

Synopsis.

```
3iaout -g nexus file.3iz [outfile]
3iaout -g compt file.3iz outfile
3iaout -g interp file.3ia file.3iz [outfile]
```

Description.

Analyse un fichier *file.3iz* produit par l'utilitaire *3ia*, et réalise diverses opérations d'extraction en fonction de la commande spécifiée par le premier argument. Les autres arguments dépendent de la commande.

Les commandes disponibles sont :

nexus	Produit un fichier au format Nexus. Le fichier Nexus est produit dans le fichier <i>outfile</i> , ou à l'écran à défaut.
compt	Produit deux fichiers au format Component (Page, 1993) : le premier comprend tous les arbres de compatibilité retenus, le second uniquement l'arbre d'intersection. Un « C » est ajouté au nom <i>outfile</i> pour le second fichier. Les fichiers sont également compatibles avec le programme TreeView (Page, 2001). Remarque : la génération de deux fichiers (au lieu d'un) est nécessitée par la contrainte Component que tous les taxons cités apparaissent dans tous les arbres.
interp	Produit une interprétation des arbres de compatibilité donnés dans le fichier <i>file.3iz</i> par rapport aux caractères les ayant générés, donnés par le fichier <i>file.3ia</i> (fichier d'entrée de <i>3ia</i>). L'interprétation est produite dans le fichier <i>outfile</i> , ou à l'écran à défaut.

Pour les commandes *nexus* et *compt*, le programme utilise des fichiers modèles pour générer le ou les fichiers en sortie. Ces modèles sont formés de texte libre, et de formes @XXX@ qui seront substituées par les données extraites de *file.3iz* et recalculées. La substitution est effectuée par un filtre *sed* généré dynamiquement par *3iaout*. Les formes disponibles sont précisées pour chaque commande ci-dessous. La partie de texte libre peut être librement adaptée par l'utilisateur pour obtenir le fichier au format désiré. Le fichier modèle n'est jamais altéré par le programme.

La livraison fournit des modèles valides. Si l'on souhaite les personnaliser, il est recommandé soit de les renommer, soit de les copier dans un autre répertoire de travail, et d'indiquer le nom et le chemin du nouveau fichier à prendre en compte sous l'entrée appropriée du fichier de configuration.

Auteur : J. Ducasse, février 2006.

Options.

g	Produit un certain nombre d'informations d'analyse, notamment le fichier de filtrage <i>./3iaout.sed</i> (fichier anonyme et transitoire normalement). Pour la commande <i>interp</i> , produit une sortie plus compacte destinée à être exploitée par programme.
---	---

Exemples.

```
% 3ia mamm.3ia mammalia
% 3iaout compt mammalia.3iz mammalia.txt
  La 1ère commande fait l'analyse 3ia du fichier mamm.3ia et produit le résultat dans le fichier
  mammalia.3iz. La seconde produit à partir de celui-ci deux fichiers au format Component :
  mammalia.txt et mammaliaC.txt.

% 3iaout nexus mammalia.3iz > mammalia.txt
% 3iaout nexus mammalia.3iz mammalia.txt
```

Les deux commandes sont équivalentes. A partir du même fichier produit par 3ia, elles produisent le fichier `mammalia.txt` au format Nexus.

```
% 3iaout interp mamm.3ia mammalia.3iz mammalia.txt
```

Produit dans `mammalia.txt` l'interprétation de l'analyse 3ia, stockée dans `mammalia.3iz`, de l'univers `mamm.3ia`.

Clefs de configuration.

Le programme utilise le fichier de configuration `tts.cfg`. Ce fichier est recherché en priorité dans le répertoire courant ou, à défaut, dans le répertoire référencé par la variable d'environnement `TTS_PKG`. Les entrées suivantes sont utilisées :

<code>SED_PROG</code>	Exécutable <code>sed</code> ou équivalent utilisé pour filtrer les fichiers modèles. Par défaut, <code>SED_PROG="sed"</code> . Le programme sera lancée avec les conventions habituelles : on peut donc soit spécifier son chemin complet, soit donner seulement son nom et laisser le système utiliser la variable d'environnement <code>PATH</code> . Note : sur Windows, si <code>sed</code> (utilitaire Unix) n'est pas disponible, on pourra utiliser l'utilitaire <code>minised</code> livré.
<code>DIR_FILE</code>	Répertoire contenant les filtres <code>sed</code> et les modèles.

Pour la commande `compt` seulement :

<code>COMPT_FILE</code>	Fichier squelette utilisé pour générer les fichiers Component. Par défaut, <code>COMPT_FILE="\$DIR_FILES/compt.mod"</code> .
-------------------------	--

Pour la commande `nexus` seulement :

<code>NEXUS_TAXO</code>	Nom du taxon virtuel (première ligne de la matrice, plein zéro) dans le fichier Nexus. Par défaut, <code>NEXUS_TAXO="Extra_Groupe"</code> .
<code>NEXUS_FILE</code>	Fichier squelette utilisé pour générer le fichier Nexus. Par défaut, <code>NEXUS_FILE="\$DIR_FILES/nexus.mod"</code> .

Particularités de la commande `nexus`.

Le fichier modèle `nexus.mod` (par défaut) est recopié intégralement, à l'exception des forme `@XXX@` qui sont substituées par les valeurs effectives provenant de l'analyse de l'univers courant. Les formes substituées sont les suivantes :

<code>@NTAX@</code>	→ nombre de taxons.
<code>@NTTS@</code>	→ nombre de 3is.
<code>@MATRICE@</code>	→ toute la matrice.
<code>@PF:TTS@</code>	→ toute la liste des PF:3is.

Particularités de la commande `compt`.

Le fichier modèle `compt.mod` (par défaut) est recopié intégralement, à l'exception des forme `@XXX@` qui sont substituées par les valeurs effectives provenant de l'analyse de l'univers courant. Les formes substituées sont les suivantes :

<code>@TITRE@</code>	→ titre de la matrice.
<code>@NTAX@</code>	→ nombre de taxons.
<code>@TAXLABELS@</code>	→ liste des taxons.
<code>@TRANSLATE@</code>	→ liste de correspondance code ↔ nom des taxons.
<code>@TREES@</code>	→ liste des arbres. Noter que les arbres sont toujours produits en notation avec le séparateur virgule, même s'ils figuraient dans le fichier <code>file.3iz</code> en notation sans séparateur.

Le même modèle est utilisé pour les deux fichiers produits.

Particularités de la commande **interp.**

Le traitement concerne chaque couple formé par un caractère *CR* et un arbre de compatibilité *CP* calculé par 3ia, y compris l'arbre d'intersection. Il n'y a aucune interférence entre les couples.

Pour chaque couple *CR/CP* :

- Un statut Accepté/Réfuté est calculé pour chaque noeud non terminal *nCR* de *CR* : voir ci-dessous la façon dont est calculé ce statut.
- Le statut du couple *CR/CP* est alors déterminé :
 - Accepté si le statut est Accepté pour tous les noeuds *nCR*.
 - Réfuté si au moins un noeud *nCR* a le statut Réfuté.

Pour chaque noeud non terminal *nCR* du caractère *CR*, vis à vis de l'arbre *CP*, le statut est calculé ainsi :

- Pour chaque noeud non terminal *nCP* de l'arbre *CP*, le couple *nCR/nCP* a le statut Réfuté si et seulement si :
 - *nCP* possède au moins deux sous-arbres (*nCP* est paralogue) contenant au moins un taxon de *nCR*.
 - et pour au moins un de ces sous-arbres, de racine *R* :
 - *R* ou l'un des descendants de *R* est réfuté.
 - ou bien :
 - *R* est orthologue (1 ou 0 sous-arbre).
 - et au moins un des taxons portés directement par *R* est absent de *nCR*.

Dans les autres cas, le couple *nCR/nCP* a le statut Accepté.

- Le statut de *nCR* est alors :
 - Accepté si le statut des couples *nCR/nCP* est Accepté pour tous les *nCP*.
 - Réfuté si le statut des couples *nCR/nCP* est Réfuté pour au moins un *nCP*.

Pour chaque noeud *nCR*, les données suivantes sont affichées.

- Si *nCR* a le statut Accepté est affichée la racine *R* du sous-arbre de *CP* englobant tous les taxons de *nCR* (éventuellement réduit à un terminal (taxon)). (1)
- Si *nCR* a le statut Réfuté sont affichées deux listes :
 - La liste de tous les *nCP* pour lesquels le couple *nCR/nCP* avait le statut Réfuté. (2)
 - Pour chaque sous-arbre *SS* de chaque noeud *nCP* réfuté (liste ci-dessus), on recherche le plus petit sous-arbre contenant tous les taxons présents dans *SS* ET présents dans *nCR*. Les noeuds affichés sont les racines *R* de ces sous-arbres, éventuellement réduits au taxon *T* lui-même. Cependant, les noeuds *X* réfutés (couple *nCR/X* réfuté), ou dont l'un des descendants est réfuté, sont éliminés. Enfin, sont aussi affichés les taxons *T'* restant, c'est-à-dire les taxons présents dans *nCR* mais absents de tous les sous-arbres ci-dessus. (3)

Exemple :

```
[1] (((((A B) ((E ((F G) H)) I)) D) C) J)
    {1} Accepte : {5}
-> Accepte
```

← affichage (1)
CP = [1] ; *nCR* = {1} ; *R* = {5}

Exemple :

```
[0] (((((A B) ((E (F G H)) I)) D) C) J)
    {1} Refute
    Noeuds refutes : {2}
    {3}
    {6}
-> Refute
```

← liste (2)
← liste (3)
← “
CP = [0] ; *nCR* = {1} ; *R* = {3} {6}

Références bibliographiques.

PAGE, R. D. M. 1993. Component : User's Guide. Natural History Museum, Londres.
id. 2001. TreeView. Natural History Museum, Londres.

Historique.

version 2.2 : décembre 2009

Changement de nom → 3iaout.

version 2.1 : avril 2008

Modification de l'algorithme du mode `interp`. Le traitement « Cependant, les noeuds X réfutés (couple nCR/X réfuté) sont éliminés. » est remplacé par « Cependant, les noeuds X réfutés (couple nCR/X réfuté) où dont l'un des descendants est réfuté sont éliminés. ».

Correction d'un défaut : pour la condition « - R ou l'un des descendants de R est réfuté. », la condition réellement traitée était « - R ou l'un des descendants de R , ou un frère de R ou l'un de ses descendants, est réfuté. » !

version 2 : août 2007

Ajout de la commande `interp`.

version 1.1 : octobre 2006

Dans le format Nexus, si les pondérations fractionnaires lues dans `file.3iz` dépassent la valeur maximale acceptable par Nexus (`PF_MAX`), elles sont automatiquement renormalisées de façon à entrer dans l'intervalle $0 - PF_MAX$. `PF_MAX` égale 32767.

version 1 : février 2006.