

TP VOTable

Cette fiche est également consultable au format HTML sur la page <http://vizier.u-strasbg.fr/~boch/tuto/> . Ouvrez cette page dans un navigateur, elle vous permettra de gagner du temps en copiant/collant les différentes commandes.

L'ensemble des fichiers et outils nécessaires au TP est téléchargeable sous forme d'archive .tar.gz : http://vizier.u-strasbg.fr/~boch/tuto/tp_votable.tar.gz .

Si vous avez déjà installé tous les outils nécessaires au tutoriel, vous pouvez charger les données uniquement : http://vizier.u-strasbg.fr/~boch/tuto/tp_votable_data_only.tar.gz

Créez un nouveau répertoire `VOTable`, chargez et décompressez l'archive en tapant `tar zxvf tp_votable.tar.gz`

Pour toute question relative à ce TP : boch@astro.u-strasbg.fr

Ce TP a pour objectif de vous familiariser avec `VOTable` au travers de l'utilisation de différents outils permettant de lire, créer ou modifier des documents `VOTable`.

Un bref descriptif des différents logiciels utilisés durant ce TP est disponible sur la page <http://vizier.u-strasbg.fr/~boch/tuto/outils.htm>

*Remarque : pour l'ensemble des exercices faisant appel à la programmation, les endroits dans le code source où effectuer les modifications sont signalés par des commentaires **A COMPLETER**.*

1 Conversion de tables ASCII en `VOTable`

Dans cette partie, nous allons voir différentes méthodes permettant de convertir des tables ASCII ou binaires en `VOTable` afin de les utiliser dans des outils VO.

Dans un premier temps, allez sur la page `VizieR` : <http://vizier.u-strasbg.fr/cgi-bin/VizieR>. Entrez `2MASS` dans le premier champ et validez. Choisissez la table `II/246/out`, validez. Entrez comme position `oph s1`, choisissez un rayon de 0.6 degrés et validez. Apparaît alors la liste des 50 premières sources répondant à votre requête. Changez le nombre maximum de réponses à 9999, et choisissez `Tab-Separated-Values` comme format de sortie. Sauvegardez le résultat sous le nom `2mass-ophs1.tsv` par exemple.

Ouvrez le fichier sauvegardé dans un éditeur de texte pour le visualiser.

Remarque : notez que Vizier propose également une sortie VOTable parmi la liste des formats de sortie disponibles.

1.1 Utilisation de conVOT pour convertir notre fichier en VOTable

Lancez conVOT (commande `conVOT` sur les postes préinstallés, commande `java -jar conVOT.jar` ou double-click sur le JAR pour les autres).

Choisissez **Convert ASCII**, et chargez le fichier précédemment sauvegardé.

conVOT essaye de deviner le nombre de lignes à ignorer. Vérifiez, puis validez en cliquant sur **OK**.

conVOT vous demande maintenant quelles sont les lignes décrivant les champs, les unités, et quelle est la première ligne de données. Attention : notre fichier possède une ligne de tirets séparant l'en-tête des données. Ajustez la valeur de "First Data line No", puis validez. L'écran suivant demande confirmation de vos choix.

conVOT vous demande alors de valider son choix de délimiteur entre les colonnes. Vous avez ensuite la possibilité d'éditer les méta données. Nous nous contenterons d'ajouter les UCDs pour quelques colonnes selon la table de correspondance suivante :

Colonne	UCD
._RAJ2000	pos.eq.ra
._DEJ2000	pos.eq.dec
RAJ2000	pos.eq.ra;meta.main
DEJ2000	pos.eq.dec;meta.main
Jmag	phot.mag;em.IR.J
e_Jmag	stat.error;phot.mag;em.IR.J
...	...

Validez, puis validez à nouveau.

Vous pouvez alors visualiser les premières lignes du document VOTable créé. Sauvegardez le (sous le nom `2mass-ohs1.xml` par exemple). Quittez conVOT.

Lancez VOPlot (commande `voplot` ou commande `java -cp voplot.jar com.jvt.applets.PlotApplication` ou double-click sur l'archive JAR), et vérifiez que vous pouvez y charger le fichier nouvellement créé. Quittez VOPlot.

1.2 Conversion d'une table FITS ASCII en VOTable, et visualisation du VOTable généré

Pour cette partie, nous disposons d'une table FITS ASCII issue d'un catalogue de modélisation d'évolution des populations stellaires (sans coordonnées). Il s'agit d'un échantillon de la table globale, seules les étoiles de métallicité 0.0002 et d'âge compris entre 10^8 et 10^9 années ayant été retenues. Le fichier se nomme `evolution_model-sample.fit`.

1.2.1 Utilisation de Topcat

Lancez Topcat (commande `topcat`, ou commande `java -jar topcat-full.jar` ou double click sur le fichier JAR). Sélectionnez **File --> Load New Table** et chargez le fichier en question.

Sélectionnez **Table Views --> Table Data**. Les données sont affichées dans une vue semblable à celle d'un tableur : vous pouvez trier par colonne, modifier leur emplacement, ajouter de nouvelles colonnes, changer la valeur des cellules, etc.

Refermez cette fenêtre. Sélectionnez **File--> Save Table**. Choisissez `votable-tabledata` comme format et sauvegardez sous le nom `evolution_model-sample.xml` par exemple. Quittez topcat

*NB : Topcat supporte un nombre important de formats de tables en entrée comme en sortie. Il inclut un utilitaire de conversion de tables numériques en ligne de commande, **tablecopy**, qui s'avère pratique pour automatiser la conversion de nombreuses tables. Tapez `tablecopy -h` sur les postes pré-installés (`java -classpath topcat-full.jar uk.ac.starlink.table.TableCopy -h` pour les autres) sur la ligne de commande pour afficher les différents formats de sortie supportés.*

*Exemple : sous **bash**, on peut convertir en VOTable tous les fichiers FITS d'un répertoire avec la commande suivante :*

```
for i in *
do
tablecopy -ofmt votable-tabledata $i $i.vot
done
```

1.2.2 Visualisation dans VOPlot

Lancez maintenant VOPlot (commande `voplot` ou `java -cp voplot.jar com.jvt.applets.PlotApplication`). Sélectionnez **File --> Open**, et chargez le fichier VOTable précédemment créé.

Nous allons pouvoir tracer un diagramme HR. Choisissez la colonne `logL` (luminosité) pour ordonnée, et la colonne `logTe` (température) pour abscisse. Appuyez sur **Plot** pour effectuer le tracé. Inversez l'axe des abscisses (case à cocher **Rev**). On distingue la séquence principale d'un diagramme HR.

Vous pouvez également tracer `logL` en fonction de `logt` pour apprécier l'évolution de la luminosité des étoiles selon leur âge (sélectionnez le menu **Mode** puis **Zoom mode** pour effectuer un zoom dans une partie du graphe).

Quittez VOPlot.

1.3 Conversion tables ASCII vers VOTable "à la main"

Nous allons voir rapidement 2 exemples de programme permettant de convertir des fichiers TSV en fichier VOTable.

1.3.1 Perl

Aucune librairie spécifique n'est nécessaire pour cet exemple.

Exécutez `tsv_to_vot.pl` avec comme argument un fichier TSV, par exemple `2mass-ophs1.tsv`. Redirigez **STDOUT** vers le fichier `test.xml`.

Vérifiez la validité du VOTable produit avec `votlint` (commande `votlint test.xml`

sur les postes pré-installés, `java -classpath topcat-full.jar uk.ac.starlink.ttools.Stilts votlint test.xml` pour les autres). Comme vous le constatez, le script produit actuellement un VOTable incorrect.

Ouvrez le script dans votre éditeur de texte, et corrigez le pour qu'il génère un VOTable valide. Relancez le script, puis testez à nouveau la validité du fichier produit avec `votlint`.

NB : La "correction" se trouve dans le fichier `tsv_to_votable-correction.pl`

1.3.2 Java

On utilisera le module "writer" du parser SAVOT pour cet exemple. Il s'agit de construire en mémoire la structure du document VOTable en suivant le modèle interne de SAVOT.

Placez-vous dans le répertoire `TSVToVOTable`. Ouvrez le fichier `cds/tuto/TSVToVOTable.java`

dans un éditeur de texte. Vous pouvez utiliser Eclipse ou votre éditeur de texte favori. Vous pouvez compiler en tapant `ant` (ou `javac -classpath cds.savot.common.jar:cds.savot.writer cds/tuto/TSVToVOTable.java` si vous ne disposez pas de `ant`) à la racine du projet. Pour lancer l'application : `./run.sh INPUT_FILE [OUTPUT_FILE]`. Vous pouvez par exemple utiliser le fichier `hip.tsv` en entrée.

La Javadoc pour le modèle interne de SAVOT et pour la partie "writer" se trouve sur

<http://cdsweb.u-strasbg.fr/cdsdevcorner/savot2.6/doc1/index.html>

Décommentez la ligne signalée par **A COMPLETER** afin de pouvoir prendre en compte les unités. Recompilez, et relancez le programme. Comparez avec le fichier obtenu précédemment.

NB : On pourrait bien entendu rendre ces exemples plus complexes en ajoutant les UCDs par le biais d'une table de correspondance, en acceptant plusieurs délimiteurs de colonnes, etc...

1.3.3 Emballage d'une table FITS binaire dans un document VOTable

Lancez Topcat (`topcat`, ou `java -jar topcat-full.jar -disk`. L'option `-disk` permet d'utiliser le disque dur comme espace de cache pour les données, ce qui permet a priori de lire des fichiers de taille arbitraire).

Chargez le fichier `xmcat.vot` dans Topcat. Puis ouvrez-le dans un éditeur de texte. Vous pouvez remarquer que le fichier VOTable ne contient que la description des métadonnées du catalogue, les données elle-même se trouvant dans un fichier extérieur pointé par une URL.

- **Avantage** : on a une séparation entre données et métadonnées. On peut ainsi fournir un document VOTable sans toucher aux données originales, mais en enrichissant les métadonnées (ajout d'UCDs par exemple).
- **Inconvénient** : peu d'applications supportent cette sérialisation. A l'heure actuelle, la sérialisation TABLEDATA est de loin la plus répandue dans le

monde VO. Heureusement, Topcat permet la conversion entre de nombreux formats tabulaires.

2 Parser des documents VOTable

Nous verrons dans cette partie comment lire des documents VOTable en utilisant un parser.

2.1 Python : utilisation du parser PyVOTable

Pour les personnes travaillant sur leurs propres portables, cette partie nécessite l'installation du module PyVOTable. Chargez l'archive depuis <http://www.eurovo.org/internal/Avo/PyVOTools/pyvotable.tar.gz> Décompressez-la, placez-vous dans le répertoire créé, et tapez la commande (en étant root) : `python setup.py install`

Exécutez le script `print_votable.py` sur le fichier `eis_I_band.xml` : `print_votable.py eis_I_band.xml`

Ce script lit un document VOTable et l'affiche dans un format lisible sur la console. Editez le script et complétez-le pour afficher non seulement le nom de chaque champ (FIELD) et son unité, mais aussi son UCD (utilisez l'attribut `ucd` de l'objet `Field`)

Le script corrigé se nomme `print_votable-correction.py`

NB : PyVOTable ne supporte actuellement que la lecture d'un fichier VOTable. L'écriture d'un document VOTable à partir d'un objet Python est prévu pour un futur proche.

2.2 Utilisation de SAVOT et introduction aux GROUP

Avant-propos : Les GROUP ont été introduits dans VOTable 1.1 afin de regrouper un ensemble de champs (FIELD) reliés logiquement, comme une valeur et son erreur par exemple.

Placez-vous dans le répertoire `PrintVOTable`. Comme précédemment, tapez simplement `ant` ou `javac -classpath cds.savot.common.jar:cds.savot.model.jar:cds.savot.pull.jar:kxml2-min.jar cds/tuto/PrintVOTable.java` pour compiler.

Exécuter le programme : `./run.sh VOTABLE_FILE`. Utilisez `eis_all_tables.xml` comme fichier d'entrée.

Décommentez les quelques lignes signalées par **A COMPLETER** et qui affichent le contenu des GROUP. Recompilez le programme et testez le à nouveau sur le fichier `eis_all_tables.xml`

3 Edition/Modification de documents VOTable

3.1 En utilisant topcat : diagramme couleur-couleur

Pour cet exemple, nous allons reprendre le fichier `2mass-ohs1.xml` produit précédemment (si vous ne l'avez plus, il est disponible dans le répertoire `generated_data`).

Lancez topcat (commande `topcat` sur les postes pré-installés, `java -jar topcat-full.jar` sinon), et ouvrez le fichier `2mass-ohs1.xml`. Allez dans **Table Views --> Columns Info**.

Nous allons ajouter 2 colonnes à ce document afin de pouvoir tracer un diagramme couleur-couleur. Cliquez sur le "+" vert. Entrez un nom pour la nouvelle colonne (J-H par exemple). Puis, entrez dans **Expression** : $J_{mag} - H_{mag}$. Cliquez sur OK. Effectuez la même opération pour ajouter la colonne H-K.

Fermez la fenêtre d'information sur les colonnes, et sélectionnez **Table views --> Table Data**. Déplacez l'ascenseur horizontal pour apercevoir les nouvelles colonnes créées.

Sauvegardez le fichier avec les 2 colonnes supplémentaires (sous le nom `2mass-ohs1_new_cols.xml` par exemple).

NB : VOPlot permet également d'effectuer ce genre d'opération sur les colonnes. Un des avantages de topcat est qu'il supporte l'ensemble des fonctions de `java.lang.Math`

3.2 En utilisant VOPlot: diagramme couleur-couleur

Lancez Aladin (commande `Aladin` ou `java -classpath Aladin.jar:voplot.jar cds.aladin.Aladin` ou double-click), et chargez-y le fichier contenant les nouvelles colonnes (**LOAD --> File**). Sélectionnez un ensemble de sources, et notez dans la fenêtre des mesures (en bas) la présence des méta-données (unités, UCD).

Interrogez le serveur d'images Aladin (**LOAD--> Aladin**) et sélectionnez l'image SERC de taille 1.7'x1.7'.

Lancez VOPlot (depuis le menu **Tools**).

Dans VOPlot, tracez J-H en fonction de H-K. Remarquez que les sources sont relativement dispersées. On va choisir un sous-ensemble en conservant uniquement les sources dont l'erreur sur la magnitude est petite. Sélectionnez le bouton **New Filter**, entrez $eK < 0.1$ comme description du filtre puis entrez $\$9 < 0.1$ comme expression. Validez (OK).

Sélectionnez le filtre nouvellement créé, et cliquez sur **Plot**. Cochez la case à cocher **Overlay**, sélectionnez le filtre **All** et validez à nouveau.

Affichez maintenant uniquement le filtre $eK < 0.1$, et sélectionnez les objets présentant un excès en K. Les objets correspondant sont également sélectionnés dans la fenêtre Aladin : ce sont des objets jeunes qui se trouvent dans la zone d'absorption.

3.3 En Java : ajout d'une colonne flux à partir de la magnitude

La conversion de flux en magnitude est donnée par la formule suivante :

$$F = 10^{0.4*(Z_p - M)}$$

où F est le flux en Jy, Z_p le point-zéro du système photométrique utilisé, et M la magnitude.

Nous allons construire un petit programme qui va modifier un document VOTable pour ajouter une colonne **flux** pour une magnitude donnée.

Pour notre exercice, nous utiliserons le fichier `eis_I_band.xml`, extrait de la table *bande I* du catalogue EIS (ESO Imaging Survey). Pour cette bande, le valeur du

point-zéro est de 8.4577. Le nom du champ magnitude est **Mtot**

En utilisant SAVOT, le mécanisme de modification d'un document VOTable est le suivant :

- chargement du document en mémoire dans un objet SavotVOTable
- modification de cet objet
- sérialisation de l'objet dans un flux de sortie

Placez-vous dans le répertoire FluxConversion. Compilez en lançant **ant**. Puis exécutez le programme en tapant **./run.sh eis_I_band.xml eis_with_flux.xml** Ouvrez le fichier **eis_with_flux.xml** dans un éditeur de texte et vérifiez la présence de la nouvelle colonne.