TP PlayMOBY
2009/26/11

Sebastien.Carrere@toulouse.inra.fr
Erika.Sallet@toulouse.inra.fr

BUT

Deployer un Web-Service BioMOBY qui à partir d'un numero d'accession d'un Cluster BIOS, recupere les lectures membres de ce cluster et fait un assemblage CAP3 de ces lectures.

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 DANS ENVIRONNEMENT BIOS

1. Installation de CAP3

   % cd /www-bios/bin/ext
   % wget http://seq.cs.iastate.edu/CAP3/cap3.linux.i686_xeon64.tar
   % tar xf cap3.linux.i686_xeon64.tar

2. Ajout de CAP3 à la configuration de BIOS

   % vi /www-bios/site/cfg/bios.cfg
   Ajouter la ligne prg_cap3=%i/bin/ext/CAP3/cap3

PREPARATION HORS BIOS

$USER = (djacob,vbrunaud,saubourg,ebiot,pbaa)

1. ssh $USER@bios-dev.toulouse.inra.fr (147.99.102.9)

2. Preparation du repertoire de travail
   % cd ~
   % mkdir -p bin/int bin/ext
   % cd bin/ext/
   % wget http://seq.cs.iastate.edu/CAP3/cap3.linux.i686_xeon64.tar
   % tar xf cap3.linux.i686_xeon64.tar
   % ln -s /www-bios/bin/int/bios_getsequences.pl .

3. Installation de PlayMOBY
   % cd ~
   % wget http://lipm-bioinfo.toulouse.inra.fr/biomoby/playmoby.tar.bz2
   % tar xjf playmoby.tar.bz2
   % cd playmoby
   % ./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
   *opencentral = This is the official test registry

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

   → http_path = mettre http://bios-dev.toulouse.inra.fr/formation/$USER/playmoby
   Cela depend de la configuration de votre serveur Apache

   → auth_uri = mettre **bios.$USER.toulouse.inra.fr**
   Permet de signer vos WS (NomWS + AuthURI)

   → registry = choisir **inra**
   Ce sera l'annuaire utilisé par défaut pour enregistrer et tester les WS

   → timeout : laisser par defaut
   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)

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

ECRITURE DU WRAPPER PERL

Nom du script: comme vous voulez, mais si c'est un programme « BIOS » voué à être partagé, nous avons choisi la convention prefixe = 'bios_'

[BIOS] écrire le script dans ~/bin/int/
[HORSBIOS] écrire le script dans ~/bin

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 ecrit sur STDERR sera capturé comme une exception/erreur

ENTREES/SORTIES/PARAMETRES

```
#
#           MANDATORY
#

 Input:
   --accessions=<accession, list,file of accessions> ex:CLUSTERTEST001 ou CL20Contig1

Outputs:
   --ace_outfile=<CAP3 ACE output file>
   --cap3_outfile=<CAP3 standard output file>

#
#           OPTIONNAL
#

   --login=s
   --password=s

#
#           CAP3 PARAMETERS
#
            -a      specify band expansion size N > 10      [20]
            -b      specify base quality cutoff for differences N > 15  [20]
            -c      specify base quality cutoff for clipping N > 5      [12]
            -d      specify max qscore sum at differences N > 20        [200]
            -e      specify clearance between no. of diff N > 10        [30]
            -f      specify max gap length in any overlap N > 1         [20]
            -g      specify gap penalty factor N > 0        [6]
            -h      specify max overhang percent length N > 2           [20]
            -i      specify segment pair score cutoff N > 20            [40]
            -j      specify chain score cutoff N > 30       [80]
            -k      specify end clipping flag N >= 0        [1]
            -m      specify match score factor N > 0        [2]
            -n      specify mismatch score factor N < 0     [-5]
            -o      specify overlap length cutoff > 15      [40]
            -p      specify overlap percent identity cutoff N > 65  [90]
            -r      specify reverse orientation value N >= 0[1]
            -s      specify overlap similarity score cutoff N > 250     [900]
            -t      specify max number of word matches N > 30       [300]
            -u      specify min number of constraints for correction N > 0      [3]
            -v      specify min number of constraints for linking N > 0         [2]
            -y      specify clipping range N > 5    [100]
            -z      specify min no. of good reads at clip pos N > 0     [3]

#
#           PLAYMOBY RELATED
#

   --mobyle
   --mobyle_dir=<Mobyle xml file target directory>
   --pmbtest
   --verbose
```

NB:

1. Pour le TP, on laisse les autres fichiers produits par CAP3 (singlets, info, contigs, qual)
2. On ne decrira que le parametre « -p » de CAP3 pour le TP (je vous donnerai les lignes de codes pour les suivants)

CORPS DU SCRIPT

Les programmes développés dans le cadre de BIOS étant voués a être partagés par les acteurs du réseau, 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 repertoires de librairies à @INC

[BIOS]: utilisation de Cwd et File::Basename pour reconstruire les chemins vers lib/ et playmoby/lib; positionnement de la variable d'environnement BIOS
[HORSBIOS] : /www-bios/lib & /www-bios/playmoby/lib

MAIN

INTERFACE

1. creation de l'objet ParamParser (source = GetOptLong)
   New ParamParser('GETOPTLONG','param1=type','param2=type', ...);

2. écriture de l'usage CLI
   ParamParser::SetUsage

3. définition des chemins d'accès aux programmes
   [BIOS] chargement de la configuration BIOS (fonction &BIOSLoadCfg de BIOS/Utils.pl) et utilisation de ParamParser
   [HORSBIOS] définition des variables pour accéder aux binaires

4. Récuperation des parametres
   ParamParser::Get
   Contrôles ParamParser::Assert*

5. Positionnement des valeurs par defaut
   ParamParser::SetUnlessDefined

METIER

6. creation de l'espace de travail temporaire
   → CAP3 ecrit les fichiers resultats la ou se trouvent les sequences

7. recuperation des sequences du cluster
   bin/int/bios_getsequences.pl --cluster_accession
   → ce programme peut déjà manipuler des accessions, listes et fichiers

8. execution systeme de CAP3
   → reconstruction de la ligne de commande

9. copie des fichiers vers les fichiers destinations ace_outfile et cap3_outfile
   File::Copy::copy

10. nettoyage
    File::Path::rmtree

TEST DU SCRIPT CLI

Input =
1. accession=CL20Contig1
2. accession=CL20Contig1,CL21Contig1
3. accession=liste.txt (contenant CL20Contig1)

DESCRIPTION DU PROGRAMME – APPLI.PM

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 Mobyle automatique : dictionnaire + OntologyMap.pm
http://lipm-bioinfo.toulouse.inra.fr/biomoby/playmoby/data/BioMobyMobyleDictionnary.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 »

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 parametres (%h_appli_parameters)

BIOMOBY → Type de données: format bas niveau [String, Integer, Float, DateTime, Boolean]
→ Min, Max, Default, enum
Ici aussi il faut veiller a 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 parametres: mapping entre le parametre du WS et la commande du CLI; on peut très bien donner un nom plus verbeux pour le parametre du WS.
exemple: **percent_identity** pour **-p** de cap3; a ce moment la on aura
$h_appli_parameters{**percent_identity**} = {cmd => '-p $value'}

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

5. L'ecriture du fichier MOBYLE-XML sera déclenchée par le paramètre --mobyle du CLI

DEPLOIEMENT DU WEBSERVICE

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

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

   source .setenv
3. Construction de l'archive du WS à partir de la description Mobyle

   $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