Pipeline Compendium

marion.verdenaud@gmail.com
jerome.gouzy@toulouse.inra.fr
sebastien.carrere@toulouse.inra.fr

Table des matières

I) Principe

1. Introduction

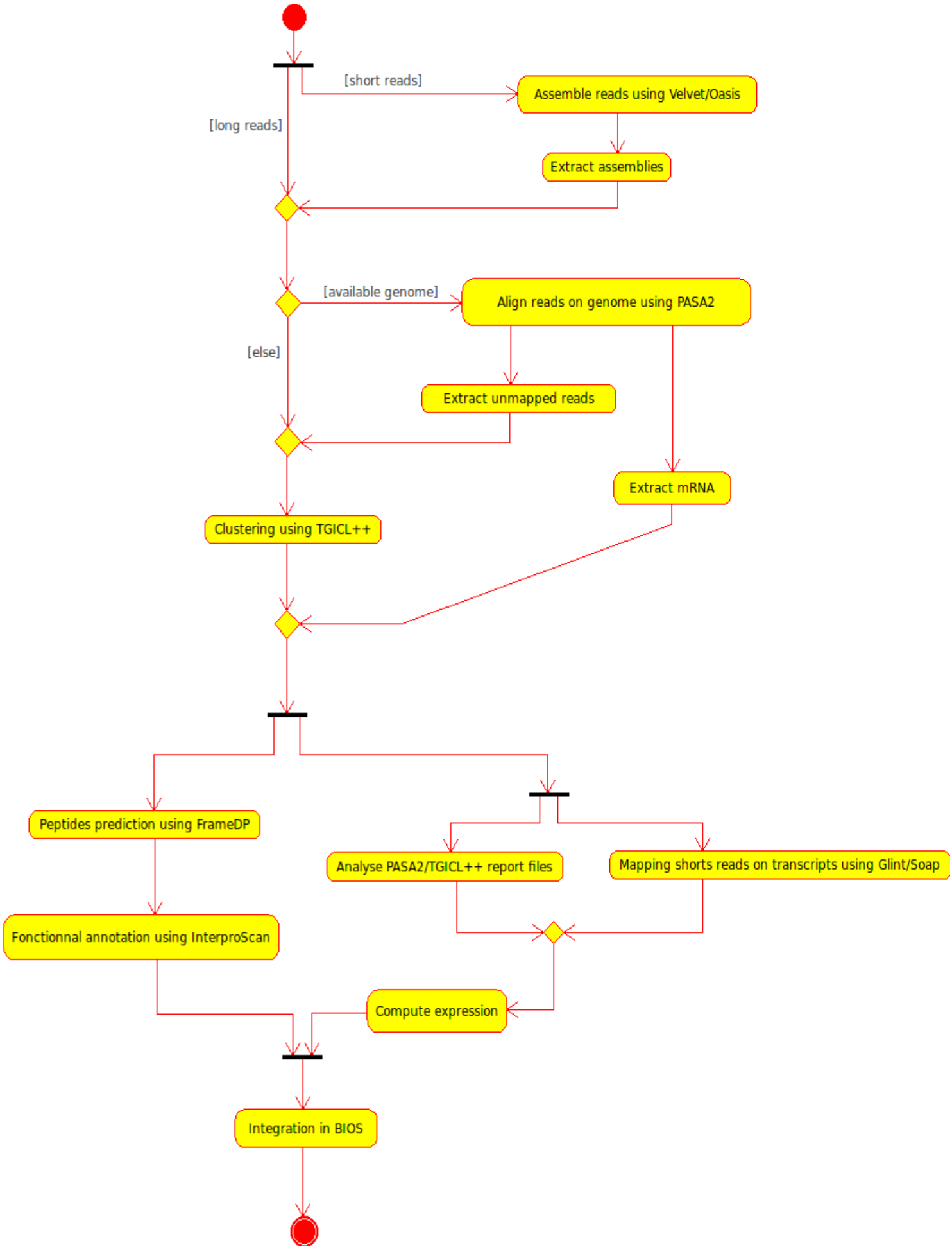
L'enjeu du pipeline compendium est de proposer une méthode qui permet de détecter les transcrits à partir d'un ensemble d'ESTs et d'un génome de référence s'il est disponible. Les clusters d'ESTs ainsi générés pourront ensuite être testés via la prédiction de peptides.

2. Description

Le pipeline se décompose en plusieurs étapes :

1) On dispose de reads Solexa, on va les assembler avec Velvet (Daniel R. Zerbino, Ewan Birney (2008) Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res* **18**:821-829) pour en faire des contigs, puis avec Oases (http://www.ebi.ac.uk/~zerbino/oases/) pour assembler ces contigs en transcrits.
2) Si l'on dispose d'un génome de référence, on va assembler des transcrits sur la base du mapping des ETS sur le génome. Pour se faire, nous allons nous servir de l'outil PASA, (Haas, B.J., Delcher, A.L., Mount, S.M., Wortman, J.R., Smith Jr, R.K., Jr., Hannick, L.I., Maiti, R., Ronning, C.M., Rusch, D.B., Town, C.D. et al. (2003) Improving the Arabidopsis genome annotation using maximal transcript alignment assemblies. *Nucleic Acids Res.* **31**(19):5654-66.)
3) Ensuite on va récupérer les ESTs non alignées sur le génome pour essayer de les clusteriser. Si bien sûr il n'y a pas de génome de référence disponible, on peut directement effectuer cette étape. Pour cela nous avons modifié le pipeline de clusterisation TGICL du TIGR. (Geo Pertea, Xiaoqiu Huang, Feng Liang,Valentin Antonescu ,Razvan Sultana, Svetlana Karamycheva, Yuandan Lee, Joseph White, Foo Cheung, Babak Parvizi, Jennifer Tsai and John Quackenbush (2003) TIGR Gene Indices clustering tools (TGICL): a software system for fast clustering of large EST datasets. *Bioinformatics*. 19(5):651-2.)
   Ce pipeline modifié est TGICL++ et propose un clustering itératif.
4) Si on a utilisé des reads Solexa, on va les aligner sur les transcrits obtenus (étape PASA + TGICL) pour avoir une idée de l'expression. Pour cela on utilise SOAP (Li et al. (2008) « SOAP: short oligonucleotide alignment program" . BIOINFORMATICS, 24 no.5,713–714, doi:10.1093/bioinformatics/btn025)
5) Enfin pour finir, on peut sur les clusters obtenus précédemment appliquer FRAMEDP qui va tenter d'en prédire les peptides. Cette étape est essentielle pour valider les clusters. (Gouzy J, Carrere S, Schiex T. (2009) FrameDP: sensitive peptide detection on noisy matured sequences. *Bioinformatics*. 25(5):670-1. Epub 2009 Jan 19.)

Voici un diagramme représentant le compendium plus en détails.


[short reads]
Assemble reads using Velvet/Oasis
[long reads]
Extract assemblies
[available genome]
Align reads on genome using PASA2
[else]
Extract unmapped reads
Extract mRNA
Clustering using TGICL++
Peptides prediction using FrameDP
Analyse PASA2/TGICL++ report files
Mapping shorts reads on transcripts using Glint/Soap
Fonctionnal annotation using InterproScan
Compute expression
Integration in BIOS

I) Quick Start

PASA sera utilisé par le compendium si l'on dispose d'un génome de référence. Pour le QUICK START nous n'en avons pas besoin. Cependant si l'on veut installer PASA, on peut se référer au chapitre « Installation de PASA ») de ce document.

1. Installation

Dans un premier temps il faut récupérer la dernière version de l'archive : compendium_latest.tar.gz (http://lipm-bioinfo.toulouse.inra.fr/download/compendium/) Il faut se placer dans le répertoire dans lequel on veut installer le pipeline.

```
% wget http://lipm-bioinfo.toulouse.inra.fr/download/compendium/compendium_latest.tar.gz
```

Ensuite on décompresse l'archive

```
% tar xvfz compendium_latest.tar.gz
```

Le répertoire compendium est crée. On va maintenant passer à son installation. On défini la variable d'environnement $COMPENDIUM qui est le chemin sur votre système du répertoire compendium que l'on vient de décompresser.

```
% setenv COMPENDIUM /path/to/rep/compendium
```

note : pour définir une variable d'environnement, on utilise :
- `export VARIABLE=/path/variable` en bash
- `setenv VARIABLE /path/variable` en tcsh

A. Installation de PARALOOP

Paraloop est le programme qui va nous servir à distribuer les calculs sur les différents cpus de la machine, ou sur un cluster de calculs si à disposition.

Site-web : http://lipm-bioinfo.toulouse.inra.fr/paraloop/

Pour installer paraloop sur une seule machine multiprocesseur il suffit de décompresser l'archive et de définir la variable d'environnement.

```
% cd $COMPENDIUM/bin/
% wget http://lipm-bioinfo.toulouse.inra.fr/download/paraloop/paraloop.tgz
% tar xvzf paraloop.tgz
% ln -s paraloop-?.? paraloop
% setenv PARALOOP `pwd`/paraloop
```

Pour l'utiliser avec une architecture plus complexe (cluster de calcul, PBS, SGE, a.s.o) il va falloir éditer certains fichiers de configuration.

```
% cd $PARALOOP/etc/templates
% cp paraloop.root.cfg_XXX ../paraloop.root.cfg
% cd ..
```

Édition des fichiers paraloop.root.cfg et paraloop.cfg.
Une documentation est disponible :$PARALOOP/documentation/user-doc.pdf, vous pouvez aussi pour toute question envoyer un mail à emmanuel.courcelle@toulouse.inra.fr

B. Installation de TGICL++

On va maintenant installer TGICL++. Pour cela on va décompresser l'archive fournie avec le compendium.

```
% cd $COMPENDIUM/bin
% tar xzvf tgicl++.tar.gz
```

On va ensuite faire tourner le script de configuration install.pl, qui va mettre à jour les fichiers de configuration par rapport à l'arborescence de votre système et vérifier que tous les modules nécessaires sont bien installés.

```
% cd tgicl_linux
% ./install.pl --verbose
```

Le statut doit être **CONFIGURATION IS OK** à la fin de l'exécution du script. Sinon les messages d'erreur vous indiquerons quels sont les modules défectueux qui faut installer. Si tous les modules ne sont pas bien installé, TGICL++ ne sera pas fonctionnel.

C. Installation de FRAMEDP

Pour installer FrameDP, il suffit aussi de récupérer l'archive de la derniere verion.

Site-web: http://iant.toulouse.inra.fr/FrameDP/

```
% cd $COMPENDIUM/bin/
% wget http://iant.toulouse.inra.fr/FrameDP/download/framedp-Linux-x86_64.1.2.0.tar.gz
% gzip -cd framedp-Linux-x86_64.1.2.0.tar.gz | tar xvf -
% ln -s framedp-1.2.0 FrameDP
% cd FrameDP
% setenv FRAMEDP `pwd`
```

Afin d'utiliser la configuration de paraloop précédemment installé (et pas celle fournie par FrameDP), il faut exécuter :

```
% cd bin/ext/
% mv paraloop paraloop.FrameDP
% ln -s $PARALOOP paraloop
```

D. Installation de Velvet et Oases

Pour installer Velvet et Oases il faut décompresser l'archive fournie.

```
% cd $COMPENDIUM/bin/
% tar xvf velvet_oases_compendium.tgz
% cd velvet_oases_compendium
% bash
% ./install.sh
% tcsh (si on veut se mettre en mode tcsh plutot que bash)
```

E. Installation de BIOS Mapreads

Pour installer BIOS_Mapreads il suffit juste de décompresser l'archive et de faire tourner le script de configuration.

```
% cd $COMPENDIUM/bin/
% tar xvf BIOS_Mapreads_*.tar.gz
% cd BIOS_Mapreads
% ./bios_mapreads_install.pl
```

F. Configuration de pipeline

Maintenant que tout ou une partie du pipeline est installée, on va lancer le script de configuration qui va mettre à jour les fichiers de configuration en fonction de votre système.

```
% cd $COMPENDIUM
% ./configure.pl --verbose
```

Les programmes que l'on souhaite utiliser doivent avoir un statut **OK**, ainsi que les pré-requis : **PRE-REQUISITE MODULES/BINARIES CONFIGURATION IS OK**
Si un des programmes du pipeline (PASA, TGICL++ ou FRAMEDP) n'a pas été installé sous l'arborescence `$COMPENDIUM`, il faut passer explicitement le chemin où il a été installé au script `./configure.pl`. Pour cela se référer à l'usage `$COMPENDIUM/configure.pl -h`
Si tout n'est pas en statut OK, il faut commencer par résoudre ces problèmes avant de continuer.

2. Lancement

Maintenant que le pipeline est installé on va l'exécuter. Voici la description de l'usage

```
./compendium.pl
     ./compendium.pl --genome_file <file> --short_reads_librairies<file(s)>
--short_reads_type<string> --short_reads_format<string> --long_reads_librairies<file(s)>
        --cfg_file <file>  --work_repository <dir> [--assemble_short_reads_lib_one_by_one]
[--full_lengths_file <file>]  [--paired_ends_file <file>] [--step  <string>]
        [--cpus <integer|filename>] [--tgicl_restart <integer>] [--annotation_file
<filename>] [--update_annotation_pasa_only]

    Mandatory
        --genome_file                          : format fasta file of the genome
        --short_reads_librairies               : path to file(s). If several file, it should
have separateb by a comma, without space
        --cfg_file                             : the configuration file, in wich parameters
of Pasa and TGICL are modifiable
        --work_repository                      : the path of the work directory
       --long_reads_librairies                : path to file(s). If several file, it should
have separateb by a comma, without space

    Optional
        --short_reads_type                     : allowed values :
short,shortPaired,short2,shortPaired2,long,longPaired [mandatory if short_reads_librairies
defined]
```

```
        --short_reads_format                    : allowed values :
fasta,fastq,fasta.gz,fastq.gz,sam,bam,eland,gerald [mandatory if short_reads_librairies
defined]
        --quality_file                          : quality file of the transcripts
        --full_length_file                      : file with the identifiers of the
transcripts known as full lengths (this ids have to be present in the transcripts file !)
        --paired_ends_file                      : file with the identifiers of couple of
transcripts known as paired ends (the ids of the couple have to be on the same line)
                           eg : AL367179  AL367180
                                MtRH15N17A1   MtRH15N17N1
        --annotation_file                       : format gff3 file annotation of the genome
        --update_annotation_pasa_only           : if defined, the script will only update the
annotation and compare it with PASA annotation
        --step                                  : run particular step of the pipeline,
allowed values : VELVET_OASES,PASA,TGICL,FRAMEDP,SOAP
        --cpus                                  : either a number of cpus or a file
containing the list of pvm nodes
        --tgicl_restart                         : if tgicl is use with the
simplification_step (see configuration file), you can define a restart step
        --assemble_short_reads_lib_one_by_one
```

Des fichiers de logs seront crées :

- pipeline.log
- pipeline.error
- pipeline.stat : contient quelques stats pour chacune des étapes du pipeline

De plus pour chaque étape des fichiers de log/error sont générés.

« nb_cpus » est le nombre de cpus que l'on souhaite utiliser. (Pour BIOS sur une seule machine 4 cpus)

```
% setenv NCPUS nb_cpus
```

On se place dans un répertoire de travail $WORK pour lequel on a bien les droits d'écriture. Le pipeline génère beaucoup de données, il faut s'assurer d'avoir suffisamment de place sur le disque dur (surtout si on a beaucoup de transcrits) . On commence par copier dans ce répertoire le fichier de configuration du compendium.

```
% cp $COMPENDIUM/cfg/compendium_pipeline.cfg $WORK
```

On va commencer le pipeline à l'étape de clustering tgicl++, en exécutant le pipeline avec notre jeu de données exemple.

```
% $COMPENDIUM/compendium.pl --work $WORK --long_reads $COMPENDIUM/data/BRADI.fasta --cfg \
compendium_pipeline.cfg --cpu $NCPUS --step 'VELVET_OASES,TGICL,FRAMEDP'
```

Note : Avec l'option -- step, on peut définir exactement quelles sont les étapes que l'on veut exécuter, il suffit de le préciser.

3. Description des fichiers générés

Voici une vue rapide des fichiers importants générés à chaque étape.
VELVET/OASES :

- `$WORK/0_VELVET_OASES/contigs` : contigs généré par Velvet
- `$WORK/0_VELVET_OASES/transcripts` : transcrits représentant les contigs assemblés de Velvet par Oases

PASA :

- `$WORK/1_PASA/validated_transcripts.gff3` : fichier contenant la description gff3 de tous les transcrits qui ont été alignés/validés sur le génome de référence.
- Après update de l'annotation :
  - `$WORK/1_PASA/*_COMPENDIUM_PASA_assemblies_post_update.MRNA.*.fasta` : les mRNAs updaté par PASA (en fonction des status du fichier de conf) et ceux pour lesquels les assemblages PASA confirme totalement la structure
  - `$WORK/1_PASA/*_COMPENDIUM_PASA_assemblies_post_update.original.MRNA.*.fasta` : les mrNAs originaux de l'annotation qui ne sont pas modifié par PASA (pas bon status ou pas assemblages)
  - `$WORK/1_PASA/*_Pasa_assemblies_not_integrated_in_annotation.*.fasta` : les assemblages PASA qui ne participent pas à la mise à jour de l'annotation (pas bon status aussi ici)

TGICL++:

- `$WORK/2_TGICL/tgicl_mes_ESTs.fasta.tgicl++.consensus.fasta` : fichier contenant les clusters générés ainsi que les singletons
- `$WORK/2_TGICL/tgicl_mes_ESTs.fasta.tgicl++.consensus.qual` : fichier contenant la qualités des clusters générés ainsi que des singletons
- `$WORK/2_TGICL/tgicl_mes_ESTs.fasta.tgicl++.consensus.report` : fichier contenant pour chaque cluster sa composition en ESTs. (ainsi que les singletons qui ne sont composés que d'une seule EST)
- `$WORK/2_TGICL/tgicl_mes_ESTs.fasta.tgicl++.consensus.correspondance` : fichier listant le nouveau et ancien nom de chaque séquence s'il y'a eu renommage.

MAPREADS (SOAP):

- `$WORK/3_SOAP/*/Soap_results.all` : fichier résultat du mapping des banques Solexa sur les transcrits. Il y a un répertoire de crée par banque Solexa
- `$WORK/3_SOAP/Result_mapping.report` : fichier report au 'format' compendium pour toutes les banques

FRAMEDP :

- `$WORK/4_FRAMEDP/framedp.*.summary` : qui contient toutes les prédictions
- `$WORK/4_FRAMEDP/framedp.*.pepdb.fa` : la banque protéique

II) Installation / Utilisation Avancée

1. Fichier de configuration de Pipeline

Voici la description des paramètres du fichier de configuration de Pipeline. Il faut faire attention au nom que l'on va donner à la banque, il doit être unique car PASA va s'en servir pour créer une base de données.

Description rapide des paramètres du fichier de configuration :

PASA:
- MYSQLDB=nom de la base de données, ne doit pas déjà avoir été utilisé

- MAX_INTRON_LENGTH=longueur maximum d'un intron
- MIN_INTRON_LENGTH=longueur minimum d'un intron
- MIN_PERCENT_ALIGNED=pourcentage minimum longueur alignée
- MIN_AVG_PER_ID=pourcentage minimum d'identité de l'alignement

TGICL++:

- tgicl_simplify_quality= utilisation des qualités dans les étapes de simplification. Quand beaucoup de données sont disponibles, nous conseillons de mettre ce parametre a « false » pour accélerer le processus.
- blast_min_percent_overlap=pourcentage minimum d'identité du chevauchement (pour étape megablast)
- blast_min_length_overlap=longueur minimum d'un chevauchement (pour étape megablast)
- cap3_min_percent_overlap=pourcentage minimum d'identité du chevauchement (pour étape cap3)
- cap3_min_length_overlap=longueur minimum d'un chevauchement (pour étape cap3)

2. Exécuter une étape sur un autre serveur

On peut exécuter une étape du compendium sur une autre machine. Pour que le pipeline prenne en compte ces résultats il faut mettre les fichiers résultats (ceux attendus par le compendium) dans le répertoire bien nommé de l'étape et ceci dans le $WORK.

Par exemple pour Velvet_Oases il faut placer les fichiers contigs.fa et/ou transcripts.fa dans un répertoire 0_VELVET_OASES, que l'on placera ensuite dans le répertoire $WORK.

Il faut ensuite lancer le compendium en précisant que l'on ne veut pas exécuter l'étape VELVET_OASES. Pour cela il faut préciser l'option --step. (par exemple on mettra --step 'PASA,TGICL,SOAP,FRAMEDP')

Les fichiers attendus après chacune des étapes sont (présence vérifiée par le pipeline):
- Velvet/Oases : les fichiers `contigs.fa` et/ou `transcripts.fa`
- PASA : le fichier `validated_transcripts.gff3`
- TGICL : le fichier `tgicl_mes_ESTs.fasta.tgicl++.consensus.fasta`

3. PASA

A. Installation

PASA est distribué sur le site web http://pasa.sourceforge.net/. La version que nous avons testé avec le compendium est la version PASA-r04042009. Nous fournissons ici un enchaînement de commandes pour installer PASA, en plus de la documentation disponible sur le site, pour pallier certains manques. Pour tout ce qui est configuration du pipeline (configuration MYSQL, url...) il faut s'y référer.
On peut installer PASA, dans le répertoire `$COMPENDIUM/bin`, sinon il faudra donner au script `configure.pl` (section II.D) le chemin du répertoire où est installé PASA.

Dans un premier temps il faut veiller à ce qu'un certain nombre de librairies soient installées sur votre système :

```
% apt-get install g++
% apt-get install makedepend
% apt-get install libghc6-zlib-dev
% apt-get install libgd-graph-perl
```

On va procéder à l'installation :

```
% cd $COMPENDIUM/bin/
% wget http://www.gene.com/share/gmap/src/gmap-2007-09-28.tar.gz
% tar xvfz gmap-2007-09-28.tar.gz
% cd gmap-2007-09-28
% ./configure
% make
% make install
% cd $COMPENDIUM/bin/
% wget
'http://sourceforge.net/projects/pasa/files/pasa/pasa_r04042009/PASA_r04042009.tar.gz/do
wnload'
% tar xvfz PASA_r04042009.tar.gz
% export PASAHOME=$COMPENDIUM/bin/PASA
% mkdir $PASAHOME/bin
% cd $PASAHOME/pasa_cpp
```

On doit ensuite éditer le `Makefile` :
il faut ajouter : `-include /usr/include/c++/4.3/cstdlib a la ligne CFLAGS = -O3`
Ce qui donne : `CFLAGS = -include /usr/include/c++/4.3/cstdlib -O3`

```
% make
% cp pasa $PASAHOME/bin
% cd $PASAHOME/SLCLUST
```

On doit éditer le `src/Makefile`.
Il faut ajouter : `-include /usr/include/c++/4.3/cstdlib a la ligne LOCAL_CFLAGS = -Wall`
Ce qui donne : `LOCAL_CFLAGS = -Wall -include /usr/include/c++/4.3/cstdlib`

```
% make clean
% make depend
% make install
% cp bin/slclust $PASAHOME/bin
% cd $PASAHOME/SIM4_MOD/sim4.2002-03-03_mod/
% make
% cp sim4-mod $PASAHOME/bin
% cd $PASAHOME/seqclean
% tar xvfz seqclean.tar.gz
% cp seqclean/* $PASAHOME/bin
% cp seqclean/bin/* $PASAHOME/bin
% cd $PASAHOME/cdbtools/
% tar xvfz cdbfasta.tar.gz
% tar xvfz tgi_cpp_library.tar.gz
% cd cdbfasta
% make
% cp cdbfasta cdbyank $PASAHOME/bin
% ln -s $COMPENDIUM/bin/PASA/bin/* /usr/local/bin/
% ln -s /usr/bin/perl /usr/local/bin/perl
% sed -i 's/^---/-- /g' $PASAHOME/schema/cdna_alignment_mysqlschema
```

On doit maintenant pour MYSQL créer des utilisateur PASA spécifique.

Pour finir on va modifier la configuration Apache pour pouvoir visualiser les résultats à l'aide d'une page web.

- Si environnement BIOS :

```
% vi /etc/apache2/sites-available/bios.conf
```

Ajouter la re-direction cgi :

```
ScriptAlias /pasa/cgi-bin/ "/www-bios/bin/ext/compendium/bin/PASA/cgi-bin/"
```

- Sinon :

Éditer le fichier : `/etc/apache2/sites-available/monserveur.conf` (fichier de configuration du serveur apache)
Ajouter la re-direction cgi : `ScriptAlias /pasa/cgi-bin/ "/mon/chemin/vers/pasa/cgi-bin/"`

Il faut maintenant redémarrer Apache :

```
% /etc/init.d/apache2 restart
```

Une fois PASA installé, on devra faire retourner le script `$COMPENDIUM/configure.pl` si le compendium a été préalablement installé.

B. Utilisation

Si l'on dispose d'un génome de référence on va lancer le pipeline en entier.
On notera « mon_genome.fasta » le fichier contenant le génome, « mes_ESTs.fasta » le fichier fasta des transcrits

```
% $COMPENDIUM/compendium.pl --work $WORK --genome_file mon_genome.fasta –transcripts \
mes_ESTs.fasta --cfg compendium_pipeline.cfg --cpu $CPUS &
```

On peut visualiser les résultats de PASA, via une interface Web dont l'adresse est donnée dans le fichier pipeline.log si lors de l'installation de PASA cette possibilité a été configurée.

Attention : Il faut s'assurer avant de lancer PASA qu'une banque du même nom que celui que l'on a précisé dans le fichier de configuration n'existe pas déjà. Pour cela il faut se connecter à la base mysql de PASA (voir fichier `$COMPENDIUM/bin/PASA/pasa_conf/conf.txt` pour récupérer le mot de passe root ainsi que le login, si PASA est installé dans le répertoire du compendium)
Si PASA plante, il faut aussi supprimer la banque qu'il a crée avant de le relancer.
Connection :

```
% mysql -u login -p passwd
```

Commandes SQL :

```
SHOW DATABASES;
DROP DATABASE nom_de_la_base;
```

4. TGICL++

A. Installation

Une façon d'optimiser TGICL++ est de recompiler pour sa machine les sources de certains des programmes utilisé. Pour cela les sources sont disponibles dans le répertoire `$COMPENDIUM/bin/tgicl++/src/`

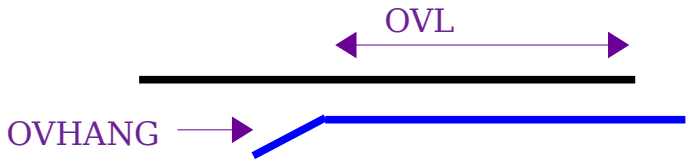
B. Utilisation

Si l'on veut affiner l'utilisation de TGICL++, on peut modifier les paramètres du fichier de configuration de tgicl. Notamment, il est par défaut défini une série d'opération de simplification, visant à réduire le nombre d'ESTs à clusteriser en regroupant avec des seuils très stringents certaines séquences que l'on va identifier comme redondantes.
Pour cela on va utiliser 2 programmes.
Nrcl: permet de clusteriser/compresser par sélection de la plus longue séquence englobante (séquence container). On peut dire que l'on « compresse » les séquences et tclust qui permet de construire les composantes connexes par clustering transitif. On peut dire que l'on « allonge » les séquences.

Ces deux programmes sont utilisés alternativement, avec des seuils très stringents en vu de masquer la redondance du jeu de données. Ces différentes étapes sont paramétrables (nombre étapes, seuils, programmes) dans le fichier de configuration `$COMPENDIUM/bin/tgicl++/cfg/tgicl++.cfg`

Voici les seuils que nous avons utiliser :

tclust :



nrcl :

OVL et OVHANG sont en pb, et SCOV en %.
Il est possible d'en définir d'autre, pour cela il faut consulter les usage de ces deux programmes :

```
% $COMPENDIUM/bin/tgicl++/bin/nrcl -h
% $COMPENDIUM/bin/tgicl++/bin/tclust -h
```

Il faut pour cela modifier

```
#thresholds for the simplify step
THRESHOLD_STEP1=nrcl_PID=99,nrcl_SCOV=99
THRESHOLD_STEP2=tclust_PID=99,tclust_OVHANG=5,tclust_OVL=200
THRESHOLD_STEP3=nrcl_PID=98,nrcl_SCOV=98
THRESHOLD_STEP4=nrcl_PID=97,nrcl_SCOV=97
THRESHOLD_STEP5=tclust_PID=98,tclust_OVHANG=5,tclust_OVL=200
THRESHOLD_STEP6=tclust_PID=97,tclust_OVHANG=10,tclust_OVL=100
```

Les règles pour définir une étape de simplification sont :
- la variable commence par THRESHOLD_STEP suivi du numéro de l'étape

- pour chaque paramètre que l'on veut donner il faut le préfixer par le nom du programme et utiliser la nomenclature de celui-ci (voir les usages précédemment)
- On n'utilise qu'un seul programme (nrcl ou tclust) à la fois

Dans ce fichier de configuration on peut aussi modifier les autres paramètres :
- quality_value=valeur donnée par défaut aux bases des séquences qui n'ont pas de qualité.
- prefix=prefix des séquences une fois le clustering effectué (les séquences vont être renommée). Si cette variable est vide les séquences ne seront pas renommées.
- cluster_minlen et cluster_mincount = affiche dans le fichier résultat uniquement les clusters dont la longueur sera supérieur ou égale à cluster_minlen et/ou le nombre de membres supérieur ou égal à cluster_mincount. Si ces variables ne sont pas définies, le fichier ne sera pas filtré.

Tgicl++ offre la possibilité de redémarré l'analyse à chacune des étapes de simplification si par exemple une erreur est levé (cap3 est susceptible de générer des erreurs qui arrête tgicl++)

Une des erreurs de cap3 est traitée automatiquement, les ESTs qui posent problème sont retirées du cluster (elles se retrouveront dans le fichier résultats de l'étape de simplification pour être de nouveau traitées à l'étape d'après) et le cap3 correspondant est relancé. Ici cette étape de traitement de l'erreur peut être longue car généralement ce sont les clusters contenant le plus de séquences qui pose problème (CL1,CL2,CL3..). L'erreur cap3 et les commandes relancées sont décrites dans le fichier erreur de tgicl++.

Pour relancer compendium à l'etape 5 (par exemple) de tgicl++ il faut :
- effacer le répertoire step5 et les suivants
- au besoin, il faut supprimer les reads ayant poser problème dans les fichier résultats de l'étape précédente step4/consensus_and_singletons_step4, step4/consensus_and_singletons_step4.report, step4/consensus_and_singletons_step4.qual.

```
% $COMPENDIUM/compendium.pl –transcripts_file=$WORKDIR/compendium/PPARA.fasta \
--cfg_file=$WORKDIR/compendium/compendium_pipeline.cfg \
--work_repository=$WORKDIR/compendium/ --step=TGICL_FRAMEDP --cpu=$NCPUS --tgicl_restart 5
```

5. FrameDP

Voir le site web : http://iant.toulouse.inra.fr/FrameDP/

A la fin du processus FrameDP, par defaut un fichier de transcrits corrigés (correction de frameshift + reverse complementation) est généré dans le repertoire 4_FRAMEDP sous le nom **`clusters.corrected.fasta`**
Un tarball contenant les images FrameD est egalement créé dans le repertoire.

6. Velvet/Oases

Si l'on veut utiliser des banque Paired-ends en entrée du compendium et que l'on utilise effectue l'étape Velvet, on doit formater correctement les fichiers.
En effet il faut 1 seul fichier ou les identifiants des couples se suivent. Pour cela il y a un script à disposition dans le package `velvet_oases_compendium` : `shuffleSequences_fasta.pl`

```
Usage: ./shuffleSequences_fasta.pl forward_reads.fa reverse_reaads.fa outfile.fa
    forward_reads.fa / reverse_reads.fa : paired reads to be merged
    outfile.fa : outfile to be created
```

```
% $COMPENDIUM/bin/velvet_oases_compendium/velvet/shuffleSequences_fasta.pl banque-pe1.fasta
banque2-pe.fasta banque_pe_velvet_format.fasta
```