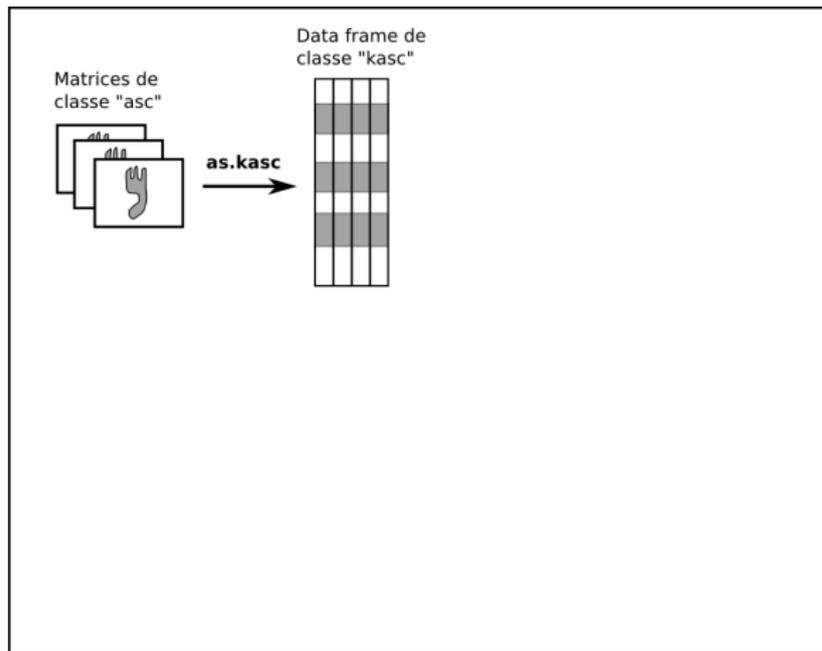
```
litab <- kasc2df(kasc)
cpt <- as.vector(cp)[litab$index]
```

En résumé

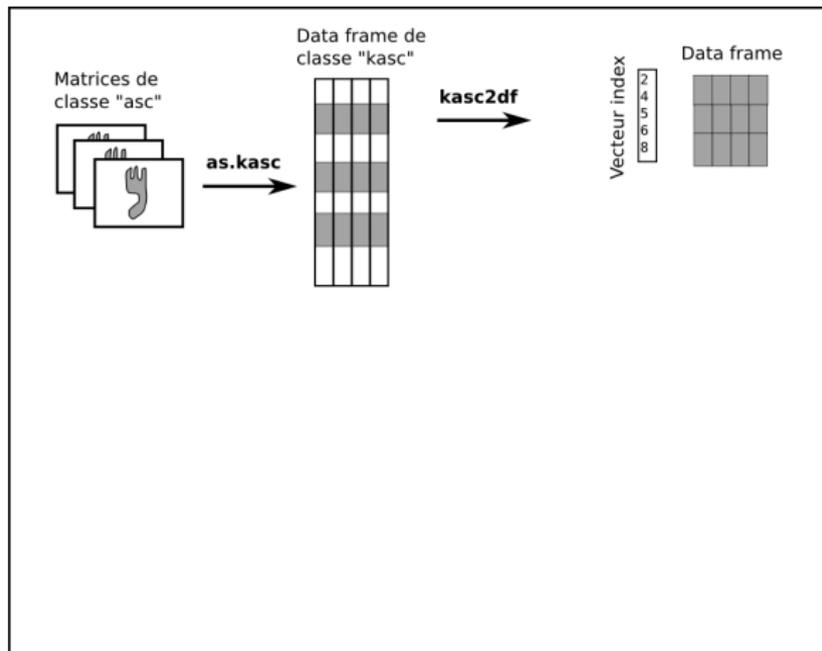
Matrices de
classe "asc"



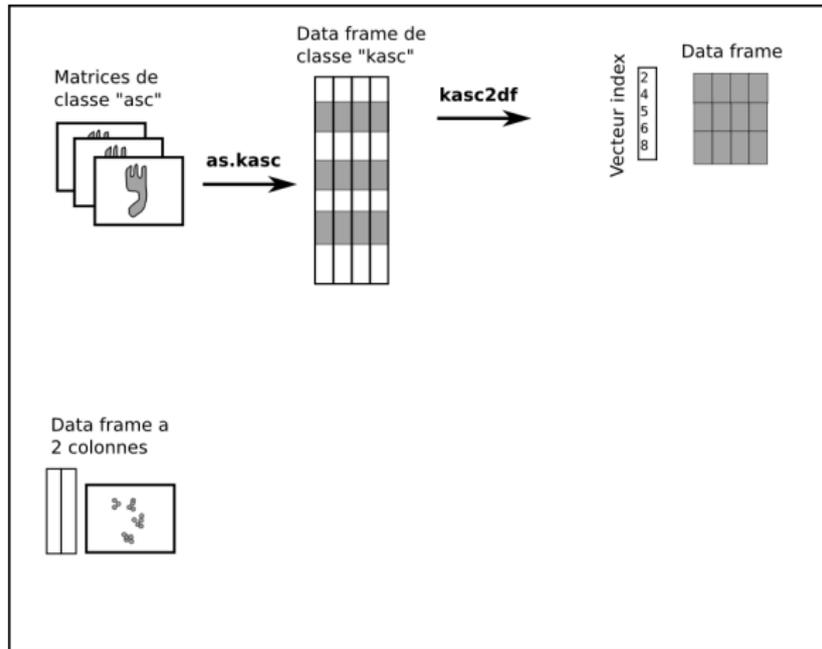
En résumé



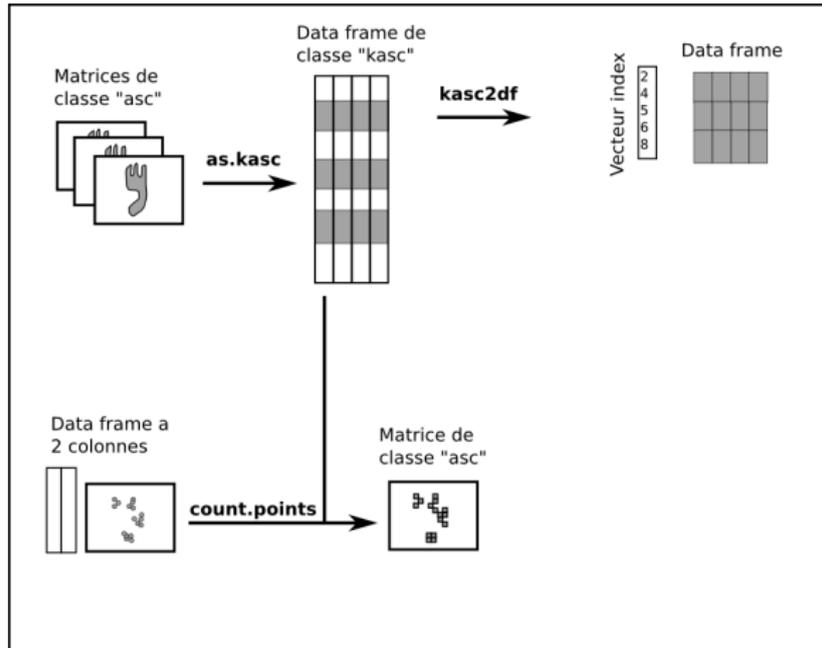
En résumé



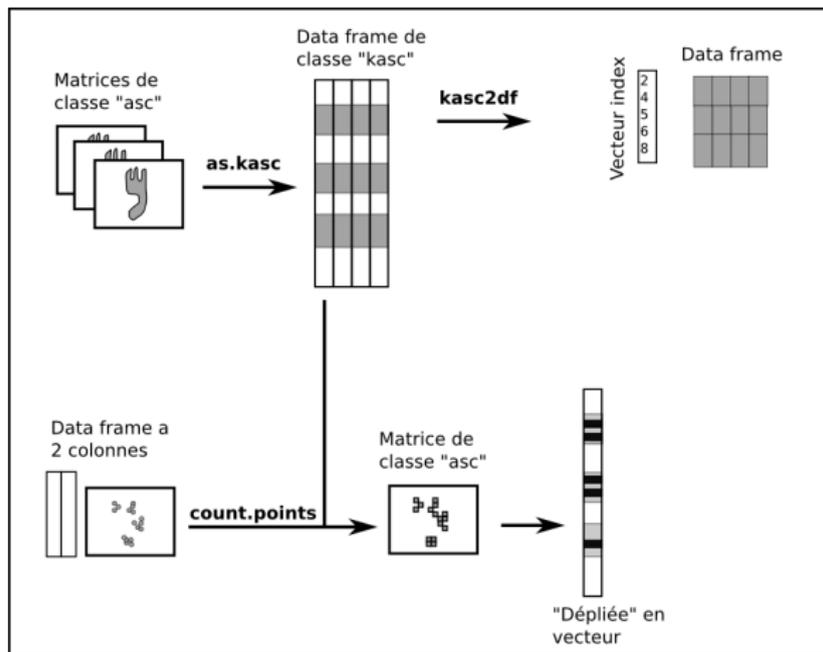
En résumé



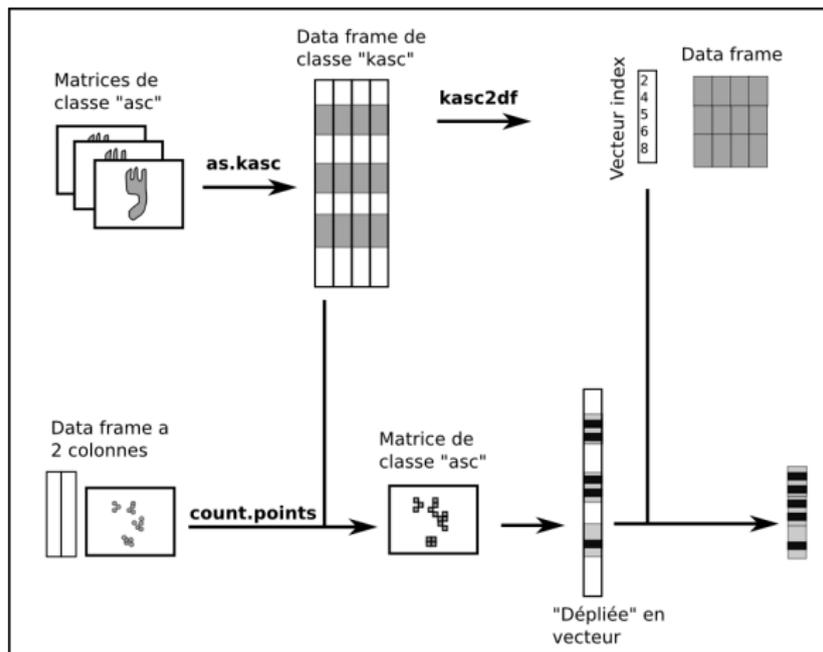
En résumé



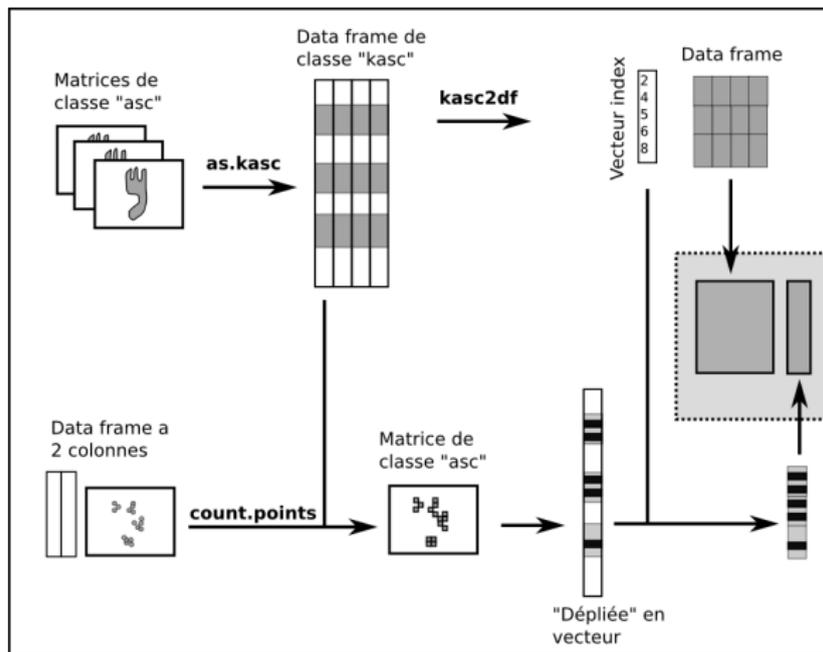
En résumé



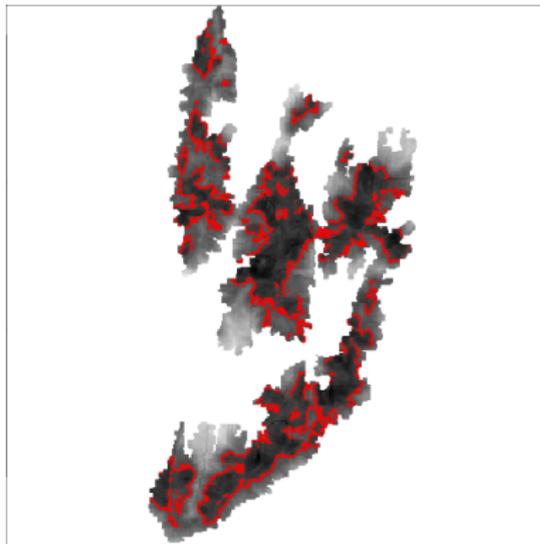
En résumé



En résumé



Cartographie des scores de la MADIFA



```
cart <- df2kasc(mad$li, litab$index, kasc)
image(getkasc(cart,1))
contour(getkasc(cart,1), add=TRUE,
        nlevels=2, col="red", lwd=2)
```

Plan

- 1 Introduction
- 2 Découverte des classes d'objet d'adehabitat
- 3 Vers une meilleure gestion du spatial : le package sp

La “suite sp”

En 2002-2004, nombreux utilisateurs commençaient à vouloir manipuler de l'info spatiale sous R, mais manque d'outils
→ spdep, ade4, maps, spatstats, adehabitat, maptools, etc.

Chaque package développait indépendamment ses propres classes de données.

Edzer Pebesma et Roger Bivand on cherché à créer un carrefour entre ces packages → **package sp**

Le coeur du package est centré sur la définition de ces classes de données → Classes de données S4

Quelques fonctions spatiales sont aussi proposées (**overlay**) ;

Les classes S4

La classe d'un objet est un attribut qui permet d'identifier le type d'un objet.

Utile pour :

- vérifier qu'un objet contient toutes les infos nécessaires à l'application d'une fonction (par exemple `cbind.data.frame` ne marche que sur des `data.frame`);
- définir le comportement de fonctions génériques (`cbind` appelle `cbind.data.frame` quand les arguments sont des `data.frames`).

Les classes S4

La classe d'un objet est un attribut qui permet d'identifier le type d'un objet.

Utile pour :

- vérifier qu'un objet contient toutes les infos nécessaires à l'application d'une fonction (par exemple `cbind.data.frame` ne marche que sur des `data.frame`);
- définir le comportement de fonctions génériques (`cbind` appelle `cbind.data.frame` quand les arguments sont des `data.frames`).

Deux mécanismes de définition de classe sous R :

- Classes S3 : les plus classiques (un attribut "class" définit la classe de l'objet)
- Classes S4 : classes **formelles** ; concept d'héritage formellement défini lui aussi : une classe A **étend** une classe B (la classe A possède les mêmes attributs que la classe B, + quelques autres).

Les classes S4

La classe d'un objet est un attribut qui permet d'identifier le type d'un objet.

Utile pour :

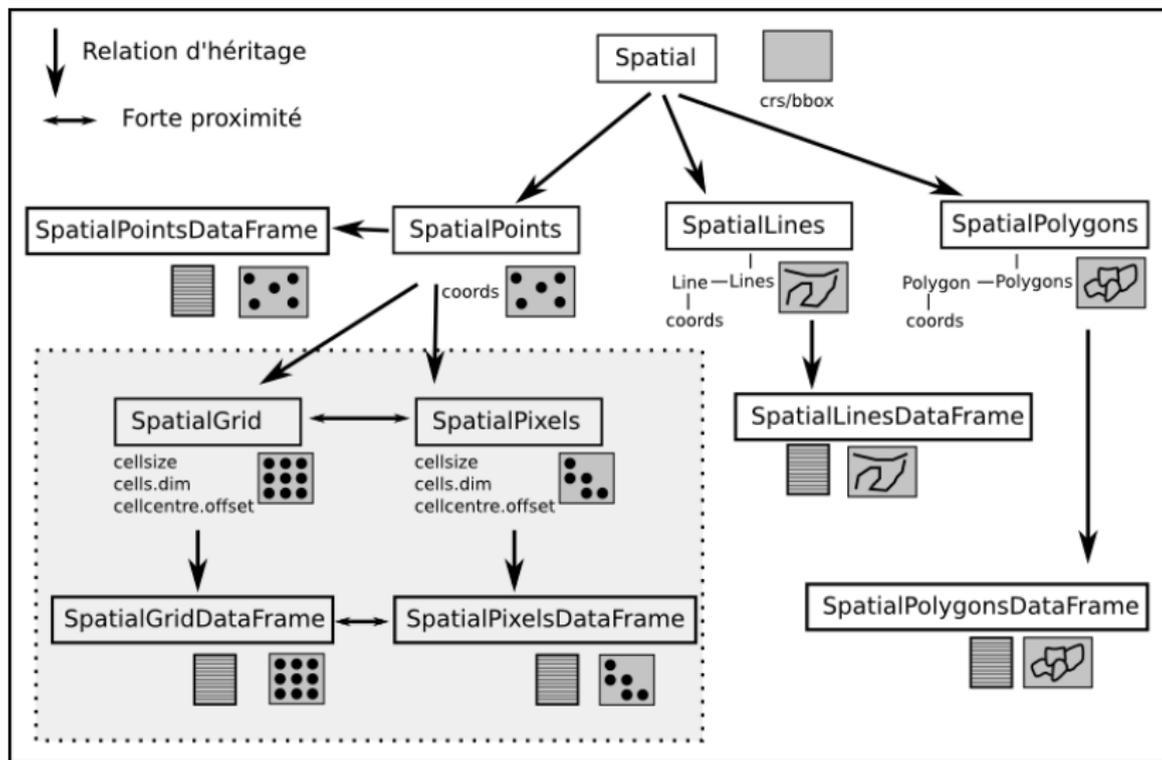
- vérifier qu'un objet contient toutes les infos nécessaires à l'application d'une fonction (par exemple `cbind.data.frame` ne marche que sur des `data.frame`);
- définir le comportement de fonctions génériques (`cbind` appelle `cbind.data.frame` quand les arguments sont des `data.frames`).

Deux mécanismes de définition de classe sous R :

- Classes S3 : les plus classiques (un attribut "class" définit la classe de l'objet)
- Classes S4 : classes **formelles** ; concept d'héritage formellement défini lui aussi : une classe A **étend** une classe B (la classe A possède les mêmes attributs que la classe B, + quelques autres).

S4 : Divise la communauté des utilisateurs : on y gagne beaucoup en rigueur, mais souvent aux dépens de la flexibilité...

Les classes de sp



Interactions entre sp et adehabitat

Fonctions de conversion :

- `kasc2spixdf` : convertir un "kasc" en "SpatialPixelsDataFrame" (vs. `spixdf2kasc`);
- `asc2spixdf` : convertir un "asc" en "SpatialPixelsDataFrame" ;

Interactions entre sp et adehabitat

Fonctions de conversion :

- `kasc2spixdf` : convertir un "kasc" en "SpatialPixelsDataFrame" (vs. `spixdf2kasc`);
- `asc2spixdf` : convertir un "asc" en "SpatialPixelsDataFrame" ;
- `area2spol` convertit un objet de la classe "area" en objet de la classe "SpatialPolygons" (vs. `spol2area`).
- `kver2spol` convertit un objet de la classe "kver" en objet de la classe "SpatialPolygons".

Interactions entre sp et adehabitat

Fonctions de conversion :

- `kasc2spixdf` : convertir un "kasc" en "SpatialPixelsDataFrame" (vs. `spixdf2kasc`);
- `asc2spixdf` : convertir un "asc" en "SpatialPixelsDataFrame" ;
- `area2spol` convertit un objet de la classe "area" en objet de la classe "SpatialPolygons" (vs. `spol2area`).
- `kver2spol` convertit un objet de la classe "kver" en objet de la classe "SpatialPolygons".
- `ltraj2spdf` convertit un objet de la classe "ltraj" en objet de la classe "SpatialPointsDataFrame".
- `ltraj2sldf` convertit un objet de la classe "ltraj" en objet de la classe "SpatialLinesDataFrame".

Intérêt central des classes de sp

- Les classes du package ont été pensées par des géographes pour une grande variété de problématiques (changement de projection, stockage cohérent du vecteur et du raster, etc.) ;

Intérêt central des classes de sp

- Les classes du package ont été pensées par des géographes pour une grande variété de problématiques (changement de projection, stockage cohérent du vecteur et du raster, etc.) ;
- Ces classes sont en général plus économes en mémoire que celles des autres packages (mais pas toujours) ;

Intérêt central des classes de sp

- Les classes du package ont été pensées par des géographes pour une grande variété de problématiques (changement de projection, stockage cohérent du vecteur et du raster, etc.) ;
- Ces classes sont en général plus économes en mémoire que celles des autres packages (mais pas toujours) ;
- Ce sont des classes standards → tremplin vers les autres packages “spatiaux” ;

Intérêt central des classes de sp

- Les classes du package ont été pensées par des géographes pour une grande variété de problématiques (changement de projection, stockage cohérent du vecteur et du raster, etc.) ;
- Ces classes sont en général plus économes en mémoire que celles des autres packages (mais pas toujours) ;
- Ce sont des classes standards → tremplin vers les autres packages “spatiaux” ;
- Un forum de discussion très actif est à la disposition des très nombreux utilisateurs (R-SIG-Geo) ;

Intérêt central des classes de sp

- Les classes du package ont été pensées par des géographes pour une grande variété de problématiques (changement de projection, stockage cohérent du vecteur et du raster, etc.) ;
- Ces classes sont en général plus économes en mémoire que celles des autres packages (mais pas toujours) ;
- Ce sont des classes standards → tremplin vers les autres packages “spatiaux” ;
- Un forum de discussion très actif est à la disposition des très nombreux utilisateurs (R-SIG-Geo) ;
- Inévitable si l'on veut travailler sur des cartes vecteur.

Quelques limites des classes de sp

- Peu flexible (c'est voulu : les classes sont "verrouillées" ; `fortune("S4")`);

Quelques limites des classes de sp

- Peu flexible (c'est voulu : les classes sont "verrouillées" ; `fortune("S4")`);
- Classes de données très instables (beaucoup de modifications d'une version à l'autre) : pas toujours simple de sortir des "rails" (Il ne faut utiliser que les fonctions fournies par le package) :

Indeed, it ought to have been that before, then the 2.3.* / 2.4.* change in how S4 classes are represented would have been hidden. Using coercion and access methods is much more robust than grabbing internal bits of objects (which are subject to change).

Quelques limites des classes de sp

- Peu flexible (c'est voulu : les classes sont "verrouillées" ; `fortune("S4")`);
- Classes de données très instables (beaucoup de modifications d'une version à l'autre) : pas toujours simple de sortir des "rails" (Il ne faut utiliser que les fonctions fournies par le package) :

Indeed, it ought to have been that before, then the 2.3.* / 2.4.* change in how S4 classes are represented would have been hidden. Using coercion and access methods is much more robust than grabbing internal bits of objects (which are subject to change).

- Conséquence : la structure interne des classes de données est parfois mal documentée (idem, c'est voulu ; heureusement, le forum est là) ;

Quelques limites des classes de sp

- Peu flexible (c'est voulu : les classes sont "verrouillées" ; `fortune("S4")`);
- Classes de données très instables (beaucoup de modifications d'une version à l'autre) : pas toujours simple de sortir des "rails" (Il ne faut utiliser que les fonctions fournies par le package) :

Indeed, it ought to have been that before, then the 2.3.* / 2.4.* change in how S4 classes are represented would have been hidden. Using coercion and access methods is much more robust than grabbing internal bits of objects (which are subject to change).

- Conséquence : la structure interne des classes de données est parfois mal documentée (idem, c'est voulu ; heureusement, le forum est là) ;
- Reste gourmand en mémoire !

Mais R n'est pas un Système d'information Géographique !

Mais réussir l'intégration d'un logiciel de statistiques dans un SIG reste un enjeu majeur !

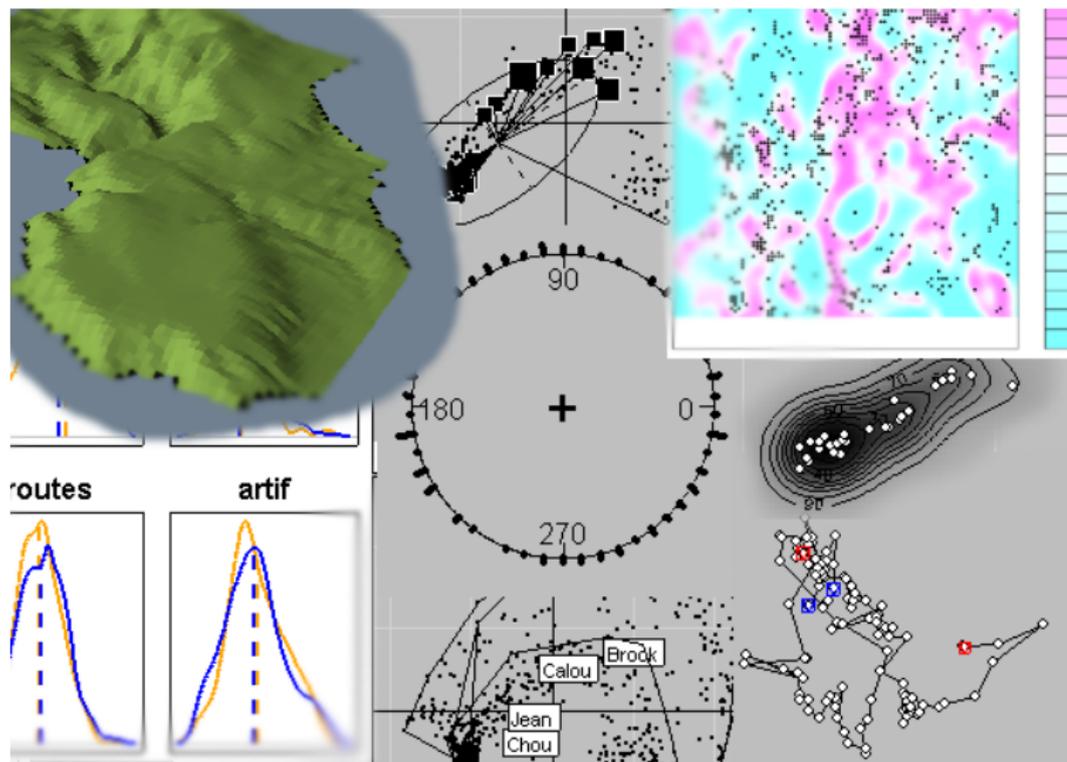
Il reste difficile de se passer des interactions avec les SIG lors d'analyses spatiales.

Nombreux SIG libres peuvent être utilisés (douloureusement pour l'utilisateur de base) en interaction avec R (PostgreSQL/PostGIS, GRASS).

→ Travail à effectuer pour intégrer R dans un environnement plus large (intégrant des logiciels de base de données, des SIG, etc.)

Urbano, Cagnacci, Calenge, Cameron, Dettki & Neteler. In prep. How not to drown in data : a new approach for managing wildlife location data.

Cagnacci & Urbano. 2008. Managing wildlife : A spatial information system for GPS collars data. Environmental Modelling & Software 23. 957-959.



Merci de votre attention !