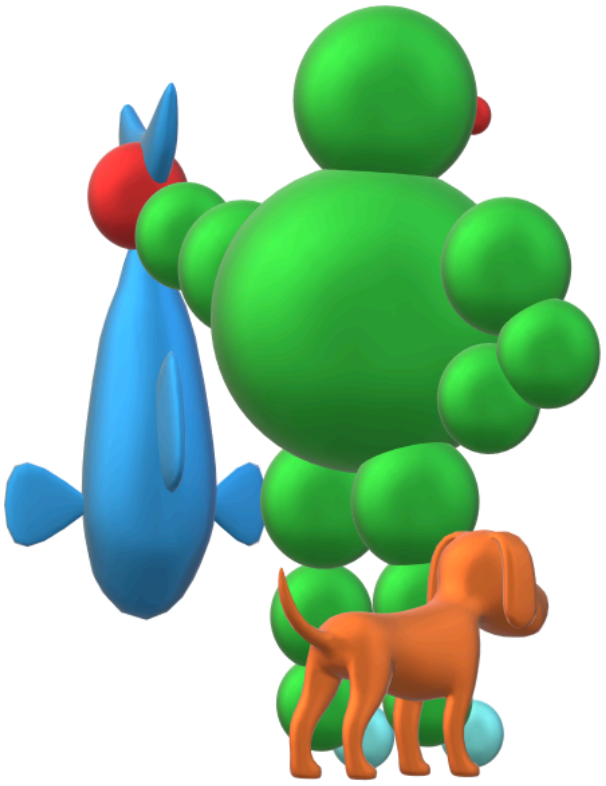


Dose Hunter
Guide d'utilisation

Oncopole Claudius Regaud, France

Créé par : **Luc SIMON** (simon.luc (at) iuct-oncopole.fr)

Contributeurs : **Luc SIMON, Bradley BEEKSMA, François-Xavier ARNAUD, Killian LACAZE, Farzam SAYAH**

Résumé

Ce document décrit comment utiliser le script DoseHunter, un outil pour automatiser la collecte de valeurs de HDV pour un grand nombre de patients dans la base de données d'Eclipse.

1. Qu’est ce que DoseHunter ?

DoseHunter est un script stand alone, c'est à dire qu'il ne nécessite pas d'ouvrir Eclipse et peut être exécuté seul par un simple “double clic”. Il permet de collecter automatiquement des index de doses (dose max, dose min, D95%, etc.) pour un grand nombre de patients. Les données récoltées sont stockées dans un fichier .csv (exploitable avec Excel, Python, etc.). **DoseHunter** est un outil qui peut être utilisé pour la recherche, des études ponctuelles ou des essais cliniques.

2. Démarrer

Vous devez être sur une station Eclipse (pas Citrix) et il faut absolument ouvrir une session windows avec un identifiant qui existe dans ARIA (ne pas utiliser les sessions Windows type Eclipse23).

Dans un dossier local de la station (sur le bureau ou sur le C:/), copier le dossier:
\\srv015\radiotherapie\SCRIPTS_ECLIPSE\dose_hunter

Merci de ne travailler qu'avec une copie de ce dossier et de ne pas le modifier.

Pour lancer DoseHunter, il suffit de double-cliquer sur le fichier DoseHunter.exe en présence des 5 fichiers suivants.
Deux fichiers .dll fournis par Varian:

- VMS.TPS.Common.Model.Types.dll
- VMS.TPS.Common.Model.API.dll

et trois fichiers texte contenant les choix de l'utilisateur:

- **id.txt** contient la liste des patients (voir section 2.1)
- **index.txt** contient la liste des index que l'on souhaite collecter (voir section 2.2)
- **planfilter.txt** contient des indications pour que DoseHunter filtre les plans des patients pour, par exemple, exclure certains plans de l'analyse (voir section 2.3)

Lorsque DoseHunter est lancé, la console Windows affiche des informations sur le déroulement de l'exécution du programme. A la fin, la console invite l'utilisateur à taper ENTER pour terminer. Les données collectées sont sauvegardées dans le dossier **out/** (voir section 3).

2.1. id.txt

id.txt est un simple fichier texte (éditable avec le bloc notes de Windows par exemple) qui contient la liste des patients que l'utilisateur souhaite étudier. Il faut simplement écrire les ID des patients, un par ligne, sans espace. Cela peut être réalisé par un copier/coller a partir d'une colonne d'un fichier Excel. Un exemple est fourni dans le dossier de travail.

2.2. index.txt

index.txt est un autre fichier texte qui indique quelles données l'utilisateur souhaite collecter pour chaque patient indiqué dans la liste de patients (voir section 2.1). Chaque ligne de index.txt doit avoir ce format general:
<struct. name 1>;<struct name 2 (opt)>;<struct. name 3 (opt)>,<index>,<index>,<index> (etc.)
Par exemple, le fichier peut contenir:

```
Heart;HEART;heart,max,min,median,D95%
PTV,max,D95cc,median
Liver;liver,max,V50cc,vol
```

Attention, veuillez noter la différence entre :

, et **;**

Chaque ligne est composée d'éléments séparés par ','. Le premier élément de la ligne indique le nom de la structure recherchée. Il est possible d'indiquer plusieurs orthographes de la structure en les séparant par des ';'.
Ici par exemple, le coeur est recherché avec les orthographes suivantes :

Heart, HEART ou theHeart.

Si une structure est trouvée avec la première orthographe (ici Heart) les autres sont ignorées. Sinon **DoseHunter** cherchera une structure dont le nom est la deuxième orthographe, puis la troisième…

DoseHunter est insensible à la casse et dans cet exemple les deux premières orthographes sont redondantes.

Les autres éléments de la ligne (après la première virgule) sont les index que l'utilisateur souhaite collecter pour la structure. Les index possibles sont :

- **vol** : volume de la structure (Vol et VOL sont tolérés)
- **min** : dose minimum de la structure (Gy) (Min, MIN tolérés)
- **max** : dose maximum de la structure (Gy) (Max, MAX tolérés)
- **mean** : dose moyenne de la structure (Gy) (Mean, MEAN tolérés)
- **median** : dose médiane de la structure (Gy) (Median, MEDIAN tolérés)
- **DXX%** or **DXXcc** : e.g. D95% ou D2.5cc : Dose (Gy) reçue par 95% ou 2.55 cc de la structure
- **VXX%** or **VXXcc** : e.g V49.6% ou V49.6cc : Volume en % ou en cc qui reçoit 49.6 Gy
- • HI or hi : Homogeneity Index in the structure:

$$HI = \frac{D2-D98}{D50}$$

where D2, D50 and D98 are the doses received by 2%, 50% and 98% of the chosen structure, respectively.

- **GI** : l'index de gradient (ne dépend pas de la structure) ; Volume du BODY qui reçoit 50% / Volume du Body qui reçoit 100% de la TotalDose du plan
- **CIxx**: e.g. CI95 : Conformity Index pour l'isodose 95%

CI95 = PIV / TV où PIV est le volume de l'isodose 95% et TV le volume de la structure.

- **PIxx** : e.g. PI95 : Paddick Conformity Index for the isodose 95%

$$PI95 = \frac{TV_{PIV}^2}{TV \times PIV}$$

where TV_{PIV} est le volume de la structure qui reçoit 95% de la dose totale, PIV est le volume de l'isodose 95% et TV le volume de la structure.

Notez que les doses sont absolues (pas de doses relatives).

2.3. planfilter.txt

Certains patients contiennent un grand nombre de plans et il peut être pratique de filtrer les plans pour lesquels l'utilisateur souhaite collecter des index. Pour cela il faut utiliser le fichier planIter.txt.
Si le fichier est absent, des valeurs de filtres par défaut seront utilisées.
Attention ! Ne pas modifier le fichier à part les valeurs indiquées à droite des ':'.
Exemple de fichiers :

```
# HOW TO FILTER YOUR PLAN
# DO NO MODIFY THIS FILE EXCEPT THE PART AFTER THE ":"
# keep only the plans with a total dose > to a value
Min Total Dose (Gy):60.0
# keep only the plans with a total dose < to a value
Max Total Dose (Gy):200.0
# Treat approved? If "no" treat approved plans will be excluded.
TreatApproved plan:no
# Planning approved? If "no" planning approved plans will be excluded.
PlanningApproved plan:no
# Unapproved? If "no" unapproved plans will be excluded.
Unapproved plan:yes
# Named plans? If "no", plans with a name will be excluded.
Named plan:yes
# Unnamed plans? If "no", plans with no name will be excluded.
Unnamed plan:yes
# keep it if it contains a string? If "yes", plans will be kept only
if they contain the string
Plan name must contain a string:no:toto
```

```
# Exclude it if it contains a string? If "yes", plans will refused
if they contain the string
Exclude if plan name contains:no:toto
# Explore Sum plans ? if no, sum plans are ignored, yes to explore them
Explore Sumplans:yes
```

Dans cet exemple, l'utilisateur souhaite garder les plans suivants :
• Total dose entre 60 et 200 Gy
• exclusion si le statut du plan est "Treat Approve"
• exclusion si le statut du plan est "Planning Approve"
• garder si le statut du plan est "Unapproved"
• garder si le plan a un nom (tous les plans ont un ID mais pas forcément un nom)
• garder si le plan n'a pas de nom (ici l'utilisateur souhaite garder tous les plans qu'ils aient un nom ou pas)
• Pas de filtre sur l'ID du plan ici. L'utilisateur pourrait vouloir par exemple exclure tous les plans pour lesquels l'ID contient DTO ou dto.
La dernière ligne du fichier serait :

```
Exclude if plan name contains:yes:DTO:dto
```

ou à l'inverse les conserver. Il faudrait alors utiliser la ligne:

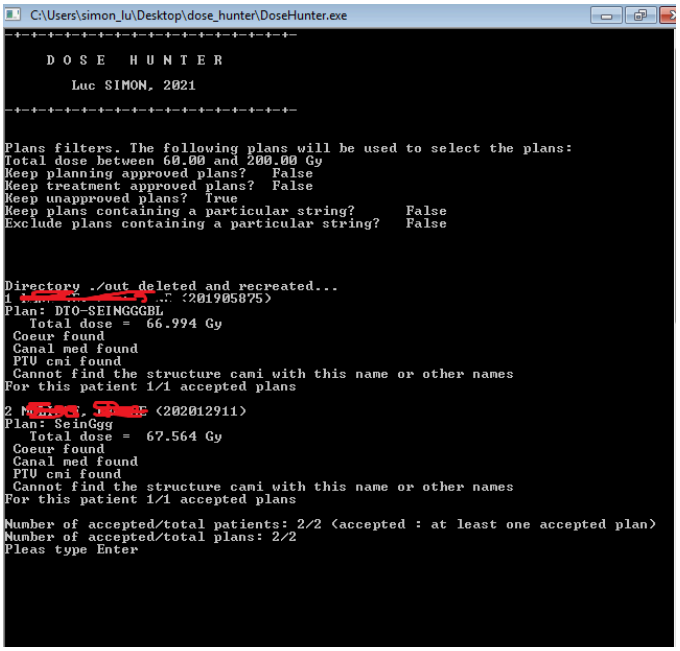
```
Plan name must contain a string:yes:DTO:dto
```

• les plans somme ne sont pas ignorés. A noter si cet option est activée, seuls quelques filtres seront utilisés pour les plans somme : les tests si le plan somme a un nom et les chaînes de caractère qu'il doit contenir pour être conservés ou exclus.

Les chaînes doivent être séparées par des ':'
D'autres filtres peuvent être développés sur demande.

2.4. Exécution de DoseHunter

Lorsque l'utilisateur double clique sur DoseHunter.exe, la liste de patients est lue (voir la description de id.txt en section 2.1). Pour chaque patient, chaque course est ouvert, et pour chaque course, chaque plan est testé (voir les filtres section 2.3). Si le plan passe la filtration, les index (voir section 2.2) sont récoltés et sauvegardés dans le fichier data.csv (voir section 3). Nul besoin d'ouvrir Eclipse pour l'exécution. Durant l'exécution, la console Windows, affiche des informations (nom du patient, nom du plan, nom des structures recherchées...) Si une structure n'est pas trouvée le programme continue. Voir ci-dessous un exemple de la console a la fin de l'exécution.



3. Données de sortie

Un dossier output/ est créé à l'exécution. Attention il est effacé et recrée à chaque exécution de DoseHunter. A la fin, l'utilisateur est invité à taper ENTER. Deux fichiers sont créés dans le dossier output/:

• **log.txt** contient plus ou moins les messages que la console a affichés pendant l'exécution.

• **data.csv** contient les données collectées. Il peut être ouvert avec Excel ou un script Python. Attention dans Excel il peut être nécessaire de remplacer tous les points par des virgules Chaque ligne contient les valeurs pour les plans étudiés (un par ligne) et il peut y avoir plusieurs lignes pour un patient. Le format est le suivant :

```
<patient ID>;<course ID>;<plan ID>;<plan date>;<author of the plan>;
<total dose>;<dose per fraction>;<number of fractions>;<MU>;<MI (MU/fraction)>;
<plan normalisation value>;<index>;<index>;<index>...
```

La première ligne contient le titre des colonnes. Les autres lignes contiennent les valeurs séparées par des ';'. Si une valeur n'a pas été récoltée (Dose Hunter n'a pas pu trouver la structure) la cellule est vide.