3area : manuel de référence. v1.2

Synopsis.

`3area -1 -2 -v filein fileout`

Description.

Produit le fichier *fileout* de format `.3ia` à partir du fichier *filein*. Le fichier *filein* a un format très semblable au format `.3ia`, spécialisé pour la biogéographie.

Le travail effectué par `3area` consiste à convertir des cladogrammes exprimés sur des taxons en caractères exprimés sur des aires, chaque aire pouvant abriter plusieurs taxons et chaque taxon pouvant occuper plusieurs aires. Le fichier d'entrée donne :

- Une liste de taxons.
- Une liste d'aires.
- La répartition des taxons sur les aires.
- Une liste de cladogrammes exprimés sur les taxons.

La conversion s'effectue en deux tranches. Par souci de clarté, les arbres générés à chaque étape sont désignés comme suit :

type E	cladogramme du fichier en entrée
type A0	pseudo-caractère généré après substitution des taxons par les aires.
type A	pseudo-caractère généré par expansion : un pseudo-caractère de type A0 génère un ou plusieurs caractères de type A.
type B	caractères généré par renormalisation d'un pseudo-caractères de type A.

Tranche A : substitution des taxons par les aires.

Cette tranche s'effectue elle-même en deux étapes. Pour chaque cladogramme :

- Chaque taxon est substitué par la liste des aires occupées par ce taxon (type E → A0).
- Chaque cladogramme génère un certain nombre de copies de lui-même (type A0 → A). Dans chaque copie, une aire seulement est prise dans chaque liste d'aires substituée aux taxons, de façon à générer toutes les combinaisons possibles d'aires.

Exemple :

Liste des taxons : `A`, `B`, `C`, `D`.

Liste des aires :
```
a1 : A B
a2 : A C D
a3 : C D
```

Le cladogramme sur taxons `(A (B (C D)))` donne par substitution l'arbre `(a1_a2 (a1 (a2_a3 a2_a3)))`, qui génère les pseudo-caractères sur aires :

```
(a1 (a1 (a2 a2)))
(a1 (a1 (a2 a3)))
(a1 (a1 (a3 a2)))
(a1 (a1 (a3 a3)))
(a2 (a1 (a2 a2)))
(a2 (a1 (a2 a3)))
(a2 (a1 (a3 a2)))
(a2 (a1 (a3 a3)))
```

Tranche B : renormalisation des pseudo-caractères.

Dans cette tranche, les pseudo-caractères générés par la tranche A, dans lesquels les « taxons » (aires) peuvent être redondants, sont renormalisés pour générer des caractères aux « taxons » non redondants (type A → B). Cette renormalisation utilise trois opérations de base : la suppressions d'une branche (= d'un sous-arbre), la duplication d'un pseudo-caractère, la suppression d'un niveau hiérarchique.

Une classe est définie comme la liste des « taxons » composants un arbre ou un sous-arbre. Dans un arbre, chaque nœud non terminal possède donc une classe associée, liste des taxons sous-tendus par la sous-arborescence de ce nœud.

Pour un pseudo-caractère donné, l'analyse s'effectue en parcourant l'arbre depuis la racine et en s'arrêtant à chaque nœud non terminal rencontré. A chaque nœud, deux opérations sont effectuées successivement : l'élimination des singletons non significatifs, puis l'élimination des paralogies. Pour le nœud N :

- On calcule la classe associée à chacun des nœuds fils de N. On obtient donc une liste de classes.
- Pour chaque classe C ne comprenant qu'un seul taxon, on recherche s'il existe une autre classe incluant ce taxon. Si c'est le cas, le sous-arbre ayant généré C est éliminé de l'arbre. Cette opération permet d'éliminer les redondances liées à la paralogie de type I.
  Exemple : l'arbre (A (**B** (C D) (**B** E) F))
  est simplifié en (A ((C D) (**B** E) F))
- Les classes comprenant au moins deux taxons sont ensuite comparées deux à deux. S'il existe des classes partageant des taxons en commun, le pseudo-caractère est dupliqué en autant de copies qu'il est nécessaire pour que chacune ne soit composée que de classes sans taxon commun. Cette opération permet d'éliminer les redondances liées à la paralogie de type II.
  Exemple : l'arbre (A ((**B C**) (**C D**) (D E) (X Y) F))
  donne les copies (A ((**B C**) (D E) (X Y) F)) et (A ((**C D**) (X Y) F))
  Dans cet exemple, (B C) et (C D) déterminent la duplication du pseudo-caractère. Ensuite, (D E) et (X Y) sont inclus dans la première copie car sans taxon communs entre eux ni avec (B C) ; par contre, (D E) n'est pas repris dans la seconde copie, car paralogique avec (C D).
- Dans les deux cas, si, à la suite des éliminations de sous-arbres par l'un ou l'autre mécanisme, il ne subsiste au nœud N qu'un seul fils non terminal, le niveau hiérarchique représenté par celui-ci, devenu inutile, est supprimé.
  Exemple : l'arbre (A (**B** (**B** E)))
  est simplifié en (A ((**B** E))) puis en (A (B E))

A l'issue du traitement, le nettoyage final suivant est effectué.

- Les derniers cas de niveau hiérarchique non significatifs sont éliminés (une partie a déjà été éliminée en cours de traitement ; voir dernier point ci-dessus) : ainsi, les formes (A (**B**) C) (non simplifiées en cours de traitement) sont simplifiées en (A B C).
- Les caractères générés non porteurs d'information sont éliminés. Il s'agit :
  - des caractères réduits à une liste de taxons sans hiérarchie, comme (A B C D E).
  - des caractères réduits à un ou deux taxons ; en réalité, ce cas se ramène au précédent par le jeu des suppressions des niveaux hiérarchiques non significatifs.
- Les redondances sont éliminées : on ne conserve qu'un seul exemplaire de tous les caractères identiques éventuellement générés à partir d'un même pseudo-caractère initial. Attention : les redondances ne sont recherchées (et éliminées) qu'entre les caractères de type B générés par un même pseudo-caractère de type A ; il peut donc subsister en fin de traitement des caractères identiques : ils auront été générés à partir de plusieurs pseudo-caractères de type A, eux-mêmes générés par un même cladogramme du fichier initial ou par plusieurs.

Les deux tranches peuvent être traitées séparément grâce aux options `-1` et `-2`. Ainsi, la série de commandes :

```
% 3area -1 filein tmp
% 3area -2 tmp fileout
```

est équivalente à la commande :

```
% 3area filein fileout
```

(dans les deux fichiers de sortie, les différents caractères de type B générés par chaque pseudo-caractère de type A peuvent cependant se présenter dans un ordre différent).

Si l'option -v n'est pas fournie, ou si la sortie standard est redirigée vers un fichier, le programme affiche la progression par une ligne de la forme :

[*cc*] *aaa*/*bbb*

avec

- *cc* Numéro du cladogramme en cours de traitement.
- *aaa* Nombre de pseudo-caractères traités pour ce cladogramme. Il s'agit de pseudo-caractères de type A, générés par la tranche A, chacun pouvant générer plusieurs caractères finaux.
- *bbb* Nombre de pseudo-caractères totaux (de type A) attendus pour le cladogramme.

Auteur : J. Ducasse, mai 2006.

Options.

- 1 3area ne procède qu'à la première phase de traitement (tranche A). Dans ce cas, les arbres produits dans le fichier *fileout* sont ceux générés par simple substitution des taxons par les aires (type A, pseudo-caractères), sans renormalisation.
- 2 3area ne procède qu'à la deuxième phase de traitement (tranche B) pour générer le fichier *fileout*. Dans ce cas, *filein* est un fichier de format .3ia. Cette option permet notamment d'analyser le processus de renormalisation (tranche B) sur des pseudo-caractères fournis explicitement, au lieu de pseudo-caractères générés indirectement par la tranche A.
- v Affiche les étapes de progression et les données intermédiaires mises en place (cf. ci-dessous).

Format du fichier d'entrée *filein*.

Ce fichier a un format semblable au fichier d'entrée .3ia (voir le manuel du programme 3ia), avec les différences suivantes :

- Les sections Dimensions et Référentiel sont ignorées et facultatives.
- Il comprend une section Aires donnant les noms des aires et la liste des taxons qui habitent cette aire.

Exemple de la section Aires :

```
Aires
  Corse :           G C K H
  France :          D L J
  Italie du nord :  D H
  Sicile :          E F C H
  Suisse :          B A I
;
```

Cette section a la syntaxe suivante :

- Chaque ligne est formée de trois groupes de termes : le nom de l'aire, le séparateur ":", la liste des taxons présents sur cette aire. Chaque terme peut être précédé ou suivi d'un nombre quelconque de blancs.
- Le nom de l'aire peut comprendre plusieurs mots ; il commence au premier caractère non blanc et se termine au dernier caractère non blanc précédant le ":". Il est formé d'un nombre quelconque de caractères quelconques.
- Les taxons sont référencés par leur code ; 3area contrôle que les codes cités ont précédemment été définis dans la section Taxons.

Format du fichier d'entrée *filein* avec l'option -2.

Avec l'option -2, 3area suppose que la tranche A a déjà été effectuée, et procède seulement à la tranche B. Le fichier d'entrée *filein* a alors le format d'un fichier .3ia : seules les sections Taxons et Descriptions sont

utilisées. Dans cette dernière, les codes utilisés référencent les taxons définis dans la section `Taxons` et représentent en principe des aires.

Une différence importante avec un fichier `.3ia` normal est que les pseudo-caractères donnés en section `Descriptions` peuvent inclure plusieurs fois le même taxon. Ce sera justement le rôle du traitement (tranche B) de renormaliser les pseudo-caractères.

Format du fichier de sortie *`fileout`*.

Ce fichier a le format de fichier `.3ia` (voir le manuel de `3ia`). Les différentes sections sont renseignées comme suit :

- La section Titre égale la section correspondante du fichier d'entrée.
- La section `Dimensions` est renseignée.
- Dans la section `Taxons`, chaque « taxon » correspond à une aire du fichier initial. Le code est généré automatiquement.
- La section `Descriptions` donne les caractères générés. Les numéros des caractères sont consécutifs. Les cladogrammes d'origine sont aussi indiqués en commentaire comme suit :

  `Source` Cladogramme (type E) du fichier d'entrée, identifié par son numéro `[`*`nn`*`]`.

  `Renormalisation`
  Pseudo-caractère de type A obtenu par substitution des taxons par les aires, identifié par le numéro du cladogramme initial *`nn`* et un numéro séquentiel *`ssss`* : `[`*`nn`*`.Exp`*`ssss`*`]`.

  Si l'option `-2` a été fournie, les lignes `Source` n'existent pas, et les lignes `Renormalisation` donnent le pseudo-caractère du fichier d'entrée.

Format du fichier de sortie *`fileout`* avec l'option `-1`.

Le fichier a le même contenu que sans l'option `-1`, avec les différences suivantes dans la section `Descriptions` :
- Les arbres représentent les pseudo-caractères issus des cladogrammes du fichier d'origine par substitution des taxons par les aires (type A), sans renormalisation.
- Les cladogrammes d'origine sont indiqués par les lignes de commentaire `Source`.
- Il n'y a pas de lignes de commentaire `Renormalisation`.

Détail des informations affichées sous l'option `-v`.

Lorsque l'option `-v` est active, le travail réalisé est affiché au fur et à mesure comme suit :

`Taxon → Aires`
Suite à l'analyse du fichier d'entrée, donne la table de correspondance inverse de celle figurant dans le fichier : liste des aires occupée par chaque taxon.

`Codes des aires`
Donne les codes générés automatiquement pour les aires, et qui seront utilisés dans toute la suite.

`Traitement caractère [`*`cc`*`]`
Débute l'analyse du cladogramme de type E *`cc`*. Tout ce qui suit, jusqu'à la prochaine ligne de même type, concerne le traitement de ce cladogramme.

`Expansion [`*`cc`*`.Exp`*`ssss`*`] : (……)`
Débute l'analyse du pseudo-caractère indiqué. Tout ce qui suit, jusqu'à la prochaine ligne de même type, concerne ce pseudo-caractère. Il s'agit d'un pseudo-caractère de type A généré par substitution des taxons par les aires à partir du cladogramme *`cc`*. Les pseudo-caractères générés sont identifiés par un numéro séquentiel *`ssss`* unique pour tout le traitement.

Analyse [*aa*] (……)

Chaque paragraphe Analyse correspond au traitement d'un pseudo-caractère, qui est affiché avec son numéro *aa*. Ce numéro est séquentiel et unique pour tout le traitement.

Le premier paragraphe Analyse suivant la ligne Expansion concerne le pseudo-caractère généré de type A lui-même. Les paragraphes Analyse suivants concernent des pseudo-caractères générés à partir de celui-ci par renormalisation. Lorsqu'un pseudo-caractère est généré par scission, l'une des copies se substitue au pseudo-caractère en cours de traitement et conserve son numéro, les autres prennent place en fin de liste et seront traitées lorsque leur tour viendra.

Seuls les nœuds dont le traitement entraîne une modification de l'arbre sont affichés, sous le terme Composante suivi du sous-arbre en question. La modification en question est ensuite affichée : suppression d'un singleton ou scission par suppression des paralogies (voir texte principal). Dans tous les cas, le sous-arbre après modification est affiché derrière ″-->″, ou ″*->″ si la modification a entraîné la suppression d'un niveau hiérarchique. Pour une scission, les sous-arbres des différentes copies sont affichés, suivis du numéro de la copie du pseudo-caractère qui emporte cette version du sous-arbre (sauf le premier, qui est emporté par le pseudo-caractère courant).

A l'issue du traitement complet du pseudo-caractère, le caractère généré est affiché derrière ″->″, et son éventuelle élimination est indiquée s'il est réduit à moins de trois taxons.

Caractère redondant

Les caractères supprimés pour redondance sont affichés.

Liste des caractères générés

Il s'agit de la liste finale des caractères de type B générés par le pseudo-caractère de type A identifié [*cc*.Exp*ssss*]. C'est cette liste qui sera reportée dans le fichier en sortie. Chaque caractère est muni de deux numéros :

- Le premier est le numéro unique utilisé jusqu'ici.
- Le second est le numéro que le caractère possèdera dans le fichier en sortie. Il peut y avoir un décalage entre les deux séries si des caractères générés ont été supprimés (redondants ou de moins de trois taxons).

Historique.

version 1.2 : décembre 2009

Changement de nom → 3area.

version 1.1 : septembre 2006

Restructuration du programme : les arbres générés ne sont plus stockés en mémoire pour une écriture finale totale dans le fichier de sortie. Au contraire, chaque arbre du fichier d'entrée, ainsi que chaque arbre généré par expansion, est traité de façon autonome et les ressources sont libérées avant de passer au suivant. Ceci amène une optimisation très conséquente pour les gros traitements.
Correction de l'algorithme de renormalisation, qui pouvait donner des résultats faux dans la version précédente !
Elimination des arbres générés sans structure hiérarchique.
Introduction de l'option -1.

version 1 : mai 2006