



TP PlayMOBY

WS: FastaCmd

2011/04/15

Sebastien.Carrere@toulouse.inra.fr

BUT

Deployer un Web-Service BioMOBY qui à partir d'un fichier tabulé permet d'exécuter fastacmd pour extraire des séquences de la banque Swiss-Prot ou Enzyme et retourner le résultat au format multi-fasta. Un parametre permettra de controler le numéro de la colonne du le fichier tabulé contenant les numero d'accession (première colonne : indice 1).

COMMENT

1. Ecrire un programme perl (*wrapper*) qui remplit cette tache en ligne de commande
2. Decrire ce programme en utilisant le module **Appli.pm**
3. Deployer le WS en utilisant l'environnement **PlayMOBY**

PLAN

PREPARATION DE L 'ENVIRONNEMENT

ECRITURE DU WRAPPER PERL

ENTREES/SORTIES/PARAMETRES

CORPS DU SCRIPT

BEGIN

MAIN

INTERFACE

METIER

TEST DU SCRIPT CLI

DESCRIPTION DU PROGRAMME - APPLI.PM

DEPLOIEMENT DU WEBSERVICE

PREPARATION

```
$USER = tp_pmb_fastacmd
```

1. Mise en place du tunnel ssh pour se connecter sur le serveur de déploiement
% ssh -N -f \$USER@snp.toulouse.inra.fr -L2222:mobyle-dev:22
2. Connexion
% ssh -X -Y -p 2222 \$USER@localhost
3. Preparation du repertoire de travail
% cd ~
% mkdir -p bin/int bin/ext
% cd bin/ext/
% wget http://lipm-bioinfo.toulouse.inra.fr/formations/biomoby/data/fastacmd_bin.tgz
% tar xfz fastacmd_bin.tgz
4. Installation de PlayMOBY
% cd ~
% wget <http://lipm-bioinfo.toulouse.inra.fr/biomoby/playmoby.tar.bz2>
% tar xjf playmoby.tar.bz2
% cd playmoby
% setenv PMBHOME \$PWD
% ./pmb_configure.pl

```
--http_path: YOUR PlayMOBY URL; same path but accessible via apache (try it with a browser)
--auth_uri : your signature URI
--email    : this PlayMoby instance administrator email
--registry : default BioMOBY registry [mobycentral]
              available values :
                *mobycentral = Public online registry
                *inra = INRA network restricted BioMOBY registry
                *opcentral = This is the official test registry

--timeout  : timeout SOAP request (default: 10 minutes)
```

→ http_path = mettre [http://mobyle-dev.toulouse.inra.fr/\\$USER/playmoby](http://mobyle-dev.toulouse.inra.fr/$USER/playmoby)

Cela depend de la configuration de votre serveur Apache



→ auth_uri = mettre test.\$USER.toulouse.inra.fr

Permet de signer vos WS : un WS dans un annuaire BioMOBY est identifié par son Nom + AuthURI, ce couple étant unique)

→ registry = choisir inra

Ce sera l'annuaire utilisé par défaut pour enregistrer et tester les WS. Par la suite,

NB: PlayMOBY vous offrira la possibilité d'enregistrer vos services sur d'autres annuaires (par exemple mobycentral qui est l'annuaire public situé au Canada. Il faut cependant s'assurer auparavant que les objets utilisés en entrée et sortie de votre WS existent bien dans cet l'annuaire de destination. Pour ce faire, cherchez les objets en question dans le navigateur d'annuaire (<http://lipm-bioinfo.toulouse.inra.fr/registry>) en sélectionnant l'annuaire de destination.

→ timeout : laisser par défaut

Il s'agit du timeout pour les WS exécutés en synchrone. Ce parametre contrôle le timeout de LWP::UserAgent, mais pas celui de Apache; il faut donc synchroniser le timeout apache en sus (httpd.conf).

NB: je vous conseille d'utiliser des services asynchrones pour les services demandant plus d'1 minute de calcul. Dans ce cas, le timeout n'a pas d'importance.

Pour vérifier que l'installation est correcte (et que le serveur apache est bien configuré (ScriptAlias) cliquez sur [http://mobyle-dev.toulouse.inra.fr/\\$USER/playmoby/cgi/index.cgi](http://mobyle-dev.toulouse.inra.fr/$USER/playmoby/cgi/index.cgi)

ECRITURE DU WRAPPER PERL

Nom du script: choisissez un nom de script « intelligent »

Ecrire le script dans ~/bin/int/

PRE-REQUIS:



- il doit manipuler des fichiers (contrainte PlayMOBY) en entree ET sortie
- tout ce qui est écrit sur STDOUT sera perdu
- tout ce qui est écrit sur STDERR sera capturé comme une exception/erreur

ENTREES/SORTIES/PARAMETRES

```
#
#   MANDATORY
#
Input:
  -i <input file>
Outputs:
  -o <multi fasta file>


#
#   fastacmd PARAMETERS
#
  -c    column index N > 0 [1]
  -l    line length for sequence  N > 0 [80]
  -d    (sp,enzyme)      [sp]

#
#   PLAYMOBY RELATED
#
--mobyte
--mobyte_dir=<Mobyte xml file target directory>
--pmbtest
```

NB:

1. on se limite a ces quelques parametres

CORPS DU SCRIPT

 Nous nous efforcerons de respecter les normes de code disponibles sur le site du CATI BIPAS: <http://cati-bipas.toulouse.inra.fr/doku.php/normesprogrammation/normesperl>

Ecriture du bloc BEGIN pour ajouter les répertoires de bibliothèques à @INC

Utilisation du module FindBin pour reconstruire les chemins vers \$PMBHOME/lib contenant les bibliothèques utilisés en routine (ParamParser, Appli)

MAIN

INTERFACE

1. création de l'objet ParamParser (source = GetOptLong)
`New ParamParser('GETOPTLONG', 'param1=type', 'param2=type', ...);`
2. définition des chemins d'accès aux programmes
définition des variables pour accéder aux binaires
3. Récupération des paramètres
`ParamParser::Get`
Contrôles `ParamParser::Assert*`
4. Positionnement des valeurs par défaut
`ParamParser::SetUnlessDefined`

METIER



5. construction de la ligne de commande blast
Attention à bien rediriger les warnings s'ils sont écrits sur STDERR
6. exécution système de blast

TEST DU SCRIPT CLI

Input = \$PMBHOME/sample/test.m8

Les banques sont déjà formatées et accessibles via :

/db/generic/sp

/db/generic/enzyme

Doc: <http://lipm-bioinfo.toulouse.inra.fr/biomoby/playmoby/doc/Appli.html>

Choix des bons types BioMOBY: <http://lipm-bioinfo.toulouse.inra.fr/registry>

Choix des types Moby automatisé : dictionnaire + OntologyMap.pm

<http://lipm-bioinfo.toulouse.inra.fr/biomoby/playmoby/data/BioMobyMobyDictionary.xml>

1. Description Generale du programme (%h_appli_general)



BIOMOBY → Type de Service: catégorie du service (Analysis, Retrieval, Sequence_Assembly)

2. Description des entrées (%h_appli_inputs)



BIOMOBY → Type de données : le format de mes données d'entrée; il faut être le plus juste possible (le plus précis : FASTA_AA vs. FASTA) dans l'optique « workflow ». Quand plusieurs types sont possibles :

– priorité à celui présent dans le dictionnaire BioMOBY ↔ Moby

<http://lipm-bioinfo.toulouse.inra.fr/biomoby/playmoby/data/BioMobyMobyDictionary.xml>

– choix de celui garantissant le maximum d'interopérabilité (nombre de services utilisant ce type, cf. score dans la fiche du type sur <http://lipm-bioinfo.toulouse.inra.fr/registry>)

3. Description des sorties (%h_appli_outputs)



BIOMOBY → Type de données : le format de mes données de sortie; il faut être le plus juste possible (le plus précis : FASTA_AA vs. FASTA) dans l'optique « workflow »

4. Description des paramètres (%h_appli_parameters)



*BIOMOBY → Type de données: format bas niveau [String, Integer, Float, DateTime, Boolean]
→ Min, Max, Default, enum*

Ici aussi il faut veiller à être précis afin que le programme soit fonctionnel et cohérent même utilisé avec les valeurs par défaut

NB: Clé **cmd** des tableaux de hash

- Pour chacun des paramètres: mapping entre le paramètre du WS et la commande du CLI; on peut très bien donner un nom plus verbeux pour le paramètre du WS.

exemple: **line_length** pour **-l** de fastacmd; à ce moment là on aura

```
$h_appli_parameters{line_length} = {cmd => '-l $value'}
```

- \$value est un mot réservé: il est substitué par la valeur du paramètre

5. L'écriture du fichier MOBY-XML sera déclenchée par le paramètre --moby du CLI

DEPLOIEMENT DU WEBSERVICE

<http://lipm-bioinfo.toulouse.inra.fr/biomoby/playmoby>

1. Préparer un fichier de test pour l'entree
2. se placer dans l'environnement PlayMOBY configuré (URI, annuaire par défaut)

source .setenv

3. Construction de l'archive du WS à partir de la description Moby

`$PMBBIN/pmb_MobyleParser.pl --xmlfile <Mobyle-xml>`



→ synchrone/asynchrone : Mon prgramme peut-t-il mettre plus de temps que le timeout ?
→ attachement des donnees de test : 1 par entrée (fichier préparé en point 1)

4. Deploiement (enregistrement) et test

`$PMBBIN/pmb_DeployWS.pl`



Pensez à :

- soit définir des paramètres par défaut qui valideront le test
- soit à placer un fichier de parametres dans le répertoire de test.

5. Mettre en place un test fonctionnel stable (--pmbtest)